

4D Write Proリファレンス

-  はじめに
-  4D Write Pro エリアを定義する
-  4D Write Proドキュメントを4Dオブジェクトフィールドに保存する
-  4D Write Pro エリアを使用する
-  4D Write Pro ドキュメントの印刷
-  4D Write Pro ドキュメントに含める式の制限
-  4D Write ドキュメントの読み込み
-  4D Write Pro標準アクションの使用
-  ピクチャーを管理する
-  4D Write Proランゲージ
-  新着
-  コマンドリスト (文字順)

はじめに

概要

4D Write Proは、4Dユーザーに対して、4Dデータベースに完全に統合した進化したワードプロセッシングツールを提供します。4D Write Proを使用すれば、プリフォーマットされたEメールや文章に画像、スキャン済みの署名、フォーマット済みのテキストやダイナミック変数用のプレースホルダーなどを含めることができます。また請求書やレポートを動的に作成し、フォーマット済みのテキストや画像を含めることができます。

この製品の主な機能は以下の通りです：

- 4D Write との互換性: 4D Write Proオブジェクトは、以前の4D Writeドキュメントを開いたり変換したりでき、また4D Writeのプロパティの大部分をサポートしています。
- ワードプロセッシング: フォームに埋め込まれた4D Write Proオブジェクトは、テキストとスタイルの操作、画像の挿入、読み込みと書き出しなどを含めた様々な標準のワープロ機能を持ち合わせます。
- データベースとの統合:
 - 4D Write Pro オブジェクトはデータベースからのデータや、4Dが計算をしたデータなどを表示する変数部分を表示することができます。
 - 4D Write Pro ドキュメントはデータベースフィールド内や、ディスク上に保存することもできます。

インストールとアクティベート

4D Write Pro はプラグインではなく、4D自身に完全に統合されており、配布と管理がより簡単になっています。追加のインストールは何も必要なく、フォーム内に4D Write Proエリアを追加するだけで、4Dアプリケーション内において4D Write Pro変数を直接管理することができます。

しかしながら、4D Write Proは4D Writeと同じライセンスを使用することに注意して下さい。機能を有効化するためには、アプリケーションにそのライセンスをインストールする必要があります。

必要要件: Windows上では、4D Write Proの機能はDirect2Dに依存しています。Windows 7またはWindows Server 2008以降のマシンでは、必要なバージョンのDirect2Dを利用可能にするために、Windowsプラットフォーム更新プログラムがインストールされていることを確認して下さい。

このマニュアルについて

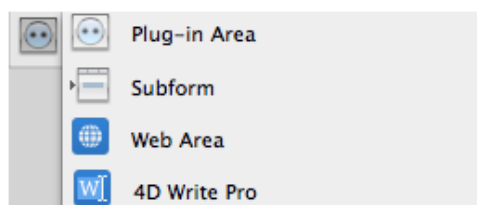
このマニュアルは4D Write Pro リファレンスガイド です。ここでは全ての4D Write Proの機能に加え、ユーザーインターフェースとランゲージコマンドについてカバーします。

4D Write Proオブジェクトは、特定のコマンド("4D Write Proランゲージ")に加えて、4Dランゲージリファレンス内にある他の4Dテーマ(**オブジェクト(フォーム)** テーマと**スタイル付きテキスト** テーマ)のコマンドを使用しても管理する事ができるという点に注意して下さい。

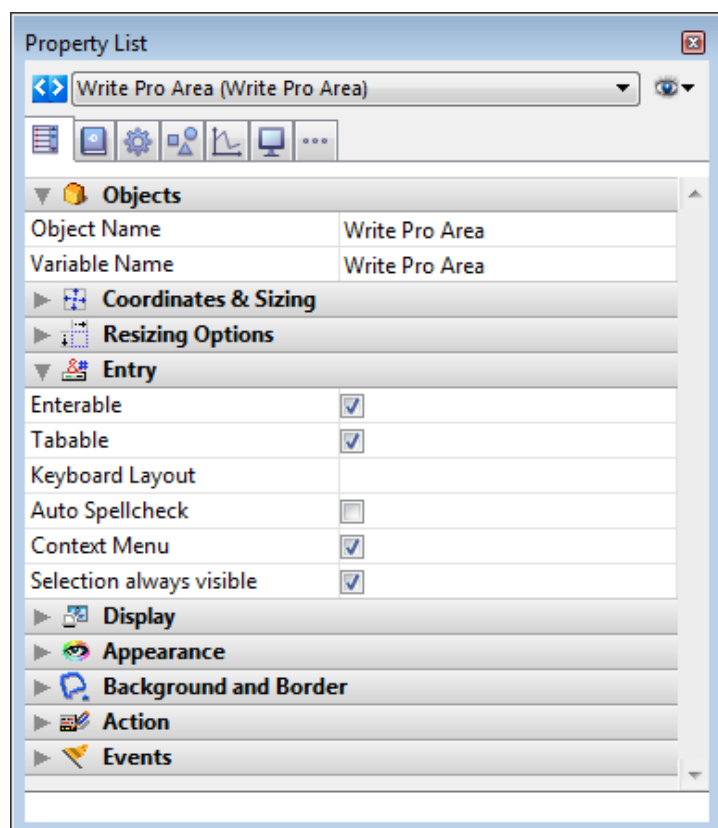
4D Write Pro エリアを定義する

エリアの作成

4Dでは、4D Write Proドキュメントは**Write Pro**という4D フォームオブジェクト内にて手動で表示・編集することができます。このオブジェクトはオブジェクトバーの最終ツールの一部として提供されています:



4D Write Proフォームエリアはプロパティリスト内の標準のプロパティを通して設定することができます。すなわち、**オブジェクト名** そして **変数名**、**座標**、**入力**、**表示**、**外見**、**イベント**です。



変数名のプロパティはランゲージにおいて4D Write Proエリアの参照として使用します。変数は*object*型でなければならないという点に注意して下さい(詳細については[C_OBJECT](#) コマンドを参照して下さい)。

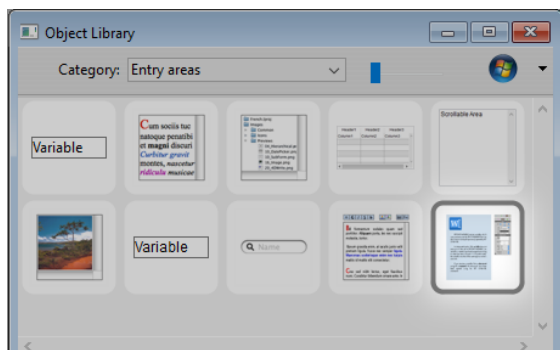
"入力" プロパティはテキスト入力の基本的な機能を管理します:

- **入力可能:** エリアをロック・アンロックすることによって編集を許可または許可しないことを選択できます。
- **自動スペルチェック:** 4D Write Pro において利用可能です。

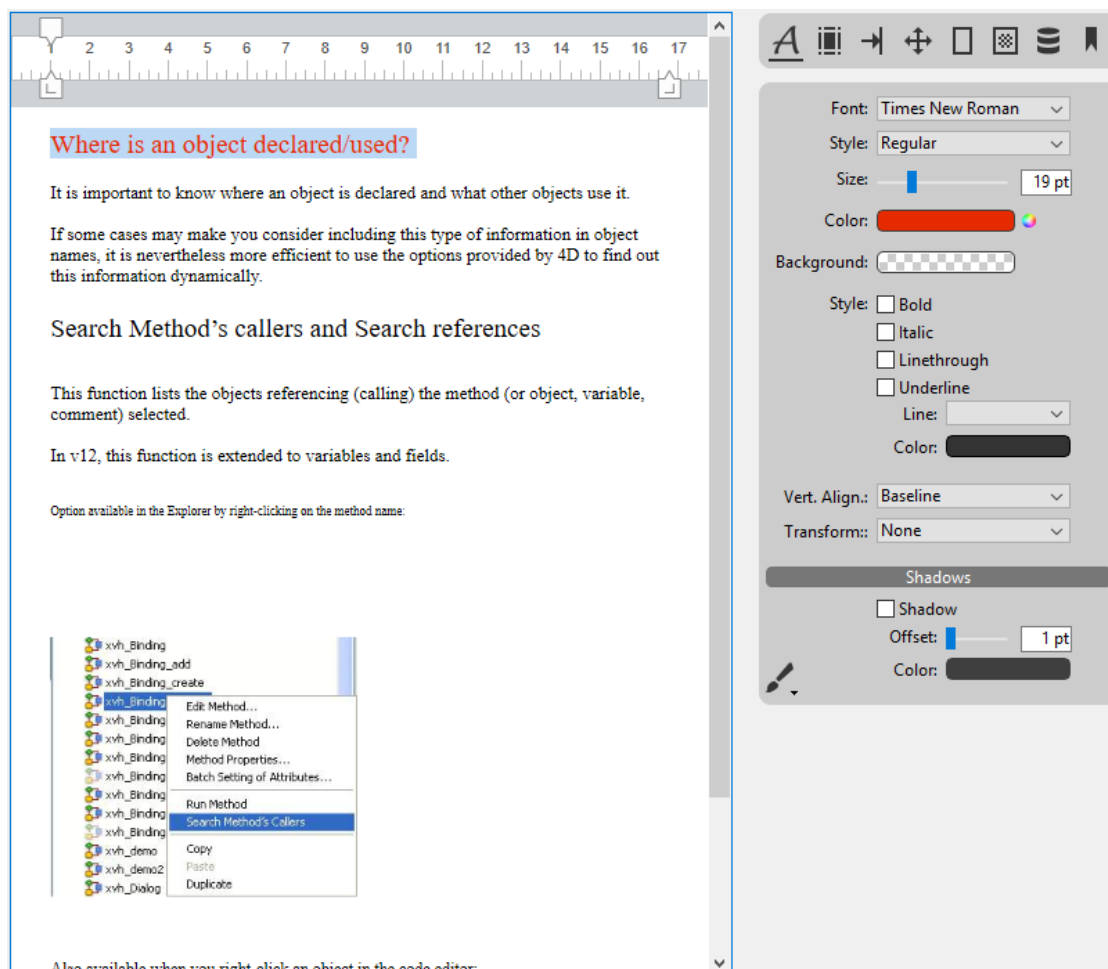
- **コンテキストメニュー:** アプリケーションモードにおいてコンテキストメニューを有効化・無効化します(**4D Write Pro エリアを使用する** の章を参照して下さい)。
- **選択を常に表示:** 標準のテキストエリアにおいてテキストの選択を管理します。

オブジェクトライブラリーの4D Write Proウィジェットの使用

オブジェクトライブラリー内("入力エリア"テーマ内)にある**4D Write Pro** オブジェクトを仕様することで設定済みの4D Write Proエリアを作成することができます:



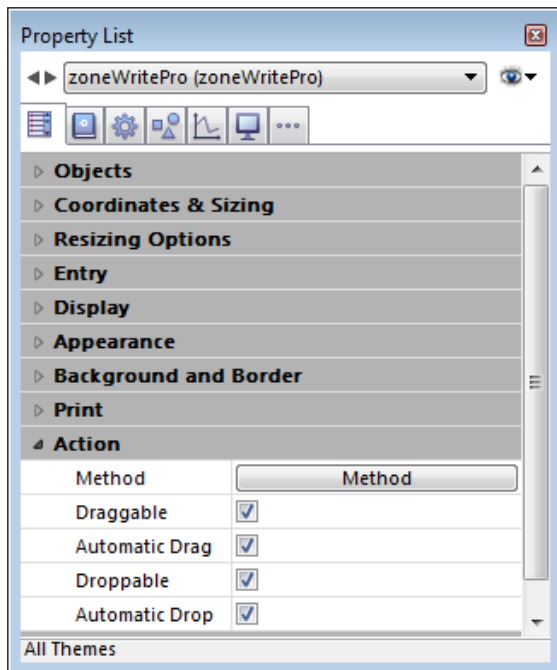
このエリアには、エリアの全ての属性(フォント、カラー、スタイル等)を管理することができるコントロールパネルが付属します:



より詳細な情報については、**4D Write Pro エリア**の章を参照してください。

ドラッグ&ドロップの設定

ご自分の4D Write Proエリアでのドラッグ&ドロップ機能の設定をするためには、プロパティリストの"アクション"テーマ内にて適切なオプションを選択する必要があります:



4D Write Proエリアは、二つのドラッグ&ドロップモードをサポートします:

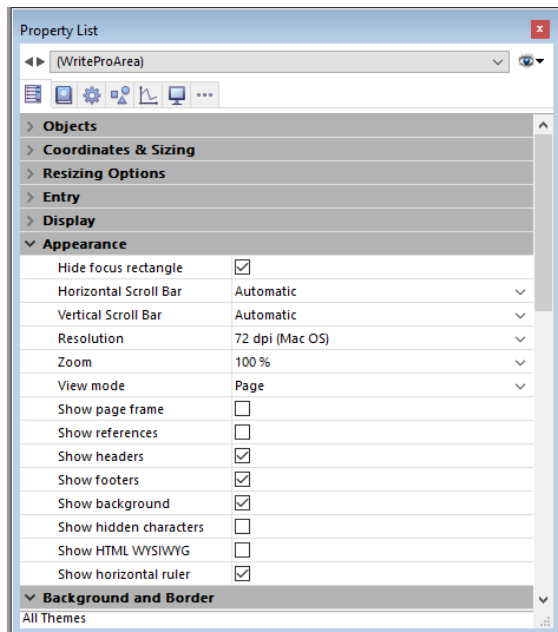
- **カスタムモード:** "ドラッグ可"と"ドロップ可"のオプションのみチェックされています。
このモードでは、テキストを選択し、移動させる事ができます。 On Begin Drag Over イベントのオブジェクトメソッドが呼び出され、このカスタムモードを使用してドロップ時の挙動を定義する事ができます。
- **自動モード:** "ドラッグ可"、"ドロップ可"、そして"自動ドラッグ"と"自動ドロップ"のオプションがチェックされています。
このモードでは、選択したテキストを移動または(**Alt/Option**キーを押しながら)コピーすることができます。 On Begin Drag Over イベントはトリガーされません。

注: "自動ドラッグ"と"自動ドロップ"のオプションしか選択していなかった場合、4D Write Proエリアには何の影響もありません。

ビュープロパティの設定

4D Write Pro エリアに 4D Write Pro ドキュメントがデフォルトでどのように表示されるかを指定するには、プロパティリストで提供されているドキュメントビュープロパティを設定します。これらのプロパティを使って、例えば 4D Write Pro ドキュメントを印刷されるとおりに表示するか、それともブラウザで見たときと同じように表示するかを設定することができます。同一のフォーム上で、同じ 4D Write Pro ドキュメントの異なるビューを表示させることも可能です。

ドキュメントビューの設定は、4D Write Pro フォームオブジェクトのプロパティリストにおいて、**アピ**
アランステーマの項目で指定します:



- **解像度:** 4D Write Pro エリアの画面解像度を設定します。デフォルト値は **72 dpi (Mac OS)** で、これはすべてのプラットフォームにおける 4D フォームの標準解像度です。この項目を**自動**に設定すると、Mac OS と Windows 間でドキュメントのレンダリングが異なることになります。特定の dpi 値を指定すると、両プラットフォームでのレンダリングを同じになります。
- **ズーム:** 4D Write Pro エリアのコンテンツ表示に使用するズーム率を設定します。デフォルト値は 100% です。
- **ビューモード:** フォームエリア内の 4D Write Pro ドキュメントの表示モードを設定します。次の値が提供されています:
 - **ページ** (デフォルト): もっとも完全といえるビューモードで、ページの枠、余白、改ページ、ヘッダー & フッターなどを含みます。詳細については [ページビュー機能](#) を参照ください。
 - **下書き:** 基本のドキュメントプロパティを含む下書きモードです。
 - **埋め込み:** 埋め込みエリアに適切なビューモードです。余白や、ヘッダー & フッター、ページフレームなどは表示されません。
このモードは Web に似た出力をするのに使用することもできます (この場合には、解像度を 96 dpi に設定のうえ、**HTML WYSIWYG を表示**オプションを有効にします)。

注: ビューモードプロパティは画面上のレンダリングにのみ使用されます。印刷設定については、特定のレンダリングルールが自動的に適用されます ([4D Write Pro ドキュメントの印刷](#) 参照)。

- **ページ枠を表示:** ページビューモードが "ページ" に設定されている場合に、ページのフレームを表示 / 非表示にします。デフォルトは非表示です。
- **参照を表示:** ドキュメントに参照として挿入された 4D 式をすべて表示します。このオプションが無効 (デフォルト) になっていると、4D Write Pro は挿入された 4D 式をカレント値で評価して、その値を表示します。もとの 4D 式を確認したいときには、このオプションを有効にします。すると、これらの参照は灰色の背景色とともにドキュメント内に表示されます。
例えば、フォーマットを指定したカレント日付を挿入していると、デフォルトでは次の表示になります:

July 11, 2016

参照を表示オプションを有効にすると、代わりにもとの参照が表示されます:

`String(Current date;Internal date long)`

注: 4D 式を挿入するには、**ST INSERT EXPRESSION** コマンドを使います。

- **ヘッダー / フッターを表示:** ページビューモードが "ページ" に設定されている場合に、ページのヘッダー & フッターを表示 / 非表示にします。デフォルトは表示です。ヘッダー & フッターについての詳細は [を参照](#)ください。
- **背景を表示:** ページの背景画像および背景色を表示 / 非表示にします。デフォルトは表示です。
- **非表示文字を表示:** 非表示の文字を表示 / 非表示にします。デフォルトは非表示です。
- **HTML WYSIWYG を表示:** HTML WYSIWYG ビューを有効 / 無効にします。このビューでは、すべてのブラウザーに対応していない 4D Write Pro の属性が取り除かれます (デフォルトは無効)。
- **水平ルーラーを表示:** ルーラーを表示/非表示にします(デフォルトでは表示)。4D Write Pro エリアでのルーラーについてのより詳細な情報については、[ルーラーの管理](#) の章を参照してください。

互換性に関する注記:

- 4D v15 R5 までのバージョンで作成された 4D Write Pro ドキュメントは、これらのプロパティはデフォルト値に設定されますが、例外として解像度のみ自動的に設定されます。
- 水平ルーラーは、4D v16 R2以降で作成されたデータベースについてはデフォルトで利用可能です。それ以前のバージョンから変換されたデータベースについては、このセッティングはデフォルトではチェックされていません。

4D Write Proドキュメントを4Dオブジェクトフィールドに保存する

4D Write Proドキュメントを自動的に4Dデータファイルに保存できるようになりました。フォーム内に4D Write Proエリアを作成し、そのエリアの中身を保存するオブジェクトフィールドを作成すれば、エリア内に入力したテキストはレコードが評価されるごとにそれぞれのレコードへと自動的に保存されます。その後**QUERY BY ATTRIBUTE**コマンドを使用して内部属性の値に基づいてレコードを選択することができます。また独自の属性を4D Write Proエリアに追加しクエリすることができます。

この章では以下の機能について説明しています：

- フォーム内の4D Write Proエリアを4D オブジェクトフィールドへと結びつける
- 標準のオブジェクトコマンド**OB SET**、**OB Get**および**QUERY BY ATTRIBUTE**を使用して、保存されている4D Write Proドキュメントのカスタムの属性を設定、取得、そしてクエリする

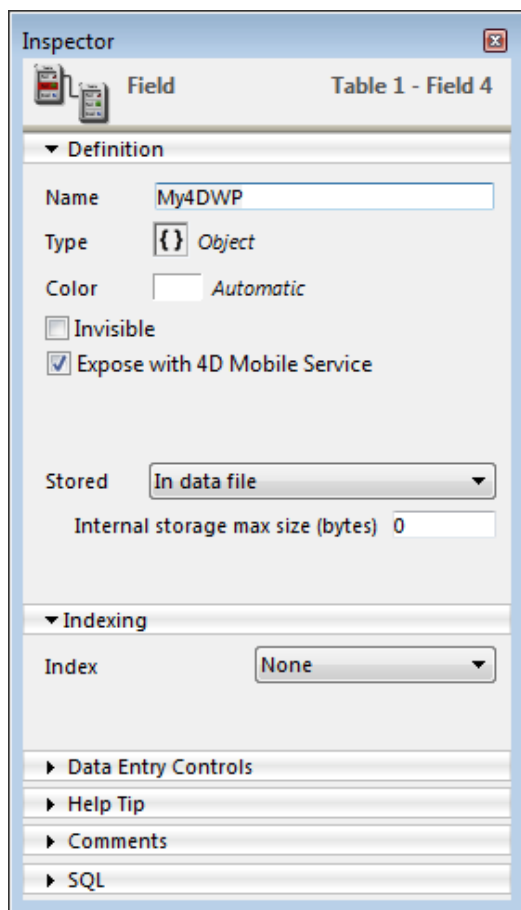
4D オブジェクトフィールドを4D Write Proエリアへと割り当てる

4D Write Proエリアを4Dオブジェクトフィールドに割り当てるためには、エリアの変数名プロパティにフィールド名を入力するだけです。

ストラクチャー内にオブジェクトフィールドを作成

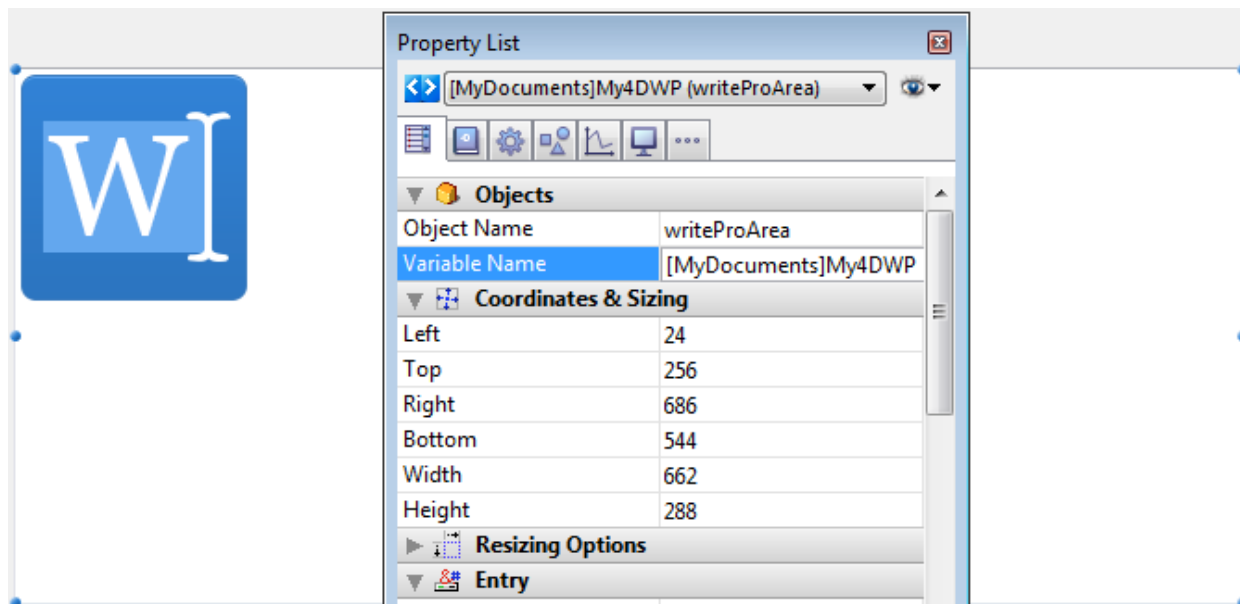
データベースのストラクチャー内において、4Dオブジェクトフィールドであればどれも4D Write Proドキュメントを保存するのに使用する事ができます。他のオブジェクトフィールドと同様、必要に応じて以下のフィールドの標準のプロパティを定義する必要があります：

- フィールド名
- "4D Mobile Serviceに公開"などの属性とインデックス
- 保存オプション(詳細についてはを**データをデータファイル外に保存**参照して下さい)



オブジェクトフィールドを4D Write Proエリアに割り当て

4D Write Proドキュメントを保存するオブジェクトフィールドを決めたら、あとはそのエリアを含んでいるフォーム内で参照するだけです。どのようなテーブルフォームもプロジェクトフォームも使用することができます。フォームエディター内において、4D Write Proエリアのプロパティリスト内の、**変数名**の欄に標準の"[Table]Field"表記を使用してフィールド名を入力して下さい:



これで4D Write Proエリアはフィールドと関連付けがなされ、エリアの中身はレコード毎に自動的に保存されるようになりました。4D自動アクションボタンを使用しない場合、4Dコマンドを使用してエリアを手動で保存しなければならない点に注意して下さい。

カスタムの属性を使用

4D Write Proエリアがオブジェクトフィールド内に保存されているとき、4D Write Proドキュメントにはカスタムの属性を保存または読み出しすることができます。例えば作者名、ドキュメントのカテゴリなど、どんな追加情報でも有用だと思えるものは使用する事ができます。そしてカスタムの属性をクエリし、条件に合致したレコードを選択することができます

カスタムの属性は **WP EXPORT DOCUMENT** または **WP EXPORT VARIABLE** コマンドで書き出されます。カスタムの属性は**JSON Stringify**コマンドを使用して4D Write Pro オブジェクトフィールドをJSONに変換する際にも書き出されます(同時に4D Write Pro のメインドキュメント属性も書き出されます)。

カスタムの属性を設定または取得するためには、オブジェクト記法を使用するか、**OB Get** と **OB SET** コマンドを使用するだけです。

たとえばフォームメソッドにおいて、以下のように書くことがでカスタムの属性を設定できます：

```
If(Form event=On Validate)
  [MyDocuments]My4DWP["myatt_Last edition by"]:=Current user
  [MyDocuments]My4DWP.myatt_Category:="Memo"
  [MyDocuments]My4DWP:=[MyDocuments]My4DWP //編集を記録
End if
```

あるいは：

```
If(Form event=On Validate)
  OB SET([MyDocuments]My4DWP;"myatt_Last edition by";Current user)
  OB SET([MyDocuments]My4DWP;"myatt_Category";"Memo")
End if
```

また、以下のように書いて、ドキュメントのカスタムの属性を読み出すことができます：

```
vAttrib:=[MyDocuments]My4DWP.myatt_Category
```

あるいは：

```
vAttrib:=OB Get([MyDocuments]My4DWP;"myatt_Last edition by")
```

カスタムの4D Write Pro属性をデータファイルに保存していた場合、これらの属性をクエリして適切な属性の値を含むレコードのセレクションを作成することができます。以下の例では、レコードを選択するためにオブジェクトフィールドを含んでいるテーブルをクエリします：

```
QUERY BY ATTRIBUTE([MyDocuments];[MyDocuments]My4DWP;"myatt_Category";="Memo")
//MyDocuments内の、(4D Write Proエリアに割り当てられている)My4DWPオブジェクトフィールド内で
//"myatt_Category"というカスタム属性が"Memo"という値を含んでいるレコードを全て選択します
```

カスタム属性の名前についての注意： カスタム属性は4D Write Pro内部属性と同じ名前空間を共有するため、内部属性とカスタム属性との衝突を避けるために、独自の属性を定義するときには必ず接頭辞をつけた名前を定義する事が強く推奨されます。接頭辞がついていない名前は4D Write Proの内部属性のために予約されているからです。接頭辞であればどのような独自の接頭辞であっても使用する事ができます(上記の例では"myatt_" を接頭辞として使用しました)。

注: 4D v15 R4以降、4D Write Pro内部属性は、標準の**QUERY BY ATTRIBUTE**と**OB Get**と**OB SET** コマンドだけではなく、**WP SET ATTRIBUTES**、**WP GET ATTRIBUTES**、**WP RESET ATTRIBUTES**コマンドを使用してもプログラムでアクセスできるようになりました。詳細な情報については、**4D Write Pro属性**の章を参照して下さい。

4D Write Pro エリアを使用する

4D Write Pro エリア内のドキュメントを管理する

4D アプリケーションでは、4D Write Proテーマ内にある特定のコマンドを使用することによって、4D Write Proドキュメントを作成、読み込み、書き出しすることができます (**WP EXPORT DOCUMENT**、**WP EXPORT VARIABLE**、**WP Import document**、**WP New**)。

4D Write Proエリアをデータベースのフォーム内のオブジェクトフィールドと関連づけることができます。これにより、各 4D Write Proドキュメントはレコードとともに自動的に保存され、データベースのデータ内に保存されます (**4D Write Proドキュメントを4Dオブジェクトフィールドに保存する** 参照)。

.4wp ドキュメントフォーマット

.4wp を使用する事によって 4D Write Proドキュメントをロスなくディスク上に保存したりディスクから開いたりすることができます。

.4wp フォーマットはドキュメント名と同じ名前を持つ ZIPフォルダーと、その中に格納された HTML テキストと画像から構成されます：

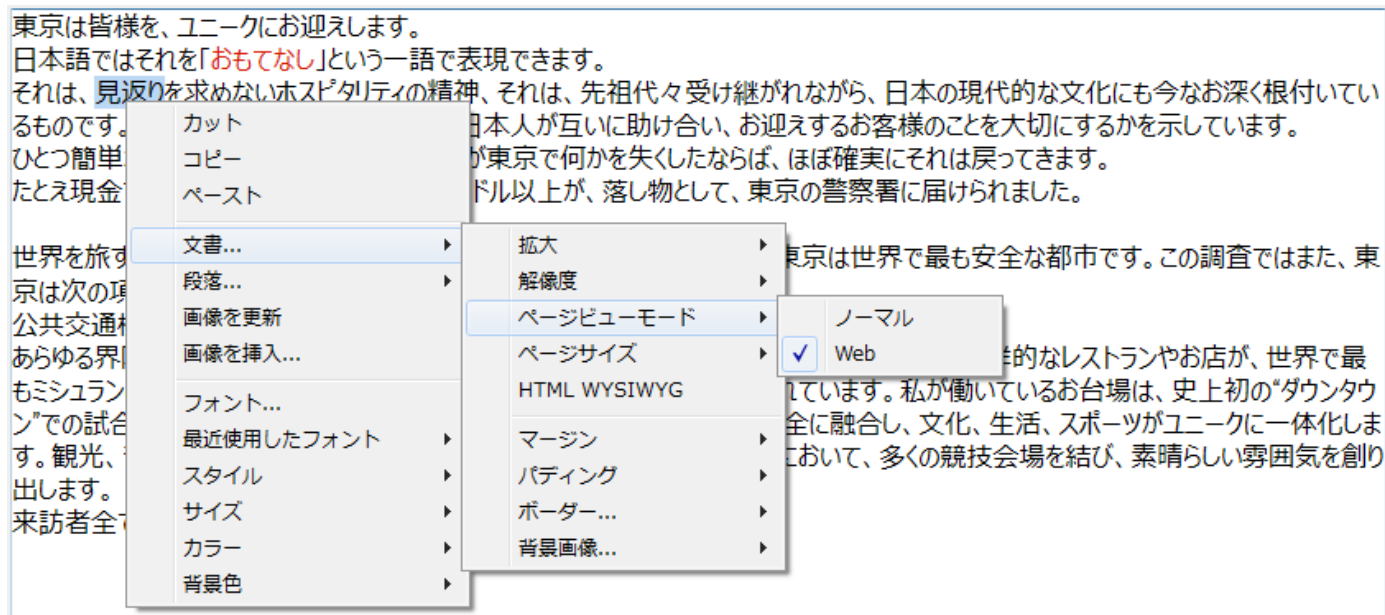
- HTMLテキストは、通常の HTML と (計算されていない) 4D式に加え、4D特有のタグを組み合わせます。
- 画像は、HTMLファイルの隣にある、ドキュメント名と同じ名前を持つフォルダー内に保存されています。

.4wp ドキュメントは HTML に基づいているので、HTMLをサポートしているものであればどんな外部アプリケーションでもそれを読み込んだり開いたりすることができます。

注: 4D Write Pro の内部ドキュメントは HTML に 4D独自の拡張を加えたもので、HTML5/XHTML5 に準拠していながら、独自の HTML/CSS属性のサブセットとタグもサポートしています。したがってデータ損失のリスクなく 4D Write Pro で開けるのは、4D Write Pro で書き出された HTMLドキュメントに限られます。外部で作成された HTMLドキュメントの読み込みはエラーを生成する可能性があります。

ユーザーインターフェース

4D Write Pro エリア内にて**コンテキストメニュー**プロパティがチェックされている場合、(**4D Write Pro エリアを定義する**を参照して下さい)、アプリケーションモードにおいてユーザーは包括的なコンテキストメニューを使用することができます：



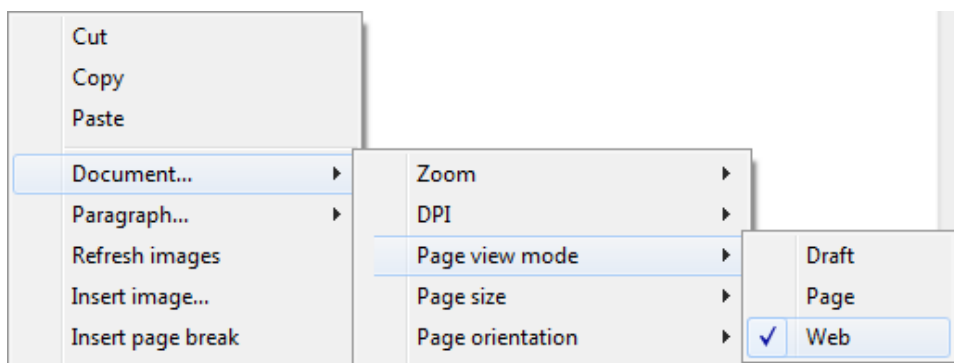
このメニューを通して、提供されているすべての4D Write Pro 機能へとアクセスすることができます。

ビューモードの選択

4D Write Proドキュメントは三種のページビューモードで表示することができます：

- **下書き**：基本的なプロパティを持つ下書きモード
- **ページ** (デフォルト)： "印刷ビュー" モード
- **埋め込み**：埋め込みエリアに最適なビューモードです。このモードでは余白、フッター、ヘッダー、カラム、ページフレーム等は表示されません。
このモードはWeb風のビュー出力を作成するのに也可以使用できます (96 dpi解像度と **HTML WYSIWYG** オプションを選択した場合)。

ページビューモードはエリアポップアップメニューを使用することで設定が可能です：



注： ページビューモードはドキュメントには保存されません。

4Dフォームに埋め込まれたエリアについては、プロパティリストを使用してデフォルトのビューモードを設定することができます。この場合、ビューモードは 4D Write Proフォームオブジェクトのプロパティとして保存されます (より詳細な情報については、[ビュープロパティの設定](#) の章を参照してください)。

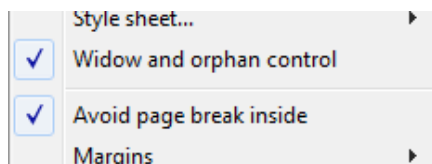
ページビュー機能

ドキュメントがページビューモードのとき、以下のドキュメントプロパティがユーザーに表示されます：

- 印刷範囲を示すページのアウトライン
- ページ幅とページの高さ (デフォルトは21x29.7 cm)

- ページの向き (デフォルト: 縦向き)
- ページの余白 (デフォルト: 2.5cm)

これに加えて、**段落...**サブメニューでは特定の段落プロパティが利用可能です:



- **ウィドウ&オーファンコントロール:** このオプションが段落に対して設定されているとき、4D Write Pro はドキュメント内にてウィドウ (段落の最終行がページ上部に取り残される状態) とオーファン (段落の先頭行がページ下部に取り残される状態) を許可しません。前者の場合には、最終行の一つ前の行を加えた 2行がページ上部に表示されます。後者の場合には、単一の先頭行は次ページへと送られます。
- **段落内の改ページを避ける:** このオプションが段落に対してチェックされている場合、4D Write Pro はその段落が 2枚以上のページに分割されないようにします。

また、**改ページを挿入/カラムを挿入/セクションブレイクを挿入、文書.../ページサイズ、文書.../ページの向き**、といったページ関連の追加のコマンドを使用する事もできます。

注: ドキュメントが **埋め込みモード**または**下書きモード**のときにページプロパティを変更しても見た目上は変化ありませんが、それでも設定は可能です。以下の段落プロパティは、**下書きモード**でも見た目上の変化があります:

- ページの高さ制限 (線が引かれます)
- カラム
- 内側での改ページを避けるプロパティ
- ウィドウ&オーファンコントロール

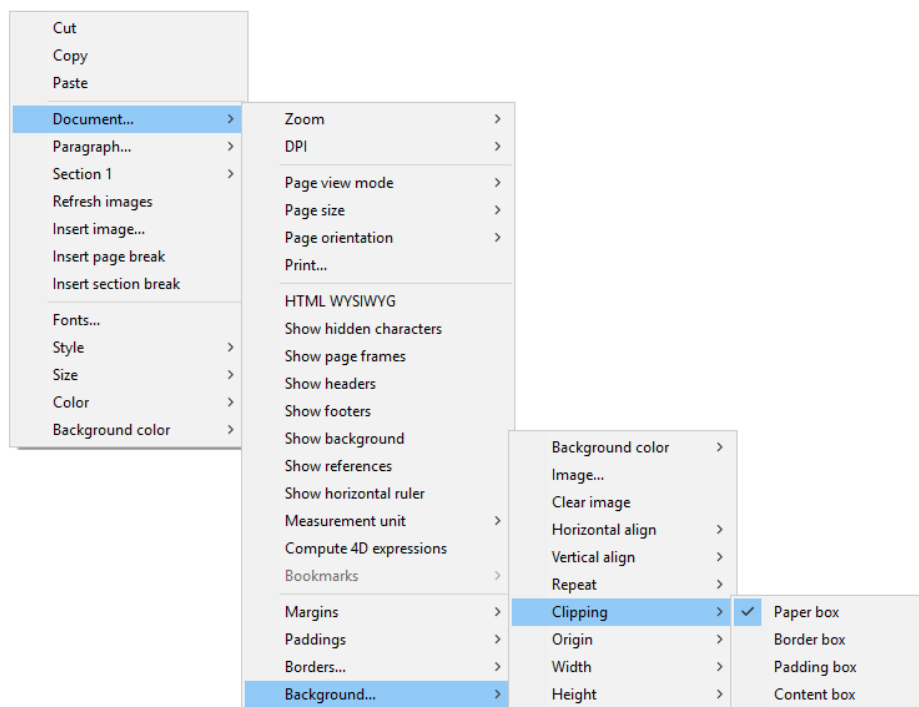
背景

4D Write Pro ドキュメントの背景とドキュメント要素(表、段落、セクション、ヘッダー/フッター等)には、以下のエフェクトを設定することができます:

- カラー
- 境界線
- 画像
- 原点、水平および垂直位置
- 塗りエリア
- 繰り返し

これらの属性は、**WP SET ATTRIBUTES** コマンドあるいは**4D Write Pro標準アクションの使用** コマンドによって、プログラムによってページ上のそれぞれの要素に対して、あるいはドキュメントの背景全体に対して定義することが可能です。利用可能な背景属性と、それがどこに適用可能であるかの一覧については、**4D Write Pro属性** の記事を参照してください。

ユーザーは以下のように、コンテキストメニューを通じて背景属性を変更することが可能です:



背景画像にフルサイズの画像を追加する例については、[こちらのHow Do I \(HDI\) デモ](#)を参照してください。

ヘッダー、フッター、セクションの管理

4D Write Pro ドキュメントはヘッダーとフッターをサポートします。ヘッダーとフッターはセクションと関連しています。

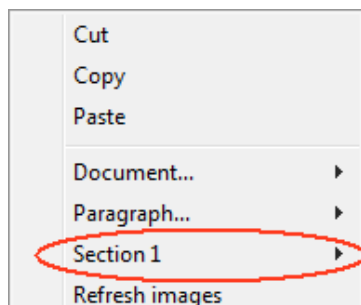
セクションとは、ページレンジによって定義されたドキュメントの一部分で、ページングと属性の設定を独自に持つことができます。ドキュメントには複数のセクションを含めることができます (1から合計のページ数まで)。それぞれのページは一つのセクションにのみ含めることができます。

各セクションに対して、ヘッダーとフッターのセットを定義することができます。

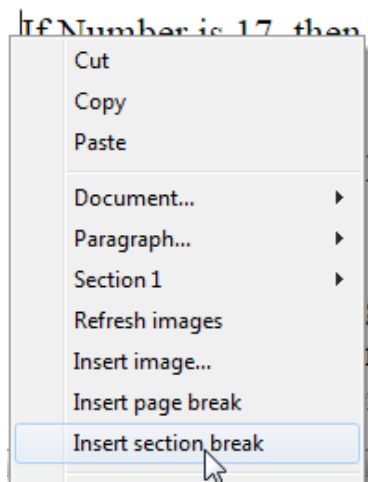
セクションの定義

セクションとは4D Write Proドキュメント内の継続したページのサブセットです。ドキュメントは1つ以上のセクションを含むことができます。ひとつのセクションには何ページでも (単一のページからドキュメント内の総ページ数まで) 含めることができます。また、ひとつのセクションにはカラムを1個から最大20個まで含めることができます。

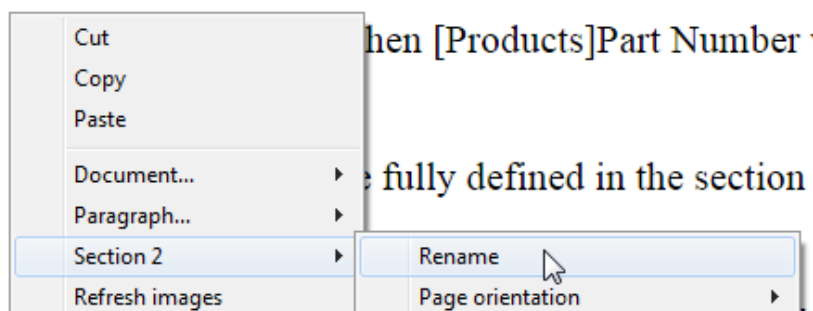
デフォルトで、ドキュメントは**セクション1**と名付けられたひとつのセクションを持ちます。ドキュメント内のどこをクリックしても、4D Write Proコンテキストメニューにそのセクション番号を表示します:



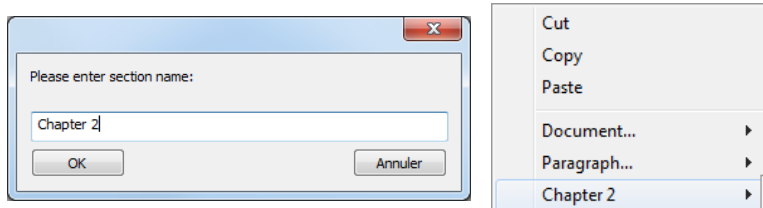
テキストの中にセクションブレイクを追加することで新しいセクションを作成することができます:



セクションブレイクが追加されたとき、コンテキストメニューはそれぞれのセクションに対しインクリメントされた数字を表示します。セクション名は任意に改名することも可能です:



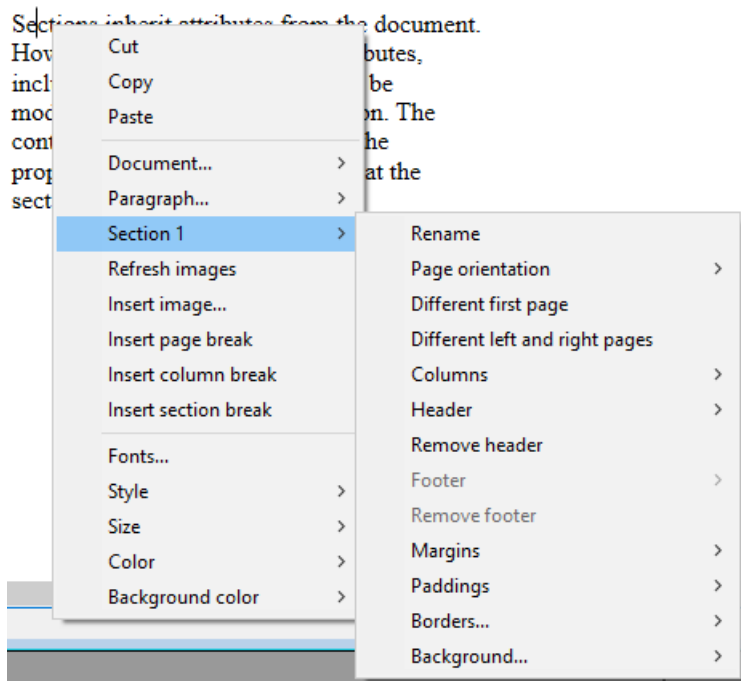
入力した名前はそのセクションの名前として、ドキュメント全体で使用されます:



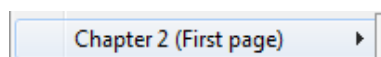
セクションに対し、異なる最初のページ、異なる左/右ページを定義していた場合、そのページタイプもまたメニューに表示されるという点に注意して下さい (後述参照)。

セクション属性

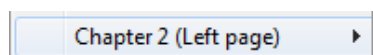
セクションはドキュメント属性を継承しますが、ヘッダーとフッターを含むドキュメント属性は、セクションごとに個別に編集することも可能です。コンテキストポップアップメニューは、セクションレベルで利用可能なプロパティと属性を表示します:



- **ページの向き:** セクションごとに特定のページの向き (横向き/縦向き)を設定することができます。
- **最初のページを個別に設定:** セクションの最初のページに対して異なる属性を設定することができます。この機能は例えば、余白を作成するのに使用することができます。この属性がチェックされている場合、セクションの最初のページはセクションのサブセクションとして管理され、独自の属性を持つことができます。



- **左右のページを個別に設定:** セクションの左ページと右ページで異なる属性を設定することができます。この属性がチェックされていると、セクションの左ページ/右ページはサブセクションとして管理され、それぞれ独自の属性を持つことができます。



- **カラムコマンド:** セクションに対してカラムの数とプロパティを定義することができます。これらのオプションは以下に詳細な説明があります。
- **ヘッダー と フッター コマンド:** これらのオプションを使用すると個別のヘッダーとフッターを定義することができます。これらのオプションは以下に詳細な説明があります。
- **マージン / パディング / 境界線 / 背景:** これらの属性は各セクションに対して個別に設定が可能です。これらの属性のより詳細な情報については、[4D Write Pro属性](#)の記事を参照してください。

ヘッダーとフッターの挿入

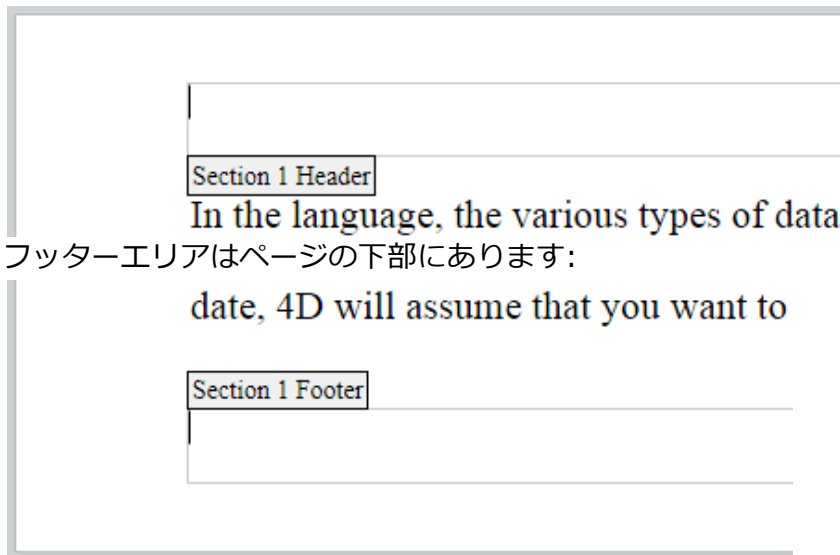
それぞれのセクションには固有のヘッダーとフッターをつけることができます。ヘッダーとフッターはページビューモードが**ページ**のときのみ表示されます。

有効化されたオプションによって、ひとつのセクションにつき最大3つまでヘッダーとフッターを定義することが可能です:

- 最初のページ
- 左ページ
- 右ページ

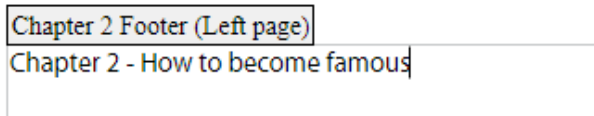
ヘッダーまたはフッターを作成する手順は以下の通りです：

1. まずドキュメントがページビューモードであることを確認してください。
2. 目的のセクションのヘッダーまたはフッターエリアをダブルクリックして、編集モードに入ります。
 - ヘッダーエリアはページの上部にあります：



- フッターエリアはページの下部にあります：

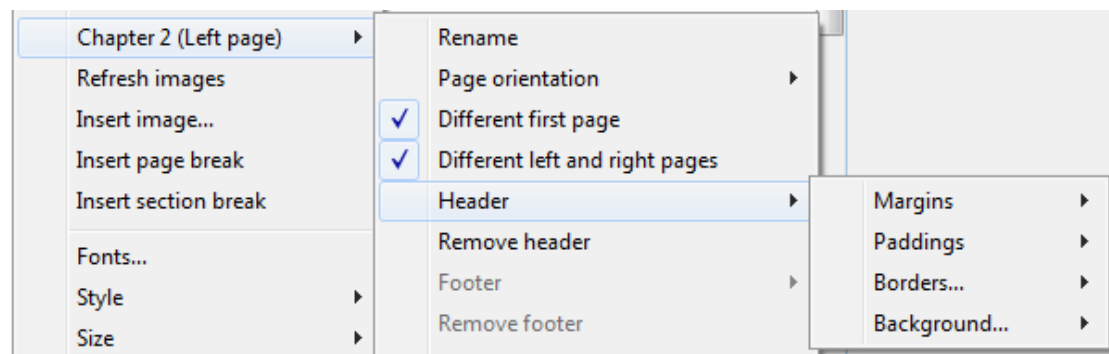
スタティックなコンテンツが入力すると、その入力した内容は各ページのセクションにて自動的に繰り返されます (ただし最初のページが別設定の場合には、そこには反映されません)。



ST INSERT EXPRESSION コマンドを使用することで、ページ番号やページ数などの動的なコンテンツを挿入することができます (より詳細な情報については、**式の挿入**の段落を参照してください)。

注: WP Get header と **WP Get footer** といった特定のコマンドを使用することで、フッターやヘッダーのをプログラミングによって管理することができます。

セクションのヘッダーとフッターが定義できたら、コンテキストメニューを使用してそれらの属性を設定することができます：



マージン、パディング、境界線、そして背景属性についてのより詳細な情報については、**4D Write Pro属性**の章を参照してください。

コンテキストメニューの**ヘッダーを削除**あるいは**フッターを削除**コマンドを選択すると、ヘッダーまたはフッターの定義全体 (コンテンツと属性) を削除することができます。

互換性

4D Write Proは、4D Writeプラグインから変換されたドキュメントのヘッダーとフッターも扱うことができます。

4D Writeプラグインのヘッダーとフッターから引き続きサポートされ変換される式とプロパティは、以下の通りです:

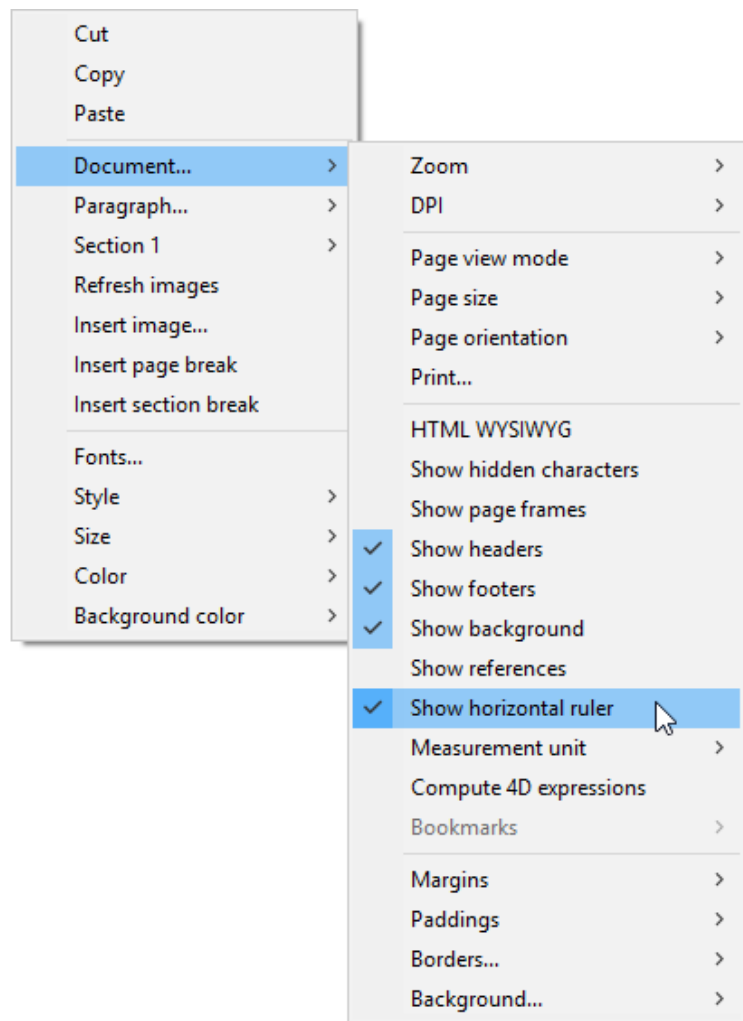
- ページ番号とページ数変数
- 個別の最初のページ
- 個別の左/右ページ

ルーラーの管理

ルーラーは、4D Write Proの全てのビューモードで利用可能で、以下のような特徴を持ちます:

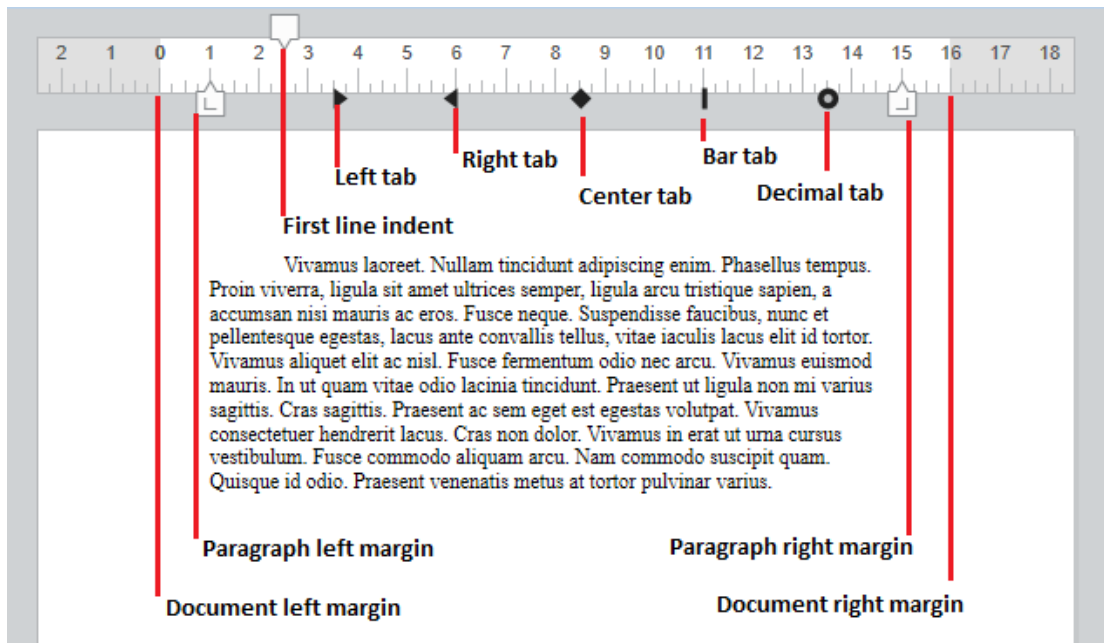
- メモリはcm、mm、inch、あるいはptのうち、4D Write Proドキュメント内で定義されているカレントのレイアウト単位に応じたものになります。単位は、コンテキストメニュー、あるいは `wk layout unit` 属性を使用して変更することができます。
- 最初の行のインデント記号
- 左の段落マージン記号
- 右の段落マージン記号
- タブはルーラーの下に表示
- 左と右のページマージンを表す視覚的なカラーコントラスト

ルーラーの表示状態は、4D Write Proエリアのコンテキストメニュー内の水平ルーラーを表示をチェックするあるいはチェックを外すことで変更可能です:

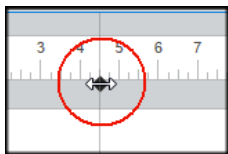


テキストのマージンとインデントを調整する

マージン、インデント、そしてタブの位置は、それぞれに対応する記号をクリックしてドラッグすることで変更することができます：



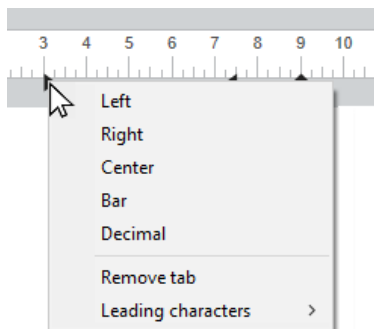
これらの記号の上にマウスをホバーさせたとき、カーソルは変わってそれが移動できることを表します。ドラッグ中は垂直のガイドラインが表示されます：



複数の段落が選択されているとき、マージンあるいはインデント記号をドラッグした場合にはそのマージンとインデントは選択されている全ての段落に適用されます。これらの記号をShiftキーを押しながらドラッグした場合には、選択された段落内でのインデント間・マージン間の現在の間隔を維持します。

タブの管理

ルーラーのコンテキストメニューを使用して、タブの作成、変更、削除をすることができます：



タブを作成するには、ルーラーを右クリックしてそのタイプをコンテキストメニューから選択してください。単一の左クリックは自動的にデフォルトの左タブを作成します。既存のタブを右クリックすることで、コンテキストメニューを使用してそのタブのタイプを変えることもできます。

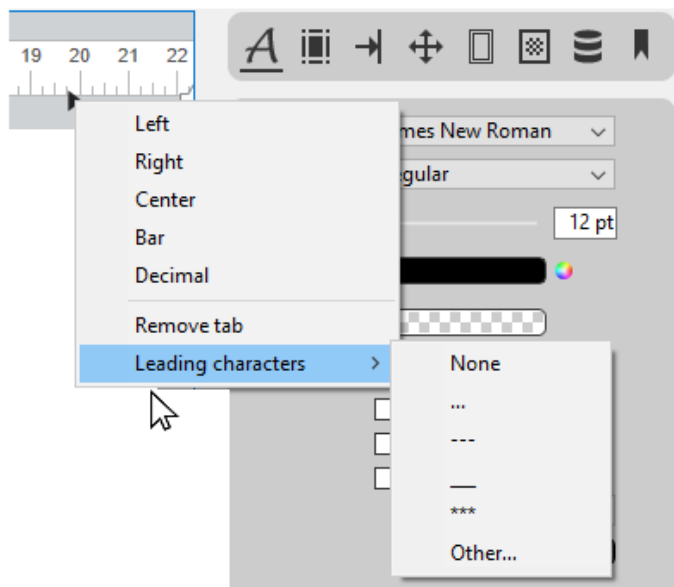
注: タブを削除は既存のタブを右クリックした場合にのみ使用可能です。また、タブをルーラーエリアの外へとドラッグすることで削除することもできます。

先頭文字を定義する

タブの前にくる文字(先頭文字)は、既定の5つの文字から選択するか、使用する特定の文字を指定することで定義する事ができます。既定の文字とは以下のとおりです：

- なし (何も文字が表示されない - デフォルト)
- (点)
- --- (破線)
- _ (アンダースコア)
- *** (アスタリスク)

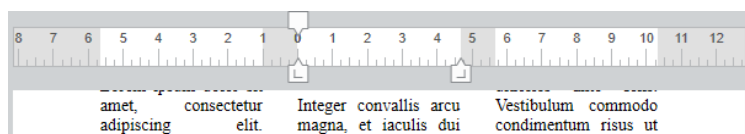
先頭文字は必ずタブの前に表示され、テキストの方向(左から右、あるいは右から左)に従います。**WP SET ATTRIBUTES**、**WP GET ATTRIBUTES**、および**WP RESET ATTRIBUTES** コマンドと **wp tab stop leadings** セレクターを使用する事でプログラミングによって定義する事も可能ですし、右クリックでのコンテキストメニューからユーザーインターフェース経由で定義する事も可能です(以下参照)。



その他...が選択された場合、カスタムの先頭文字を定義可能なダイアログが表示されます。

マルチカラムルーラー

ドキュメントあるいはセクションに対して二つ以上のカラムが定義されている時、ルーラーは各カラムに対するそれぞれの特定のエリアを表示します:



注: マルチカラム昨日は、**埋め込み**ビューモードでは利用できません。

On After Edit イベント

どの種類のタブやマージンコントロールでも、それらが(ドラッグあるいはコンテキストメニューを使用して)移動、追加、削除されたときには、4D Write Proエリアフォームオブジェクトに対して**On After Edit** フォームイベントがトリガーされます。

カラムの管理

4D Write Pro ではドキュメント内にカラムを管理することができます。カラムは最も左のカラムから最も右のカラムへと順番につながっています。言い換えると、テキストを入力していくとき、テキストは左にあるカラムを埋めていき、そのあとすぐ右にあるカラムへと続き、それがページの終わりに達する

まで続いていきます。ページの終わりまで達すると、テキストは次のページへと続いていきます。ページ設定のバランスをとるために、4D Write Pro ではかカラムブレイクを挿入することができます。

カラムはドキュメントレベル(ドキュメント全体に表示される)、あるいはセクションレベル(各セクションはそれぞれ独自のカラム設定を持つことができる)において定義することができます。

注: カラムはページビューモードと下書きビューモードにおいてのみサポートされます(埋め込みビューモードでは表示されず、また**WP EXPORT DOCUMENT** と wk web page complete フォーマットでは書き出されません)。

カラムは以下の方法を用いて設定することも可能です:

- 4D Write Pro エリアコンテキストメニューの**カラム**サブメニュー
- 4D Write Pro 属性(**4D Write Pro属性**参照)
- 4D Write Pro 標準アクション(**4D Write Pro標準アクションの使用**参照)

カラムに対しては、以下のプロパティを設定/取得することができます:

プロパティ	詳細	ドキュメント属性	標準アクション
カラム数	ドキュメント/セクションに対して最大20個までカラムを定義可能です カラム間の間隔をpt、インチ、cm単位で指定。全てのカラムに同じ間隔が適用されることに注意してください。	<u>wk column count</u>	columnCount
カラム間隔	各カラムの幅は、カラム数、ページ幅、そしてカラム間隔に応じて、4D Write Proによって自動的に算出されます。 (読込のみ属性) 各カラムのカレントの(自動算出された)幅。	<u>wk column spacing</u>	columnSpacing
カラム幅	カラムのカレントの(自動算出された)幅。	<u>wk column width</u>	-
区切り線スタイル、カラー、幅	カラムの間には垂直のセパレーター(区切り線)を追加することができます。これらのオプションを	<u>wk column rule style</u> 、 <u>wk column rule color</u> 、 <u>wk column rule width</u>	columnRuleStyle 、 columnRuleColor 、 columnRuleWidth

使用する
と、その区
切り線のス
タイル、カ
ラー、幅を
デザインす
ることがで
きます。



区切り線を
削除するた
めには、ス
タイルから
なしを選ん
でくださ
い。

ブレーク
ーク
を挿
入

wk column
break、WP
INSERT
BREAKも
参照してく
ださい

カラ
ムメ
ニュ
ー

コラムのサ
ブメニュー
を作成しま
す

insertColumnBreak

columns

テーブルの管理

4D Write Proドキュメントにはテーブルを含めることができます。4D Write Proテーブルはプログラミングによって作成・定義されますが、そのコンテンツについては変更可能で、ユーザーによって管理されます。行の高さ、並び、マージン、テキストスタイル、カラー、境界線を含め、様々な4D Write Proテーブル属性が編集可能です。

Name	Date	Value
Alpha	03/04/2017	867
Bravo	05/04/2017	579
Charlie	02/06/2017	501
Delta	06/06/2017	354
Echo	09/04/2017	463
Foxtrot	31/05/2017	912
Golf	17/03/2017	756
Hotel	11/04/2017	437
India	20/04/2017	64
Juliett	05/04/2017	684
Total		5617

テーブルの作成

4D Write Proテーブルは**WP Insert table** コマンドを呼び出すことによって作成されます。その後、**WP Table append row** コマンドを使用して行を追加することができます。

注: ユーザーはセルの範囲をコピー・ペーストすることによってテーブルを作成することができます:

Golf	17/03/2017	756
Hotel	11/04/2017	437
India	20/04/2017	64
Juliett	05/04/2017	684
Total		5617

Golf	17/03/2017
Hotel	11/04/2017

テーブルの編集

セルの中身は**WP Table append row** コマンドを使用することでプログラミングによって追加することができます。

テーブル作成後は、セルの中身はユーザーによってランタイムで編集可能です。ユーザーはセル内をクリックし、通常の段落同様、選択、編集、コピー/ペースト、あるいはテキストまたはピクチャーの削除をすることが可能です。また**Tab**キーを使用してセルを移動していくことも可能です(**Shift+Tab**で逆方向に移動していきます)。

セルの幅は固定であることに注意して下さい。ユーザーがテキストを入力するかピクチャーを貼り付けると、行の高さが必要に応じて自動的に拡張され、テキストは自動的に折り返されます:

Charlie	20/05/2017	844
Delta is the fourth letter of the most known spelling alphabet	06/05/2017	188
Echo	29/05/2017	579

ユーザーはまた、カラム、行、あるいはセルを選択し、ビルトインのポップアップメニューあるいはカスタマイズされたインターフェースを使用して、テキストスタイル、カラー、並び、などの利用可能な属性を適用することができます。4D Write Proではテーブルの任意の部分を選択する複数のコマンドを提供します:

- **WP Table get rows** は行のレンジを取得します。
- **WP Table get columns** はカラムのレンジを取得します。(*)
- **WP Table get cells** はセルのレンジを取得します。

(*) カラムは、それに相当するものがhtmlにはありません。4D Write Proでは、カラムのレンジは実際にはセルのレンジであり、これはつまりカラムはセルと同じプロパティを持つことを意味します。

レンジを選択したら、**WP SET ATTRIBUTES** コマンドを使用して適切な属性を適用することができます。セルの中では、そのコンテンツに応じて、属性は段落、章、あるいはピクチャーへと適用されます。例えば、テーブルあるいはテーブルのセルに対しては、高さ、フォントサイズ、境界線、パディング、などを設定することができます(詳細な情報については、**4D Write Pro属性** の章を参照して下さい)。

異なる属性がテーブルの要素に同時に適用された場合、レンダリングには以下の優先順位が適用されます:

1. テーブルが最初にレンダリングされます
2. 行がレンダリングされます(このときテーブル属性を上書きします)
3. セル/カラムがレンダリングされます(このとき行属性を上書きします)

4D Write Pro ドキュメントの印刷

4D Write Pro ドキュメントの印刷には次の方法があります：

- 4D フォームの一部として
- 独立したドキュメントとして

4D フォーム内ドキュメントの印刷

PRINT SELECTION や **PRINT RECORD** などの標準の 4D印刷コマンドを使用して、どのような 4D フォームでも (プロジェクトフォーム、テーブルフォーム、入力フォーム、出力フォーム)、その中に埋め込まれた 4D Write Pro オブジェクトを印刷することができます。

標準の **印刷時可変** オプションは 4D Write Pro エリアでもサポートされている (*) ため、印刷時にサイズを操作することができます。このオプションがチェックされているとき、マージン (内側・外側) と上マージンは先頭のページにのみ適用されます。マージン (内側・外側) と下マージンは最終ページにのみ適用されます。ドキュメントのページ付けプロパティは無視されます。ウィドウ・オーファンコントロールは無効化され、改ページは適用されません (これらのプロパティはスクリーン上でのページレンダリングとドキュメントのスタンドアロン印刷においてのみ使用されます)。印刷時可変オプションが選択されていた場合、フォームエリアより上に位置しているオブジェクトのみが印刷されます。このオプションについてのより詳細な情報については、デザインリファレンスマニュアルの "**印刷時可変**" を参照してください。

(*) **Print object** と **Print form** コマンドは、このオプションと併用できません。

印刷とビューモード

4D Write Pro エリアに設定した**ビューモード** (**ビュープロパティの設定** 参照) にかかわらず、**Print form** などの 4D 印刷コマンドをしようすると、常に **埋め込み** モードで印刷されます。つまり、4D Write Pro フォームオブジェクトの次のアピアランス設定が無視されます：ページビューモード (常に **埋め込み**)、ヘッダーを表示、フッターを表示、ページ枠を表示 (常に "No")、非表示の文字を表示 (常に "No")。

例題

以下の例は、デフォルト出力フォームに埋め込まれた4D Write Proエリアにおける**印刷時可変**オプションの効果を示したものです。ここでは以下のコードが実行されています：

```
ALL RECORDS([Movies])
ORDER BY([Movies]Title)
PRINT SELECTION([Movies])
```

- 以下は印刷時可変のオプションがチェックされていない (off) 場合の結果です:

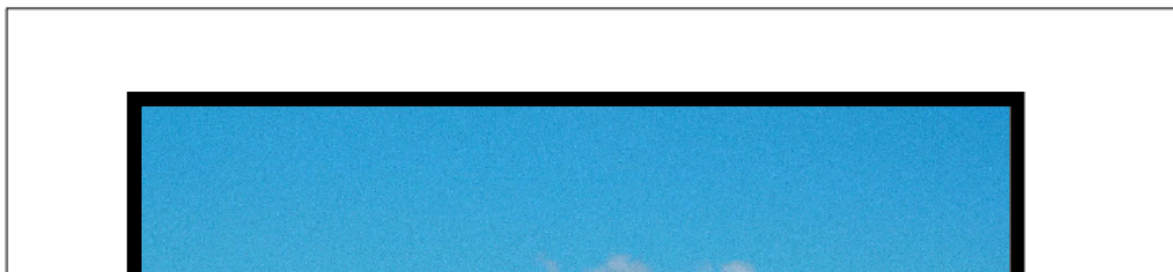
Mountains of the world

Name :

Everest

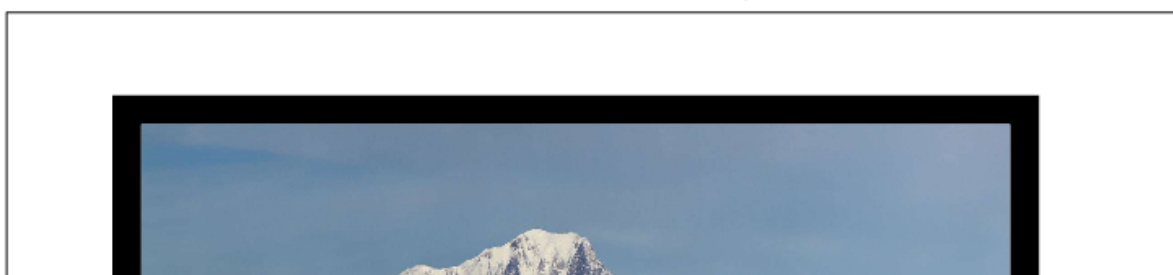
Continent :

Asia



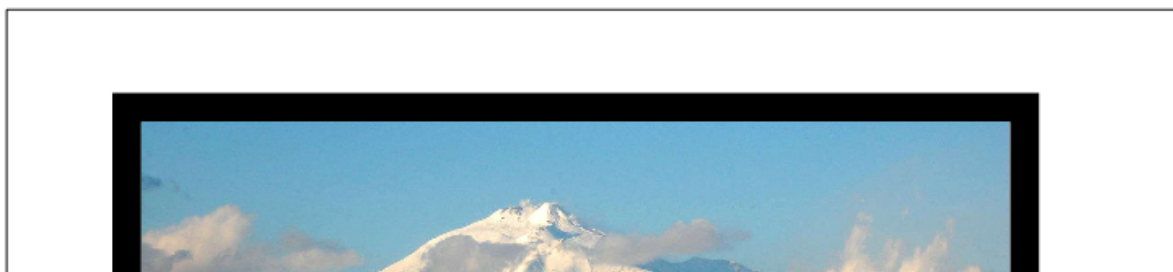
Mont Blanc

Europe



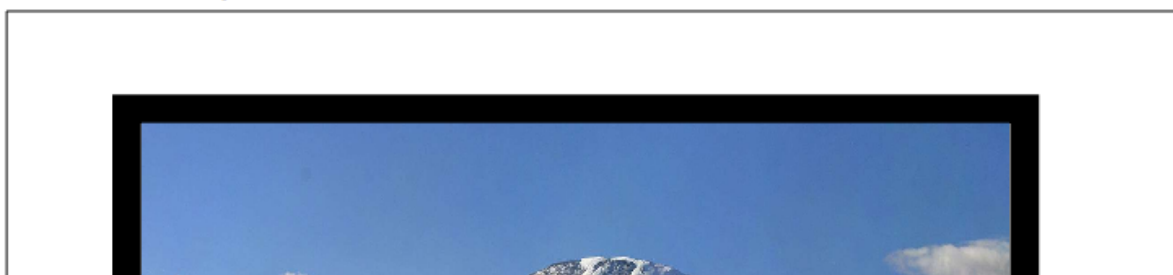
Mount Etna

Europe



Mount Kilimanjaro

Africa



- 以下は印刷時可変のオプションがチェックされている (on) 場合の結果です:

Mountains of the world	
Name :	Continent :
Everest	Asia



Mount Everest, also known in Nepal as Sagarmāthā and in Tibet as Chomolungma, is Earth's highest mountain. It is located in the Mahalangur mountain range in Nepal.[8][9] Its peak is 8,848 metres (29,029 ft) above sea level.[1] It is not the furthest summit from the centre of the Earth. That honour goes to Mount Chimborazo, in the Andes.[10] The international border between China and Nepal runs across Everest's precise summit point. Its massif includes neighbouring peaks Lhotse, 8,516 m (27,940 ft); Nuptse, 7,855 m (25,771 ft) and Changtse, 7,580 m (24,870 ft).

(Sample text source: Wikipedia)

独立したドキュメントの印刷

4D v15 R5以降、4D Write Pro に含まれる新機能によって、独立した 4D Write Pro ドキュメントを印刷できるようになったのに加え、フォーマット、ページの向き、ページ番号と言った標準の印刷オプションの操作が可能になりました。

4D Write Pro コマンド

4D Write Proの印刷機能は、主に次の二つのコマンドによって管理されます。**WP PRINT** と **WP USE PAGE SETUP**です。

- **WP PRINT** は 4D Write Pro ドキュメントの印刷ジョブをローンチするか、ドキュメントをカレントの印刷ジョブに追加します。
- **WP USE PAGE SETUP** はカレントプリンターのページ設定を、4D Write Proドキュメント属性のページサイズとページの向きに変更します。

設定に関する注意:

- 4D Write Proの印刷コマンドは、64-bit版4D用にデザインされた新しい内部アーキテクチャーに基づいています。これらは32-bit版でもサポートはされますが、**OPEN PRINTING JOB**でローンチされた印刷ジョブ内ではサポートされません(**WP PRINT**を参照してください)。
- Windows 7 あるいは Windows Server 2008 R2のマシン上では、*Platform Update for Windows 7* がインストールされており、印刷機能がサポートされていることを確認してください。

標準の 4D コマンド

以下の 4Dコマンドは 4D Write Pro 印刷機能をサポートします:

- **SET PRINT OPTION** と **GET PRINT OPTION**: すべてのオプションが、**WP PRINT** によって印刷される 4D Write Proドキュメントに対してサポートされています。Paper option と Orientation optionに関しては、**WP USE PAGE SETUP** を呼び出してページサイズと向きの属性を 4D Write Pro ドキュメントの設定と同期させる方が効率的な場合もあります。Page range option (15) を使って、印刷するページ範囲を指定することができます。
- **PRINT SETTINGS**: カレントのプリンターに対して印刷設定を設定します。**WP PRINT** がこの後に呼び出された場合、プリンター設定が PRINT SETTINGS ダイアログで変更されていれば、その設定を使用します (ただし余白ページ設定だけは常に4D Write Proドキュメントの設定を使用します)
- **OPEN PRINTING JOB** と **CLOSE PRINTING JOB**: これらのコマンドの間に **WP PRINT** を呼び出すことによって、一つ以上の4D Write Proドキュメントを一件の印刷ジョブに挿入することができます。

4D Write Pro ドキュメントに含める式の制限

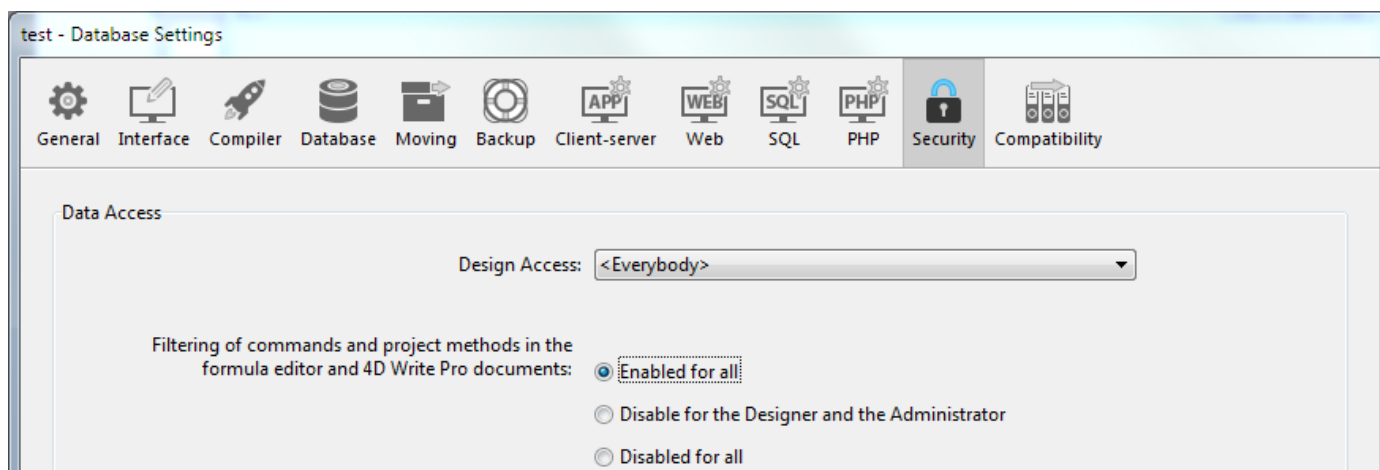
4D Write Pro ドキュメントに使用できる式の制限

4D Write Pro ドキュメントには、変数やフィールドに限らず、フォーミュラやプロジェクトメソッド、4D コマンドなどの動的な 4D 式への参照を含めることができます。ドキュメントの表示および印刷の際に、これらの参照は評価されます。セキュリティ上の理由により、適切でない式が使用されてしまったり、データベースに想定外の変更を加えるなどの影響を与えてしまったりしないよう、動的な式の評価は開発者によって制限する必要があります。例えば、**DELETE SELECTION** のようなコマンドや "DeleteOrders" といったメソッドの実行は制限したほうがよいでしょう。

注: 式を挿入するには、**ST INSERT EXPRESSION** コマンドを使うか、HTML エディターで 4D Write Pro ドキュメントを編集します。

4D のフィルタリング機能をサポート

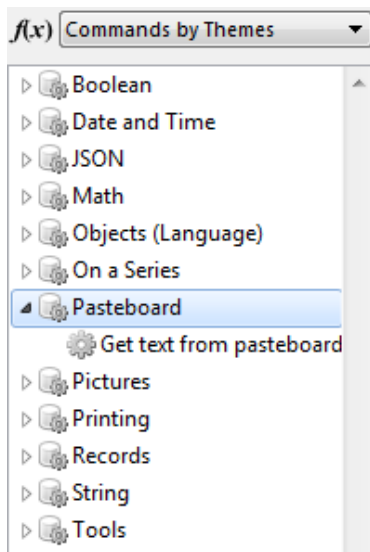
4D v16 より、4D Write Pro ドキュメントに挿入される 4D コマンドやプロジェクトメソッドの評価は、データベースレベルで設定される通常の 4D のフィルタリング機能によって管理されます。このオプションは "データベース設定" ダイアログの**セキュリティ**ページで提供されています:



デフォルトでは、すべてのユーザーに対して制限が課されるため、明示的に許可されたコマンドやメソッド以外を使用すると、評価エラーが発生します。この制限を部分的に緩和することも (Designer と Administrator は制限しない)、誰も制限しないことも可能です。このオプションは 4D データベース全体に対して有効で、ユーザーによるフォーミュラの評価をすべて管理します。このオプションについての詳細は、デザインリファレンスマニュアルを参照ください。

制限が有効になっている場合:

- ユーザーはフォーミュラエディターの右端にあるコマンドリストに表示されているコマンドしか使用できません:



- ユーザーはプロジェクトメソッドを実行できません。4D Write ドキュメントで使用するには、**SET ALLOWED METHODS** コマンドを使って明示的に許可する必要があります。

4D Write Pro ドキュメント内で式の評価中に許可されていない 4D コマンドやメソッドが見つかった場合、値が " ## Error # 48" に置き換わります。

例題

4D Write Pro ドキュメントに次の式を挿入します:

```
ST INSERT EXPRESSION(*;"WriteProArea";"Gender")
```

セキュリティの設定で制限がかかっていると、**Gender** メソッドは評価されません:

```
Dear ## Error # 48 Smith,
```

その場合には次のようなコードを実行します:

```
ARRAY TEXT(aTallow;1)
aTallow{1}:="Gender"
SET ALLOWED METHODS(aTallow)
```

すると **Gender** メソッドが評価され、値が返されます:

```
Dear Mr. Smith,
```

互換性

以前のリリースでは、4D Write Pro ドキュメントのフィルタリング機能は有効になっていませんでした。4D Write Pro ドキュメントで 4D メソッドを参照しているデータベースを 4D v16 以上に変換した場合、これらは評価されなくなり、" ## Error # 48" メッセージが表示されます。このような場合には、**SET ALLOWED METHODS** コマンドを使用して、当該メソッドを許可リストに追加します。

4D Write ドキュメントの読み込み

新しい4D Write Proオブジェクトの主要な機能の一つとして、既存の4D Writeドキュメントの読み込みと変換機能が挙げられます。これによって4D Write プラグインに依存しているアプリケーションを移行させることができます。

互換性に関する注意:

- サポートされるのは、4DWite ドキュメントのうち最後の世代 ("4D Write v7") のドキュメントに限られます。
- 4D Write ドキュメントから 4D Write Pro エリアへのコピー・ペーストは現時点ではまだサポートされていません。4D Writeドキュメントの読み込みは 4D Write Pro ランゲージコマンドの使用によってのみ可能です。

4D Write ドキュメントを読み込むには

4D Write Proオブジェクトに 4D Writeドキュメントを読み込む方法は二つあります:

- ディスク上に保存されている 4D Write ファイルに関しては、**WP Import document** コマンドを使用してください。
- BLOBフィールドに格納されている 4D Write エリアに関しては、**WP New** コマンドを使用してください。

より詳細な情報に関しては、それぞれのコマンドの詳細を参照してください。

4D Write のプロパティで復元されるもの

4D Write プラグインから4D Write Proへの移行を簡単にするために、4D Write の機能のうち、出来る限り多くのものが4D Write Proオブジェクトでサポートされようとしています。

以下の段落では、4D Write プラグインのプロパティのうち、**WP Import document** または **WP New** コマンドを使用して4D Write Proエリアへと読み込んだときに復元されるプロパティについてまとめています。

しかしながら一部の機能において、バグとはみなされない小さな差異が生じることがあります。これらは例えば、4D Write Proで使用するデフォルトの行頭の記号や、下線のタイプの小さな変化などによるものです。

ドキュメント情報

4D Writeプラグイン 4D Write Pro

作成日時	利用可能
修正日時	利用可能
ロック	利用不可(読み込み専用オブジェクトプロパティを使用して下さい)
タイトル	利用可能
件名	利用可能(標準テキストのみ)
作者	利用可能
会社名	利用可能
注記	利用可能

ドキュメントの表示の引数

4D Writeプラグイン

4D Write Pro

ページモード	読み込まれません(コンテキストメニューの、ドキュメント/ページモードを使用して下さい)
ルーラー	読み込まれません(オブジェクトプロパティを使用して下さい)
枠	読み込まれません(オブジェクトプロパティを使用して下さい)
ヘッダー	読み込まれません(オブジェクトプロパティを使用して下さい)
フッター	読み込まれません(オブジェクトプロパティを使用して下さい)
最初のページのヘッダー	利用不可
最初のページのフッター	利用不可
画像	利用不可
縦スクロールバー	読み込まれません(オブジェクトプロパティの縦スクロールバーを使用して下さい)
横スクロールバー	読み込まれません(オブジェクトプロパティの横スクロールバーを使用して下さい)
非表示文字	読み込まれません(オブジェクトプロパティを使用して下さい)
参照	利用不可(ST SET OPTIONS を参照して下さい)
カラム分割	利用可能(v17以降)
縦スプリッター	利用不可
横スプリッター	利用不可
Wysiwyg	利用不可
ズーム	読み込まれません(コンテキストメニューの、ドキュメント/ズームを使用して下さい)

ドキュメントの引数

4D Write プラグイン

4D Write Pro

単位	読み込まれません(コンテキストメニューのドキュメント/計測単位を使用して下さい)
言語	利用不可
カラム数	利用可能(v17以降)
カラム間隔	利用可能(v17以降)
ウィンドウ&オーバーラップ	利用可能
デフォルトのタブ	利用可能
先頭のタブ	利用可能
URLカラー	利用不可
アクセス済みのURLカラー	利用不可

ドキュメントのページ付の引数

4D Writeプラグイン

4D Write Pro

ページ幅	利用可能
ページの高さ	利用可能
最初のページ番号	利用可能(v16以降)
最初のページのヘッダーとフッターを別にする	利用可能(v16以降)
左右のページでヘッダーとフッターが異なる	利用可能(v16以降)
ページバインディング	利用可能(v16以降)
反対側のページ	利用可能(v16以降)
ページマージン	利用可能
ヘッダーの上マージン	利用可能(v16以降)
ヘッダーの下マージン	利用可能(v16以降)
フッターの上マージン	利用可能(v16以降)
フッターの下マージン	利用可能(v16以降)
最初のページの上マージン	利用可能(v16以降)
最初のページの下マージン	利用可能(v16以降)
最初のページのヘッダーの上マージン	利用可能(v16以降)
最初のページのヘッダーの下マージン	利用可能(v16以降)
最初のページのフッターの上マージン	利用可能(v16以降)
最初のページのフッターの下マージン	利用可能(v16以降)
最初のページを右側にする	利用可能(v16以降)

ドキュメントの印刷引数

4D Write プラグイン 4D Write Pro

用紙の種類	利用不可
横向きに印刷	利用不可
幅	利用不可
高さ	利用不可
ユーザー設定マージン	利用不可
倍率	利用不可
X 解像度	利用不可
Y 解像度	利用不可

画像

互換性に関する注意:Windows用では、Mac OSのPICTフォーマットの画像は、4D Write Pro (32-bit 版、64-bit版とも)において読み込みはしますが、レンダリングはされません。この画像タイプが含まれたドキュメントを読み込むには、あらかじめそれらの画像を異なるフォーマットへと変換しておく必要があります。また、PICTフォーマットは廃止予定であり、今後使用されるべきではない点にも注意してください (**PICT フォーマットのピクチャー** を参照してください)。

4D Write プラグイン

X (左)
Y (上)
幅
高さ
ページ番号
背景画像
最初のページには含めない
ビューポートモード(縮小して画像を入れる)
式である
サイズを保持

4D Write Pro

(& position :absolute) (ページ内の画像のみ)
(& position :absolute) (ページ内の画像のみ)
利用可能
利用可能
利用可能(v16 R6 以降)
利用可能(v16 R6 以降)
利用可能(v16 R6 以降)
利用可能
インライン画像で利用可能(v16 R5 以降)
読み込まれません(4D Write Proでは自動)

文字プロパティ

ハイパーリンク

4D Write プラグイン

URL リンク

4D メソッドリンク

ドキュメントを開くリンク

4D Write Pro

利用可能 (*)

利用不可

利用可能(ローカルファイルのURLに変換されます)

(*) 4D v16 R4以降、URLリンクは編集可能な属性に変更されました。詳細については [URL](#) を参照ください。

4D 式

4D Write プラグイン

4D 式

日付と時間

HTML 式

RTF 式

4D Write Pro

利用可能

利用可能

利用不可

利用不可

テキストデータ

4D Write プラグイン

メインテキストデータ

ヘッダーテキストデータ

フッターテキストデータ

4D Write Pro

利用可能

利用可能

利用可能

4D Write Pro標準アクションの使用

概要

4D Write Proエリアのユーザーインターフェースは、多様な**標準アクション**を使用して管理することができます。標準アクションは次のようなものに割り当てることが可能です:

- **メニューバー**あるいは**Dynamic pop up menu** コマンドを通して利用可能なメニューコマンド
- リスト項目、ポップアップ・ドロップダウンリスト、階層ポップアップメニュー
- ボタン、チェックボックス
- あるいは**INVOKE ACTION** コマンドによって実行

インターフェースオブジェクトに割り当てた場合、標準アクションはコンテキストに応じてオブジェクトの有効化/無効化を管理します。詳細な情報については**標準アクション** の章を参照してください。

4D Write Proエリアでは二種類のアクションが利用可能です:

- 4D Write Pro 特有の、4D Write Proエリアに対してのみ使用可能なアクション
- フォント、式、スペルチェック、編集アクションを含む、4D Write Proエリアと他の4Dエリアで使用可能な**その他のアクション**(**標準アクション**参照)。

4D Write Pro アクション

4D Write Proエリアでは以下の標準アクションが利用可能です。

注:

- 4D Write Pro アクションには、対応する定数はありません。
- 自動メニュー/リストを表示するアクションは、メニューコマンド、ポップアップ/ドロップダウンリスト、あるいは階層ポップアップメニューオブジェクトにのみ割り当て可能です(* 参照)。
- チェックボックスと3Dチェックボックスはステータスアクション("section/differentFirstPage"あるいは"visibleHorizontalRuler"など)にのみ割り当て可能です。スリーステートオプションは標準のチェックボックスでのみサポートされます。
- セクションのアクションステータスは、常にその選択されたセクションの実際のステータスと一致します(属性が親セクションあるいはデフォルト属性から継承される場合にはステータスが継承されます)。選択されたセクションのステータス(セクションに現在適用されているステータス)を反映する必要があるからです。しかしながら標準アクションを使用してセクション属性を変更した場合には、選択されたセクションの属性のみが上書きされます。

(*) *Submenu*: 引数を渡さず、メニューコマンド、ポップアップ/ドロップダウンリスト、あるいは階層ポップアップメニューに割り当てて使用した場合、これらのアクションは自動サブメニュー/リストを表示します。例えば、"backgroundColor"アクションをメニューコマンドに割り当てた場合、このメニューコマンドをランタイムで選択するとBackground color サブメニュー項目が表示されます。"zoom"アクションを階層ポップアップメニューに割り当てた場合、このメニューには既定のズーム値の一覧が自動的に含まれます。これらのアクションはボタンに対しては割り当てられない点に注意し

てください。

(**) *showDialog*: アクション名に/showDialog の文字列を追加することで、アクションに割り当てられた標準のダイアログボックスを表示します。例えば、"paragraph/styleSheet/showDialog"と入力することで新しいスタイルシート名を入力する入力ダイアログを開くことができます。

アクション名	ド キ ュ メ ン ト	段 落	画 像	セ ク シ ヨ ン	ヘ ッ ダ ー	フ ッ タ ー	Submenu(*)	showDialog(**)
anchorHorizontalAlign			○				○	
anchorLayout			○				○	
anchorOrigin			○				○	
anchorPage			○				○	
anchorSection			○				○	
anchorVerticalAlign			○				○	
avoidPageBreakInsideEnabled		○						
background	○	○	○	○	○	○	○	
backgroundColor	○	○	○	○	○	○	○	○
backgroundClip	○	○	○	○	○	○	○	
backgroundImage	○	○	○	○	○	○	○	○
backgroundOrigin	○	○	○	○	○	○	○	
backgroundPositionH	○	○	○	○	○	○	○	
backgroundPositionV	○	○	○	○	○	○	○	
backgroundRepeat	○	○	○	○	○	○	○	
backgroundSizeH	○	○	○	○	○	○	○	
backgroundSizeV	○	○	○	○	○	○	○	
bookmark							○	
borderCollapse		○						
borderColor	○	○	○	○	○	○	○	○
borderRadius	○	○	○	○	○	○	○	
borders	○	○	○	○	○	○	○	
borderStyle	○	○	○	○	○	○	○	
columnCount	○			○			○	
columnRuleColor	○			○			○	○
columnRuleStyle	○			○			○	
columnRuleWidth	○			○			○	
columns	○			○			○	
columnSpacing	○			○			○	
borderWidth	○	○	○	○	○	○	○	
deleteStyleSheet		○	○				○	
direction		○					○	
doc	○						○	
dpi							○	
fontStyleWrite							○	

fontSubscript	0							
fontSuperscript	0							
footer							0	
footer/remove						0		
header							0	
header/remove					0			
htmlWYSIWIGEnabled	0							
image		0					0	
image/height		0					0	
image/verticalAlign		0					0	
insertColumnBreak	0		0					
insertImage	0							
insertPageBreak	0							
insertSectionBreak	0							
lineHeight		0					0	
listStartNumber							0	
listStyleImage								0
listStyleType		0					0	
margin	0	0	0	0	0	0	0	
merge		0						
minHeight		0	0				0	
minWidth		0	0				0	
moveToFront			0					
moveToBack			0					
newLineStyleSheet		0					0	
padding	0	0	0	0	0	0	0	
pageMode							0	
pageOrientation				0			0	
pageSize							0	
paragraph		0					0	
print	0							
refreshImages			0					
section				0			0	
section/differentFirstPage				0				
section/differentLeftRightPages				0				
section/name/showDialog				0				0
section/reset				0				
split		0						
styleSheet		0	0				0	0
styleSheets		0	0				0	

anchorOrigin

シンタックス: anchorOrigin?value={paper-box | header-box | footer-box}

アンカーされた画像の絶対位置に対してのエリアの相対的な位置を定義します。このアクションは水平・垂直方向のオフセットを0へとリセットします。画像がページ上で非表示のヘッダーボックス/フッターボックスに対して相対的な位置で定義されているとき、この画像は表示されません。

ページモードのみ有効、インライン画像に対しては有効化されていません。

anchorPage

シンタックス: anchorPage?value={all | current | currentSubSection}

画像が表示されるページを定義します(インライン画像に対しては有効化されていません)。ページモードのみ。

anchorSection

シンタックス: anchorSection?value={all | current}

選択された画像が表示されるセクションを定義します(インライン画像に対しては有効化されていません)。

ページモードのみ。

anchorVerticalAlign

シンタックス: anchorVerticalAlign?value={top | center | bottom}

画像の垂直方向並びを定義します。ページモードの画像はanchorOrigin に対して相対的に、埋め込みモードの画像はレイアウトボックスに対して相対的に設定されます。このアクションは垂直方向のオフセットを0へとリセットします(インライン画像に対しては有効化されていません)。

avoidPageBreakInsideEnabled

シンタックス: avoidPageBreakInsideEnabled

段落内での改ページを許可する、あるいは許可しません。

background

シンタックス: {doc | paragraph | image | section | header | footer}/background

メニューコマンドに対してのみ。全ての背景属性に対してのデフォルトのサブメニューです。

backgroundClip

シンタックス: {doc | paragraph | image | section | header | footer}/backgroundClip?
value={paper-box | border-box | padding-box | content-box}

ターゲットの背景のクリッピングボックスを変更します(デフォルトのターゲットは段落)。

Paper-boxターゲットドキュメントとセクションのみ

backgroundColor

シンタックス: {doc | paragraph | image | section | header | footer}/backgroundColor?value={<Css_color> | transparent}
ターゲットの背景を変更します。

backgroundImage

シンタックス: {doc | paragraph | image | section | header | footer}/backgroundImage?value=none
ターゲットの背景画像を消去します(デフォルトのターゲットは段落)

backgroundOrigin

シンタックス: {doc | paragraph | image | section | header | footer}/backgroundOrigin?value={paper-box | border-box | padding-box | content-box}
ターゲットの背景画像のオリジンボックスを変更します(デフォルトのターゲットは段落)。
Paper-boxはドキュメントとセクションのみターゲットとして指定できます。

backgroundPositionH

シンタックス: {doc | paragraph | image | section | header | footer}/backgroundPositionH?value={left | right | center}
ターゲットの背景画像の水平方向揃えを変更します(デフォルトのターゲットは段落)。
例: *paragraph/backgroundPositionH?value=left*

backgroundPositionV

シンタックス: {doc | paragraph | image | section | header | footer}/backgroundPositionV?value={top | bottom | center}
ターゲットの背景画像の垂直方向揃えを変更します(デフォルトのターゲットは段落)。
例: *paragraph/backgroundPositionV?value=top*

backgroundRepeat

シンタックス: {doc | paragraph | image | section | header | footer}/backgroundRepeat?value={no-repeat | repeat | repeat-x | repeat-y}
ターゲットの背景画像の繰り返しモードを変更します(デフォルトのターゲットは段落)。

backgroundSizeH

シンタックス: {doc | paragraph | image | section | header | footer}/backgroundSizeH?value={<Css_length> | <Css_percentage> | auto | cover | contain}
ターゲットの背景画像の幅を変更します(デフォルトのターゲットは段落)。
例: *paragraph/backgroundSizeH?value=100%. section/backgroundSizeH?value=cover*

backgroundSizeV

シンタックス: {doc | paragraph | image | section | header | footer}/backgroundSizeV?value={<Css_length> | <Css_percentage> | auto}
ターゲットの背景画像の高さを変更します(デフォルトのターゲットは段落)。
例: *paragraph/backgroundSizeV?value=50%*. Ex: *section/backgroundSizeV?value=40pt*

bookmark

シンタックス: bookmark?index=<number>
N 番目のブックマークを選択します。
例: *bookmark?index=2* // 2番目のブックマークを選択

borderCollapse

シンタックス: borderCollapse
選択された段落の境界線と内部余白を折り畳みます。

borderColor

シンタックス: {doc | paragraph | image | section | header | footer}/{inside | outside}/{borderColor | borderColorLeft | borderColorRight | borderColorTop | borderColorBottom | borderColorLeftRight | borderColorTopBottom}?value=<Css_color>
ターゲットの境界線カラーを変更します(デフォルトのターゲットは段落)。
例: *paragraph/borderColorLeft?value=green*

borderRadius

シンタックス: {doc | paragraph | image | section | header | footer}/borderRadius?value=<Css_length>
ターゲットの境界線の半径を変更します(デフォルトのターゲットは段落)。
例: *paragraph/borderRadius?value=4pt*

borders

シンタックス: {doc | paragraph | image | section | header | footer}/{inside | outside/}borders
ターゲットの境界線のデフォルトのサブメニュー

borderStyle

シンタックス: {doc | paragraph | image | section | header | footer}/{inside | outside/}/{borderStyle | borderStyleLeft | borderStyleRight | borderStyleTop | borderStyleBottom | borderStyleLeftRight | borderStyleTopBottom}?value={none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset}
ターゲットの境界線スタイルを変更します(デフォルトのターゲットは段落)。
例: *paragraph/borderStyleLeft?value=double*. Ex: *outside/borderStyle?value=solid* // 選択された段落の外部境界線をソリッドスタイルに設定します。 例: *inside/borderStyle?value=none* // 選択された段落の内部境界線を取り除きます。

borderWidth

シンタックス: {doc | paragraph | image | section | header | footer/}{inside | outside/}{borderWidth | borderWidthLeft | borderWidthRight | borderWidthTop | borderWidthBottom | borderWidthLeftRight | borderWidthTopBottom}?value=<Css_length>

ターゲットの境界線の幅を変更します(デフォルトのターゲットは段落)。

例: *paragraph/borderWidthLeft?value=4pt*

columnCount

シンタックス: {section/}columnCount?value={1<=number<=20}

カレントドキュメントあるいはカレントセクション内にあるカラム数

例: *section/columnCount?value=3*

columnRuleColor

シンタックス: {section/}columnRuleColor?value={CSS color}

ドキュメントのカラムあるいはセクションのカラム間の区切り線のカラー

例: *columnRuleColor?value="#FFFFFF"*

columnRuleStyle

シンタックス: {section/}columnRuleStyle?value={none | dotted | dashed | solid | double | groove | ridge | inset | outset}

ドキュメントのカラムあるいはセクションのカラムの間の区切り線のスタイル

例: *columnRuleStyle?value="solid"*

columnRuleWidth

シンタックス: {section/}columnRuleWidth?value={CSS length}

ドキュメントのカラムあるいはセクションのカラム間の区切り線の幅

例: *columnRuleWidth?value="2pt"*

columns

シンタックス: {section/}columns

メニュー項目に対してのみ: 完全なカラムサブメニューを自動的に作成します。中には全てのカラムアクションサブメニューが含まれます(ただし[insertColumnBreak](#)を除く)

columnSpacing

シンタックス: {section/}columnSpacing?value={CSS length}

カレントのドキュメントあるいはセクション内で二つのカラム間の間隔

例: *columnSpacing?value="1cm"*

deleteStyleSheet

シンタックス: {paragraph | image}/deleteStyleSheet?index={1<= number <= number of target style sheets}

選択されたターゲットの保存されたスタイルシートの中から、N 番目のスタイルシートを取り除きます(デフォルトのターゲットは段落)。メニュー項目名はランタイムでスタイルシート名に置き換えられます。

direction

シンタックス: direction?value={ltr | rtl}

段落の方向

doc

シンタックス: doc

ドキュメントの本文のレイアウトのデフォルトのサブメニュー(埋め込みモード属性あるいはセクションのデフォルト属性、ただし余白については埋め込みモード専用なので除く)

dpi

シンタックス: カレントのビューのdpiを変更します(ピクセル <-> ポイント変換に内部的に使用される wk dpi ドキュメント属性からは独立)

ビューのdpiを変更します。

例: *dpi?value=72*

fontStyleWrite

シンタックス: fontStyleWrite

メニューコマンドに対してのみ。4D Write Proのデフォルトのフォントスタイルサブメニューを表示します。

fontSubscript

シンタックス: fontSubscript

下付き文字のフォント属性を切り替えます。

fontSuperscript

シンタックス: fontSuperscript

上付き文字のフォント属性を切り替えます。

footer

シンタックス: footer

フッターのサブメニューを表示します。

footer/remove

シンタックス: footer/remove

選択されたフッターを削除します。

header

シンタックス: header

ヘッダーサブメニューを表示します。

header/remove

シンタックス: header/remove

選択されたヘッダーを削除します。

htmlWYSIWIGEnabled

シンタックス: htmlWYSIWIGEnabled

html wysiwig モードを切り替えます。

image

シンタックス: image

画像レイアウトサブメニューを表示します。

image/height

シンタックス: image/height?value={<Css_length> | auto}

ターゲットの高さ(画像に対してのみサポートされます)。

例: *image/height?value=50pt* image/widthについては、*width* を参照して下さい。

image/verticalAlign

シンタックス: image/verticalAlign?value={top | middle | bottom | baseline | super | sub}

画像の垂直方向並び。

例: *image/verticalAlign?value=super*

insertImage

シンタックス: insertImage

ピクチャー選択ダイアログボックスを開き、選択されたピクチャー(あれば)を文字としてエリアに挿入します。

insertColumnBreak

シンタックス: insertColumnBreak

セクションにカラムブレイクを挿入します

insertPageBreak

シンタックス: insertPageBreak

セクションの位置に改ページを挿入します。

insertSectionBreak

シンタックス: insertSectionBreak

セクションの位置に改セクションを挿入します。

lineHeight

シンタックス: lineHeight?value={<Css_length> | <Css_percentage>}

段落の行の高さ。

例: *lineHeight?value=120%*

listStartNumber

シンタックス: listStartNumber?value={<number> (longint) | auto}

リストの開始番号。

例: *listStartNumber?value=10*

listStyleImage

シンタックス: listStyleImage/showDialog

リストの行頭記号として表示する画像を選択するファイルダイアログを開きます。またリストスタイルの型をディスクに設定します(これは画像が見つからなかった時に使用されるスタイルです)。

listStyleType

シンタックス: listStyleType?value={none | disc | circle | square | hollow-square | diamond | club | decimal | decimal-leading-zero | lower-latin | lower-roman | upper-latin | upper-roman | lower-greek | decimal-greek | armenian | georgian | hebrew | hiragana | katakana | cjk-ideographic}

段落リストスタイル型

margin

シンタックス: {doc | paragraph | image | section | header | footer/}{inside | outside/}{margin | marginLeft | marginRight | marginTop | marginBottom | marginLeftRight | marginTopBottom}?value={<Css_length> | auto}

ターゲットの余白を変更します(デフォルトのターゲットは段落)。

例: *margin?value=4pt* //全ての段落の余白を4ptに設定します。

例: *outside/margin?value=4pt* //段落の外側の余白を4ptに設定します。

例: *doc/marginLeft?value=1cm* //埋め込みモードで使用されているドキュメントに対して左側の余白を1cmに設定します。

例: *section/marginLeft?value=1cm* //選択されている最初のセクションに対して左側の余白を1cmに設定します。

merge

シンタックス: merge

段落を結合します。

minHeight

シンタックス: {paragraph | image}/minHeight?value=<Css_length>

ターゲットの最小の幅(デフォルトのターゲットは段落)。

例: *paragraph/minWidth?value=50pt*

minWidth

シンタックス: {paragraph | image}/minWidth?value=<Css_length>

ターゲットの最小の高さ(デフォルトのターゲットは段落)。

例: *paragraph/minHeight?value=50pt*

moveToBack

シンタックス: moveToBack

画像を背景へと移動します(インラン画像に対しては有効化されていません)

moveToFront

シンタックス: moveToFront

画像を前面へと移動します(インラン画像に対しては有効化されていません)

newLineStyleSheet

シンタックス: newLineStyleSheet?index={1<=number<=number of paragraph stylesheets | newLineStyleSheet?value=auto

選択された段落の要素は、Returnキーで新規に作成されたあるいは分割された段落に対して、N番目のスタイルシートをスタイルシートとして使用します。value=autoの場合、新規段落は同じスタイルシートを使用します(デフォルト)。(メニュー項目名はランタイムでスタイルシート名に置き換えられます)

padding

シンタックス: {doc | paragraph | image | section | header | footer/}{inside | outside/}{padding | paddingLeft | paddingRight | paddingTop | paddingBottom | paddingLeftRight | paddingTopBottom }?value=<Css_length>

ターゲットのパディングを変更します(デフォルトのターゲットは段落)。

pageMode

シンタックス: pageMode?value={embedded | page | draft}

ビューモードを変更します。

例: *pageMode?value=page*

pageOrientation

シンタックス: {section/}pageOrientation?value={landscape | portrait}

ドキュメントあるいは最初の選択されたページの変更を変更します(セクションと指定されていない場合、変更は全てのセクションに適用されます。指定されている場合には最初に選択

されているセクションに適用されます)。

例: `pageOrientation?value=portrait` //ドキュメント全体に対して方向を縦向きに変更します。

例: `section/pageOrientation?value=landscape` //最初に選択されているセクションに対して方向を横向きに変更します。

pageSize

シンタックス: `pageSize?index=<number>`

ドキュメントのページサイズを変更します。numberでカレントのプリンターで利用可能なページサイズのリストの中のN番目のページサイズを指定します。

例: `pageSize?index=1`

paragraph

シンタックス: `paragraph`

段落レイアウトのデフォルトのサブメニュー

print

シンタックス: `print`

ドキュメントをカレントビューのページモードでレンダリングされている状態で印刷します。

refreshImages

シンタックス: `refreshImages`

URL経由の画像に対して、ネットワークからイメージを再読み込みします。

section

シンタックス: `section`

最初に選択されたセクションのページレイアウト用のデフォルトのサブメニューです。

section/differentFirstPage

シンタックス: `section/differentFirstPage`

セクションに対して異なる1ページ目を持っている(あるいは持っていない)かどうか

section/differentLeftRightPages

シンタックス: `section/differentLeftRightPages`

セクションに対して、異なる左右ページを持っている(あるいは持っていない)かどうか

section/name/showDialog

シンタックス: `section/name/showDialog`

最初の選択されたセクション名を入力するダイアログを表示します。

section/reset

シンタックス: section/reset

最初の選択されたセクションの全ての属性をデフォルトの属性へとリセットします。デフォルトの属性は、背景、境界線、パディング(埋め込みモードと同じ)はドキュメントから継承されており、ページの余白については2.5cmとなっています。最初の選択されたセクションのカラムのプロパティも削除されます(結果的に、デフォルトのセクションプロパティのみが適用されます)。

split

シンタックス: split

段落を分割します。

styleSheet

シンタックス: {paragraph | image}/styleSheet?value={1<=number<=number of target style sheets}

選択されたターゲットの要素に対し、N番目のスタイルシートを適用します。ダイアログを使用すると、新しいスタイルシートは最初に選択されたターゲット(デフォルトのターゲットは段落)のカレントの属性を使用します。

例: *styleSheet?value=1* //最初の段落スタイルシートを選択された段落に適用し、それを段落スタイルシートとして設定します。

styleSheets

シンタックス: {paragraph | image}/styleSheets

ターゲットのスタイルシートを編集するためのデフォルトのサブメニュー

textAlign

シンタックス: textAlign?value={left | right | center | justify | initial}

段落のテキストの並び(initialに対してはrtlではrightを、ltrではleftを使用)

textIndent

シンタックス: textIndent?value=<Css_length> (may be negative)

段落のテキストのインデントを開きます(最初の行のインデント)。

例: *textIndent?value=2cm*

textLinethrough

シンタックス: textLinethrough

メニューコマンドのみ。打ち消し線のサブメニュー(スタイルとカラー付き)

textLinethroughColor

シンタックス: textLinethroughColor?value={<Css_color> | currentColor}

例: *textLinethroughColor?value=red*

textLinethroughStyle

シンタックス: textLinethroughStyle?value={none | solid | dotted | dashed | double | semi-transparent | word}

例: *textLinethroughStyle?value=dotted*

textShadow

シンタックス: textShadow

メニューコマンドのみ。テキストシャドウのデフォルトのサブメニュー(カラーとオフセット付き)

textShadowColor

シンタックス: textShadowColor?value={<Css_color> | none}

文字のシャドウのステータスあるいはカラーを変更します。

例: *textShadowColor?value=green*

textShadowOffset

シンタックス: textShadowOffset?value=<Css_length>

pt単位の長さのみがサポートされます。

例: *textShadowOffset?value=2pt*

textTransform

シンタックス: textTransform?value={none | capitalize | uppercase | lowercase | small-uppercase}

テキストをフォーマットします

例: *textTransform?value=capitalize*

textUnderline

シンタックス: textUnderline

メニューコマンドのみ。下線のサブメニュー(スタイルとカラー付き)

textUnderlineColor

シンタックス: textUnderlineColor?value={<Css_color> | currentColor}

例: *textUnderlineColor?value=red*

textUnderlineStyle

シンタックス: textUnderlineStyle?value={none | solid | dotted | dashed | double | semi-transparent | word}

例: *textUnderlineStyle?value=dotted*

updateStyleSheet

シンタックス: {paragraph | image}/updateStyleSheet

最初の選択されたターゲットのスタイルシートを、選択されたターゲットのカレントの属性で置き換えます(デフォルトのターゲットは段落)。

userUnit

シンタックス: userUnit?value={cm | mm | in | pt}

ドキュメントのレイアウトの単位を変更します(変更はルーラーに反映されます)。

verticalAlign

シンタックス: verticalAlign?value={top | middle | bottom}

段落の垂直方向の並び。

例: *verticalAlign?value=top* 効果は段落の*minHeight* の値に応じます。

view

シンタックス: view

メニューコマンドのみ。ビュー設定のデフォルトのサブメニュー。

visibleBackground

シンタックス: visibleBackground

背景を表示/非表示にします。

visibleFooters

シンタックス: visibleFooters

フッターを表示/非表示にします。

visibleHeaders

シンタックス: visibleHeaders

ヘッダーを表示/非表示にします。

visibleHiddenChars

シンタックス: visibleHiddenChars

非表示の文字を表示/非表示にします。

visibleHorizontalRuler

シンタックス: visibleHorizontalRuler

垂直ルーラーを表示/非表示にします。

visiblePageFrames

シンタックス: visiblePageFrames

ページ枠を表示/非表示にします。

widowAndOrphanControlEnabled

シンタックス: widowAndOrphanControlEnabled

ウィドウ&オーファンコントロールモードを切り替えます。

width

シンタックス: {paragraph | image}/width?value=<Css_length> | auto

ターゲットの幅(段落と画像のみサポートされます。デフォルトのターゲットは段落)。

例: *image/width?value=50pt*

zoom

シンタックス: zoom?value={25% <= percentage in Css <= 400%}

ビューのズームを変更します。

例: *zoom?value=120%*

その他のアクション

4D フォームで使用可能な他の複数の標準アクションも、4D Write Proエリアで使用することができます:

- **編集アクション**、例えばコピー/ペーストアクションなど
- **フォント** アクション、例えばfontBold あるいは fontSize など
- **動的な式** アクション。挿入された式などを管理することができます。
- **スペルチェック** アクション。

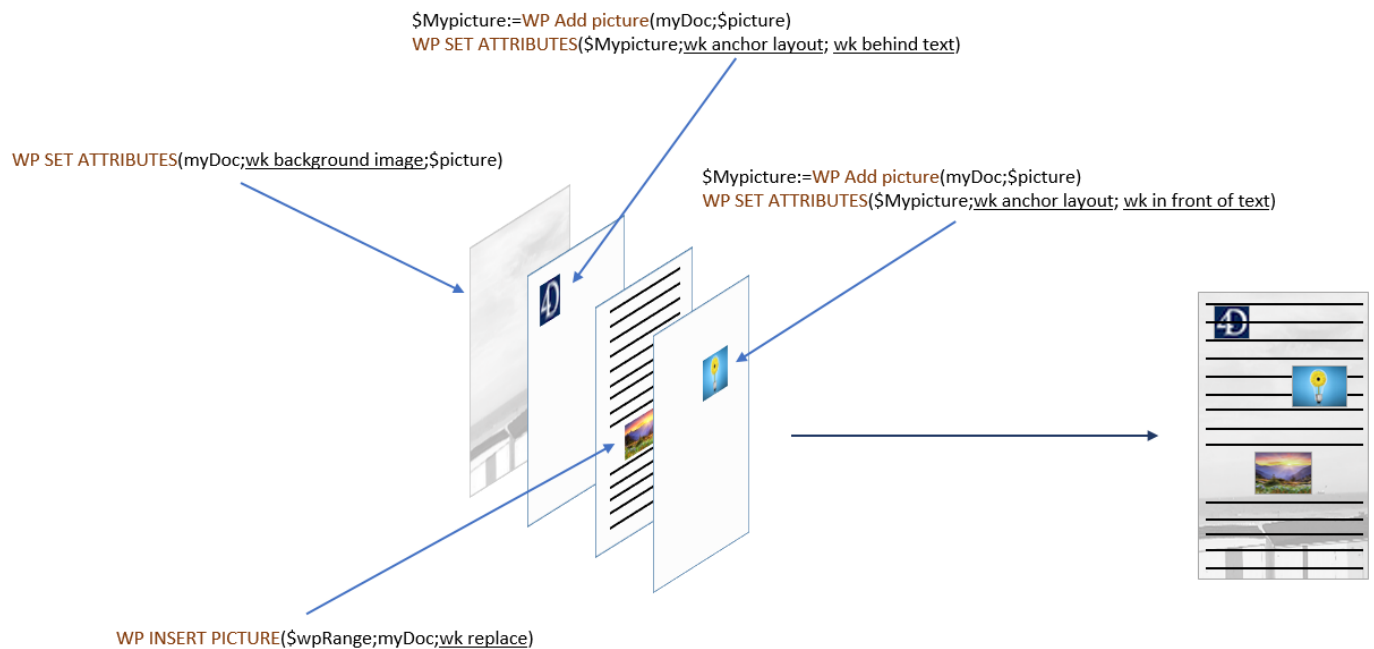
📄 ピクチャーを管理する

ピクチャーの追加

4D Write Pro ドキュメントへのピクチャーの追加は、必要に応じて複数の方法で行うことができます:

- **背景画像**を追加するためには、**WP SET ATTRIBUTES** コマンドと wk background image 属性あるいは wk background image url 属性を使用します。
- **インライン画像**(文字のようにテキストフローに挿入された画像)を追加するためには、**WP INSERT PICTURE** あるいは **ST INSERT EXPRESSION** コマンドを使用します。
- **アンカー画像**をページ内(テキストの後ろあるいは前面)に追加するためには、**WP Add picture** コマンドを使用して wk anchor layout 属性を設定します。

ピクチャーの追加方法によって画像が位置するレイヤーが決定されます。以下の図を参照してください:



背景画像

ピクチャーは4D Write Pro ドキュメントあるいはドキュメント要素(テーブル、段落、セクション、ヘッダー/フッター等)の、背景画像として設定することが可能です。

以下の図は、ピクチャーをドキュメントの背景として使用する二つの異なる方法を示したものです:



Background image
clipped to Border box.



Background image
clipped to Paper box.

背景画像はプログラミングによって、あるいはコンテキストメニューによって設定することが可能です。詳細な情報については、[4D Write Pro エリアを使用する](#) の記事内の[背景](#) の章を参照してください。

アンカーされたピクチャーの位置と表示

アンカーされたピクチャーは絶対位置で、テキストの前あるいは後ろに追加されます。また、ページあるいはドキュメントの特定のパーツ(ヘッダー、フッター、セクションなど)にアンカーすることもできます。ピクチャーに絶対位置を設定するためには、[WP Add picture](#) and [WP SET ATTRIBUTES](#) コマンドを使用します。

アンカーされた画像の位置は以下の特定の属性あるいは標準アクションを使用することで変更可能です:

プロパティ(定数)

[wk anchor layout](#)

[wk anchor horizontal offset](#)

[wk anchor horizontal align](#)

[wk anchor vertical offset](#)

[wk anchor vertical align](#)

[wk anchor origin](#)

[wk anchor page](#)

[wk anchor section](#)

標準アクション

[anchorLayout](#)

[anchorHorizontalAlign](#)

[anchorVerticalAlign](#)

[anchorOrigin](#)

[anchorPage](#)

[anchorSection](#)

[moveToBack](#)

[moveToFront](#)

アンカーされたピクチャーは全てページビューモードでのみ表示されます。以下の場合には表示されません:

- ピクチャーが非表示のヘッダーあるいはフッターにアンカーされている
- ビューモードが下書きモードである
- 中央揃えになっているかセクションにアンカーされている状態で、**HTML WYSIWYG**で表示オプションがチェックされている
- "背景を表示"オプションが選択されていない

ピクチャープロパティ

全てのピクチャーには高さ、幅、境界線、などといったプロパティ(属性)があります。これら4D Write Pro ランゲージ(**WP GET ATTRIBUTES** あるいは **WP SET ATTRIBUTES**)あるいは標準アクションを用いて取得あるいは設定することが可能です。

- ピクチャーに対して使用可能なプロパティの完全な一覧は、**4D Write Pro属性** のページに記載されています。
- **画像** の章にはピクチャー専用の属性が含まれています。
- **4D Write Pro標準アクションの使用** のページにも利用可能な画像プロパティの一覧があります。

ピクチャー参照またピクチャーURL

ピクチャー参照(ピクチャー変数、フィールド、式)を使用する事もできますし、ピクチャーURL(ピクチャーのローカルあるいはネットワークアドレスを表すテキスト)を使用する事もできます。

2組の属性を使用する事で、ピクチャー参照を使用したいか、ピクチャーURLを使用したいかを定義する事ができます:

- wk image、wk background image、wk list style image: ピクチャー参照を設定あるいは取得します。
これらの属性を使用してピクチャーを取得した場合、ピクチャーの定義方法に関係なく、4Dピクチャーを受け取ります。
- wk image url、wk background image url、wk list style image url: ピクチャーURLを設定あるいは取得します。
これらの属性を使用してピクチャーを取得した場合、テキストのみを受け取ります。ピクチャーが参照を通して定義されていた場合(例: ピクチャー変数)、ローカルのURI と、そのあとにBase64の画像を受け取ります。

例:

```
$range:=WP Get selection(WPArea) //ユーザーが選択した画像を取得
$range:=WP Create picture range($range) //レンジを作成
$url:="http://doc.4d.com/image/logo/poweredby4D_web.png"

WP SET ATTRIBUTES($range;wk image;$url) //URLから画像参照を設定

//画像を取得
C_PICTURE(vPictureGet)
WP GET ATTRIBUTES($range;wk image;vPictureGet) //vPictureGet に画像が格納される
C_TEXT(vPictureURLGet)
WP GET ATTRIBUTES($range;wk image url;vPictureURLGet) //vPictureURLGet=$url
```

ピクチャーの取得

以下のコマンドを使用するとピクチャーが返されます:

- **WP Create picture range** - インライン画像に対してのみ適用可能
- **WP Get selection** - ユーザーが選択した画像に対してのみ適用可能

マウス/キーボードアクション

ピクチャーはマウスあるいはキーボードで編集することができます。利用可能なアクションは以下の通りです:

- **選択**

- 前面レイヤーにあるピクチャーはクリックすることで選択することが可能です。
- 背景画像はCTRL+クリック(Windows)あるいはCommand+クリック(Mac)で選択することが可能です。
- 絶対位置をもつピクチャーを一つだけ選択することができます。複数選択することはできません。

- **移動**

- ピクチャーはマウスを使用して移動およびリサイズすることができます(インライン画像はドキュメントの外へとドラッグすることはできません)。

- **挿入**

- 絶対位置を持つピクチャーが下書きモードに貼り付け場合、カレントのキーボードセレクションの先頭にインラインとして表示されます。

- **カット/貼り付け/コピー**

- ピクチャーは選択されていればクリップボードへとコピーしたり、カット/貼り付けなどを行うことができます。
- 絶対位置を持つピクチャーをページモードで貼り付けすると、そのピクチャーは選択されている最初のページに貼り付けられ、元のアンカー設定に関係なく、そのページへとアンカーされます。貼り付けられたピクチャーは元のオフセットは保持しますが、元のモードとは異なるページモードへと貼り付けられた場合には保持しません(例えばピクチャーがページモードでコピーされ、埋め込みモードで貼り付けされた場合には、オフセットは0へとリセットされます)。

- **削除** -ピクチャーを選択した状態で以下の操作を行うと、ピクチャーを削除することができます:

- deleteキー
- クリアアクションのキーボードショートカット

4D Write Proランゲージ

-  4D Write Pro オブジェクトについて
-  オブジェクト(フォーム)テーマのコマンドの使用
-  スタイル付テキストテーマのコマンドの使用
-  ドキュメントのコンテンツにプログラムでアクセスする
-  WP Add picture
-  WP CREATE BOOKMARK
-  WP Create paragraph range
-  WP Create picture range
-  WP Create range
-  WP Create table range New 17.0
-  WP DELETE BOOKMARK
-  WP EXPORT DOCUMENT
-  WP EXPORT VARIABLE
-  WP GET ATTRIBUTES
-  WP Get body
-  WP Get bookmark range
-  WP GET BOOKMARKS
-  WP Get element by ID New 17.0
-  WP Get elements New 17.0
-  WP Get footer
-  WP Get frame
-  WP Get header
-  WP Get page count
-  WP Get position New 17.0
-  WP Get selection
-  WP Get text New 17.0
-  WP Import document
-  WP INSERT BREAK Updated 17.0
-  WP INSERT DOCUMENT
-  WP INSERT PICTURE
-  WP Insert table
-  WP Is font style supported
-  WP New
-  WP PRINT
-  WP RESET ATTRIBUTES
-  WP SELECT
-  WP SET ATTRIBUTES
-  WP SET FRAME

- WP SET TEXT New 17.0
- WP Table append row
- WP Table get cells Updated 17.0
- WP Table get columns Updated 17.0
- WP Table get rows Updated 17.0
- WP USE PAGE SETUP
- 4D Write Pro
- 4D Write Pro属性

4D Write Pro オブジェクト

4Dアプリケーションにおいて、4D Write Proドキュメントは`object` 型の変数を使用して管理されます(詳細な情報に関しては、[C_OBJECT](#) コマンドを参照して下さい)。

4D Write Pro オブジェクトを一度メモリで参照した後は、以下の様な事をすることができます:

- "Write Pro エリア"というフォームオブジェクトを使用して、4Dフォームを通して表示・編集ができます。その場合、フォームエリアはオブジェクト変数と同じ名前である必要があります([4D Write Pro エリアを定義する](#)を参照して下さい)。
- 複数のテーマのコマンドによって、プログラムで管理することができます(以下参照)。
- [WP EXPORT DOCUMENT](#) コマンドを使用して、ディスク上にHTMLドキュメントとして書き出す事ができます。

4D Write Pro オブジェクトを扱う事ができるコマンド

4D Write Pro オブジェクトは、複数のテーマの4Dコマンドを使用して管理することができます:

- **4D Write Pro** テーマの4D Write Pro専用のコマンド。これらのコマンドはこの章内にて詳細な説明があります。
- [オブジェクト\(フォーム\)](#) コマンド。フォーマット機能・オブジェクトプロパティ機能を管理します。詳細な情報に関しては、[オブジェクト\(フォーム\)テーマのコマンドの使用](#) の章を参照して下さい。
- [スタイル付きテキスト](#) コマンド。コンテンツ機能を管理します。詳細な情報に関しては、[スタイル付テキストテーマのコマンドの使用](#) の章を参照して下さい。
- [オブジェクト\(ランゲージ\)](#) コマンド。4D Write Pro オブジェクト属性を管理します。[4D Write Proドキュメントを4Dオブジェクトフィールドに保存する](#) の章を参照してください。

オブジェクト(フォーム)テーマのコマンドの使用

以下のコマンドは4D Write Pro フォームオブジェクトをサポートします:

コマンド

補足

OBJECT DUPLICATE

OBJECT Get auto spellcheck/OBJECT SET
AUTO SPELLCHECK

OBJECT Get border style/OBJECT SET
BORDER STYLE

OBJECT Get context menu/OBJECT SET
CONTEXT MENU

OBJECT GET COORDINATES/OBJECT SET
COORDINATES

OBJECT Get data source/OBJECT SET
DATA SOURCE

OBJECT GET DRAG AND DROP
OPTIONS/OBJECT SET DRAG AND DROP
OPTIONS

OBJECT Get enabled/OBJECT SET
ENABLED

OBJECT Get enterable/OBJECT SET
ENTERABLE

OBJECT GET EVENTS/OBJECT SET
EVENTS

OBJECT Get focus rectangle
invisible/OBJECT SET FOCUS RECTANGLE
INVISIBLE

OBJECT Get font/OBJECT SET FONT

カレントセクションがあればそこに適用

OBJECT Get font size/OBJECT SET FONT
SIZE

カレントセクションがあればそこに適用

OBJECT Get font style/OBJECT SET FONT
STYLE

カレントセクションがあればそこに適用

OBJECT Get horizontal
alignment/OBJECT SET HORIZONTAL
ALIGNMENT

カレントセクションがあればそこに適用。4D
Write Proエリアの[wk.justify](#)定数をサポート

OBJECT GET RESIZING
OPTIONS/OBJECT SET RESIZING
OPTIONS

OBJECT SET COLOR

カレントセクションがあればそこに適用

OBJECT GET RGB COLORS/OBJECT SET
RGB COLORS

カレントセクションがあればそこに適用

OBJECT Get type

OBJECT Get vertical alignment/OBJECT
SET VERTICAL ALIGNMENT

段落の垂直方向の行揃え:段落の高さが段落のテキ
ストの高さより高い場合にのみ効力を持ちます。

OBJECT Get visible/OBJECT SET VISIBLE

OBJECT Is styled text

trueを返します

OBJECT MOVE

**OBJECT GET SCROLL POSITION/OBJECT
SET SCROLL POSITION**

OBJECT GET SUBFORM CONTAINER SIZE

OBJECT Get name

OBJECT Get pointer

上記にないOBJECT コマンドはどれも4D Write Pro エリアに対しては使用できません。

例題

4D Write Pro エリアの垂直方向の行揃えを中央に設定したい場合を考えます：

Case of

:(**Form event**=On Clicked)

OBJECT SET HORIZONTAL ALIGNMENT(*;"myObject";Align center)

hAlignLeft:=0 //次回表示のためにプロパティを保存

hAlignRight:=0

End case

スタイル付テキストテーマのコマンドの使用

以下のコマンドは4D Write Pro オブジェクトをサポートします。これらは4D Write Pro フォームオブジェクト名(文字列)あるいは4D Write Pro ドキュメント参照(*Object* 型の変数)の両方に対して使用できます:

コマンド	補足
------	----

ST COMPUTE EXPRESSIONS	
-------------------------------	--

ST FREEZE EXPRESSIONS	
------------------------------	--

ST GET ATTRIBUTES / ST SET ATTRIBUTES	
--	--

ST Get content type	画像コンテンツの型に新しいタイプ(6)が追加されました
----------------------------	-----------------------------

ST Get expression / ST INSERT EXPRESSION	4D v16 より、4D Write Pro ドキュメントの式にはデフォルトで制限がかけられており、明示的に許可する必要があります。詳細については、 4D Write Pro ドキュメントに含める式の制限 を参照ください。v16 R5 以降、画像式を挿入することが可能になりました。
---	--

ST GET OPTIONS / ST SET OPTIONS	
--	--

ST Get plain text / ST SET PLAIN TEXT	4D v17以降、 WP Get text and WP SET TEXT コマンドを使用することで4D Write Pro ドキュメントの標準テキストを取得/設定することができるようになりました。
--	---

ST Get text / ST SET TEXT	
----------------------------------	--

ST GET URL / ST INSERT URL	4D v16 R4 以降、画像やセルを含むすべての4D Write Proレンジで、 WP SET ATTRIBUTES および WP GET ATTRIBUTES コマンドと <code>wk link url</code> 属性を使って、URLを取り扱うことができます。
-----------------------------------	---

例題

4D Write Pro エリアの選択範囲を変数の中身で置き換えたい場合を考えます:

```
C_TEXT(fullName)
```

```
Case of
```

:(Form event=On Clicked)

ST INSERT EXPRESSION(myArea;"fullName";ST Start highlight;ST End highlight)

End case

式の挿入

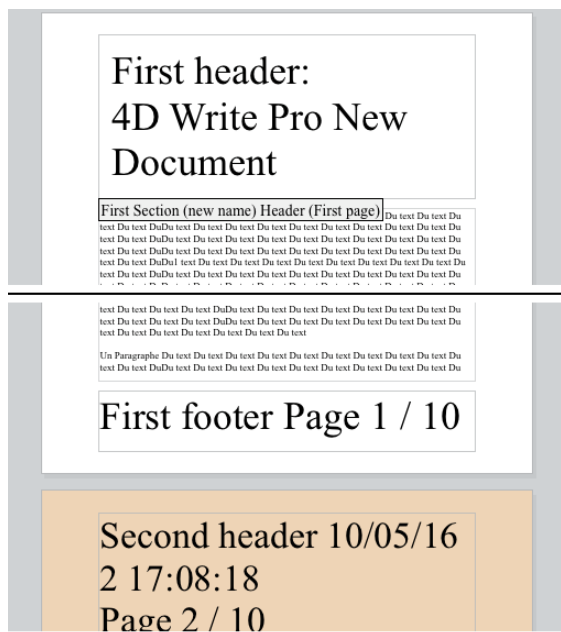
ST INSERT EXPRESSION コマンドを使って、ドキュメントやページの属性に関連づいた式を挿入することができます。

式のシンタックス	有効範囲	型	説明
\$wp_title	ドキュメント全域	テキスト	wk title 属性が定義するタイトル
\$wp_author	ドキュメント全域	テキスト	wk author 属性が定義する著者名
\$wp_subject	ドキュメント全域	テキスト	wk subject 属性が定義する題名
\$wp_company	ドキュメント全域	テキスト	wk company 属性が定義する会社
\$wp_notes	ドキュメント全域	テキスト	wk notes 属性が定義するコメント
\$wp_dateCreation	ドキュメント全域	日付	wk date creation 属性が定義する作成日
\$wp_dateModified	ドキュメント全域	日付	wk date modified 属性が定義する最終更新日
\$wp_pageNumber	ヘッダー & フッター (それ以外はエラー)	倍長整数	ドキュメントの先頭 (デフォルト) あるいはセクションの先頭からのページ数
\$wp_pageCount	ヘッダー & フッター (それ以外はエラー)	倍調整数	ページ総数

式を挿入するには、まずカーソルが任意のエリア (ヘッダー、フッター、ドキュメント本文) にあることを確認し、ST INSERT EXPRESSION コマンドをコールします。例えば、選択したフッターエリアにページ番号を挿入する場合:

ST INSERT EXPRESSION("*;4DWPArea";"\$wp_pageNumber")

例えば、下のようなドキュメントデザインを定義することができます：



画像式の挿入

ST INSERT EXPRESSION コマンドを使用することで、4D Write Pro エリアにピクチャーを返す4D 式を挿入することができます。式には変数、フィールドあるいは4D メソッドが使用可能です(*)。

(*)4D メソッドを式として呼び出すためには、そのメソッドが事前に明示的に許可されている必要があるということに注意してください([4D Write Pro ドキュメントに含める式の制限](#)を参照して下さい)。

例題：

//ピクチャー変数を挿入

ST INSERT EXPRESSION("*;WParea";"vPicture";ST Start highlight;ST End highlight)

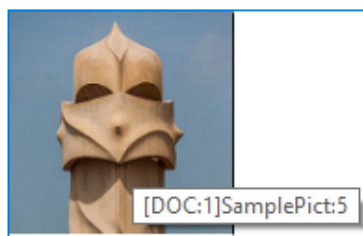
//フィールドを挿入

ST INSERT EXPRESSION("*;WParea";"[DOC]SamplePic";ST Start highlight;ST End highlight)

//4D メソッドを挿入

ST INSERT EXPRESSION("*;WParea";"M_ComputeChart";ST Start highlight;ST End highlight)

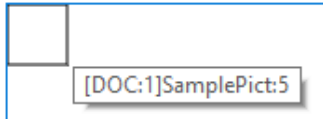
ピクチャーのtipに、式参照を見ることができます：



ピクチャー式にはすべての画像属性を適用することができます([wk image](#) ただし属性は読み込みのみです)。しかしながらピクチャーには特有の属性があることから、4D Write Proは式の結果が画像であると検知し、ピクチャー式として適切に管理するために少なくとも一度は式を評価する必要があります。

これはピクチャー式が**ST INSERT EXPRESSION** コマンドで挿入されている場合、ピクチャー属性を設定する前に**ST COMPUTE EXPRESSIONS** コマンドを呼び出す必要があるということです。

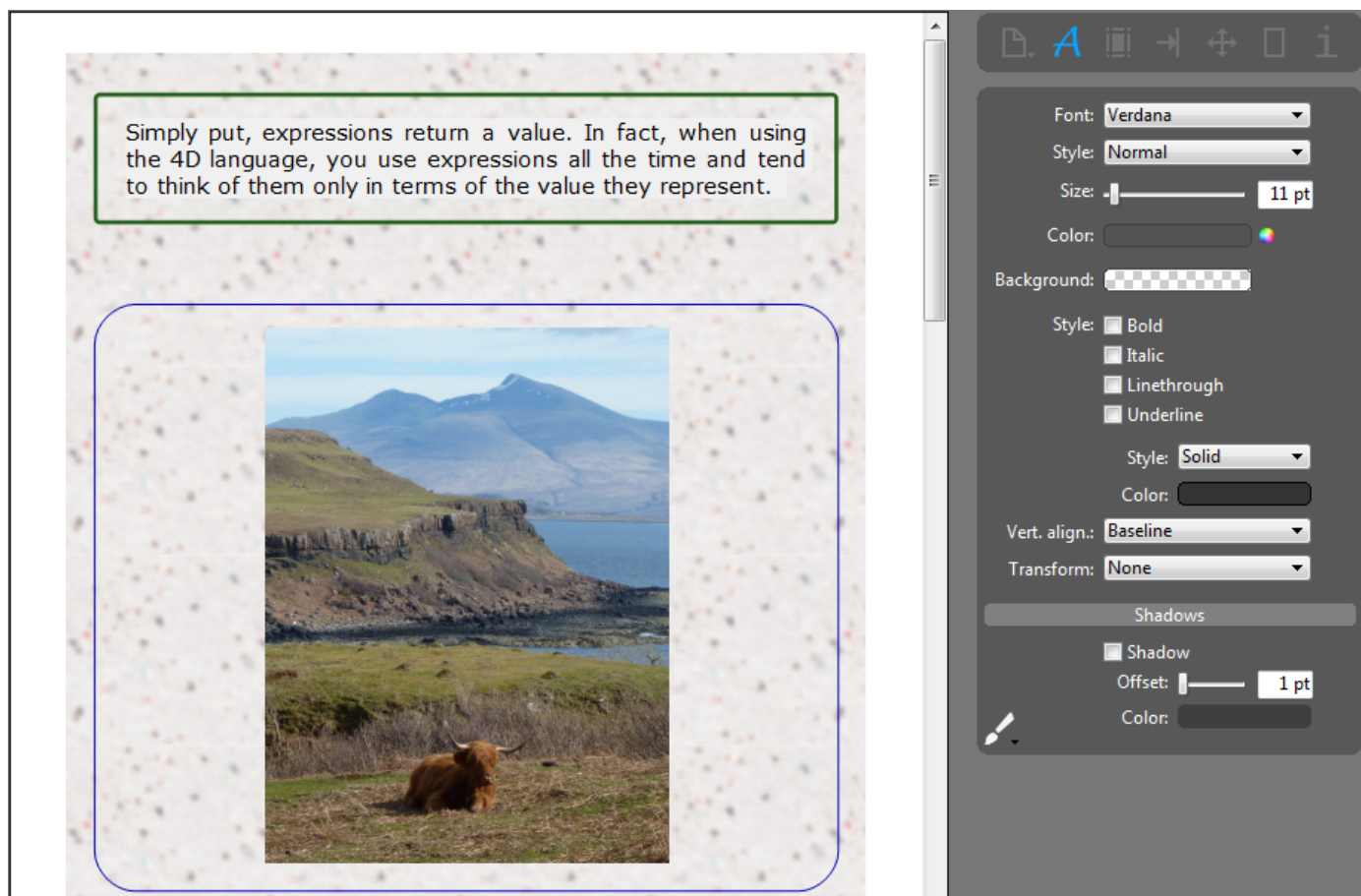
式の結果の画像が計算あるいは読み込みが不可能な場合、4D Write Pro は、未定義あるいは見つからないURLのためのデフォルトの黒いフレームを表示します：



📄 ドキュメントのコンテンツにプログラムでアクセスする

4D Write Proではドキュメント内でテキストと画像の属性を変更するためのコマンドをフルセット提供しています。これらの機能のおかげで、4Dデベロッパーは4D Write Proドキュメントに対して(ボタン、メニュー、チェックボックス等を使用した)独自のインターフェースをデザインすることができます。コマンドはドキュメント全体、または特定の部分(レンジ)に対して適用可能で、どちらもユーザー選択あるいはカスタムの値に基づきます。利用可能なプロパティには、ドキュメント単位、マージン、パディング、背景、段落設定、カラー、フォント、フォントスタイル、そして画像プロパティなどが含まれます。

例えば、**4D Write Pro エリア**ライブラリオブジェクトはこれらの機能を存分に活用し、洗練された4D Write Proインターフェースを提供しています。



セレクションレンジコマンド

4D Write Proでは、ドキュメントのコンテンツをプログラムによって選択し操作することができます。選択されたコンテンツにはテキスト、画像、表、そして(非表示の)フォーマット用タグなどが含まれることから、4D Write Proではレンジと呼ばれるオブジェクト型を使用することができます。

レンジとは、4D Write Pro ドキュメントの一部を表すオブジェクト型です。これを使用することで、選択する要素を指定したり、(**WP GET ATTRIBUTES** および **WP SET ATTRIBUTES**を使用して)ドク

メントの一部の要素を操作することができます。

レンジには異なるタイプのレンジがあります。wk type 属性(読み込みのみ)を用いて、レンジのタイプを決定することができます。それぞれのレンジには、そのレンジを定義する複数の非公開の属性が含まれます:

定数	値	コメント
wk end	end	(読み込み専用のレンジ属性)
wk id	id	(読み込みのみ属性)要素のID。値:文字列
wk owner	owner	(読み込み専用のレンジ属性)
wk start	start	(読み込み専用のレンジ属性) (読み込みのみ可能なレンジ属性) 4D Write Pro オブジェクトの型。取りうる値: • <u>wk type default</u>: 未定義の型のレンジ • <u>wk type paragraph</u>: 段落型のレンジ • <u>wk type image</u>: 画像(アンカーされた画像およびインライン画像) • <u>wk type image anchored</u>: アンカーされた画像 • <u>wk type image inline</u>: インライン画像 • <u>wk type container</u>: 例えばヘッダーやフッターなど • <u>wk type table</u>: 表参照 • <u>wk type table row</u>: 表の行参照 • <u>wk type table cell</u>: 表のセル参照 • <u>wk type table column</u>: 表のカラム参照(カラムレンジに対してのみ) • <u>wk type body</u>: 本文参照
wk type	type	

注: 配列に関連付けられたレンジ(セル、カラム、行のレンジ)は、特別な属性を含みます(以下の段落参照)。

レンジを定義するために、複数のコマンドが用意されています:

- **ClientMacFolderToWin**(*wpArea ; startRange ; endRange*) -> *rangeObj*: 引数として渡した境界に対応する新しい範囲を返します。
- **ClientMacIncludeIt**({*; } *wpArea*) -> *rangeObj*: カレントのユーザーセクションに対応した新しいレンジを返します。
- **ClientMacIconForWinPath**(*rangeObj*) -> *rangeObj*: ピクチャーのみを含んだ新しいレンジを返します。
- **ClientWinIconForMacPath**(*rangeObj*) -> *rangeObj*: 段落のみを含んだ新しいレンジを返します。

表コマンド

4D Write Pro の表は以下のコマンドを使用することでプログラムによって作成することができます:

- **WP Insert table** によって表を挿入し、その参照を取得
- **WP Table append row** によって行を表に追加し、対応するレンジを取得
- **WP Table get rows** によって行のレンジを取得
- **WP Table get columns** によってカラムのレンジを取得(*)

- **WP Table get cells** によってセルのレンジを取得

(*) htmlにはカラムに相当するものがありません。4D Write Proでは、カラムのレンジは実際にはセルのレンジです。

行、カラムとセルのレンジには、それらを定義するための非公開の属性が存在します:

定数	値	コメント
		(テーブル、ドキュメント、セクション用の属性)カラムの数。値:倍長整数
wk column count	columnCount	• テーブルの場合: 読み込みのみ属性 • ドキュメントあるいはセクションの場合: 読み込み/書き込み可能属性。デフォルト値=1(単一の行)。最大値=20
wk first column	firstColumn	(読み込みのみ属性)テーブル内にある最初のカラムの数。値:倍長整数
wk first row	firstRow	(読み込みのみ属性)テーブル内の最初の行の数。値:倍長整数
wk row count	rowCount	(読み込みのみ属性)行の総数。値:倍長整数
wk table ID	tableID	(読み込みのみ属性)親テーブルのID。値:文字列

ブックマークコマンド

4D Write Proでは**ブックマーク**と呼ばれる、ドキュメントの一部に対して動的な参照を作成し、使用することができます。ブックマークとは、4D Write Proドキュメント内の特定のレンジに関連づけられた名前付き参照です。

ブックマークは動的です。これはつまり、ブックマークに関連付いた文章をユーザーが移動または削除した場合には、紐付いているレンジも自動的に更新され、ブックマークはドキュメント内の同じ内容を引き続き参照し続けるという事です。例えば:

- ドキュメント内の20ページ目にある"Hello world"を参照する、"MyBM"というブックマークを作成したとします。
- 次に、ドキュメントの最初に50ページ挿入したとします。
- "MyBM"ブックマークを使用する事で、今度はドキュメント内の70ページ目にある、同じ"Hello world"というテキストに自動的にアクセスすることができます。

ドキュメント内には無制限のブックマークを含める事ができます。複数のブックマークが同じレンジを参照することもできますし、ブックマークのレンジは重複することもできます。しかしながら、それぞれブックマーク名はドキュメント内において固有でなければなりません。ブックマークは、**WP INSERT DOCUMENT** コマンドを使用した場合にはインポートされません(保存先のドキュメントのブックマークは上書きされません)。

一度作成すると、ブックマークはドキュメント内に保存されています。ブックマークはドキュメントとともに保存され、複数のコマンドによって管理することができます。また、テンプレートドキュメントの一部を参照するのにも使用できます。これらの部分はデータベース内のドキュメントと自動的に組み

合わせることもでき、その結果請求書やカタログといった動的な出力ドキュメントを生成することもできます。

複数のコマンドを使用して、ブックマークを作成、削除、使用することができます：

- **WP CREATE BOOKMARK** はレンジから新しいブックマークを作成します。
- **WP GET BOOKMARKS** はドキュメント内で定義されている全てのブックマークを取得します。
- **WP Get bookmark range** は既存のブックマークからレンジを取得します。
- **WP DELETE BOOKMARK** はブックマークを削除します。

属性管理コマンド

以下のコマンドはドキュメントの属性を取得あるいは設定します：

- **WP SET ATTRIBUTES**(*rangeObj* | *wpDoc* ; *attribName* ; *attribValue* { ; *attribName2* ; *attribValue2* ; ... ; *attribNameN* ; *attribValueN* }) : ドキュメントまたはレンジ内の一つ以上の属性/値のペアを設定します。
- **WP GET ATTRIBUTES**(*rangeObj* | *wpDoc* ; *attribName* ; *attribValue* { ; *attribName2* ; *attribValue2* ; ... ; *attribNameN* ; *attribValueN* }) : ドキュメントまたはレンジ内の属性のカレントの値を取得します。
- **WP RESET ATTRIBUTES**(*rangeObj* ; *attribName* { ; *attribName2* ; ... ; *attribNameN* }) : ドキュメントまたはレンジ内の属性の値を再設定します。

属性は**4D Write Pro属性**の章に詳細な説明があります。

フォント管理コマンド

このコマンドはレンジのスタイルサポートに関する情報を取得します：

- **WP Is font style supported**(*rangeObj* ; *wpFontStyle*) -> *true* または *false*: レンジが特定のスタイルをサポートするかどうかを調べます(インターフェースのデザインに有用です)。

URL

4D Write Pro では任意のドキュメントレンジ (テキスト、段落、画像、テーブルの行、セル) に対してハイパーリンクを設定することができます。たとえば、画像レンジにハイパーリンクを設定した場合、その4D Write ProドキュメントをHTMLに出力すれば、ユーザーはその画像をクリックすることで指定されたアドレスのページを開くことができます。

注: 4D Write Pro ドキュメント内でも **Ctrl+クリック** (Windows) または **Cmd+クリック** (macOS) 操作でハイパーリンクを実行することができます。

ハイパーリンクはレンジ属性として扱われるため、**WP SET ATTRIBUTES** および **WP RESET ATTRIBUTES** コマンドを wk link url 定数と組み合わせて設定します。たとえば、ユーザーが選択したテキストをリンク化したい場合：

Please refer to the **4D documentation**
for more information

次のように書きます：

```
$range:=WP Get selection(*;"WParea")
WP SET ATTRIBUTES($range;wk link url;"http://doc.4d.com")
```

Please refer to the [4D documentation](#)
for more information

注: URLに特殊文字が含まれている場合、それらの変換は自動ではおこなわれなため、別途エンコードする必要があります。URLが有効でない場合には、エラーが発生します。

対象レンジにテキストスタイルが定義されていない場合、挿入されたURLにはデフォルトのハイパーリンク形式 (下線付き青文字) が設定されます。

ドキュメント内のURLに対して、4D Write Proは次のルールを適用します:

- リンクの対象レンジに複数の段落が含まれている場合、リンクは複数作成されます。
- すでにリンクを含んでいるレンジに対してリンクを設定した場合:
 - 既存リンクは新しいリンクで上書きされます。
 - 既存リンクのレンジが新しいリンクの対象レンジと交差している場合、対象レンジ内は新しいリンクで上書きされ、その外では既存リンクが維持されます。
- **WP GET ATTRIBUTES** コマンドでリンクを取得する場合、次のケースが考えられます:
 - レンジにハイパーリンクが1つ含まれている: コマンドはそのURL文字列を返します。
 - レンジにハイパーリンクが複数含まれている: コマンドは最初のURL文字列を返します。
 - レンジにハイパーリンクが含まれていない: コマンドは空文字列を返します。
- レンジからリンクを削除するには、次の方法があります:

```
WP RESET ATTRIBUTES($range;wk link url)
```

または

```
WP SET ATTRIBUTES($range;wk link url;"")
```

WP Add picture

WP Add picture (wpDoc ; picture) -> 戻り値

引数	型	説明
wpDoc	Object	→ 4D Write Pro ドキュメント
picture	ピクチャー, テキスト	→ ピクチャー(フィールドまたは変数)あるいはピクチャーへのパス(テキスト)
戻り値	Object	→ ピクチャーを参照するオブジェクト

説明

WP Add picture コマンドは *wpDoc* 引数で指定したドキュメント内に、*picture* 引数で指定したピクチャーを固定位置でアンカーし、その参照を返します。返された参照はコマンドに渡すことで *picture* 引数のピクチャーを *wpDoc* 引数のドキュメント内のどんな位置(ページ、セクション、ヘッダー、フッター)にも、定義されたレイヤー、サイズで移動させることができます。

wpDoc 引数には、4D Write Pro ドキュメントオブジェクト名を渡します。

picture 引数には、以下のいずれかを渡すことができます：

- 4D ピクチャー。
- システムシンタックスを使用して表現された、ディスク上に保存されたピクチャーファイルへのパスを含む文字列。

文字列を使用する場合、完全なパス名を使用することもできますし、データベースのストラクチャーファイルからみた相対的なパス名を使用することもできます。ファイル名のみを渡すこともできますが、その場合ファイルはデータベースのストラクチャーファイルの隣に置かれていなければなりません。ファイル名を渡す場合、ファイルの拡張子の指定も必要です。

注意： 4D でサポートされているフォーマットであればどのようなものでも使用可能です(**ピクチャー** の章を参照してください)。**PICTURE CODEC LIST** コマンドを使用すると、利用可能なピクチャーフォーマットの一覧を取得することができます。ピクチャーが複数のフォーマット(コーデック)を格納している場合、4D Write Pro は表示用に1つのみ、(フォーマットが異なる場合)印刷用に1つのみドキュメント内に保持します。その際のフォーマットは"最適な"フォーマットが自動的に選択されます。

デフォルトでは追加されたピクチャーは：

- テキストの背後に埋め込まれ、
- ペーパーボックスの左上端に表示され、
- 全てのページに表示されます。

picture 引数で指定したピクチャーの位置、レイヤー(インライン、テキストの前面/背景)、表示状態などのプロパティは **WP SET ATTRIBUTES** コマンドを使用する、あるいは標準アクションを使用する (**4D Write Pro 標準アクションの使用参照**) ことで変更可能です。

注： **WP Get selection** コマンドは、アンカーされたピクチャーが選択されていた場合にはピクチャー参照オブジェクトを返し、インライン画像が選択されていた場合にはレンジオブジェクトを返しま

す。wk type 属性を使用すると、選択されたオブジェクトがピクチャーオブジェクトかどうかをチェックすることができます。値 = 2の場合、オブジェクトはピクチャーオブジェクトであり、それ以外の場合にはレンジオブジェクト(値 = 0)となります。

例題 1

ファイルパスを使用してデフォルトの設定でピクチャーを追加する場合を考えます。

```
C_OBJECT($obPict)
$obPict:=WP Add picture(myDoc;"C:\\Users\\John.Doe\\Pictures\\Sunrise.jpg")
```

結果は以下のようになります:



例題 2

ピクチャーをリサイズし、中央揃えにしてヘッダーにアンカーして追加したい場合を考えます:

```
C_OBJECT($oblImage)
$oblImage:=WP Add picture(myDoc;"C:\\Users\\John.Doe\\Pictures\\Saved
Pictures\\Sunrise.jpg")
WP SET ATTRIBUTES($oblImage;wk anchor origin;wk header box)
WP SET ATTRIBUTES($oblImage;wk anchor horizontal align;wk center)
WP SET ATTRIBUTES($oblImage;wk anchor vertical align;wk center)
WP SET ATTRIBUTES($oblImage;wk width;"650px";wk height;"120px")
```

結果は以下のようになります:



WP CREATE BOOKMARK

WP CREATE BOOKMARK (targetObj ; bkName)

引数	型		説明
targetObj	Object	→	4D Write Pro レンジ または 要素の参照
bkName	文字	→	作成するブックマーク名

説明

WP CREATE BOOKMARKコマンドは、親ドキュメント内の *targetObj* 引数で指定された4D Write Proオブジェクトに基づいた、*bkName* 引数で指定された名前の新しいブックマークを作成します。ブックマークとはレンジや参照への命名された参照で、これによって例えばテンプレート目的などでドキュメントの特定の部分にアクセスして再利用することができるようになります。詳細な情報については、[ブックマークコマンド](#) の章を参照してください。

targetObj 引数には、以下のものを渡すことができます：

- レンジ
- 要素 (本文 / テーブル / 行 / インラインピクチャー / 段落) の参照

注: 要素の参照を*targetObj* 引数に渡した場合、ブックマークはその指定された要素のみを含みます。*bkName* 引数には新しいブックマークの名前を渡して下さい。ブックマーク名はHTML/CSS名に準拠している必要があります(スペース文字などの無効な文字は自動的に削除されます)。ブックマーク名はドキュメント内において固有でなければなりません。同じ名前のブックマークが既にドキュメント内に存在していた場合、それは上書きされます。

ブックマークは同ドキュメント内において無制限に作成することができます。同一のレンジを使用した複数のブックマークを作成することもできます。作成したあとは、ブックマークは自動的に親ドキュメントに保存され、そのドキュメント自身と共に保存されます。

例題 1

ドキュメント内で現在選択されているテキストを参照する新しいブックマークを作成したい場合を考えます。以下のように書く事ができます：

```
C_OBJECT($range)
$range:=WP Get selection(*;"WPDocument")
WP CREATE BOOKMARK($range;"my_bookmark")
```

例題 2

既存のブックマークの名前を変更したい場合を考えます。そのためには、同じレンジを使用した新しいブックマークを作成し、古い方を削除する必要があります：

```
C_TEXT($bookmarkOldName)
C_TEXT($bookmarkNewName)
C_LONGINT($p)
C_OBJECT($wpRange)
```

```
$bookmarkOldName:="MyBookmark"
$bookmarkNewName:="MyNewBookmark"
```

```
ARRAY TEXT($_bookmarks;0)
WP GET BOOKMARKS(WParea;$_bookmarks)
```

```
$p:=Find in array($_bookmarks;$bookmarkOldName)
If($p>0)
    $wpRange:=WP Get bookmark range(WParea;$bookmarkOldName)
    WP DELETE BOOKMARK(WParea;$bookmarkOldName)
    WP CREATE BOOKMARK($wpRange;$bookmarkNewName)
End if
```

⚙ WP Create paragraph range

WP Create paragraph range (targetObj) -> 戻り値

引数	型		説明
targetObj	Object	➡	4D Write Pro レンジ または 要素の参照
戻り値	Object	↩	段落のみを格納したレンジ

互換性に関する注意

WP Create paragraph range は以前のバージョンの4D Write Pro では**WP Get paragraphs** という名前でした。分かりやすさのために名前が変更されています。

説明

新しい**WP Create paragraph range** コマンドは、*targetObj* 引数内の含まれる段落のみだけを内包したレンジオブジェクトを返します。返されたレンジオブジェクトの段落は**WP GET ATTRIBUTES** や **WP SET ATTRIBUTES** コマンドなどで使用でき、それによって段落属性のみを管理することができます。

targetObj 引数には、以下のいずれかを渡すことができます：

- レンジ
- 要素 (テーブル / 行 / 段落 / ヘッダー / フッター / 本文) の参照

例題

段落に対してのみパディングを設定したい場合を考えます：

```
$oParagraphs:=WP Create paragraph range($oSelection)
WP SET ATTRIBUTES($oParagraphs;wk_padding;20)
```

⚙ WP Create picture range

WP Create picture range (targetObj) -> 戻り値

引数	型		説明
targetObj	Object	→	4D Write Pro レンジ または 要素の参照
戻り値	Object	↩	ピクチャーのみを含んだレンジオブジェクト

互換性に関する注意

WP Create picture range は以前のバージョンの4D Write Pro では**WP Get pictures** という名前でした。分かりやすさのために名前が変更されています。

説明

新しい**WP Create picture range**コマンドは、*targetObj* 引数に渡したオブジェクト内に含まれるピクチャーのみを格納したレンジオブジェクトを返します。返された画像レンジオブジェクトは、**WP GET ATTRIBUTES** と**WP SET ATTRIBUTES** によってピクチャーの属性のみを管理するのに使用することができます。

targetObj 引数には、以下のいずれかを渡すことができます：

- レンジ
- 要素 (テーブル / 行 / 段落 / ヘッダー / フッター / 本文) の参照

例題

ピクチャーの境界線カラーのみを変えたい場合を考えます：

```
$oPicts:=WP Create picture range($oSelection)
WP SET ATTRIBUTES($oPicts;wk border color;"blue")
```

WP Create range

WP Create range (targetObj ; startRange ; endRange) -> 戻り値

引数	型	説明
targetObj	Object	→ 4D Write Pro ドキュメント、レンジ または 要素の参照
startRange	倍長整数	→ エリア内でのレンジの開始のオフセット
endRange	倍長整数	→ エリア内でのレンジの終わりのオフセット
戻り値	Object	↪ レンジオブジェクト

互換性に関する注意

WP Create range は、以前のバージョンの4D Write Proでは**WP Get range** という名前でした。分かりやすさのために名前が変更されています。

説明

WP Create range コマンドは *targetObj* 引数の4D Write Proエリア内から、*startRange* と *endRange* の間に含まれるセクションを新しいレンジオブジェクト(*rangeObj*)として返します。

4D Write Pro レンジオブジェクトを使用するとテキストのセクションの属性を管理することができます(具体的には**WP GET ATTRIBUTES** and **WP SET ATTRIBUTES** コマンドを使用します)。詳細な情報については、**セクションレンジコマンド** の段落を参照してください。

targetObj 引数には以下のものを渡すことができます:

- レンジ
- 要素 (テーブル / 行 / 段落 / インラインピクチャー / 本文 / ヘッダー / フッター) の参照
- 4D Write Pro ドキュメント

*startRange*と*endRange*引数には、ドキュメント内で選択したい文字の最初の位置と最後の位置に対応する値を渡します。*startRange*引数に`wk start text`定数を渡す事でドキュメントの最初を、*endRange*引数に`wk end text`定数を渡す事でドキュメントの最後を指定する事ができます。4D Write Proドキュメントは表示されているテキストだけではなく範囲内に含まれるフォーマットタグもあるという点に注意して下さい。

注: *targetObj* 引数に要素の参照を渡した場合、返されたオブジェクトは親要素のコンテンツに対して相対的であるという点に注意してください。レンジオブジェクトは親要素への内部的な参照を常に持ち続けます。

例題

4D Write Proフィールドの、最初から数えて12文字のレンジを選択したい場合を考えます。フィールドはフォームオブジェクト内に表示されているとします:

Description

The **DIALOG** command presents the form form to the user. This command is often used to get information from the user through the use of variables, or to present information to the user, such as options for performing an operation. It is common to display the form inside a modal window created with the **Open window** command.

以下のコードを実行した場合:

```
$range2:=WP Get range([SAMPLE]WP;wk start text;12)  
WP SELECT(*;"WParea";$range2)
```

...結果は以下の画像のようになります:

Description

The **DIALOG** command presents the form form to the user. This command is often used to get information from the user through the use of variables, or to present information to the user, such as options for performing an operation. It is common to display the form inside a modal window created with the **Open window** command.

⚙ WP Create table range

WP Create table range (targetObj) -> 戻り値

引数	型	説明
targetObj	Object	→ 4D Write Pro ドキュメント、レンジ または 要素の参照
戻り値	Object	→ 表のみを格納したレンジオブジェクト

説明

WP Create table range コマンドは、*targetObj* 引数内に格納されている表のみを指定するレンジオブジェクトを返します。渡した*targetObj* 引数に表が含まれていなかったとしても、wk type tableのレンジが返されます。返されたレンジオブジェクトは**WP GET ATTRIBUTES** および **WP SET ATTRIBUTES** で使用することで、表及び表の中身に適用される属性を管理することができます。

targetObj 引数には、以下のものを渡すことができます：

- レンジ
- 要素 (本文 / ヘッダー / フッター / 段落) の参照
- 4D Write Pro ドキュメント

例題

テーブルレンジ内に格納されている表の外見を変更したい場合を考えます：

```
tableRange:=WP Create table range(myDoc)
//テーブル属性
WP SET ATTRIBUTES(tableRange;wk border style;wk solid;wk border color;"Blue";wk border
width;"3px")
WP SET ATTRIBUTES(tableRange;wk table align;wk center)
//テキスト属性
WP SET ATTRIBUTES(tableRange;wk font size;12)
WP SET ATTRIBUTES(tableRange;wk text transform;wk capitalize)
WP SET ATTRIBUTES(tableRange;wk font bold;wk true)
```

WP DELETE BOOKMARK

WP DELETE BOOKMARK (wpDoc ; bkName)

引数	型		説明
wpDoc	Object	➡	4D Write Proドキュメント
bkName	文字	➡	削除するブックマークの名前

説明

WP DELETE BOOKMARK コマンドは*wpDoc* 引数で指定した4D Write Proドキュメント内の、*bkName* 引数で指定した名前のブックマークを削除します。

bkName 引数のブックマークがの*wpDoc* ドキュメント内に存在しない場合、コマンドは何もしません。

例題

既存のブックマークの名前を変更したい場合を考えます。そのためには、同じレンジを使用した新しいブックマークを作成し、古い方を削除する必要があります：

```
C_TEXT($bookmarkOldName)
C_TEXT($bookmarkNewName)
C_LONGINT($p)
C_OBJECT($wpRange)

$bookmarkOldName:="MyBookmark"
$bookmarkNewName:="MyNewBookmark"

ARRAY TEXT($_bookmarks;0)
WP GET BOOKMARKS(WParea;$_bookmarks)

$p:=Find in array($_bookmarks;$bookmarkOldName)
If($p>0)
    $wpRange:=WP Get bookmark range(WParea;$bookmarkOldName)
    WP DELETE BOOKMARK(WParea;$bookmarkOldName)
    WP CREATE BOOKMARK($wpRange;$bookmarkNewName)
End if
```

⚙ WP EXPORT DOCUMENT

WP EXPORT DOCUMENT (wpDoc ; filePath {; format {; option}})

引数	型		説明
wpDoc	Object	➡	4D Write Pro 変数
filePath	文字	➡	書き出されたファイルへのパス
format	倍長整数	➡	ドキュメントの出力フォーマット
option	倍長整数	➡	書き出しオプション

説明

WP EXPORT DOCUMENT コマンドは *wpDoc* 引数に渡したディスク上の4D Write Proオブジェクトを、*filePath* 引数で指定した場所に、他の任意の引数に基づいて書き出します。

wpDoc 引数には、書き出したい4D Write Proオブジェクトを渡します。

filePath 引数には、書き出し先の場所と、書き出したドキュメント名を渡します。ドキュメント名のみを渡した場合、4Dストラクチャーファイルと同階層に保存されます。デフォルトで、*format* 引数を省略した場合、コマンドはドキュメントの拡張子を見て書き出しのフォーマットを選択します。

format 引数には**4D Write Pro** テーマからの定数を渡す事もできます。この場合、4Dは必要に応じて適切な拡張子をファイル名に追加します。サポートされるフォーマットは以下の通りです：

定数	型	値	コメント
wk 4wp	倍長整数	4	4D Write Proドキュメントは、ネイティブなアーカイブフォーマット(zipになったHTMLと、別のフォルダに保存された画像)にて保存されています。4D特有のタグも含まれ、4D式は計算はされていません。このフォーマットは、特に4D Write Proドキュメントをディスク上に損失なく保存してアーカイブする事に適しています。
wk web page complete	倍長整数	2	.htm または .html 拡張子。このドキュメントは標準HTMLとして保存され、そのリソースは別に保存されます。4Dタグは除去され、式は値が計算されます。このフォーマットは特に4D Write Pro ドキュメントWeb ブラウザで表示したい場合に特に適しています。

注:

- "4D 特有のタグ"とは、4Dネームスペースと4D CSSスタイルを含めた4D XHTMLのことです。
- 式は、**ST FREEZE EXPRESSIONS**コマンドを使用することで、書き出す前でもいつでも値を確定させることができます。
- 4D Write Proフォーマットに関するより詳細な情報については、**.4wp ドキュメントフォーマット**を参照して下さい。

options 引数には、書き出しの設定を指定するオプションを渡します。倍長整数 の値を渡す事で、HTMLコードのスタイルを定義することができます。使用できる定数は以下の通りです：

定数	型	値	コメント
wk html debug	倍長整数	1	フォーマット済みのHTMLコード("整形済みフォーマット")。デバッグが容易。
wk normal	倍長整数	0	標準のHTML コード

- HTML デバッグオプションなし(デフォルト):

```
<html xmlns="http://www.w3.org/1999/xhtml"> <head> <title>New 4D Write Pro Document</title> <style type="text/css">body { background-color:#FFFFFF }ul, ol { margin:0;padding:0 }p,li { white-space:pre-wrap;margin:0pt;padding:0pt;font-family:'Times New Roman' }p.Normal,li.Normal { text-align:left }p._p1,li._p1 { font-family:'Arial';font-size:18pt;color:#1D1D1D }img._img1 { width:51pt;height:51pt }</style> </head> <body> <p class="Normal _p1"> <span>23/12/14</span>,</p> <p class="Normal _p1">Dear <span style="color:#BE0E00">Shaun Stoltz</span><span style="color:#800062"> <span style="color:#000000">from
```

- HTML デバッグオプションあり:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>New 4D Write Pro Document</title>
<style type="text/css">
body { background-color:#FFFFFF }
ul, ol { margin:0;padding:0 }
</style>
</head>
<body>
<p>23/12/14,</p>
<p>Dear Shaun Stoltz,</p>
<p>from
```

例題 1

myArea という4D Write Proオブジェクトの中身をデータベースフォルダー内のドキュメントへと書き出したい場合を考えます。'pprint'ボタンを使用してデバッグオプションを設定することができます:

```
C_TEXT($filePath)
$filePath:=Get 4D folder(Database folder)+"Exported files"+Folder separator+"WriteProExport.html"
If(pprint=0) //デバッグオプションがoffの場合
    WP EXPORT DOCUMENT(myArea;$filePath;wk web page complete;wk normal)
Else
    WP EXPORT DOCUMENT(myArea;$filePath;wk web page complete;wk html debug)
End if
```

例題 2

myArea 4D Write Proオブジェクト内のコンテンツを、.4wpフォーマットで書き出したい場合を考えます:

```
C_TEXT($path)
C_LONGINT($docRef)

Case of
: (Form event=On Clicked)

    $path:=Get 4D folder(Database folder)+"Export"+Folder separator
    $path:=Select document($path;".4wp";" title";File name entry)

    If($path# "")
```

WP EXPORT DOCUMENT(myArea;document;wk 4wp;wk normal)

Else

ALERT("An error occurred.")

End if

End case

⚙ WP EXPORT VARIABLE

WP EXPORT VARIABLE (wpDoc ; destination ; format {; option})

引数	型		説明
wpDoc	Object	➡	4D Write Pro変数
destination	テキスト変数	←	書き出したコンテンツを受け取る変数
format	倍長整数	➡	変数出力フォーマット
option	倍長整数	➡	書き出しオプション

説明

WP EXPORT VARIABLE コマンドはwpDoc 変数内の4D Write Proオブジェクトをformat 引数で指定したフォーマットでdestination 引数で指定した4D変数へと書き出します。

wpDoc 引数には、書き出したい4D Write Proオブジェクトを渡します。

destination 引数には、書き出された4D Write Pro オブジェクトを受け取る変数を渡します。必要な変数の型は、format 引数で指定した書き出しフォーマットに応じて変化します：

- ネイティブな.4wpフォーマットを渡した場合、必要な変数はBLOB型になります。
- HTMLフォーマットを渡した場合、必要な変数はテキスト型になります。

format 引数には、**4D Write Pro** テーマから、使用する書き出しフォーマットを設定する定数を渡します。それぞれのフォーマットは特定の用法と関連しています。サポートされているフォーマットは以下の通りです：

定数	型	値	コメント
wk 4wp	倍 長 整 数	4	4D Write Proドキュメントは、ネイティブなアーカイブフォーマット(zipになったHTMLと、別のフォルダに保存された画像)にて保存されています。4D特有のタグも含まれ、4D式は計算はされていません。このフォーマットは、特に4D Write Proドキュメントをディスク上に損失なく保存してアーカイブする事に適しています。
wk mime html	倍 長 整 数	1	4D Write Proドキュメントは標準のMIME HTMLとして保存され、htmlドキュメントと画像はMIMEパーツとして埋め込まれます(base64でエンコードされます)。式は計算され4D特有のタグは除去されます。このフォーマットは SMTP_QuickSend コマンドを使用してHTML Eメールを送信するのに特に適しています。
wk web page html 4D	倍 長 整 数	3	4D Write ProドキュメントはHTMLとして保存され、4D独自のタグを含みます。それぞれの式はノンブレイクのスペースとして挿入されます。このフォーマットはロスレスであるため、テキストフィールドへの保存目的に適しています。

注:

- "4D独自のタグ" とは4Dネームスペースと4D CSSスタイルを含めた4D XHTMLのことです。

- 式は、書き出しをする前であれば**ST FREEZE EXPRESSIONS** コマンドを使用することでいつでも値を確定させることができます。
- 4D Write Pro ドキュメントフォーマットについてのより詳細な情報については、**.4wp ドキュメントフォーマット** を参照して下さい。

options 引数には、書き出しの詳細を設定するオプションを渡します。詳細は以下の通りです:

- 倍長整数 の値を渡してHTMLコードのスタイルを定義します。以下の定数を使用することができます。

定数	型	値	コメント
wk html debug	倍長整数	1	フォーマット済みのHTMLコード("整形済みフォーマット")。デバッグが容易。
wk normal	倍長整数	0	標準のHTML コード

- *HTML* デバッグオプションなし(デフォルト):

```
<html xmlns="http://www.w3.org/1999/xhtml"> <head> <title> New 4D Write Pro Document</title> <style type="text/css"> body { background-color:#FFFFFF }ul, ol { margin:0;padding:0 }p,li { white-space:pre-wrap;margin:0pt;padding:0pt;font-family:'Times New Roman' }p.Normal,li.Normal { text-align:left }p._p1,li._p1 { font-family:'Arial';font-size:18pt;color:#1D1D1D }img._img1 { width:51pt;height:51pt }</style> </head> <body> <p class="Normal _p1"><span>23/12/14</span>,</p> <p class="Normal _p1">Dear <span style="color:#BE0E00">Shaun Stoltz</span><span style="color:#800062"> <span style="color:#000000">from
```

- *HTML* デバッグオプションあり:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title> New 4D Write Pro Document</title>
  <style type="text/css">
    body { background-color:#FFFFFF }
    ul, ol { margin:0;padding:0 }
  </style>
</head>
<body>
  <p>23/12/14,</p>
  <p>Dear Shaun Stoltz</p>
  <p>from
```

- または文字列を渡す事もできます。以下のプロパティがサポートされています(ただし wp_mime_html フォーマットが使用されている場合に限りです):
 - CID ホストドメイン名: 生成されたCIDのURLに追加されるホストドメイン(区切り文字として'@'を含む)。例えば、"gmail.com"を渡した場合、CIDユニークIDが123のときには'123@gmail.com' が挿入されます。省略された場合、デフォルトで、CIDユニークIDのみが使用されます(ほとんどのメールサーバーで使用可能です)。

例題

スタイル付テキストと4D参照と画像を含んだEメールを送信したい場合を考えます。この場合、MIME HTMLフォーマットで書き出された4D Write Proエリアを、4D Internet Commandを使用して送信することができます:

```
C_LONGINT($smtpid_l;$err_l;$smtpOption_l;$smtpPort_l)
C_TEXT($str;$emailBody_t;$smtpHost_t;$emailTo_t;$emailFrom_t;$smtpPass_t)
```

```
//エリアを適切なフォーマットで書き出す
```

```
WP_EXPORT VARIABLE(myWPArea;$str;wk_mime_html)
```

```
$emailTo_t:="johnsmith@4d.com"
```

```
$emailFrom_t:"testWritePro@gmail.com"
```

```
$emailBody_t:=$str
$smtpHost_t:="smtp.gmail.com"
$smtpOption_l:=9
$smtpPort_l:=465
$smtpPass_t:="QRN_on_bretzelburg"

$err_l:=SMTP_QuickSend($smtpHost_t;$emailFrom_t;$emailTo_t;$emailTitle_t;\
$emailBody_t;$smtpOption_l;$smtpPort_l;$smtpUser_t;$smtpPass_t)

If(($err_l=0))
    ALERT("Email sent to "+emailTo_t)
Else
    ALERT("Error in parameters, please try again.")
End if
```

WP GET ATTRIBUTES

WP GET ATTRIBUTES (targetObj ; attribName ; attribValue {; attribName2 ; attribValue2 ; ... ; attribNameN ; attribValueN})

引数	型	説明
targetObj	Object	➡ 4D Write Pro ドキュメント、レンジ または要素の参照
attribName	テキスト	➡ 取得する属性の名前
attribValue	テキスト, Number, 倍長整数配列, 実数配列, ピクチャー, 日付	➡ テキストのレンジの属性のカレント値

説明

WP GET ATTRIBUTES コマンドは、4D Write Pro レンジ、ヘッダー/フッター/ボディ、あるいはドキュメント内のあらゆる属性の値を返します。このコマンドを使用すると4D Write Proのどのような内部属性(文字、段落、ドキュメント、テーブル、画像など)にもアクセスすることができます。

targetObj 引数には、以下のいずれかを渡すことができます:

- レンジ
- 要素 (ヘッダー / フッター / 本文 / テーブル / 段落 / アンカー画像あるいはインライン画像) の参照
- 4D Write Pro ドキュメント

attribName 引数に渡すことのできる属性の包括的な一覧と、そのスコープおよび値については、**4D Write Pro 属性** の章を参照してください。

引数として渡した要素内の同じ属性に対して異なる値があった場合、コマンドは以下のように対応します:

- 数値に対しては、wk mixed を返します。
- 配列の場合、空の配列を返します (*attribValue* が配列として定義されていた場合にはタブストップ、カラー)。ただし wk text shadow offset に関しては、配列の値は常に2つのエントリーを含みます。これらのエントリーは水平方向あるいは垂直方向のオフセット(あるいはその両方)がミックスされていた場合には wk mixed が返されます。
- 文字列の値に対しては、空の文字列を返します。
- ピクチャーの値に対しては、空のピクチャーを返します。

例題

選択されたエリアの背景色を取得したい場合を考えます:

```
$range:=WP Get selection(*;"WParea")
WP GET ATTRIBUTES($range;wk background color;$bcol)
```

⚙ WP Get body

WP Get body (wpDoc) -> 戻り値

引数	型		説明
wpDoc	Object	➡	4D Write Pro ドキュメント
戻り値	Object	↩	4D Write Pro ボディ参照

説明

WP Get body コマンドは *wpDoc* 引数で指定した4D Write Pro ドキュメントのボディ(本文)への参照を返します。

この参照は以下のコマンドに対して渡すことができます:

- **WP SELECT** コマンドに渡してドキュメントのボディを選択
- **WP Create range** コマンドに渡してボディをレンジとして定義
- **WP SET ATTRIBUTES** と **WP GET ATTRIBUTES** コマンドに渡してボディの属性を管理

例題

ある4D Write Pro ドキュメントのボディを他の4D Write Pro ドキュメントにコピーする場合を考えます:

```
C_OBJECT($bodySource;$rangeSource;$tempoc;$bodyTarget;$rangeTarget)
```

```
$bodySource:=WP Get body([TEMPLATES]WPtemplate)
```

```
$rangeSource:=WP Create range($bodySource;wk start text;wk end text)
```

```
$tempoc:=WP New($rangeSource)
```

```
$bodyTarget:=WP Get body([Docs]MyWPDoc)
```

```
$rangeTarget:=WP Create range($bodyTarget;wk start text;wk end text)
```

```
WP INSERT DOCUMENT($rangeTarget;$tempoc;wk replace)
```

⚙ WP Get bookmark range

WP Get bookmark range (wpDoc ; bkName) -> 戻り値

引数	型		説明
wpDoc	Object	➡	4D Write Proドキュメント
bkName	テキスト	➡	レンジを取得したいブックマーク名
戻り値	Object	↩	ブックマークのレンジ

説明

WP Get bookmark range コマンドは、*wpDoc*引数で指定した4D Write Proドキュメント内の、*bkName*引数で指定されたブックマークのレンジを格納したテキストレンジオブジェクト(*rangeObj*)を返します。

bkName のブックマークが*wpDoc* のドキュメント内に存在しない場合、空の*rangeObj*オブジェクトが返されます。

例題

ドキュメント内の"MyBookmark"ブックマークのレンジを表示したい場合を考えます:

```
C_OBJECT($wpRange)
$wpRange:=WP Get bookmark range(WParea;"MyBookmark")
WP SELECT(WParea;$wpRange)
```

WP GET BOOKMARKS

WP GET BOOKMARKS (wpDoc ; arrBKNames)

引数	型		説明
wpDoc	Object	→	4D Write Proドキュメント
arrBKNames	テキスト配列	←	ブックマーク名の配列

説明

WP GET BOOKMARKS コマンドは`wpDoc` 引数で指定した4D Write Proドキュメント内で定義されている全てのブックマーク名を含む配列を返します。

コマンド実行後、`arrBKNames` 変数にはドキュメント内の全てのブックマーク名が格納されます。配列内ではブックマーク名はドキュメント内での位置によって並べ替えられています。

同じ位置から始まる複数のブックマークが存在する場合、それらはアルファベット順に並べられます。

例題

ドキュメント内で定義されているブックマークの数を知りたい場合を考えます：

```
ARRAY TEXT($_bookmarks;0)
WP GET BOOKMARKS(WParea;$_bookmarks)
ALERT("The document contains "+Size of array($_bookmarks)+" bookmarks.")
```

⚙ WP Get element by ID

WP Get element by ID (wpDoc ; ID) -> 戻り値

引数	型		説明
wpDoc	Object	➡	4D Write Pro ドキュメント
ID	文字	➡	取得する要素のID
戻り値	Object	↩	4D Write Pro 要素への参照

説明

WP Get element by ID コマンドは、*wpDoc* 引数で指定したドキュメント内の、*ID* 引数で指定された要素への参照を格納しているオブジェクトを返します。

wpDoc 引数には、4D Write Pro ドキュメント参照を渡します。

ID 引数には、取得する要素の属性IDを渡します。要素はドキュメント内のもの、例えば画像、段落、表、などを指定できますし、あるいはドキュメント自身を指定することもできます。ドキュメント内に指定されたIDを持つ要素がなかった場合、null 値が返されます。

例題 1

"img1"というID を持つピクチャーへの参照を取得する場合を考えます：

```
C_OBJECT(obImage)
obImage:=WP Get element by ID(myDoc;"img1")
```

例題 2

表要素を探し、その境界線を太い紫の境界線に変更したい場合を考えます：

```
C_OBJECT($element)
$element:=WP Get element by ID(myDoc;"Table1")
WP SET ATTRIBUTES($element;wk border style;wk solid)
WP SET ATTRIBUTES($element;wk border width;"4px")
WP SET ATTRIBUTES($element;wk border color;"purple")
```

例題 3

要素を取得し、そのID を変更したい場合を考えます：

```
$colTable:=WP Create table range([INFO]Sample)
$elements:=WP Get elements($colTable)
```

```
$elem:=WP Get element by ID([INFO]Sample;"Paris")  
If($elem#Null)  
    $elem.ID:="Lyon"  
End if
```

WP Get elements

WP Get elements (targetObj {; elementType}) -> 戻り値

引数	型	説明
targetObj	Object	➡ 4D Write Pro ドキュメント、レンジ または 要素の参照
elementType	倍長整数	➡ 取得したい要素の型
戻り値	Collection	➡ 要素の参照を格納するコレクション

説明

WP Get elements コマンドは、段落、画像、表および行などの参照を格納したオブジェクトのコレクションを返します。

targetObj 引数には、以下のものを渡すことができます：

- レンジ
- 要素 (表 / 行 / 段落 / 本文 / ヘッダー / フッター) の参照
- 4D Write Pro ドキュメント

型指定されたレンジあるいは参照が渡された場合、コマンドはそれに対応する型の要素のみを格納したコレクションを返します(ただし*elementType* 引数で型を指定した場合を除きます。以下参照)。そうでない場合、コマンドは*targetObj* 引数内にある利用可能な全ての要素を格納したコレクションを返します。

オプションとして、*elementType* 引数を渡すことで、コレクションに返される要素の型を指定することができます。以下の型を使用することができます：

定数	値	コメント
wk type default	0	デフォルトのレンジ型
wk type image	2	アンカーされた画像/インライン画像
wk type image anchored	200	ドキュメントオブジェクトに対するアンカーされた画像の参照
wk type image inline	100	ドキュメントオブジェクトに対するインライン画像の参照
wk type paragraph	1	段落タイプレンジ
wk type table	4	テーブル参照
wk type table row	5	テーブルの行参照

注：

- wk type default セレクターを渡した場合、コマンドは何もフィルターされていないコレクションを返します。
- wk type paragraph, wk type image あるいは wk type table セレクターを渡した場合、返されるコレクションにはその型の要素のみが格納されます。これは*elementType* 引数での指定より優先されます。
- 返されたコレクション内の要素は*targetObj* 引数内の出現順に並びますが、4D Write Pro ドキュメント参照が渡された場合は要素はそのID 順に並びます。

- *targetObj* 引数で渡されたオブジェクトに格納される要素がレンジ内に完全に格納されていない場合、返される結果にはその要素全体が含まれます。例えば、レンジ内に表の一部(全体ではない)が含まれる場合、表全体への参照が返されますが、含まれるのはセル(段落)参照のみとなります。

例題 1

ドキュメント内にある全ての段落のコレクションを返したい場合を考えます:

```
C_COLLECTION($docElements)
$docElements:=WP Get elements($myDoc;wk type paragraph)
//段落要素のみを格納するコレクションを返す
```

例題 2

ドキュメント内の段落で、文字の揃え方とカラーを交互に切り替えたい場合を考えます:

```
C_COLLECTION($col)
C_OBJECT($obj)
C_OBJECT($body)
C_LONGINT($i;$n)
$col:=New collection
$body:=WP Get body(myDoc)
$col:=WP Get elements($body;wk type paragraph)
$i:=0
$n:=$col.length
For($i;0;$n-1)
    $obj:=$col[$i]
    WP RESET ATTRIBUTES($obj)
    WP SET ATTRIBUTES($obj;wk font bold;wk false)
    WP SET ATTRIBUTES($obj;wk font italic;wk false)
    If($i%2=0)
        WP SET ATTRIBUTES($obj;wk text align;wk left)
        WP SET ATTRIBUTES($obj;wk font bold;wk true)
        WP SET ATTRIBUTES($obj;wk text color;"#804040")
        WP SET ATTRIBUTES($obj;wk margin right;"5cm")
        WP SET ATTRIBUTES($obj;wk margin left;"1cm")
    Else
        WP SET ATTRIBUTES($obj;wk text align;wk right)
        WP SET ATTRIBUTES($obj;wk font italic;wk true)
        WP SET ATTRIBUTES($obj;wk text color;"#404020")
        WP SET ATTRIBUTES($obj;wk margin left;"5cm")
        WP SET ATTRIBUTES($obj;wk margin right;"1cm")
    End if
End for
```

⚙ WP Get footer

WP Get footer (rangeObj) | (wpDoc ; sectionIndex {; sectionSubType}) -> 戻り値

引数	型	説明
rangeObj	Object	➡ 4D Write Pro レンジ
wpDoc	Object	➡ 4D Write Pro ドキュメント
sectionIndex	倍長整数	➡ セクションインデックス
sectionSubType	倍長整数	➡ サブセクションのタイプ(wk first page、wk left page、あるいは wk right page)
戻り値	Object	➡ 4D Write Pro フッター参照

説明

WP Get footer コマンドは *wpDoc* 引数で指定したドキュメントの、*rangeObj* 引数で指定したレンジ、あるいは *sectionIndex* 引数で指定したインデックスに関連付けられたフッターへの参照を返します。

- 第一シンタックス(*rangeObj* 引数を使用)では、コマンドはレンジ内の最初のページのフッターを返します。ページ内においてサブセクションがアクティブになっている場合、コマンドはアクティブなサブセクションのフッターを返します(そのため常に表示されているフッターを返します)。

- 第二シンタックス(*wpDoc* 引数)では、*sectionIndex* コマンドは引数で指定したセクションのフッターを返します(*sectionIndex* 引数の値は1以上でなければなりません)。

subSectionType 引数が渡された場合、コマンドは指定されたサブセクションタイプに関連付けられたフッターを返します。*subSectionType* 引数には以下の定数のどれか一つを渡すことができます:

定数	型	値
wk first page	倍長整数	1
wk left page	倍長整数	2
wk right page	倍長整数	3

subSectionType 引数が指定されなかった場合、メインセクションのフッターが返されます。この場合、サブセクションがアクティブになっている場合、メインセクションのフッターオブジェクトは *undefined* となります(メインセクションフッターはアクティブな wk first page あるいは wk left page サブセクションがない場合にのみ利用可能です)。

例題

ある4D Write Pro ドキュメントの最初のページのフッターを、他の4D Write Pro ドキュメントのすべてのページのフッターにコピーしたい場合を考えます:

```
C_OBJECT($footerSource;$rangeSource;$tempoc;$footerTarget;$rangeTarget)
```

```
$footerSource:=WP Get footer([TEMPLATES]WPtemplate;1;wk first page) //セクション1の最初のページのフッターを取得
```

```
$rangeSource:=WP Create range($footerSource;wk start text;wk end text)
```

```
$tempoc:=WP New($rangeSource)
```

```
$footerTarget:=WP Get footer([Docs]MyWPDoc;1) //セクション1の全てのページ
```

```
$rangeTarget:=WP Create range($footerTarget;wk start text;wk end text)
```

```
WP INSERT DOCUMENT($rangeTarget;$tempoc;wk replace)
```

WP Get frame

WP Get frame ({ * ; } wpArea) -> 戻り値

引数	型	説明
*	演算子	➡ 指定時、wpArea はフォームオブジェクト(文字列)。省略時、wpArea はオブジェクトフィールドあるいは変数
wpArea	文字	➡ フォームオブジェクト名(* 指定時)あるいは4D Write Pro オブジェクト変数あるいはフィールド(* 省略時)
戻り値	倍長整数	↻ カースルが現在設定されているフレーム

説明

WP Get frame コマンドはwpArea 引数で指定した4D Write Pro エリア内で、カーソルが現在設定されているフレームを返します。

任意の * 引数を渡した場合、wpArea 引数はフォームオブジェクト名(文字列)です。省略時、wpArea 引数は4D Write Pro オブジェクト変数あるいはフィールドです。

返された値は、**4D Write Pro** テーマ内にある以下の定数のどれか一つと比較することができます:

定数	型	値	コメント
wk body	倍長整数	0	セクションのボディフレーム
wk current section default footer	倍長整数	10	セクションのメインのフッター(アクティブなサブセクションのフッターがない場合にのみ利用可能)
wk current section default header	倍長整数	9	セクションのメインのヘッダー(アクティブなサブセクションのヘッダーがない場合にのみ利用可能)
wk current section first footer	倍長整数	4	セクションの最初のページのフッター
wk current section first header	倍長整数	3	セクションの最初のページのヘッダー
wk current section left footer	倍長整数	6	セクションの左ページのフッター
wk current section left header	倍長整数	5	セクションの左ページのヘッダー
wk current section right footer	倍長整数	8	セクションの右ページのフッター
wk current section right header	倍長整数	7	セクションの右ページのヘッダー

例題

ユーザーがカーソルをヘッダーあるいはフッターフレームに設定したかどうかを確認したい場合を考えます:

```
C_LONGINT($frameGet)
$frameGet:=WP Get frame(*,"WParea")
If($frameGet=wk body)
    ALERT("フッターあるいはヘッダーを選択してください")
End if
```

⚙ WP Get header

WP Get header (rangeObj) | (wpDoc ; sectionIndex {; subSectionType}) -> 戻り値

引数	型	説明
rangeObj	Object	➡ 4D Write Pro レンジ
wpDoc	Object	➡ 4D Write Pro ドキュメント
sectionIndex	倍長整数	➡ セクションインデックス
subSectionType	倍長整数	➡ サブセクションのタイプ(wk first page、wk left page、あるいは wk right page)
戻り値	Object	➡ 4D Write Pro ヘッダー参照

説明

WP Get header コマンドはwpDoc 引数で指定したドキュメントの、rangeObj 引数で指定したレンジ、あるいはsectionIndex 引数で指定したインデックスに関連付けられたヘッダーへの参照を返します。

- 第一シンタックス(rangeObj 引数を使用)では、コマンドはレンジ内の最初のページのヘッダーを返します。ページ内においてサブセクションがアクティブになっている場合、コマンドはアクティブなサブセクションのヘッダーを返します(そのため常に表示されているヘッダーを返します)。
- 第二シンタックス(wpDoc 引数)では、sectionIndex コマンドは引数で指定したセクションのヘッダーを返します(sectionIndex 引数の値は1以上でなければなりません)。subSectionType 引数が渡された場合、コマンドは指定されたサブセクションタイプに関連付けられたヘッダーを返します。subSectionType 引数には以下の定数のどれか一つを渡すことができます:

定数	型	値
wk first page	倍長整数	1
wk left page	倍長整数	2
wk right page	倍長整数	3

subSectionType 引数が指定されなかった場合、メインセクションのヘッダーが返されます。この場合、サブセクションがアクティブになっている場合、メインセクションのヘッダーオブジェクトはundefined となります(メインセクションのヘッダーはアクティブなwk first page あるいは wk left page サブセクションがない場合にのみ利用可能です)。

例題

ある4D Write Pro ドキュメントの最初のページのヘッダーを、他の4D Write Pro ドキュメントのすべてのページのヘッダーにコピーしたい場合を考えます:

C_OBJECT(\$headerSource;\$rangeSource;\$tempoc;\$headerTarget;\$rangeTarget)

\$headerSource:=**WP Get header**([TEMPLATES]WPtemplate;1;wk first page) //セクション1の最初のページのヘッダーを取得

\$rangeSource:=**WP Create range**(\$headerSource;wk start text;wk end text)

\$tempoc:=**WP New**(\$rangeSource)

\$headerTarget:=**WP Get header**([Docs]MyWPDoc;1) //セクション1の全ページ

\$rangeTarget:=**WP Create range**(\$headerTarget;wk start text;wk end text)

WP INSERT DOCUMENT(\$rangeTarget;\$tempoc;wk replace)

⚙ WP Get page count

WP Get page count (wpDoc) -> 戻り値

引数	型		説明
wpDoc	Object	→	4D Write Pro ドキュメント
戻り値	倍長整数	↩	ドキュメントのページ数

説明

WP Get page count コマンドは *wpDoc* に指定した 4D WritePro ドキュメントの総ページ数を返します。

例題

Items テーブルのカレントセクション内で、"Manual" フィールドに格納されている 4D Write Pro ドキュメントの総ページ数を確認します：

```
C_LONGINT($pageCount)
C_LONGINT($totalCount)
FIRST RECORD([Items])
While(Not(End selection([Items])))
    $pageCount:=WP Get page count([Items]Manual)
    $totalCount:=$totalCount+$pageCount
NEXT RECORD([Items])
End while
ALERT("マニュアルの総ページ数: "+String($totalCount))
```

⚙ WP Get position

WP Get position (targetObj {; layout}) -> 戻り値

引数	型	説明
targetObj	Object	➡ 4D Write Pro ドキュメント、レンジ または 要素の参照
layout	倍長整数	➡ 評価の際に使用される4D Write Pro ドキュメントのレイアウト: 0(デフォルト)=4D Write Pro レイアウト、 1=HTML WYSIWYG
戻り値	Object	➡ 位置情報

説明

WP Get position コマンドは、*targetObj* 引数で指定したターゲットのカレントの位置を説明するオブジェクトを返します。

targetObj 引数には、以下のものを渡すことができます:

- レンジ
- 要素参照(表 / 行 / 段落 / 本文 / ヘッダー / フッター / ピクチャー)
- 4D Write Pro ドキュメント

このコマンドは位置の情報を、以下のプロパティを持つオブジェクトとして返します:

プロパティ名	型	詳細
section	数値	<i>targetObj</i> 引数と交錯する最初のセクションの番号
page	数値	<i>targetObj</i> 引数と交錯する最初のページの番号
column	数値	<i>targetObj</i> 引数と交錯する最初のカラムのインデックス(1-based)
line	数値	<i>targetObj</i> 引数と交錯する最初の行のインデックス(カラム(あるいはカラムがない場合にはページ)に対して相対的 - 1-based)
position	数値	<i>targetObj</i> 引数の最初の文字の、行の中での位置(1-based)

targetObj 引数が空レンジの場合、**WP Get position** はレンジの開始位置情報を返します。

targetObj 引数がヘッダーあるいはフッター内のテキストを参照する場合、**WP Get position** はヘッダー/フッターが最初のセクション/ページ内にて表示されている位置情報を返します。

targetObj 引数が(複数のページに複製可能な)アンカーされた画像を参照する場合、**WP Get position** は画像が最初のセクション/ページ内にてアンカーされている位置情報を返します。

任意の*layout* 引数を使用すると、*targetObj* の評価にHTML wysiwyg ビューを設定することができます。"4D Write Pro"テーマ内にある、以下の定数のいずれか1つを渡すことができます:

定数	型	値	コメント
wk 4D Write Pro layout	倍長 整数	0	標準の4D Write Proレイアウト。いくつかの特定のスタイル属性を含める事ができます。
wk html wysiwyg	倍長 整数	1	このレイアウトでは、4D Write Proの高度な属性のうち、全てのブラウザで互換性があるもの以外は除去されます。

layout 引数が省略された場合、デフォルトでは4D Write Pro レイアウトが使用されます。

WP Get position では、**WP PRINT** コマンドと同じビュー設定が使用されます:

- ヘッダー: 表示
- フッター: 表示
- 式: 計算されて表示
- ページモード: ページあるいは下書き (HTML wysiwyg ビューはコマンドによって定義された通りです)

targetObj が4D Write Pro フォームエリアに表示されている場合、必ずエリアのビュー設定とコマンドのビュー設定が同じ情報を返すように設定が一致しているようにしてください。また、これらの情報がエリア内で異なっていた場合、4D Write Pro はコマンドを呼び出すたびにレイアウトを計算するためにドキュメントを"クローン"しなければならない、そのため時間がかかる可能性があることに注意してください。パフォーマンス上の理由から、こういった場合にはドキュメントをオフラインでビルドし、ビルドが完成したあとにドキュメントをフォームオブジェクトにコピーすることが推奨されます。

例題 1

レンジのページ番号を調べたい場合を考えます:

```
C_OBJECT($range;$rangePosition)
$range:=WP Get selection(WParea) //ユーザーセレクションからレンジを取得
$rangePosition:=WP Get position($range;wk 4D Write Pro layout)
ALERT("First page number in selection is "+String($rangePosition.page))
```

例題 2

ドキュメントの最初のページと最後のページにおいて、テキストカラーを青に設定したい場合を考えます:

```
C_LONGINT($nbPages)
C_OBJECT($body)
C_COLLECTION($_paragraphs)

$nbPages:=WP Get page count([DOC]Sample)
$body:=WP Get body([DOC]Sample)
$_paragraphs:=WP Get elements($body;wk type paragraph)
For each($paragraph;$_paragraphs)
    $info:=WP Get position($paragraph)
    If($info.page=1)|($info.page=$nbPages) //最初のページと最後のページのテキストカラーは青に
        WP SET ATTRIBUTES($paragraph;wk text color;"blue")
```

```
Else //他のページはテキストカラーは黒に
    WP SET ATTRIBUTES($paragraph;wk text color;"black")
End if
End for each
```

例題 3

ドキュメントをビルドする際、段落がページで分割されることを避けたい場合を考えます：

```
C_OBJECT(WParea;$body)
C_COLLECTION($_paragraphs)
WParea:=WP New

$body:=WP Get body([DOC]Sample)
$_paragraphs:=WP Get elements($body;wk type paragraph)

For each($paragraph;$_paragraphs)
    $insert:=WP New($paragraph)

    //カレントのレンジとページ番号を記憶
    $rangeBefore:=WP Create range(WParea;wk end text;wk end text)
    $info:=WP Get position($rangeBefore)
    $memoPage:=$info.page

    //コンテンツを挿入、レンジには含めない
    WP INSERT DOCUMENT($rangeBefore;$insert;wk append;wk exclude from range)

    //挿入後の位置をチェック
    $rangeAfter:=WP Create range(WParea;wk end text;wk end text)
    $info:=WP Get position($rangeAfter)

    If($memoPage# $info.page) // ページが変化していた場合、改ページを追加
        WP INSERT BREAK($rangeBefore;wk page break;wk replace)
    End if

End for each
```

⚙ WP Get selection

WP Get selection ({ * ; } wpArea) -> 戻り値

引数	型	説明
*	演算子	➡ 指定時、wpAreaはオブジェクト名(文字列)。省略時はwpAreaはオブジェクトフィールドあるいは変数。
wpArea	文字	➡ フォームオブジェクト名(*指定時)、または4D Write Proオブジェクト変数またはフィールド(*省略時)
戻り値	Object	🔄 レンジオブジェクト または ピクチャーオブジェクト

説明

WP Get selection コマンドはwpArea引数で指定した4D Write Proエリア内で現在選択されているコンテンツに基づいた、新しいレンジオブジェクトあるいはピクチャーオブジェクトを返します。

任意の * 引数を渡した場合、wpAreaがフォームオブジェクト名(文字列)である事を指示します。この引数を渡さなかった場合、wpAreaが4D Write Proオブジェクト変数あるいはフィールドであることを指示します。wpArea引数に有効な4D Write Proエリアが渡されなかった場合、空のオブジェクトが返されます。

注: このコマンドを使用するには、wpArea がフォームに関連付けられている(つまりフォーム上に表示されている)必要があります。

アンカーされた画像(位置が固定されたピクチャー)がwpArea 引数内で選択されている場合、**WP Get selection** コマンドはレンジオブジェクトではなく、ピクチャー参照オブジェクトを返します(詳細な情報については、**WP Add picture** コマンドの詳細を参照してください)。それ以外の場合には、コマンドは新しいレンジオブジェクトを返します。レンジオブジェクトの詳細については、**セレクションレンジコマンド** を参照してください。

例題

4D Write Proエリアから選択されたテキストを取得したい場合を考えます:

```
$range:=WP Get selection(*;"WParea")
```

⚙ WP Get text

WP Get text (targetObj {; expressions}) -> 戻り値

引数	型	説明
targetObj	Object	➡ 4D Write Pro ドキュメント、レンジ または 要素の参照
expressions	Number	➡ 式の評価の指定
戻り値	テキスト	🔄 targetObjのテキスト

説明

WP Get text コマンドは*targetObj* 引数で指定した4D Write Pro ドキュメントのテキストのコンテンツを取得します。

WP Get text は標準テキストのみを返します。適用されたスタイル(太字、イタリック、など)は全て無視されます。

targetObj 引数には、以下のいずれかを格納したオブジェクトを渡します:

- レンジ
- 要素参照(表 / 段落 / 本文 / ヘッダー / フッター)
- 4D Write Pro ドキュメント

expressions 任意の引数は、*targetObj* 引数の中にある4D 式の評価の仕方を定義します。指定できるオプションは以下のとおりです:

定数

wk expressions as source

wk expressions as space

wk expressions as value

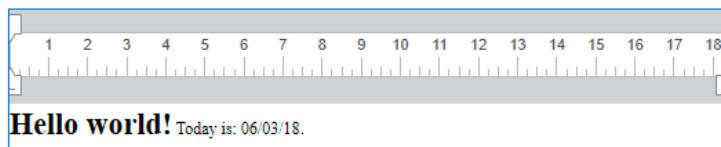
重要: WP や ST コマンドで使用されるテキストレンジインデックスと同じになるのは、標準テキストにwk expressions as space を使用した場合のみです。WP やST コマンドでは常に式のサイズ=1 文字とみなすからです。

注:

- *targetObj* 引数のオブジェクトにピクチャーが含まれる場合、そのピクチャーは無視され、スペース文字(" ")が返されます。
- *targetObj* 引数のオブジェクトに表が含まれる場合、各セルのコンテンツが個別の段落として扱われ、タブ区切りされたテキストとして返されます。行はキャリッジリターンで分割されます。

例題

以下のドキュメント内のテキストを取得したい場合を考えます:



以下のように書くことができます:

```
$range:=WP Create range(myDoc2;wk start text;wk end text)
vText:=WP Get text($range;wk expressions as value) // "Hello world! Today is: 06/03/18." を返す
vText2:=WP Get text($range;wk expressions as source) // "Hello world! Today is: Current date." を返す
vText3:=WP Get text($range;wk expressions as space) // "Hello world! Today is: ." を返す
```

⚙ WP Import document

WP Import document (filePath) -> 戻り値

引数	型	説明
filePath	文字	→ 4D Write ドキュメント(.4w7または.4wt)へのパス、または 4D Write Pro ドキュメント(.4wp)へのパス
戻り値	Object	↩ 4D Write Proオブジェクト

説明

WP Import document コマンドは既存の4D Write Proドキュメントまたは4D Write ドキュメント(.4w7 または .4wt)を新規の4D Write Proオブジェクトへと変換します。

filePath 引数には、ディスク上に保存されたドキュメントへのパスを渡します。以下のタイプのドキュメントがサポートされています:

- 以前の4D Write ドキュメント(.4w7 or .4wt)。4D Write Proオブジェクト内で現在サポートされている4D Writeの昨日の詳細な一覧に着いては、[4D Write ドキュメントの読み込み](#) の章を参照して下さい。
- 4D Write Pro(.4wp)フォーマットドキュメント。4D Write Proドキュメントフォーマットについてのより詳細な情報に関しては、[.4wp ドキュメントフォーマット](#) を参照して下さい。

パスは完全なパスを渡す必要があります。ドキュメントがストラクチャーファイルと同等の階層の保存されている場合に限り、ドキュメント名のみを渡す事もできます。

実行後、コマンドはこの変換の結果を格納した4D Write Proオブジェクトを返します。

注: 4D BLOBフィールドに保存されたドキュメントを読み込みたい場合には、**WP New** コマンドの使用も検討してみてください。

filePath 引数に渡したパスが無効な場合、またはファイルが見つからない場合、または指定したファイルがサポートされていないフォーマットだった場合、エラーが返されます。

例題

```
C_OBJECT(WPDoc)
WPDoc:=WP Import document("C:\\documents\\4DWriteDocs\\Letter.4w7")
```

WP INSERT BREAK

WP INSERT BREAK (rangeObj ; breakType ; mode {; rangeUpdate})

引数	型		説明
rangeObj	Object	➡	4D Write Proレンジオブジェクト
breakType	倍長整数	➡	挿入するブレイクの型
mode	倍長整数	➡	挿入モード
rangeUpdate	倍長整数	➡	レンジ更新モード

説明

WP INSERT BREAK コマンドは*mode* 引数や*rangeUpdate* 引数で指定された形で、*rangeObj* 引数で指定されたレンジ内に*breakType* 引数で指定された型の新しいブレイクを挿入します。

rangeObj 引数には、有効な4D Write Pro標準レンジオブジェクトを渡します。

breakType 引数には、**4D Write Pro**テーマからの以下の定数のどれか一つを渡し、挿入するブレイクのタイプを定義します：

定数	型	値	コメント
wk column break	倍長整数	3	カラムブレイク:カラムブレイクを挿入します
wk line break	倍長整数	0	(同じ段落内での)改行
wk page break	倍長整数	2	改ページ：新しいページを定義します。
wk section break	倍長整数	1	セクションブレイク:新しいセクションを定義します。

mode 引数には、対象となる*rangeObj* レンジ内で使用されるブレイクの挿入モードを指定する定数を渡します：

定数	型	値	コメント
wk append	倍長整数	2	コンテンツをレンジの終わりに挿入する
wk prepend	倍長整数	1	コンテンツをレンジの始めに挿入する
wk replace	倍長整数	0	レンジのコンテンツを置き換える

任意の*rangeUpdate* 引数内には、以下の定数のいずれか一つを渡し、挿入したコンテンツが挿入後のレンジに含まれるかどうかを指定します：

定数	型	値	コメント
wk exclude from range	倍長整数	1	更新されたレンジには、以前のコンテンツのみを含める
wk include in range	倍長整数	0	更新されたレンジに挿入コンテンツを含める

rangeUpdate 引数を渡さなかった場合、デフォルトで、挿入されたコンテンツは挿入後のレンジに含まれます。

例題

請求書を作成する際、最後のページを除いてページブレイク(改ページ)を挿入したい場合を考えます：

```
$nbInvoices:=Records in selection([INVOICE])
For($j;1;$nbInvoices)
... //請求書进行处理
If($j# $nbInvoices) //最後のページを除いて改ページを挿入
    WP INSERT BREAK($buildRange;wk_page break;wk_append;wk_exclude from range)
End if
End for
```

WP INSERT DOCUMENT

WP INSERT DOCUMENT (rangeObj ; wpDoc ; mode { ; rangeUpdate })

引数	型		説明
rangeObj	Object	➡	4D Write Pro レンジオブジェクト
wpDoc	Object	➡	4D Write Pro ドキュメント
mode	倍長整数	➡	挿入モード
rangeUpdate	倍長整数	➡	レンジ更新モード

説明

WP INSERT DOCUMENT コマンドは *mode* や *rangeUpdate* で指定した方法で、*rangeObj* に指定されたレンジに *wpDoc* のドキュメントを挿入します。

rangeObj には4D Write Pro レンジを渡します。

挿入する *wpDoc* ドキュメントは **WP New** または **WP Import document** で作成された 4D Write Pro ドキュメントオブジェクトを受け渡すことができます。挿入されるのは本文の要素のみです (挿入先のセクションおよびブックマークは変更されません)。挿入は実質的にはコピー処理されるため、*wpDoc* は再利用することができます。

mode パラメーターには、*rangeObj* レンジにドキュメントを挿入する際の挿入モードを指定するために、**4D Write Pro** テーマから一つ以上の定数を受け渡します:

定数	型	値	コメント
wk append	倍長整数	2	コンテンツをレンジの終わりに挿入する
wk prepend	倍長整数	1	コンテンツをレンジの始めに挿入する
wk replace	倍長整数	0	レンジのコンテンツを置き換える

上に紹介した定数を、次の挿入オプションと組み合わせることができます:

定数	型	値	コメント
wk freeze expressions	倍長整数	64	挿入の瞬間に式を評価します。
wk keep paragraph styles	倍長整数	128	移行先の段落スタイルを保持します。

任意の *rangeUpdate* パラメーターには、次の定数のいずれか一つを渡して、挿入コンテンツを処理後のレンジに含めるかどうかを指定します:

定数	型	値	コメント
wk exclude from range	倍長整数	1	更新されたレンジには、以前のコンテンツのみを含める
wk include in range	倍長整数	0	更新されたレンジに挿入コンテンツを含める

rangeUpdate パラメーターを省略した場合 (デフォルト)、挿入コンテンツは処理後のレンジに含まれます。

例題 1

ドキュメントの内容を、他のドキュメント内で選択されたテキストで置き換える場合を考えます：

```
$tempRange:=WP Get selection(WPTemplate) //WPTemplateドキュメント内のユーザーセレクションを  
取得  
$doctoCopy:=WP New($tempRange) //WPTemplateに基づいて新しいドキュメントを作成  
WP INSERT DOCUMENT(WPDoc;$doctoCopy;wk replace) //新しいドキュメントの中身でWPDocの中身  
を置き換え
```

例題 2

あらかじめフォーマットされ、それぞれがブックマークとして保存された複数のパーツで構成された、テンプレートドキュメントを定義しました。このテンプレートから任意のブックマークを新規ドキュメントとして抽出し、作成中のドキュメントに挿入することができます。

```
ARRAY TEXT($_BookmarkNames;0)  
WP GET BOOKMARKS([TEMPLATES]WP;$_BookmarkNames) // テンプレートからブックマークを取得し  
ます  
$targetRange:=WP New //空のドキュメントを作成(これが最終的なドキュメントになります)  
  
$p:=Find in array($_BookmarkNames;"Main_Header") // 使用したいブックマークを名称で探します  
If($p>0)  
    $Range:=WP Get bookmark range(WParea;$_BookmarkNames{$p}) // ブックマークからレンジを  
取得します  
    $RangeDoc:=WP New($Range) // レンジから新規ドキュメントを作成します  
    WP INSERT DOCUMENT($targetRange;$RangeDoc;wk append+wk freeze expressions) // wk  
append = $RangeDocドキュメントは $targetRange の先頭に挿入されます  
End if
```

⚙ WP INSERT PICTURE

WP INSERT PICTURE (rangeObj ; picture ; mode {; rangeUpdate})

引数	型	説明
rangeObj	Object	➡ 4D Write Proレンジ
picture	ピクチャ —	➡ ピクチャーフィールドまたは変数、あるいはディスク上のピクチャーファイルへのパス
mode	倍長整数	➡ 挿入モード
rangeUpdate	倍長整数	➡ レンジ更新モード

説明

WP INSERT PICTURE コマンドは、指定した挿入モード *mode* に従って、*picture* のピクチャーを *rangeObj* で指定したレンジに挿入します。ピクチャーは文字として *rangeObj* に挿入されます。*rangeObj* にはレンジを受け渡します。

picture には次のものを渡すことができます：

- ピクチャーフィールドまたは変数
- ディスク上のピクチャーファイルへのパス (システムシンタックス、文字列)
文字列を受け渡す場合には、フルパス名またはデータベースのストラクチャーファイルを起点とした相対パス名を指定することができます。データベースのストラクチャーファイルと同じ場所にピクチャーが保存されている場合には、ファイル名だけを渡すこともできます。ファイル名を渡した場合には、ファイル拡張子も指示しなければなりません。

このピクチャのフォーマットは4Dがネイティブにサポートされるものであればいずれの形式でも構いません (**ピクチャ** 参照)。**PICTURE CODEC LIST** コマンドを使用すると、使用可能なフォーマットのリストを取得できます。ピクチャーファイルに複数のフォーマット (コーデック) が含まれていると、4D Write Pro は表示用にひとつ、そして (異なる場合には) 印刷用にひとつ、"最適な" フォーマットを自動的に選択して保存します。

mode パラメーターには、ピクチャーの挿入モードを指定する定数を受け渡します：

定数	型	値	コメント
wk append	倍長整数	2	コンテンツをレンジの終わりに挿入する
wk prepend	倍長整数	1	コンテンツをレンジの始めに挿入する
wk replace	倍長整数	0	レンジのコンテンツを置き換える

任意の *rangeUpdate* パラメーターには、挿入したピクチャーをレンジに含めるか否かを指定する定数を受け渡すことができます：

定数	型	値	コメント
wk exclude from range	倍長整数	1	更新されたレンジには、以前のコンテンツのみを含める
wk include in range	倍長整数	0	更新されたレンジに挿入コンテンツを含める

rangeUpdate パラメーターを省略した場合には、挿入したピクチャーはレンジに含められます。

例題

この例題では、ユーザーが選択したピクチャーがレンジオブジェクトに挿入できなかった場合に警告します:

```
C_OBJECT($wpRange)
$wpRange:=WP Get selection([EXAMPLES]wpDoc)

C_BOOLEAN($fail)
$fail:=False

// ユーザーに挿入するピクチャーを選択してもらいます
$imgRef:=Open document("")
// ユーザーがキャンセルしなかった場合
If(OK=1)
// サポートされている形式のピクチャーファイルの場合
If(Is picture file(document))
// ユーザーが選択したピクチャーを挿入します
WP INSERT PICTURE($wpRange;document;wk.replace)
Else
$fail:=True
End if
Else
$fail:=True
End if
// 挿入に失敗した場合には、ユーザーに警告します
If($fail)
ALERT("Picture insertion failed")
End if
```

⚙ WP Insert table

WP Insert table (rangeObj ; mode {; rangeUpdate {; nbCols {; nbRows}} }) -> 戻り値

引数	型		説明
rangeObj	Object	➡	レンジオブジェクト
mode	倍長整数	➡	挿入モード
rangeUpdate	倍長整数	➡	レンジ更新モード
nbCols	倍長整数	➡	テーブル内に設定するカラム数
nbRows	倍長整数	➡	テーブル内に設定する行数
戻り値	Object	↺	テーブル参照オブジェクト

説明

WP Insert table コマンドは *rangeObj* 引数で指定したレンジに、*mode* 引数で指定した挿入モード新しいテーブルを挿入し、それに対応した新しいテーブルオブジェクトを返します。オプションとして、*rangeUpdate* 引数や、規定のカラム数や行数を指定する *nbCols* や *nbRows* 引数を渡すことができます。

rangeObj 引数には、レンジを渡します。

注: テーブル内に別のテーブルを挿入することはできません(エラーが返されます)。

mode 引数には、ドキュメント内のテーブルに対して使用される挿入モードを指定する以下の定数のいずれか一つを渡します:

定数	型	値	コメント
wk append	倍長整数	2	コンテンツをレンジの終わりに挿入する
wk prepend	倍長整数	1	コンテンツをレンジの始めに挿入する
wk replace	倍長整数	0	レンジのコンテンツを置き換える

任意の *rangeUpdate* 引数には、以下の定数のいずれか一つを渡し、挿入されたテーブルがレンジに含まれるかどうかを指定します:

定数	型	値	コメント
wk exclude from range	倍長整数	1	更新されたレンジには、以前のコンテンツのみを含める
wk include in range	倍長整数	0	更新されたレンジに挿入コンテンツを含める

rangeUpdate 引数を渡さなかった場合、挿入されたテーブルはデフォルトではレンジに含まれます。

nbCols および *nbRows* 引数を渡して、テーブル内に定義したいカラムの数と行の数を指定することができます。デフォルトでは、引数が省略された場合、0列のカラムと0行を持った新しいテーブルが作成されます。すると **WP Table append row** コマンドを使用してこれらの要素を追加することができるようになります。

このコマンドは新しいテーブルをテーブルオブジェクトとして返し、これはその後他の WP table コマンドへと渡すことができます。テーブルはデフォルトのフォント、境界線、カラーで作成されますが、これらは **WP SET ATTRIBUTES** コマンドを使用してカスタマイズすることが可能です。

例題

空のドキュメントに、3つのカラムと2つの行を持つ新しいテーブルを作成したい場合を考えます:

```
C_OBJECT($WPRange;$WPTable)
$WPRange:=WP Create range(WParea;wk start text;wk end text)
$WPTable:=WP Insert table($WPRange;wk append;wk exclude from range;3;2)
```

結果:

⚙ WP Is font style supported

WP Is font style supported (targetObj ; wpFontStyle) -> 戻り値

引数	型	説明
targetObj	Object	➡ 解析するレンジ、要素への参照
wpFontStyle	倍長整数	➡ フォントスタイル定数
戻り値	ブール	➡ wpFontStyleがサポートされている部分がレンジ内にあればTrue、それ以外はFalse

説明

WP Is font style supported コマンドは引数で指定したレンジオブジェクトのどこかが *wpFontStyle* スタイルをサポートする場合、Trueを返します。

targetObj 引数には有効な4D Write proレンジオブジェクトを渡して下さい。

- レンジ
- 要素(表/行/段落/インライン画像/本文/ヘッダー/フッター)の参照

wpFontStyle 引数には、"4D Write Pro"定数テーマ内にある以下のスタイル定数のうちいずれか一つを渡します:

定数	型	値	コメント
wk font bold	文字列	fontBold	テキストの太さを指定します(利用可能なフォントスタイルによります)。取り得る値は以下の通りです: • <u>wk true</u> を指定すると、選択された文字を太字フォントスタイルへと設定します。WP GET ATTRIBUTESコマンドを使用するとき、選択された文字のうち少なくとも1文字が太字フォントスタイルをサポートする場合、<u>wk true</u>が返されます。 • <u>wk false</u> (デフォルト)を指定すると、選択された文字から太字フォントスタイルを除去します。WP GET ATTRIBUTESコマンドを使用するとき、選択された文字がどれも太字フォントスタイルをサポートしない場合、<u>wk false</u>が返されます。 テキストの、イタリックスタイルを指定します(利用可能なフォントスタイルによります)。取り得る値は以下の通りです: • <u>wk true</u> は選択された文字をイタリック体あるいは斜体スタイルに設定します。WP GET ATTRIBUTESコマンドは選択された文字のうち少なくとも一つ以上の文字がイタリック体あるいは斜体フォントスタイルをサポートする場合、<u>wk true</u>が返されます。 • <u>wk false</u> (デフォルト)は選択された文字のうちイタリック体あるいは斜体フォントスタイルがある場合にはそれらを除去します。コマンドは選択された文字のうちイタリック体あるいは斜体フォントスタイルをサポートする文字がない場合、<u>wk false</u>が返されます。
wk font italic	文字列	fontItalic	

定数 型 値

wk text linethrough style	文 字 列	textLinethroughStyle
---------------------------------	-------------	----------------------

コメント

テキストの打ち消し線のスタイルを(あれば)指定します。取り得る値は以下の通りです:

- wk none (デフォルト): 打ち消し線エフェクトはありません。
- wk solid: 選択されたテキストに対して実線を描画します。
- wk dotted: 選択されたテキストに対して点線を描画します。
- wk dashed: 選択されたテキストに対して破線を描画します。
- wk double: 選択されたテキストに対して二重線を描画します。
- wk semi transparent: 選択されたテキストに対してグレーの線を描画します。他のスタイルと組み合わせる事が可能です。
- wk word: 単語にのみ線を描画します(空白のスペースは描画しません)。他のスタイルと組み合わせる事が可能です。

テキストの下線のスタイルを(あれば)指定します。取り得る値は以下の通りです:

wk text underline style	文 字 列	textUnderlineStyle
-------------------------------	-------------	--------------------

- wk none (デフォルト): 下線エフェクトはありません。
- wk solid: 実線の下線を描画します。
- wk dotted: 点線の下線を描画します。
- wk dashed: 破線の下線を描画します。
- wk double: 二重線の下線を描画します。
- wk semi transparent: グレーの下線を描画します。他の線スタイルと組み合わせる事が可能です。
- wk word: 単語にのみ下線を描画します(空白のスペースは描画しません)。他の線スタイルと組み合わせる事が可能です。

一般的にこのコマンドは、選択されたテキストに基づいたスタイルオプションを提供するポップアップメニューなどのカスタムのインターフェースオブジェクトをデベロッパーが実装する目的で提供されています。

WP New {(source)} -> 戻り値

引数	型	説明
source	文字, BLOB, Object	→ 文字列: 4D HTML ソース BLOB: 4D Write Blob ドキュメント(.4w7/.4wt) あるいは4D Write Pro ドキュメント(.4wp) Object: 4D Write Pro レンジあるいは要素参照
戻り値	Object	↩ 4D Write Pro オブジェクト

説明

WP New コマンドは4D Write Pro オブジェクトを作成して返します。

source 引数を省略した場合、コマンドはデフォルトで空の4D Write Proオブジェクトを返します。

source 引数を使用した場合、新しい4D Write Pro オブジェクトは*source* 引数の中身をコンテンツとして返されます。渡せる内容は以下の通りです:

- 文字列 の引数: この場合、4D HTMLソースを渡します。つまり、[wk web page html 4D](#) オプションを使用した**WP EXPORT VARIABLE** で書き出されたテキストです。このテキストは参照(4Dタグと式)と埋め込まれた画像を含むことができます。
- *blob* 引数: この場合、いかのどちらかを渡す事ができます:
 - BLOBに保存された4D Write Pro(.4wp)フォーマットドキュメント。
4D Write Proドキュメントフォーマットの詳細については、[.4wp ドキュメントフォーマット](#) を参照して下さい。
 - BLOBに読み込まれている以前の4D Writeエリア(.4w7 または .4wt を含んだBlobがサポートされます)。
4D Write Proオブジェクト内で現在サポートされている4D Writeの機能の詳細な一覧については、[4D Write ドキュメントの読み込み](#) の章を参照して下さい。

ディスク上に保存されている4D Write ドキュメント(.4w7 or .4wt) を読み込みたい場合、**WP Import document** コマンドの使用も検討してみてください。

- *object* 引数: この場合、以下のいずれかを渡します。
 - レンジ。
この場合、**WP New** は指定したレンジから作成した新規ドキュメントを返します。指定レンジがドキュメントの全レンジでない場合、最初のセクションだけが書き出され、ブックマークなどが存在する場合には、これらは書き出されないという点に注意してください。
 - 要素(テーブル/行/段落/インラインピクチャー/ヘッダー/フッター)参照
ヘッダーあるいはフッターの参照オブジェクト(**WP Get header** および **WP Get footer**参照)。
この場合、**WP New** は要素の参照のみを本文部分に含んだ新しいドキュメントを返します(行の場合、ドキュメントにはその行を含んだテーブル要素が格納されます)。

返されるオブジェクトは、例えば **WP INSERT DOCUMENT** コマンドに受け渡し可能な、完全なドキュメントです。

例題 1

空の4D Write Proオブジェクトを作成したい場合を考えます:

```
myWPObject:=WP New
```

例題 2

簡単な4D式の参照を含んだ4D Write Proオブジェクトを作成したい場合を考えます:

```
C_TEXT(myText)
myText:="Today is "
ST INSERT EXPRESSION(myText;"string(current date;System date long)";ST End text)
myWPA:=WP New(myText)
```

例題 3

以前作成したテンプレートを使用して4D Write Proエリアを初期化したい場合を考えます:

```
//既存のエリアからテンプレートを書き出し
C_TEXT(wpTemplate)
WP EXPORT VARIABLE(myWPArea;wpTemplate;wk web page html 4D)

// 新規エリアに対してテンプレートを使用
C_OBJECT(myNewWPA)
myNewWPA:=WP New(wpTemplate)
```

例題 4

4Dフィールドに保存されている4D Write ドキュメントを新しい4D Write Proエリア内に読み込みたい場合を考えます:

```
C_OBJECT(wpArea)
wpArea=WP New([Templates]Reference_)
```

例題 5

あらかじめフォーマットされ、それぞれがブックマークとして保存された複数のパーツで構成された、テンプレートドキュメントを定義しました。このテンプレートから任意のブックマークを新規ドキュメントとして抽出し、作成中のドキュメントに挿入することができます。

```
ARRAY TEXT($_BookmarkNames;0)
WP GET BOOKMARKS([TEMPLATES]WP;$_BookmarkNames) // テンプレートからブックマークを取得し
```

ます

```
$targetRange:=WP New //空のドキュメントを作成(これが最終的なドキュメントになります)
```

```
$p:=Find in array($_BookmarkNames;"Main_Header") // 使用したいブックマークを名称で探します
```

```
If($p>0)
```

```
    $Range:=WP Get bookmark range(WParea;$_BookmarkNames{$p}) // ブックマークからレンジを  
取得します
```

```
    $RangeDoc:=WP New($Range) // レンジから新規ドキュメントを作成します
```

```
    WP INSERT DOCUMENT($targetRange;$RangeDoc;wk append+wk freeze expressions) // wk
```

```
append = $RangeDocドキュメントは $targetRange の先頭に挿入されます
```

```
End if
```

WP PRINT

WP PRINT (wpDoc {; printLayout})

引数	型	説明
wpDoc	Object	➡ 4D Write Pro ドキュメント
printLayout	倍長整数	➡ 4D Write Pro ドキュメントの印刷レイアウト: 0 (デフォルト)=4D Write Pro レイアウト、1=HTML WYSIWYG

説明

WP PRINT コマンドは、*wpDoc* で指定した 4D Write Pro ドキュメントの印刷ジョブをローンチするか、あるいは**OPEN PRINTING JOB** と **CLOSE PRINTING JOB** の間に呼び出された場合には、ドキュメントをカレントの印刷ジョブに追加します (64-bit版 4Dのみ。後述参照)。**WP PRINT** は通常の **PRINT SETTINGS** あるいは **SET PRINT OPTION** コマンドで定義された印刷設定を使用しますが、ページの余白に関しては常に 4D Write Pro ドキュメントの設定を使用します。また、**WP PRINT** はカレントのページ設定オプション (ページサイズやページの向きなど) を使用しますが、**WP USE PAGE SETUP** が先に呼び出されていた場合には、指定ドキュメントにおけるこれらの設定を使用します。

任意の *printLayout* パラメーターを使って、印刷出力に HTML WYSIWYG ビューを適用することができます。"4D Write Pro" テーマの次の定数を受け渡すことができます:

定数	型	値	コメント
wk 4D Write Pro layout	倍長整数	0	標準の4D Write Proレイアウト。いくつかの特定のスタイル属性を含める事ができます。
wk html wysiwyg	倍長整数	1	このレイアウトでは、4D Write Proの高度な属性のうち、全てのブラウザで互換性があるもの以外は除去されます。

printLayout を省略した場合のデフォルトは (0) です。

注: **WP PRINT** を使用して印刷した場合、エリアのビュープロパティ (**ビュープロパティの設定** 参照) の設定にかかわらず、4D Write Pro ドキュメントは常にページビューモードのとおり印刷されます。

32-bit版に関する注記

WP PRINT コマンドは 32-bit版の 4D でもサポートされていますが、32-bit版では **OPEN PRINTING JOB** によって開始された 4D のプリントジョブ内で呼び出すことはできません。32-bit版の 4D でプリントジョブ内でコールした場合にはエラーが生成されます。

例題

変数の値に応じて、4D Write Pro ドキュメントを標準レイアウト、あるいは HTML wysiwyg レイアウトで印刷します:

```
// HTML wysiwyg レイアウトまたは4D Write Pro レイアウトで印刷します
If(rb_htmlwysiwyg=1)
  WP PRINT(writeProDoc;wk html wysiwyg)
Else
  WP PRINT(writeProDoc;wk 4D Write Pro layout)
End if
```

⚙ WP RESET ATTRIBUTES

WP RESET ATTRIBUTES (targetObj ; attribName {; attribName2 ; ... ; attribNameN})

引数	型	説明
targetObj	Object	→ 4D Write Pro ドキュメント、レンジ または 要素の参照
attribName	文字	→ 除去したい属性の名前

説明

WP RESET ATTRIBUTES コマンドは引数として渡された4D Write Proレンジ、要素の参照あるいはドキュメント内の、一つ以上の属性の値をリセットします。このコマンドは4D Write Proのどのような内部(文字、段落、ドキュメント、表または画像)の属性でも除去する事ができます。

第一引数には、以下のいずれかを渡すことができます:

- レンジ
- 要素 (ヘッダー / フッター / 本文 / テーブル / 段落 / アンカーされた画像あるいはインライン画像) の参照
- 4D Write Pro ドキュメント

WP RESET ATTRIBUTES コマンドを使用して属性の値が除去されたとき、値はデフォルト値にリセットされます。デフォルト値については[4D Write Pro属性](#)の章に記されています。

引数として渡された要素内にリセットしたい属性が定義されていなかった場合、コマンドは何もしません。

例題

以下のセレクションから複数の属性を除去したい場合を考えます:

Description

The DIALOG command presents the form form to the user. *This command is often used to get information from the user through the use of variables, or to present information to the user, such as options for performing an operation.*

It is common to display the form inside a modal window created with the Open window command.

以下のコードを実行します:

```
$range:=WP Get selection(*;"WParea")
WP RESET ATTRIBUTES($range;wk padding)
WP RESET ATTRIBUTES($range;wk background color)
```

WP RESET ATTRIBUTES(\$range;wk text underline style)

WP RESET ATTRIBUTES(\$range;wk margin)

WP RESET ATTRIBUTES(\$range;wk border style)

その結果ドキュメントは以下ようになります:

Description

The DIALOG command presents the form form to the user. *This command is often used to get information from the user through the use of variables, or to **present information to the user**, such as options for performing an operation.*

It is common to display the form inside a modal window created with the Open window command.

WP SELECT

WP SELECT ({ * ; } wpArea { ; targetObj } | { ; startRange ; endRange })

引数	型	説明
*	演算子	➡ 指定時、wpAreaはフォームオブジェクト名(文字列)。省略時、wpAreaはオブジェクトフィールドまたは変数。
wpArea	文字, Object	➡ フォームオブジェクト名(*指定時)または4D Write Proオブジェクト変数またはフィールド(*省略時)
targetObj	Object	➡ レンジ、または要素の参照
startRange	倍長整数	➡ テキストレンジの開始のオフセット
endRange	倍長整数	➡ テキストレンジの終わりのオフセット

説明

WP SELECT コマンドは *wpArea* 引数で指定した4D Write Proエリアから、*targetObj* に基づいた、あるいは *startRange* と *endRange* で定義された新しいレンジに基づいた新しいセクションを作成します。

任意の * 引数を渡した場合、*wpArea* 引数がフォームオブジェクト名(文字列)であることを指示します。この引数を省略した場合、*wpArea* 引数が4D Write Proオブジェクト変数あるいはフィールドであることを指示します。*wpArea* 引数に有効な4D Write Proエリアが渡されなかった場合、このコマンドは何もしません。

注: このコマンドを使用するには、*wpArea* がフォーム上に表示されている必要があります。

セクションを定義するためには、既存のオブジェクトを *targetObj* 引数に渡すか、範囲を指定する *startRange* / *endRange* のペアを渡します:

- **第一シンタックス:** WP SELECT({ * ; } wpArea ; targetObj)

targetObj は、以下のいずれかです:

- レンジ
- 要素(テーブル/行/段落/アンカーされたピクチャーあるいはインラインピクチャー/ヘッダー/フッター)の参照

- **第二シンタックス:** WP SELECT({ * ; } wpArea ; startRange ; endRange)

startRange と *endRange* 引数にはエリアで選択したい最初の文字と最後の文字の位置に対応する値を渡します。*startRange* 引数に wk start text 定数を渡す事でエリアの最初を定義する事ができ、また *endRange* 引数に wk end text 定数を渡す事でエリアの最後を定義する事ができます。4D Write Proドキュメントは表示されているテキストだけではなく範囲内に含まれるフォーマットタグもあるという点に注意して下さい。

例題

以下のコードは:

```
$range2:=WP Create range([SAMPLE]WP;wk start text;12)  
WP SELECT(*;"WParea";$range2)
```

... このコードと同じ意味になります:

```
WP SELECT(*;"WParea";wk start text;12)
```

WP SET ATTRIBUTES

WP SET ATTRIBUTES (targetObj ; attribName ; attribValue {; attribName2 ; attribValue2 ; ... ; attribNameN ; attribValueN})

引数	型	説明
targetObj	Object	➡ 4D Write Pro ドキュメント、レンジ または 要素の参照
attribName	テキスト	➡ 設定する属性名
attribValue	テキスト, Number, 倍長整数配列, 実数配列, ピクチャー, 日付	➡ 新しい属性の値

説明

WP SET ATTRIBUTESコマンドを使ってレンジ、要素の参照、あるいはドキュメントのあらゆる属性の値を設定することができます。このコマンドを使ってあらゆる4D Write Pro内部の属性(文字、段落、ドキュメント、またはピクチャー)にアクセスすることができます。

第一引数には、以下のいずれかを渡すことができます：

- レンジ
- 要素の参照(ヘッダー/フッター/本文/テーブル/行/段落/アンカーされた画像あるいはインライン画像)
- 4D Write ドキュメント

*attribName*には設定する属性名を、*attribValue*には設定する値を渡します。*attribName*引数に渡す属性の包括的な一覧と、それに対応する値については、[4D Write Pro属性](#)の章を参照してください。

受け渡すことのできる *attribName* / *attribValue* ペアの数に制限はありません。

注: 同じ対象について複数の属性を設定する場合、**WP SET ATTRIBUTES** を複数回呼び出すよりも、一回の呼び出しで複数の*attribName* / *attribValue* ペアを一括指定するほうが処理が最適化されます。

例題 1

この4D Write Proエリアにて、以下のように単語を選択した場合を考えます：

Description

The **DIALOG** command presents the form form to the user. This command is often used to get information from the user through the use of variables, or to present information to the user, such as options for performing an operation.

It is common to display the form inside a modal window created with the Open window command.

Here is a typical example of a dialog:

以下のコードを実行した場合:

```
$range:=WP Get selection(*;"WParea") //選択されたレンジを取得

// 選択されたテキストに対して影のオフセットをpt単位で設定
WP SET ATTRIBUTES($range;wk text shadow offset;1)
//段落のパディングを設定
WP SET ATTRIBUTES($range;wk padding;1)
//10 ptの境界線を定義
WP SET ATTRIBUTES($range;wk border style;wk solid;wk border width;10)
//境界線カラーを設定
WP SET ATTRIBUTES($range;wk border color;"blue";wk border color bottom;"#00FA9A";wk
border color right;"#00FA9A")
```

以下の様な結果が得られます:

Description

The **DIALOG** command presents the form form to the user. This command is often used to get information from the user through the use of variables, or to present information to the user, such as options for performing an operation.

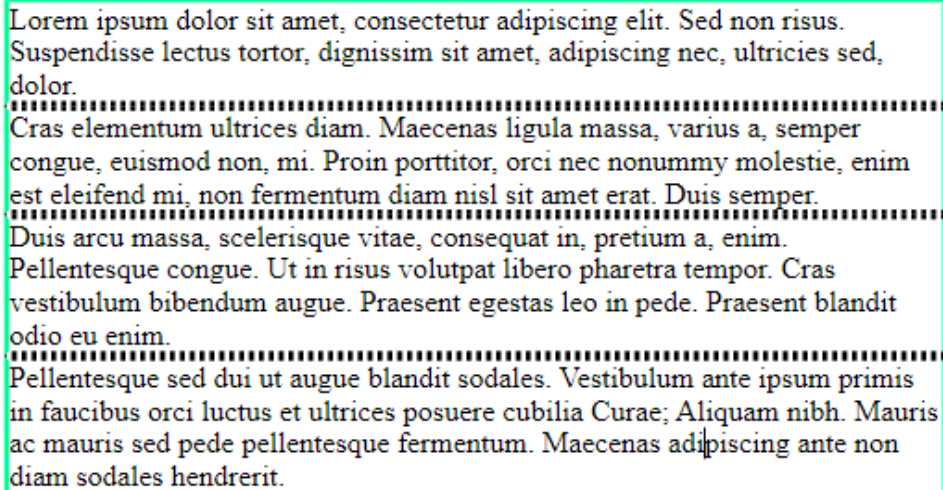
It is common to display the form inside a modal window created with the Open window command.

Here is a typical example of a dialog:

以下の例では、wk insideとwk outside定数の使用を説明します:

```
$wpRange:=WP Get selection(writeProdoc)
WP SET ATTRIBUTES($wpRange;wk border style+wk inside;wk dotted)
WP SET ATTRIBUTES($wpRange;wk border style+wk outside;wk solid)
WP SET ATTRIBUTES($wpRange;wk border color+wk outside;"#00FA9A")
```

ドキュメント全体が選択されていた場合、結果は以下のようになります:



>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus.
Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed,
dolor.
.....
Cras elementum ultrices diam. Maecenas ligula massa, varius a, semper
congue, euismod non, mi. Proin porttitor, orci nec nonummy molestie, enim
est eleifend mi, non fermentum diam nisl sit amet erat. Duis semper.
.....
Duis arcu massa, scelerisque vitae, consequat in, pretium a, enim.
Pellentesque congue. Ut in risus volutpat libero pharetra tempor. Cras
vestibulum bibendum augue. Praesent egestas leo in pede. Praesent blandit
odio eu enim.
.....
Pellentesque sed dui ut augue blandit sodales. Vestibulum ante ipsum primis
in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam nibh. Mauris
ac mauris sed pede pellentesque fermentum. Maecenas adipiscing ante non
diam sodales hendrerit.

例題 3

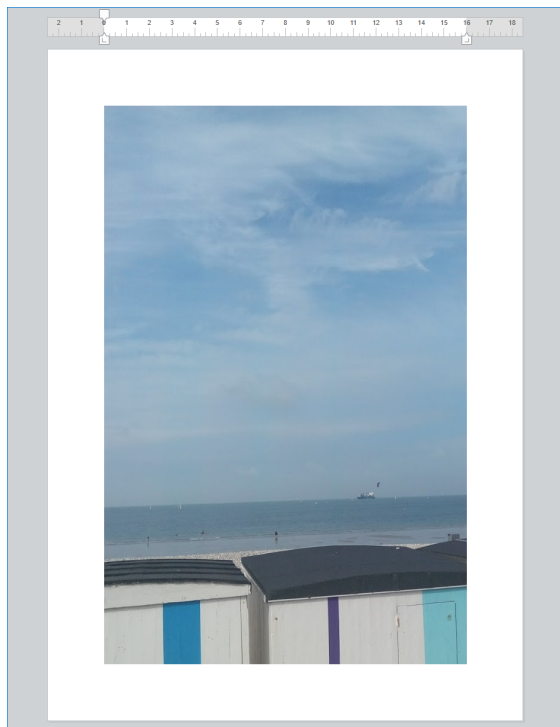
ドキュメントに対して背景画像を設定したい場合を考えます:

```
C_OBJECT(WParea)
WParea:=WP New

READ PICTURE FILE("C:\\Pictures\\boats.jpg";$picture)

WP SET ATTRIBUTES(WParea;wk background image;$picture)
```

結果は以下のようになります:



背景画像を、印刷可能エリア全体に対して設定したい場合を考えます：

```
C_OBJECT(WParea)
```

```
WParea:=WP New
```

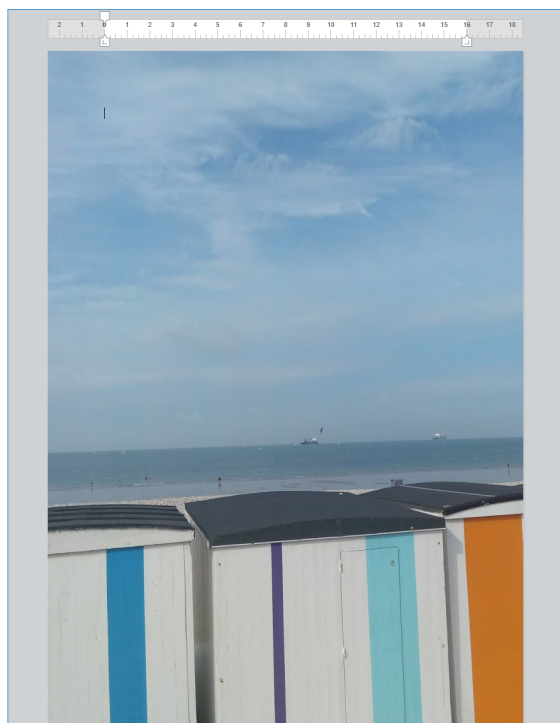
```
READ PICTURE FILE("C:\\Pictures\\boats.jpg";$picture)
```

```
WP SET ATTRIBUTES(WParea;wk background image;$picture)
```

```
WP SET ATTRIBUTES(WParea;wk background clip;wk paper box)
```

```
WP SET ATTRIBUTES(WParea;wk background origin;wk paper box)
```

結果は以下ようになります：



注: paper box の値はドキュメントあるいはセクションに対してのみ適用可能です。

例題 4

異なる間隔のタブを設定し、別々の文字を先頭文字として定義したい場合を考えます:

```
C_OBJECT(WParea)
```

```
ARRAY TEXT(_position;0)
```

```
ARRAY LONGINT(_type;0)
```

```
ARRAY TEXT(_leadings;0)
```

```
APPEND TO ARRAY(_position;"3cm")
```

```
APPEND TO ARRAY(_type;wk left)
```

```
APPEND TO ARRAY(_leadings;".")
```

```
APPEND TO ARRAY(_position;"7.5cm")
```

```
APPEND TO ARRAY(_type;wk right)
```

```
APPEND TO ARRAY(_leadings;".")
```

```
APPEND TO ARRAY(_position;"9cm")
```

```
APPEND TO ARRAY(_type;wk center)
```

```
APPEND TO ARRAY(_leadings;".")
```

```
APPEND TO ARRAY(_position;"12cm")
```

```
APPEND TO ARRAY(_type;wk decimal)
```

```
APPEND TO ARRAY(_leadings;"~")
```

```
APPEND TO ARRAY(_position;"2cm")
```

```
APPEND TO ARRAY(_type;wk left)
```

```
APPEND TO ARRAY(_leadings;"")
```

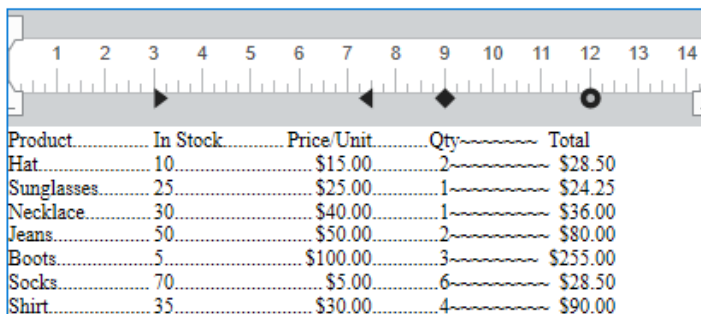
```
$range:=WP Get selection(WParea)
```

```
WP SET ATTRIBUTES($range;wk tab stop offsets;_position)
```

```
WP SET ATTRIBUTES($range;wk tab stop types;_type)
```

```
WP SET ATTRIBUTES($range;wk tab stop leadings;_leadings)
```

結果は以下のようになります:



Product.....	In Stock.....	Price/Unit.....	Qty.....	Total
Hat.....	10.....	\$15.00.....	2.....	\$28.50
Sunglasses.....	25.....	\$25.00.....	1.....	\$24.25
Necklace.....	30.....	\$40.00.....	1.....	\$36.00
Jeans.....	50.....	\$50.00.....	2.....	\$80.00
Boots.....	5.....	\$100.00.....	3.....	\$255.00
Socks.....	70.....	\$5.00.....	6.....	\$28.50
Shirt.....	35.....	\$30.00.....	4.....	\$90.00

WP SET FRAME

WP SET FRAME ({ * ; } wpArea ; frameSelector)

引数	型	説明
*	演算子	→ 指定時、wpArea はフォームオブジェクト(文字列)。省略時、wpArea はオブジェクトフィールドあるいは変数
wpArea	文字	→ フォームオブジェクト名(* 指定時)あるいは4D Write Pro オブジェクト変数あるいはフィールド(* 省略時)
frameSelector	倍長整数	→ カーソルを設定するフレーム数

説明

WP SET FRAME コマンドは *wpArea* 引数で指定した4D Write Pro エリア内の、*frameSelector* 引数で指定したフレーム内にカーソルを設定します。

任意の * 引数を渡した場合、*wpArea* 引数はフォームオブジェクト名(文字列)です。省略時、*wpArea* 引数は4D Write Pro オブジェクト変数あるいはフィールドです。

frameSelector 引数には、**4D Write Pro** テーマ内にある以下の定数のどれか一つを渡すことができます：

定数	型	値	コメント
wk body	倍長 整数	0	セクションのボディフレーム
wk current page footer	倍長 整数	2	
wk current page header	倍長 整数	1	
wk current section default footer	倍長 整数	10	セクションのメインのフッター(アクティブなサブセクション のフッターがない場合にのみ利用可能)
wk current section default header	倍長 整数	9	セクションのメインのヘッダー(アクティブなサブセクション のヘッダーがない場合にのみ利用可能)
wk current section first footer	倍長 整数	4	セクションの最初のページのフッター
wk current section first header	倍長 整数	3	セクションの最初のページのヘッダー
wk current section left footer	倍長 整数	6	セクションの左ページのフッター
wk current section left header	倍長 整数	5	セクションの左ページのヘッダー
wk current section right footer	倍長 整数	8	セクションの右ページのフッター
wk current section right header	倍長 整数	7	セクションの右ページのヘッダー

- *frameSelector* 引数に wk current page header or wk current page footer が渡された場合、**WP SET FRAME** はサブセクションタイプにかかわらず、選択された最初のページのヘッダーあるいはフッターにフォーカスを移します(ページにヘッダーあるいはフッターがない場合、コマンドは何もしません)。
- *frameSelector* 引数に wk current section[...] 系の定数が渡された場合、**WP SET FRAME** は選択された最初のセクション内の指定されたサブセクションヘッダーあるいはフッターにフォーカスを移します(そこにヘッダーあるいはフッターがない場合、コマンドは何もしません)。
- 選択されたセクション内に *frameSelector* 引数で指定されたフレームが存在しない場合、**WP SET FRAME** は何もしません(例えば、*wpArea* 引数で指定したエリア内に左ページと右ページの区別がない場合に渡された[#cst id="3479773"/]) など)。

例題

カレントセクションの左のフッターにカーソルを設定したい場合を考えます:

```
WP SET FRAME(*;"WPArea";wk current section left footer)
```

⚙ WP SET TEXT

WP SET TEXT (wpRange ; newText ; textPosition {; rangeUpdate})

引数	型	説明
wpRange	Object	→ レンジオブジェクト
newText	テキスト	→ 挿入するテキスト
textPosition	Number	→ 挿入位置(置き換え、前、後)
rangeUpdate	Number	→ 挿入されたコンテンツをレンジに含める、あるいは含めない

説明

WP SET TEXT コマンドは、*newText* 引数に渡されたテキストを、4D Write Proドキュメント内の *wpRange* および *textPosition* 引数で指定された位置に標準テキストとして挿入します。.

wpRange 引数には、テキストが挿入される位置を指定するレンジオブジェクトを渡します。レンジは 4D Write Pro ドキュメント内の明示的な位置も取れますし、ユーザーの選択に基づいた位置も選択できます。

newText 引数には *wpRange* 引数で指定したレンジ内に挿入されるテキストを渡します。**WP SET TEXT** は標準テキストのみを挿入します。書かれた通りにテキストを挿入しますが、スタイルタグは除外されます。

textPosition 引数には、挿入されるテキストの位置を指定する、以下のセレクトアーのいずれかを渡すことができます:

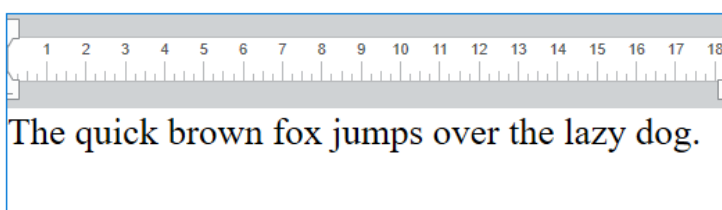
定数	コメント
wk append	コンテンツをレンジの終わりに挿入する
wk prepend	コンテンツをレンジの始めに挿入する
wk replace	レンジのコンテンツを置き換える

任意の *rangeUpdate* 引数には、挿入されたコンテンツが返されたレンジに含まれるかどうかを指定する、以下の定数のいずれかを渡すことができます:

定数	コメント
wk exclude from range	更新されたレンジには、以前のコンテンツのみを含める
wk include in range	更新されたレンジに挿入コンテンツを含める

例題

以下のドキュメントにテキストを追加したい場合を考えます:



既存のテキストの前に挿入:

```
$range:=WP Create range(myDoc;wk start text;wk end text)  
WP SET TEXT($range;" Hello world!";wk prepend)  
//書き換えたあとは " Hello world! The quick brown fox jumps over the lazy dog."となる
```

既存のテキストの後に挿入:

```
$range:=WP Create range(myDoc;wk start text;wk end text)  
WP SET TEXT($range;" Hello world!";wk append)  
//書き換えたあとは "The quick brown fox jumps over the lazy dog. Hello world!"となる
```

既存のテキストを置き換える:

```
$range:=WP Create range(myDoc;wk start text;wk end text)  
WP SET TEXT($range;" Hello world!";wk replace)  
//書き換えたあとは " Hello world!"となる
```

⚙ WP Table append row

WP Table append row (tableRef ; value {; value2 ; ... ; valueN}) -> 戻り値

引数	型	説明
tableRef	Object	→ テーブル参照オブジェクト
value	テキスト, Number, 時間, 日付, ピクチャー	→ 追加行の各セルに設定する値
戻り値	Object	↪ 行レンジオブジェクト

説明

WP Table append row コマンドは *tableRef* で指定したテーブルの最後に新規行を追加し、各セルに *value* で指定した値を設定して、当該行レンジオブジェクトを戻り値として返します。

value に受け渡した引数の件数分のセルが、このコマンドによって追加されます。様々な型の値を引数として渡すことができ、その件数に制限はありません。セル内のデフォルト水平位置は値の型に依存します:

- テキスト: 左揃え
- ピクチャー: 中央揃え
- その他 (数値、日付、時間): 右揃え

注: 配列型の値には対応していません。

このコマンドは新規追加行を行レンジオブジェクトとして返します。

例題 1

新規テーブルを作成し、様々な大きさの行を複数追加します:

```
C_OBJECT($wpTable;$wpRange;$wpRow1;$wpRow2;$wpRow3)
$wpRange:=WP Create range(WParea;wk start text;wk end text)
$wpTable:=WP Insert table($wpRange;wk append)
$wpRow1:=WP Table append row($wpTable;"Paul";"Smith";25)
$wpRow2:=WP Table append row($wpTable;"John";"Richmond";40)
$wpRow3:=WP Table append row($wpTable;"Mary";"Trenton";18;"New!")
```

Paul	Smith	25	
John	Richmond	40	
Mary	Trenton	18	New!

例題 2

請求書において、関連する請求項目のテーブルを自動作成します:

C_OBJECT(\$wpTable;\$wpRange)

\$wpRange:=WP Create range(4DWPArea;wk start text;wk end text)

\$wpTable:=WP Insert table(\$wpRange;wk append) // テーブルを作成します

// ヘッダー行を追加します

\$row:=WP Table append row(\$wpTable;"Name";"Quantity";"Unit Price";"Discount Rate";"Total")

WP SET ATTRIBUTES(\$row;wk font bold;wk true;wk text align;wk center)

// apply to selection コマンドを使います

APPLY TO SELECTION([INVOICE_LINES];**WP Table append row**(\$wpTable;

[INVOICE_LINES]ProductName;[INVOICE_LINES]Quantity;[INVOICE_LINES]ProductUnitPrice;

[INVOICE_LINES]DiscountRate;[INVOICE_LINES]Total))

// フッター行を追加します

\$row:=WP Table append

row(\$wpTable;"Total:";**Sum**([INVOICE_LINES]Quantity);"";"";**Sum**([INVOICE_LINES]Total))

// テーブルの書式を設定します

\$range:=WP Table get columns(\$wpTable;1;5)

WP SET ATTRIBUTES(\$range;wk width;"80pt")

WP SET ATTRIBUTES(\$wpTable;wk font size;10)

Name	Quantity	Unit Price	Discount Rate	Total
Stylo Bille Granger Bright Yellow	7	20	10	126
Stylo Multifonction Voldermont Green	4	5	4	19,2
Stylo Bille Dumbledore Bright Red	4	5	7	18,6
Stylo Bille Rogue Green	3	35	7	97,65
Stylo Plume Granger Yellow	1	30	6	28,2
Stylo Multifonction Voldermont Red	3	15	0	45
Total:	22			334,65

⚙ WP Table get cells

WP Table get cells (tableRef ; startCol ; startRow {; numCols {; numRows}}) -> 戻り値

引数	型		説明
tableRef	Object	➡	テーブル参照オブジェクト
startCol	倍長整数	➡	先頭セルの位置 (カラム)
startRow	倍長整数	➡	先頭セルの位置 (行)
numCols	倍長整数	➡	取得するカラムの数
numRows	倍長整数	➡	取得する行の数
戻り値	Object	↺	指定セルを含む新規レンジ

説明

WP Table get cells コマンドは *tableRef* で指定したテーブルより、*startCol*、*startRow*、*numCols*、および *numRows* で定義したセル範囲を新規レンジオブジェクトとして返します。

startCol および *startRow* パラメーターは取得するセル範囲の先頭セル (左上)の位置を定義します。任意の*numCols* および *numRows* パラメーターを渡すことで、取得するセル範囲のカラム数および行数を指定することもできます。*numRows* が省略された場合は一行だけ取得します。*numCols* と *numRows* パラメーターを両方省略した場合には、一つのセルだけを取得します。

startCol と *numCols*、または *startRow* と *numRows* の和が *tableRef* のカラム数、または行数を超過する場合や、先頭セルを指定する*startCol* および*startRow* の値が *tableRef* のカラム数および行数を超えている場合には、返されたレンジには最後のカラム/行までが含まれます。

例題 1

特定のセル範囲の属性を変更します:

```
C_OBJECT($wpTable;$wpRange;$wpRow1;$wpRow2;$wpRow3;$wpRow4;$wpRow5;$cells)
$wpRange:=WP Create range(WParea;wk start text;wk end text)
```

```
$wpTable:=WP Insert table($wpRange;wk append)
$wpRow1:=WP Table append row($wpTable;"Paul";"Smith";25)
$wpRow2:=WP Table append row($wpTable;"John";"Richmond";40)
$wpRow3:=WP Table append row($wpTable;"Mary";"Trenton";18)
$wpRow4:=WP Table append row($wpTable;"Christopher";"Lee";53)
$wpRow5:=WP Table append row($wpTable;"Henry";"Cartier";42)
```

```
$cells:=WP Table get cells($wpTable;2;2;2;2)
WP SET ATTRIBUTES($cells;wk background color;0x00E0E0E0)
WP SET ATTRIBUTES($cells;wk text color;"blue")
WP SET ATTRIBUTES($cells;wk border color;0x00E000E0)
WP SET ATTRIBUTES($cells;wk text align;wk center)
```

結果:

Paul	Smith	25
John	Richmond	40
Mary	Trenton	18
Christopher	Lee	53
Henry	Cartier	42

例題 2

"5x7"のセルから行の最後までセルのレンジを取得する場合を考えます:

```
$cellRange:=WP Table get cells(tableRef;5;7;MAXLONG;1)
```

"5x7"のセルからカラムの最後までセルのレンジを取得する場合を考えます:

```
$cellRange:=WP Table get cells(tableRef;5;7;1;MAXLONG)
```

"5x7"のセルからテーブルの最後までセルのレンジを取得する場合を考えます:

```
$cellRange:=WP Table get cells(tableRef;5;7;MAXLONG;MAXLONG)
```

⚙ WP Table get columns

WP Table get columns (tableRef ; startCol {; numCols}) -> 戻り値

引数	型		説明
tableRef	Object	➡	テーブル参照オブジェクト
startCol	倍長整数	➡	最初のカラムの位置
numCols	倍長整数	➡	取得するカラムの位置
戻り値	Object	➡	選択されたカラムを含む新しいテーブルレンジ

説明

WP Table get columns コマンドは *startCol* と *numCols* 引数で指定した、*tableRef* 引数のテーブルの新しいセルレンジオブジェクトを返します。

startCol 引数は、返す最初のテーブルを指定し、任意の *numCols* 引数は返すカラムの列数を指定します。*numCols* 引数が省略された場合、単一の *startCol* カラムだけが返されます。

startCol 引数と *numCols* 引数の合計が *tableRef* 引数で参照するテーブルのカラムの列数を超える場合、あるいは *startCol* 引数が *tableRef* 引数で参照するテーブルのカラムの列数を超える場合、返されたレンジには最後のカラムまでが含まれます。

例題 1

テーブルの、最後の2列の属性を変更したい場合を考えます：

```
C_OBJECT($wpTable;$wpRange;$wpRow1;$wpRow2;$wpRow3;$wpRow4;$wpRow5;$cols)
$wpRange:=WP Create range(WParea;wk start text;wk end text)
$wpTable:=WP Insert table($wpRange;wk append)
$wpRow1:=WP Table append row($wpTable;"Paul";"Smith";25)
$wpRow2:=WP Table append row($wpTable;"John";"Richmond";40)
$wpRow3:=WP Table append row($wpTable;"Mary";"Trenton";18)
$wpRow4:=WP Table append row($wpTable;"Christopher";"Lee";53)
$wpRow5:=WP Table append row($wpTable;"Henry";"Cartier";42)

$cols:=WP Table get columns($wpTable;2;2)
WP SET ATTRIBUTES($cols;wk background color;0x00E0E0E0;wk text color;"red";wk
height;"36pt";wk text align;wk center)
```

結果：

Paul	Smith	25
John	Richmond	40
Mary	Trenton	18
Christopher	Lee	53
Henry	Cartier	42

例題 2

10列目のカラムから最後のカラムまでのレンジを取得したい場合を考えます:

WP Table get columns(tableRef;10;MAXLONG)

⚙ WP Table get rows

WP Table get rows (tableRef ; startRow {; numRows}) -> 戻り値

引数	型		説明
tableRef	Object	➡	テーブル参照オブジェクト
startRow	倍長整数	➡	最初の行の位置
numRows	倍長整数	➡	取得したい行数
戻り値	Object	↪	選択された行を含む新しいテーブルレンジ

説明

WP Table get rows コマンドは、*startRow* と *numRows* 引数で指定した、*tableRef* 引数のテーブルの新しいセルレンジオブジェクトを返します。

startRow 引数は、返す最初のテーブルを指定し、任意の *numRows* 引数は返す行数を指定します。

numRows 引数が省略された場合、単一の *startRow* 行だけが返されます。

startRow 引数と *numRows* 引数の合計が *tableRef* 引数で参照するテーブルの行数を超える場合、あるいは *startRow* 引数が *tableRef* 引数で参照するテーブルの行数を超える場合、返されたレンジには最後の行までが含まれます。

例題 1

テーブルの最初の2行に対して特定の背景色を設定し、3行目の境界線を変更したい場合を考えます：

```
C_OBJECT($wpTable;$wpRange;$wpRow1;$wpRow2;$wpRow3;$wpRow4;$wpRow5;$rows;$rows2)
$wpRange:=WP Create range(WParea;wk start text;wk end text)

$wpTable:=WP Insert table($wpRange;wk append)
$wpRow1:=WP Table append row($wpTable;"Paul";"Smith";25)
$wpRow2:=WP Table append row($wpTable;"John";"Richmond";40)
$wpRow3:=WP Table append row($wpTable;"Mary";"Trenton";18)
$wpRow4:=WP Table append row($wpTable;"Christopher";"Lee";53)
$wpRow5:=WP Table append row($wpTable;"Henry";"Cartier";42)

$rows:=WP Table get rows($wpTable;1;2)
WP SET ATTRIBUTES($rows;wk background color;0x00E0F0FF)
$rows2:=WP Table get rows($wpTable;3)
WP SET ATTRIBUTES($rows2;wk border style;wk solid)
WP SET ATTRIBUTES($rows2;wk border width;4)
```

Paul	Smith	25
John	Richmond	40
Mary	Trenton	18
Christopher	Lee	53
Henry	Cartier	42

例題 2

10行目の行から最後の行までのレンジを取得したい場合を考えます:

WP Table get rows(tableRef;10;MAXLONG)

WP USE PAGE SETUP

WP USE PAGE SETUP (wpDoc)

引数	型		説明
wpDoc	Object	➡	4D Write Proドキュメント

説明

WP USE PAGE SETUP コマンドはカレントのプリンターページ設定を、4D Write Proドキュメント属性のページサイズとページの向きに変更します。このコマンドは、カレントのプリンターページ設定を4D Write Proドキュメントページ設定と同期させるために、**WP PRINT**の直前に呼び出される必要があります。

その他の設定は **PRINT SETTINGS** 4Dコマンドによって定義されます。カレントの印刷設定は 4Dセッション全体に対して設定されます。

例題

ドキュメントを印刷する前に、そのサイズと向きをドキュメント内に保存されている値に設定したい場合を考えます：

WP USE PAGE SETUP(writeProDoc)

定数	型	値	コメント
wk 4D Write Pro layout	倍長整数	0	標準の4D Write Proレイアウト。いくつかの特定のスタイル属性を含める事ができます。
wk 4wp	倍長整数	4	4D Write Proドキュメントは、ネイティブなアーカイブフォーマット(zipになったHTMLと、別のフォルダに保存された画像)にて保存されています。4D特有のタグも含まれ、4D式は計算はされていません。このフォーマットは、特に4D Write Proドキュメントをディスク上に損失なく保存してアーカイブする事に適しています。
wk anchor all	倍長整数	-1	<u>wk anchor section</u> 及び <u>wk anchor page</u> 用の画像アンカー値
wk anchor embedded	倍長整数	-2	<u>wk anchor section</u> 用の画像アンカー値
wk anchor first page	倍長整数	-2	<u>wk anchor page</u> 用の画像アンカー値
wk anchor horizontal align	文字列	anchorHorizontalAlign	画像の、原点からの相対的な水平方向揃えを定義します(<u>wk anchor origin</u> 参照)。取り得る値は以下の通りです: • <u>wk left</u> - 左揃え • <u>wk center</u> - 中央揃え(<i>HTMLでは互換性がなく、Webでは画像が表示されません</i>) • <u>wk right</u> - 右揃え <u>wk layout unit</u> からの画像の水平方向のオフセットを、CSS位置文字列あるいは倍長整数(cmまたはptあるいはpixel単位)で表現された長さで定義します。取り得る値: • <u>wk anchor horizontal align</u> に相対的なページの左端、あるいは右端 • 埋め込みモードの本文の左端、あるいは右端(<u>wk anchor section</u> = <u>wk anchor embedded</u>の場合)
wk anchor horizontal offset	文字列	anchorHorizontalOffset	

デフォルトの値 = 0

定数	型	値	コメント
wk anchor layout	文字列	anchorLayout	画像のレイアウト位置を、ページのテキストに対しての相対位置で定義します。取り得る値: • <u>wk behind text</u> - 画像はテキストの背後にアンカーされている • <u>wk in front of text</u> - 画像はテキストの前面にアンカーされている • <u>wk inline with text</u> - 画像はテキストのインライン画像になっている(WP INSERT PICTURE で挿入された画像に対してのデフォルト)。読み込みのみ属性(インライン画像はプログラミングによってアンカー画像へと変換することはできません)。
wk anchor left page	倍長整数	-3	<u>wk anchor page</u>用の画像アンカー値 画像がページ、ヘッダー、フッターのどれに対してアンカーされているかを定義します。取り得る値:
wk anchor origin	文字列	anchorOrigin	• <u>wk paper box</u> (デフォルト) - 画像はページの端に対してアンカーされています。 • <u>wk header box</u> - 画像はドキュメントのヘッダーに対してアンカーされています。ヘッダーが非表示の場合には画像は表示されません。 • <u>wk footer box</u> - 画像はドキュメントのフッターに対してアンカーされています。フッターが非表示の場合には画像は表示されません。 このセレクターは埋め込みモードでは無視されます。

定数

型 値

コメント

画像がアンカーされているページインデックス、あるいはページのタイプを定義します。取り得る値:

wk anchor
page

文
字
列

anchorPage

- wk anchor all - 画像を、wk anchor section で定義されたセクションの全てのページにアンカーします。
- wk anchor embedded - 画像を、埋め込みモードのドキュメント本文にアンカーします(画像は埋め込みモードでのみ表示されます)。
- wk anchor first page - 画像を、wk anchor section で定義されたセクションの最初のページにアンカーします。
- wk anchor left page - 画像を、wk anchor section で定義されたセクションの左ページにアンカーします。
- wk anchor right page - 画像を、wk anchor section で定義されたセクションの右ページにアンカーします。
- 画像をアンカーするページを示す番号(0以上の倍長整数)。この場合、wk anchor section = wk anchor allとなります。セクションへのアンカーは、画像が単一のページにアンカーされている時は無視されません。

注: ページモードでの画像はブラウザでは表示されません。

wk anchor
right page

倍
長
整
数

-4

wk anchor page 用の画像アンカー値

定数	型	値	コメント
			画像がアンカーされているセクションインデックスあるいはセクションのタイプを定義します。取り得る値: • <u>wk anchor all</u> (デフォルト) - 画像をドキュメント内の全てのセクションにアンカーします(画像はページモードでのみ表示されます)。 • <u>wk anchor embedded</u> - 画像を埋め込みモードのドキュメント本文にアンカーします(画像は埋め込みモードでのみ表示されます)。 • 画像をアンカーするセクションを示す番号(1以上の倍長整数、この場合画像はページモードでのみ表示されます)。 注: ページモードでの画像はブラウザでは表示されません。
wk anchor section	文字列	anchorSection	
			画像の、原点からの相対的な垂直方向揃えを定義します(<u>wk anchor origin</u> 参照)。取り得る値は以下の通りです: • <u>wk top</u> - 上揃え • <u>wk center</u> - 中央揃え(<i>HTMLでは互換性がなく、ブラウザでは画像が表示されません</i>) • <u>wk bottom</u> - 下揃え
wk anchor vertical align	文字列	anchorVerticalAlign	
			wk layout unit からの画像の垂直方向のオフセットを、CSS位置文字列あるいは倍長整数(cmまたはptあるいはpixel単位)で表現された長さで定義します。取り得る値: • ページの上、中央、下方向の限界(<u>wk anchor origin</u> 参照) • 埋め込みモードの本文のページの上、中央、下方向の限界(<u>wk anchor section</u> = <u>wk anchor embedded</u> の場合) デフォルト値 = 0
wk anchor vertical offset	文字列	anchorVerticalOffset	
wk append	倍長整数	2	コンテンツをレンジの終わりに挿入する

定数	型	値	コメント
wk armenian	倍 長 整 数	19	伝統的なアルメニア数字スタイルを使用(wk list style type 用の値)
wk author	文 字 列	author	ドキュメントの著者名を指定(文字列)
wk auto	倍 長 整 数	0	プロパティの値(定数)で、適用されると要素の中身またはコンテキストに応じて自動的に適応されます。
wk background clip	文 字 列	backgroundClip	背景の塗りエリアを指定します。取り得る値: • wk border box (デフォルト): 背景は境界線の外側の端まで塗られます。 • wk content box: 背景はコンテンツボックスの内部まで塗られます。 • wk padding box: 背景はパディングの端の外側まで塗られます(もしくはあった場合には境界線の内側まで)。 • wk paper box: 背景は端まで塗られます (ドキュメントあるいはセクションのみ) 要素の背景色を指定します。取り得る値: • CSS カラー("#010101" または "#FFFFFF" または "red") • 4D カラーの倍長整数値(コマンドを参照して下さい) • R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列 ドキュメントのデフォルトは"#FFFFFF" と wk transparentで、段落と画像に対しては"transparent"がデフォルトです。
wk background color	文 字 列	backgroundColor	

定数

型 値

wk background height	文 字 列	backgroundHeight
----------------------------	-------------	------------------

コメント

背景画像の垂直方向のサイズを指定します。取り得る値は以下の通りです:

- wk auto (デフォルト): 背景画像はその高さを保ちます。
- wk contain: 背景画像のアスペクト比を保ったまま、画像全体が表示される範囲で最大のサイズへと拡大/縮小します。このオプションは他のサイズ属性の値も変更します。
- wk cover: 背景画像のアスペクト比を保ったまま、背景全体を画像でカバーできる大きさまで拡大/縮小します。このとき背景画像の一部は見切れる可能性があります。このオプションは他のサイズ属性の値も変更します。
- 定義されたサイズ: 背景画像の垂直方向のサイズを実数値あるいは文字列値を使用して表現します:
 - 実数: サイズはwk layout unitで指定します。
 - 文字列: 連結された値と単位のCSS文字列。例: 12ptは12ポイント、1.5cmは1.5センチを意味します。
最小値: 0pt、最大値: 1000pt。相対値(パーセンテージ、%)はサポートされています。

背景画像参照を指定します。4Dピクチャー変数あるいは式などの有効な画像であれば可能です。

wk background image	文 字 列	backgroundImage
---------------------------	-------------	-----------------

- 返された値(**WP GET ATTRIBUTES**): 画像がネットワークURLを通して定義された場合には、その画像がすでに読み込まれている場合にはターゲット画像が返されます。そうでない場合には空の画像が返されます。

ピクチャーをURLあるいはローカルのURIを通して扱いたい場合にはwk background image urlを使用してください。

定数

型 値

wk background image url	文 字 列	backgroundImageURL
-------------------------------	-------------	--------------------

コメント

URL(文字列)を通して定義された背景画像です。
ネットワークURL あるいはデータURI あるいは
ローカルファイルURLです(絶対パスあるいはス
トラクチャーからの相対パス)。

- 返された値(**WP GET ATTRIBUTES**): ネットワークURL あるいはデータURI。ネットワークURLで参照されたものではない画像に関しては、最初のURLと同じではないことがあります(ネットワークURL のみが保存されます)。ローカルファイルのURLに関しては、画像ストリームそのものはドキュメント内に保管されていて、そのため返されたURLはデータURIと、base64にエンコードされた画像ストリームです。

背景画像をピクチャー式として扱いたい場合には、wk background image を使用してください。

背景画像どのような位置におかれるかを指定します。取りうる値は以下の通りです:

wk background origin	文 字 列	backgroundOrigin
----------------------------	-------------	------------------

- wk padding box (デフォルト): 背景画像はパディング(または内側の境界線の端)からスタートします。
- wk border box: 背景画像は境界線(外側の境界線の端)の四角形からスタートします。
- wk content box: 背景画像はコンテンツ四角形からスタートします。
- wk paper box: 背景画像は端からスタートします(ドキュメントあるいはセクションのみ)。

背景画像の水平方向のスタート位置を指定します。取り得る値は以下の通りです:

wk background position horizontal	文 字 列	backgroundPositionHorizontal
--	-------------	------------------------------

- wk left (デフォルト): 背景画像は水平方向に対して要素の左側からスタートします。
- wk center: 背景画像は水平方向に対して要素の中央からスタートします。
- wk right: 背景画像は水平方向に対して要素の右側からスタートします。

定数	型	値	コメント
wk background position vertical	文 字 列	backgroundPositionVertical	背景画像の垂直方向のスタート位置を指定します。取り得る値は以下の通りです: • <u>wk top</u> (デフォルト): 背景画像は垂直方向に対して要素の上側からスタートします。 • <u>wk middle</u>: 背景画像は垂直方向に対して要素の中央からスタートします。 • <u>wk bottom</u>: 背景画像は垂直方向に対して要素の下側からスタートします。 背景画像をどのように繰り返すか(あるいは繰り返さないか)を指定します。取り得る値は以下の通りです:
wk background repeat	文 字 列	backgroundRepeat	• <u>wk repeat</u> (デフォルト): 背景画像は垂直方向にも水平方向にも繰り返されます。 • <u>wk no repeat</u>: 背景画像は繰り返されません。 • <u>wk repeat x</u>: 背景画像は水平方向にのみ繰り返されます。 • <u>wk repeat y</u>: 背景画像は垂直方向にのみ繰り返されます。

定数	型	値	コメント
wk background width	文字列	backgroundWidth	背景画像の水平方向のサイズを指定します。取り得る値は以下の通りです: • <u>wk auto</u> (デフォルト): 背景画像はその幅を保ちます。 • <u>wk contain</u>: 背景画像のアスペクト比を保ったまま、画像全体が表示される範囲で最大のサイズへと拡大/縮小します。このオプションは他のサイズ属性の値も変更します。 • <u>wk cover</u>: 背景画像のアスペクト比を保ったまま、背景全体を画像でカバーできる大きさまで拡大/縮小します。このとき背景画像の一部は見切れる可能性があります。このオプションは他のサイズ属性の値も変更します。 • 定義されたサイズ: 背景画像の水平方向のサイズは実数値あるいは文字列値を使用して表現されます: ◦ 実数: サイズは<u>wk layout unit</u>で指定します。 ◦ 文字列: 連結された値と単位のCSS文字列。例: 12ptは12ポイント、1.5cmは1.5センチを意味します。 最小値: 0pt、最大値: 1000pt。相対値(パーセンテージ、%)はサポートされています。
wk bar	倍長整数	4	タブ位置に垂直のバーを挿入します(<u>wk tab stop types</u> 用の値)
wk baseline	倍長整数	4	要素のベースラインを親要素のベースラインと合わせる(<u>wk vertical align</u> 用の値)
wk behind text	倍長整数	0	<u>wk anchor layout</u> 用の値
wk body	倍長整数	0	セクションのボディフレーム

定数	型	値	コメント
wk border box	倍長整数	0	背景は境界線ボックスに合わせて見切れます(wk background clip用の値) 境界線4つともにカラーを設定します。取りうる値は以下の通りです: - CSSカラー("#010101"または"#FFFFFF"または"red")。 4Dカラーの倍長整数値(OBJECT SET COLORコマンドを参照して下さい) R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列 デフォルト値は"#000000"です(文字列値の場合)。複数のカラーがある場合、WP GET ATTRIBUTESは空の文字列を返します。 下の境界線のカラーを設定します。取りうる値は以下の通りです: - CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。 4Dカラーの倍長整数値(OBJECT SET COLORコマンドを参照して下さい) R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列 左の境界線のカラーを設定します。取りうる値は以下の通りです: - CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。 4Dカラーの倍長整数値(OBJECT SET COLORコマンドを参照して下さい) R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列
wk border color	文字列	borderColor	
wk border color bottom	文字列	borderColorBottom	
wk border color left	文字列	borderColorLeft	

定数	型	値	コメント
wk border color right	文字列	borderColorRight	右の境界線のカラーを設定します。取りうる値は以下の通りです: - CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。 4Dカラーの倍長整数値(OBJECT SET COLORコマンドを参照して下さい) R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列 上の境界線のカラーを設定します。取りうる値は以下の通りです: - CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。 4Dカラーの倍長整数値(OBJECT SET COLORコマンドを参照して下さい) R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列 角の丸い境界線を指定します。取り得る値は以下の通りです: <u>wk none</u> (デフォルト): 境界線は丸い角を持ちません。 - 数値または文字列値を使用して表現された半径: - 数値: <u>wk layout unit</u>での半径 - 文字列値: 値と単位が連結されたCSS文字列。例: 12ptは12ポイント、1.5cmは1.5センチを意味します。
wk border color top	文字列	borderColorTop	
wk border radius	文字列	borderRadius	

定数	型	値	コメント
			境界線の4辺全てのスタイルを指定します。取り得る値は以下の通りです:
wk border style	文字列	borderStyle	• <u>wk none</u> (デフォルト): 境界線なし • <u>wk hidden</u>: <u>wk none</u>と同様、ただし境界線の競合解決時を除く • <u>wk solid</u>: 実線の境界線 • <u>wk dotted</u>: 点線の境界線 • <u>wk dashed</u>: 破線の境界線 • <u>wk double</u>: 二重境界線 • <u>wk groove</u>: 3D くぼみ境界線(実際の効果は境界線カラーによります) • <u>wk ridge</u>: 3D 隆起境界線(実際の効果は境界線カラーによります) • <u>wk inset</u>: 3D インセット境界線(実際の効果は境界線カラーによります) • <u>wk outset</u>: 3D アウトセット境界線(実際の効果は境界線カラーによります)

			下部の境界線を指定します。取り得る値は以下の通りです:
wk border style bottom	文字列	borderStyleBottom	• <u>wk none</u> (デフォルト): 下部境界線なし • <u>wk hidden</u>: <u>wk none</u>と同様、ただし境界線の競合解決時を除く • <u>wk solid</u>: 実線の下部境界線 • <u>wk dotted</u>: 点線の下部境界線 • <u>wk dashed</u>: 破線の下部境界線 • <u>wk double</u>: 二重下部境界線 • <u>wk groove</u>: 3D くぼみ下部境界線(実際の効果は下部境界線カラーによります) • <u>wk ridge</u>: 3D 隆起下部境界線(実際の効果は下部境界線カラーによります) • <u>wk inset</u>: 3D インセット下部境界線(実際の効果は下部境界線カラーによります) • <u>wk outset</u>: 3D アウトセット下部境界線(実際の効果は下部境界線カラーによります)

定数	型	値	コメント
			左の境界線を指定します。取り得る値は以下の通りです:
wk border style left	文字列	borderStyleLeft	• <u>wk none</u> (デフォルト): 下部境界線なし • <u>wk hidden</u>: <u>wk none</u>と同様、ただし境界線の競合解決時を除く • <u>wk solid</u>: 実線の左境界線 • <u>wk dotted</u>: 点線の左境界線 • <u>wk dashed</u>: 破線の左境界線 • <u>wk double</u>: 二重左境界線 • <u>wk groove</u>: 3D くぼみ左境界線(実際の効果は左境界線カラーによります) • <u>wk ridge</u>: 3D 隆起左境界線(実際の効果は左境界線カラーによります) • <u>wk inset</u>: 3D インセット左境界線(実際の効果は左境界線カラーによります) • <u>wk outset</u>: 3D アウトセット左境界線(実際の効果は左境界線カラーによります)
			右の境界線を指定します。取り得る値は以下の通りです:
wk border style right	文字列	borderStyleRight	• <u>wk none</u> (デフォルト): 右境界線なし • <u>wk hidden</u>: <u>wk none</u>と同様(境界線の競合解決時を除く) • <u>wk solid</u>: 実線の右境界線 • <u>wk dotted</u>: 点線の右境界線 • <u>wk dashed</u>: 破線の右境界線 • <u>wk double</u>: 二重右境界線 • <u>wk groove</u>: 3D くぼみ右境界線(実際の効果は右境界線カラーによります) • <u>wk ridge</u>: 3D 隆起右境界線(実際の効果は右境界線カラーによります) • <u>wk inset</u>: 3D インセット右境界線(実際の効果は右境界線カラーによります) • <u>wk outset</u>: 3D アウトセット右境界線(実際の効果は右境界線カラーによります)

定数	型	値	コメント
			上部境界線を指定します。取り得る値は以下の通りです: • <u>wk none</u> (デフォルト): 上部境界線なし • <u>wk hidden</u>: <u>wk none</u>と同様(境界線の競合解決時を除く) • <u>wk solid</u>: 実線の上部境界線 • <u>wk dotted</u>: 点線の上部境界線 • <u>wk dashed</u>: 破線の上部境界線 • <u>wk double</u>: 二重上部境界線 • <u>wk groove</u>: 3D くぼみ上部境界線(実際の効果は上部境界線カラーによります) • <u>wk ridge</u>: 3D 隆起上部境界線(実際の効果は上部境界線カラーによります) • <u>wk inset</u>: 3D インセット上部境界線(実際の効果は上部境界線カラーによります) • <u>wk outset</u>: 3D アウトセット上部境界線(実際の効果は上部境界線カラーによります)
wk border style top	文字列	borderStyleTop	
			四辺全ての境界線の幅を指定します。境界線幅を指定する前に境界線スタイルを指定する必要があります。取り得る値は以下の通りです: • 数値あるいは文字列値を使用して表現された幅: ◦ 数値: <u>wk layout unit</u>での幅 ◦ 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。 • デフォルト値: 2pt
wk border width	文字列	borderWidth	
			下部境界線の幅を指定します。取り得る値は以下の通りです: • 数値あるいは文字列値を使用して表現された幅: ◦ 数値: <u>wk layout unit</u>での幅 ◦ 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。 • デフォルト値: 2pt
wk border width bottom	文字列	borderWidthBottom	

定数	型	値	コメント
wk border width left	文字列	borderWidthLeft	左境界線の幅を指定します。取り得る値は以下の通りです: - 数値あるいは文字列値を使用して表現された幅: - 数値: <u>wk layout unit</u>での幅 - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。 - デフォルト値: 2pt
wk border width right	文字列	borderWidthRight	右境界線の幅を指定します。取り得る値は以下の通りです: - 数値あるいは文字列値を使用して表現された幅: - 数値: <u>wk layout unit</u>での幅 - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。 - デフォルト値: 2pt
wk border width top	文字列	borderWidthTop	上部境界線の幅を指定します。取り得る値は以下の通りです: - 数値あるいは文字列値を使用して表現された幅: - 数値: <u>wk layout unit</u>での幅 - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。 - デフォルト値: 2pt
wk bottom	倍長整数	1	背景画像の位置を設置(<u>wk background position y</u>の値の場合)、あるいは要素の下部を行での最低の要素に合わせる(<u>wk vertical align</u>の値の場合)
wk capitalize	倍長整数	1	それぞれの単語の最初の文字を大文字に変換します(<u>wk text transform</u>用の値)
wk cell count	文字列	cellCount	行内にあるセルの総数。値の型: 倍超整数(<u>wk type table row</u>用の値)

定数	型	値	コメント
wk center	倍長整数	2	テキストまたは画像を中央に配置します(wk background position h 、 wk text align 、または wk tab stop types 用の値)
wk circle	倍長整数	11	円形の絵文字が使用されます(wk list style type 用の値)
wk cjk ideographic	倍長整数	24	標準の表意文字の数字が使用されます(wk list style type 用の値)
wk club	倍長整数	27	クラブ型の絵文字が使用されます(wk list style type 用の値)
wk column break	倍長整数	3	カラムブレイク:カラムブレイクを挿入します
wk column count	文字列	columnCount	(テーブル、ドキュメント、セクション用の属性) カラムの数。値:倍長整数 • テーブルの場合: 読み込みのみ属性 • ドキュメントあるいはセクションの場合: 読み込み/書き込み可能属性。デフォルト値=1(単一の行)。最大値=20
wk column rule color	文字列	columnRuleColor	垂直方向のカラムルーラーのカラー。取りうる値は以下の通り: • CSSカラー("#010101"あるいは"#FFFFFF"あるいは"red")。デフォルトは"#000000"(black)です。 • 4D カラー倍超整数値(OBJECT SET COLORコマンド参照のこと) • RGBコンポーネントの値をそれぞれ格納している倍超整数配列(0-255)

定数	型	値	コメント
			垂直方向のカラムルーラーのスタイル。取りうる値は以下の通り: • wk none (デフォルト): ルーラーなし • wk hidden: wk none と同じですが、ルーラー干渉の解決の場合を除く • wk solid: 実線のルーラー • wk dotted: 点線のルーラー • wk dashed: 破線のルーラー • wk double: 二重線のルーラー • wk groove: 3Dルーラー(実際のエフェクトはルーラーのカラーによります) • wk ridge: 3Dくぼみルーラー(実際のエフェクトはルーラーのカラーによります) • wk inset: 3Dインセットルーラー(実際のエフェクトはルーラーのカラーによります)
wk column rule style	文字列	columnRuleStyle	
			垂直方向のルーラーの幅。取りうる値は以下の通り: • 実数: wk layout unit で指定した単位での幅 • 文字列: 連結された幅の値と単位(例.: "12pt"で12ポイント、"1.5cm"で1.5cmを指定) デフォルト値="2.5pt"
wk column rule width	文字列	columnRuleWidth	
			(ドキュメントあるいはセクションのみ)二つのカラム間の間隔。取り得る値は以下のとおり: • 実数: wk layout unit 定数で指定した単位での幅 • 文字列: 連結された幅の値と単位(例.: "12pt"で12ポイント、あるいは"1.5cm"で1.5cmを指定) デフォルト値="12pt"
wk column spacing	文字列	columnSpacing	

定数	型	値	コメント
wk column width	文字列	columnWidth	(ドキュメントとセクションのみ)読み込みのみ属性。各カラムのカレントの幅、つまり実際のページ幅、ページマージン、カラム数およびカラム間隔に基づいて計算された幅です。 ドキュメントの場合、デフォルトのセクションカラム幅を使用するので、セクション内で一部属性が上書きされていた場合にはセクションのカラム幅と異なる場合があります。 取りうる値: - 実数: <u>wk layout unit</u> で指定された単位での幅 - 文字列: 連結された幅の値と単位(例: "12pt"で12ポイントを、"1.5cm"で1.5cmを指定します)
wk company	文字列	company	ドキュメントに関連づけられる会社を指定します(文字列)
wk contain	倍長整数	-1	画像の幅と高さがコンテンツエリアの内側に合致する最大のサイズまで画像を拡大/縮小します(<u>wk background size h</u> と <u>wk background size v</u> 用の値)
wk content box	倍長整数	2	背景はコンテンツボックスのサイズで見切れる(<u>wk background clip</u> 用の値)、または背景画像はコンテンツの左上端からスタートする(<u>wk background origin</u> 用の値)
wk cover	倍長整数	-2	背景画像を、背景エリアがその画像で完全に隠れる最小のサイズまで拡大/縮小します(<u>wk background size h</u> または <u>wk background size v</u> 用の値)
wk current page footer	倍長整数	2	
wk current page header	倍長整数	1	
wk current section default footer	倍長整数	10	セクションのメインのフッター(アクティブなサブセクションのフッターがない場合にのみ利用可能)

定数	型	値	コメント
wk current section default header	倍長整数	9	セクションのメインのヘッダー(アクティブなサブセクションのヘッダーがない場合にのみ利用可能)
wk current section first footer	倍長整数	4	セクションの最初のページのフッター
wk current section first header	倍長整数	3	セクションの最初のページのヘッダー
wk current section left footer	倍長整数	6	セクションの左ページのフッター
wk current section left header	倍長整数	5	セクションの左ページのヘッダー
wk current section right footer	倍長整数	8	セクションの右ページのフッター
wk current section right header	倍長整数	7	セクションの右ページのヘッダー
wk custom	倍長整数	29	カスタムのマーカーを使用します(wk list style type 用の値)
wk dashed	倍長整数	3	破線を使用(wk text linethrough style または wk text underline style 用の値)
wk date creation	文字列	dateCreation	ドキュメントの作成日(日付)を返します。この値は読み込みのみで、設定する事はできません。
wk date modified	文字列	dateModified	ドキュメントの最終更新日(日付)を返します。この値は読み込みのみであり、設定する事はできません。

定数	型	値	コメント
wk decimal	倍長整数	3	小数点の桁揃え(wk tab stop types 用の値)または使用される数字(wk list style type 用の値)
wk decimal greek	倍長整数	28	ギリシャ数字が使用されます(wk list style type 用の値)
wk decimal leading zero	倍長整数	13	小数は先頭にゼロがつけられます(wk list style type 用の値)
wk default	倍長整数	-1	プロパティ(定数)のデフォルト値が使用されます。
wk diamond	倍長整数	26	ダイヤ型の絵文字が使用されます(wk list style type 用の値)
wk direction	文字列	direction	段落のテキストの方向を指定します。取り得る値は以下の通りです: • wk left to right (デフォルト) • wk right to left
wk disc	倍長整数	10	黒丸のマーカーが使用されます(wk list style type 用の値)
wk dotted	倍長整数	2	破線あるいは境界線が使用されています(wk border style 、 wk text linethrough style と wk text underline style 用の値)
wk double	倍長整数	4	二重線あるいは境界線が使用されています(wk border style 、 wk text linethrough style と wk text underline style 用の値)
wk dpi	文字列	dpi	内部でのピクセル <-> ポイント変換に使用されるDPI(整数)。常に96です(読み込みのみ)変更あるいは読み出し可能であるドキュメントのカレントのビューのDPIを変更あるいは読み出す"dpi"標準アクションと、この内部属性を混同しないようにしてください。

定数	型	値	コメント
wk end	文 字 列	end	(読み込み専用のレンジ属性)
wk end text	倍 長 整 数	0	ドキュメントの終わりをテキストレンジの終わりに設定します。
wk exclude from range	倍 長 整 数	1	更新されたレンジには、以前のコンテンツのみを含める
wk expressions as source	倍 長 整 数	1	The original string of 4D expression references is returned.
wk expressions as space	倍 長 整 数	2	Each reference is returned as a non-breaking space character.
wk expressions as value	倍 長 整 数	0	4D expression references are returned in their evaluated form. (default if option if not passed).
wk false	倍 長 整 数	0	
wk first column	文 字 列	firstColumn	(読み込みのみ属性)テーブル内にある最初のカラムの数。値:倍長整数
wk first page	倍 長 整 数	1	
wk first row	文 字 列	firstRow	(読み込みのみ属性)テーブル内の最初の行の数。値:倍長整数
wk font	文 字 列	font	FONT STYLE LIST コマンドによって返される、完全なフォント名とスタイル名を指定します。無効なフォント名を設定した場合、コマンドは何もしません。デフォルト値:"Times New Roman"

定数

型 値

コメント

テキストの太さを指定します(利用可能なフォントスタイルによります)。取り得る値は以下の通りです:

wk font
bold

文
字
列

fontBold

- wk true を指定すると、選択された文字を太字フォントスタイルへと設定します。**WP GET ATTRIBUTES**コマンドを使用するとき、選択された文字のうち少なくとも1文字が太字フォントスタイルをサポートする場合、wk trueが返されます。
- wk false (デフォルト)を指定すると、選択された文字から太字フォントスタイルを除去します。**WP GET ATTRIBUTES**コマンドを使用するとき、選択された文字がどれも太字フォントスタイルをサポートしない場合、wk falseが返されます。

wk font
family

文
字
列

fontFamily

wk fontで定義されたフォントファミリー名を指定します。デフォルトの値は"Times New Roman"です。
選択された文字に異なるフォントファミリープロパティが含まれる場合、**WP GET ATTRIBUTES**コマンドは空の文字列を返します。
テキストの、イタリックスタイルを指定します(利用可能なフォントスタイルによります)。取り得る値は以下の通りです:

wk font
italic

文
字
列

fontItalic

- wk true は選択された文字をイタリック体あるいは斜体スタイルに設定します。**WP GET ATTRIBUTES**コマンドは選択された文字のうち少なくとも一つ以上の文字がイタリック体あるいは斜体フォントスタイルをサポートする場合、wk trueが返されます。
- wk false (デフォルト)は選択された文字のうちイタリック体あるいは斜体フォントスタイルがある場合にはそれらを除去します。コマンドは選択された文字のうちイタリック体あるいは斜体フォントスタイルをサポートする文字がない場合、wk falseが返されます。

定数	型	値	コメント
wk font size	文字列	fontSize	テキストのフォントサイズを指定します。取り得る値は以下の通りです(単位はポイントのみ): • 実際の値(デフォルト = 12) • 値と単位が連結されたCSS文字列。例: 12pt (12 ポイントを意味する)
wk footer box	倍長整数	1	<u>wk anchor origin</u> 用の画像アンカー値
wk freeze expressions	倍長整数	64	挿入の瞬間に式を評価します。
wk georgian	倍長整数	20	伝統的なグルジア数字が使用されます(<u>wk list style type</u> 用の値)
wk groove	倍長整数	6	3Dくぼみ境界線が使用されます(<u>wk border style</u> 用の値)
wk header box	倍長整数	0	<u>wk anchor origin</u> 用の画像アンカー値
wk hebrew	倍長整数	21	伝統的なヘブライ数字が使用されます(<u>wk list style type</u> 用の値)

定数	型	値	コメント
wk height	文字列	height	要素の高さを設定します。高さのプロパティには、パディング、境界線、マージンは含まれません。このプロパティは要素のパディング、境界線、マージンの内側のエリアの高さを設定します。取り得る値は以下の通りです: • <u>wk auto</u> (デフォルト): 高さは要素の中身に基づいて決められます。 • 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズです: ◦ 実数: <u>wk layout unit</u>で設定された単位でのサイズ ◦ 文字列: 値と単位が結合されたCSS文字列。例: 12ptは12ポイントを意味し、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 10,000pt <u>wk height</u>属性は(定義されていれば)<u>wk min height</u>によって上書きされます。
wk hidden	倍長整数	5	境界線は使用されません。なし(境界線なし)と同じですが、境界線上での対立があった場合にはこちらが優先されます(<u>wk border style</u> 用の値)
wk hiragana	倍長整数	22	ひらがなの数字が使用されます(<u>wk list style type</u> 用の値)
wk hollow square	倍長整数	25	空の四角の絵文字が使用されます(<u>wk list style type</u> 用の値)
wk html debug	倍長整数	1	フォーマット済みのHTMLコード("整形済みフォーマット")。デバッグが容易。
wk html wysiwyg	倍長整数	1	このレイアウトでは、4D Write Proの高度な属性のうち、全てのブラウザで互換性があるもの以外は除去されます。
wk id	文字列	id	(読み込みのみ属性)要素のID。値:文字列

定数	型	値	コメント
wk image	文字列	image	画像参照を指定します。4Dピクチャー変数あるいは式などの有効な画像であれば可能です。 返された値(WP GET ATTRIBUTES): 画像がネットワークURLを通して定義された場合には、その画像がすでに読み込まれている場合にはターゲット画像が返されます。そうでない場合には空の画像が返されます。 ピクチャーをURLあるいはローカルのURIを通して扱いたい場合には<code>wk image url</code> を使用してください。
wk image alternate text	文字列	imageAlternateText	画像が表示できない際の、画像の代替テキストを指定します。 URL(文字列)を通して定義された背景画像です。ネットワークURL あるいはデータURI (絶対パスあるいはストラクチャーからの相対パス)です。 返された値(WP GET ATTRIBUTES): ネットワークURL あるいはデータURI。ネットワークURLで参照されたものではない画像に関しては、最初のURLと同じではないことがあります(ネットワークURL のみが保存されます)。ローカルファイルのURLに関しては、画像ストリームそのものはドキュメント内に保管されていて、そのため返されたURLはデータURIと、base64にエンコードされた画像ストリームです。 画像をピクチャー式として扱いたい場合には、<code>wk image</code> を使用してください。
wk image url	文字列	imageURL	<code>wk anchor layout</code> 用の画像レイアウト値
wk in front of text	倍長整数	1	
wk include in range	倍長整数	0	更新されたレンジに挿入コンテンツを含める
wk inherit style from paragraph	倍長整数	32	挿入されたコンテンツは、段落のデフォルト文字スタイルを継承します。

定数	型	値	コメント
wk inline with text	倍長整数	-1	wk anchor layout 用の画像レイアウト値
wk inset	倍長整数	8	3Dインセット境界線が使用されます(wk border style 用の値)
wk inside	文字列	Inside	選択されたエリアに複数の段落が含まれる場合、属性は、対応して被っている段落のプロパティに対してのみ適用することを指定します(範囲の外側は含まれません)。境界線、パディング、マージン属性に対してのみ適用可能で、指定された属性に追加されなければなりません。 WP SET ATTRIBUTES コマンドの例2を参照して下さい。
wk justify	倍長整数	5	4D Write Proエリアに対してのみ適用可能です。
wk katakana	倍長整数	23	カタカナの数字が使用されます(wk list style type 用の値)
wk keep paragraph styles	倍長整数	128	移行先の段落スタイルを保持します。
wk layout unit	文字列	userUnit	値が整数あるいは実数として設定・取得された際の一部の属性の次元の単位を指定します。取り得る値は以下の通り: • wk unit cm (デフォルト): センチメートル • wk unit pt: ポイント • wk unit px: ピクセル • wk unit percent (wk line height、wk background size h / wk background size vに対してのみ使用可能) • wk unit mm: ミリメートル • wk unit inch: インチ
wk left	倍長整数	0	テキストまたはタブを左側に揃える(wk text align または wk tab stop types 用の値)か、背景画像の開始位置を設定します(wk background position h 用の値)

定数	型	値	コメント
wk left page	倍長整数	2	
wk left to right	倍長整数	0	左から右へのテキストの読み書き方向が使用されます(wk direction 用の値)
wk line break	倍長整数	0	(同じ段落内での)改行
wk line height	文字列	lineHeight	行間スペースを指定します。取り得る値は以下の通り: • wk normal (デフォルト): テキストサイズに基づいた値を使用 • 数値あるいは文字列値で表現された高さを使用: ◦ 数値: wk layout unit の単位での高さ ◦ 文字列値: 値と単位が結合されたCSS 文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。相対値(パーセンテージ%)はサポートされています。 レンジに設定されたハイパーリンクです。取りうる値は以下のとおり: • 絶対URL。例: "http://www.4d.com/" • 相対URL。例: "/test/page.html" (データベースストラクチャーファイルを起点とします) • 空文字列 = リンクなし
wk link url	文字列	linkUrl	
wk list auto	倍長整数	2147483647	自動リストスタイル値を保存・適用します。
wk list font	文字列	listFont	FONT STYLE LIST コマンドで返される、完全なフォント名を指定してリスト項目マーカーを表示するようにします(しかし段落のテキストは表示しません)。システムがフォント名を認識しない場合、その代理を使用します。無効なフォント名を設定した場合、コマンドは何もしません。デフォルト値: "Times"

定数	型	値
wk list font family	文字列	listFontFamily
wk list start number	文字列	listStartNumber

wk list string format LTR	文字列	listStringFormatLtr
---------------------------	-----	---------------------

wk list string format RTL	文字列	listStringFormatRtl
---------------------------	-----	---------------------

コメント

wk list fontによって定義されたフォントファミリー名を使用してリスト項目マーカーを表示します(しかし段落のテキストは表示しません)。デフォルト値は"Times New Roman"です。

並べられたリストの始まる値を設定します。取り得る値は以下の通りです:

- wk auto (デフォルト): 始まる値を0に設定します。
- 整数値: 始まる値

左から右への段落方向二対してのリスト項目マーカー文字列フォーマット。定義されると、リストのデフォルトのリスト項目マーカー文字列を上書きします。

- 順番に並んでいないリスト: 文字列がリスト項目マーカーとして使用されます(通常は単一文字による文字列、例: "-")
- 順番に並んでいるリスト: "#"文字を含んだ文字列。"#"は計算された数字や文字のプレースホルダーになります。デフォルトは"#."なので、例えばカレントのリスト項目番号が15でリストスタイルタイプが小数だった場合、リスト項目マーカー文字列は"15."になります。

右から左への段落方向に対してのリスト項目マーカー文字列フォーマット。定義されると、リストのデフォルトのリスト項目マーカー文字列を上書きします。

- 順番に並んでいないリスト: 文字列がリスト項目マーカーとして使用されます(通常は単一文字による文字列、例: "-")
- 順番に並んでいるリスト: "#"文字を含んだ文字列。"#"は計算された数字や文字のプレースホルダーになります。デフォルトは"#."なので、例えばカレントのリスト項目番号が15でリストスタイルタイプが小数だった場合、リスト項目マーカー文字列は"15."になります。

定数	型	値	コメント
			番号なしリストにおいて、画像をリスト項目マーカーとして指定します。取り得る値は以下の通りです:
wk list style image	文字列	listStyleImage	• <u>wk none</u> (デフォルト): リスト項目マーカーは画像によっては定義されていません。 • 4Dピクチャー変数あるいは式などの有効な画像 • 返された値(WP GET ATTRIBUTES): 画像がネットワークURLを通して定義された場合には、その画像がすでに読み込まれている場合にはターゲット画像が返されます。そうでない場合には空の画像が返されます。
			ピクチャーをURLあるいはローカルのURIを通して扱いたい場合には <u>wk list style image url</u> を使用してください。
			リスト項目マーカーとして使用される画像の高さを設定します。取り得る値は以下の通りです:
wk list style image height	文字列	listStyleImageHeight	• <u>wk auto</u> (デフォルト): 高さは画像のサイズに基づいて決められます。 • 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズ: ◦ 実数: <u>wk layout unit</u>の単位でのサイズ ◦ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 1000pt。

定数	型	値	コメント
			URL(文字列)を通して定義された、番号なしリストのリスト項目マーカの画像。取り得る値:
			• <u>wk none</u> (デフォルト): リスト項目マーカは画像によって定義されていません。 • ネットワークURL あるいはデータURI(絶対パスあるいはストラクチャーからの相対パス) • 返された値(WP GET ATTRIBUTES): ネットワークURL あるいはデータURI。ネットワークURLで参照されたものではない画像に関しては、最初のURLと同じではないことがあります(ネットワークURL のみが保存されます)。ローカルファイルのURLに関しては、画像ストリームそのものはドキュメント内に保管されていて、そのため返されたURLはデータURIと、base64にエンコードされた画像ストリームです。
wk list style image url	文字列	imageURL	リスト項目マーカ画像をピクチャー式として扱いたい場合には、 <u>wk list style image</u> を使用してください。

定数	型	値	コメント
wk list style type	文字列	listStyleType	順番付けされた、あるいは順番付けされていないリスト項目マーカのタイプを指定します。取り得る値は以下の通りです: • wk disc (デフォルト) • wk circle • wk square • wk decimal: 1 2 3 • wk decimal leading zero: 01 02 03 • wk lower latin: a b c • wk lower roman: i ii iii iv • wk upper latin: A B C • wk upper roman: I II III IV • wk lower greek: α、β、γ、等 • wk armenian • wk georgian • wk hebrew • wk hiragana • wk katakana • wk cjk ideographic • wk hollow square • wk diamond • wk club • wk decimal greek • wk custom: 順番付けされていないリストで "-" をデフォルトのリスト項目マーカとして使用します。これは標準のリスト項目マーカを変更する事なく wk list string format LTR または wk list string format RTL のリスト項目マーカをカスタマイズするのに便利なスタイルです。 • wk none
wk lower greek	倍長整数	18	小文字の伝統的なギリシャ文字が使用されます (wk list style type 用の値)
wk lower latin	倍長整数	14	小文字のASCII文字が使用されます (wk list style type 用の値)
wk lower roman	倍長整数	15	小文字のローマ数字が使用されます (wk list style type 用の値)

定数	型	値	コメント
wk lowercase	倍長整数	2	全ての文字を小文字に変換します(<u>wk text transform</u>用の値) 要素の全てのマージンのサイズを指定します。取り得る値は以下の通りです: - 数値あるいは文字列値を使用して表現されたサイズ: - 数値: <u>wk layout unit</u>の単位でのサイズ - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。 <u>wk none</u> (デフォルト): 特にマージンを指定しません。
wk margin	文字列	margin	
wk margin bottom	文字列	marginBottom	要素の下部のマージンのサイズを指定します。取り得る値は以下の通りです: - 数値あるいは文字列値を使用して表現されたサイズ: - 数値: <u>wk layout unit</u>の単位でのサイズ - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。 <u>wk none</u> (デフォルト): 特にマージンを指定しません。
wk margin left	文字列	marginLeft	要素の左のマージンのサイズを指定します。取り得る値は以下の通りです: - 数値あるいは文字列値を使用して表現されたサイズ: - 数値: <u>wk layout unit</u>の単位でのサイズ - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。 <u>wk none</u> (デフォルト): 特にマージンを指定しません。

定数	型	値	コメント
wk margin right	文字列	marginRight	要素の右のマージンのサイズを指定します。取り得る値は以下の通りです: - 数値あるいは文字列値を使用して表現されたサイズ: - 数値: wk layout unitの単位でのサイズ - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。 wk none (デフォルト): 特にマージンを指定しません。
wk margin top	文字列	marginTop	要素の上部のマージンのサイズを指定します。取り得る値は以下の通りです: - 数値あるいは文字列値を使用して表現されたサイズ: - 数値: wk layout unitの単位でのサイズ - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。 wk none (デフォルト): 特にマージンを指定しません。
wk middle	倍長整数	2	背景画像の位置を設定(wk background position y用の値)、あるいは要素を親要素の中央に配置します(wk vertical align用の値)。
wk mime html	倍長整数	1	4D Write Proドキュメントは標準のMIME HTMLとして保存され、htmlドキュメントと画像はMIMEパーツとして埋め込まれます(base64でエンコードされます)。式は計算され4D特有のタグは除去されます。このフォーマットは コマンドを使用してHTML Eメールを送信するのに特に適しています。

定数	型	値	コメント
wk min height	文字列	minHeight	要素の最小限の高さを設定します。これは<code>wk height</code>プロパティの値が<code>wk min height</code>プロパティより小さくなることを防ぎます。取り得る値は以下の通りです: • <code>wk auto</code> (デフォルト): 最小限の高さは要素の中身に基づいて決められます。 • 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズ: ◦ 実数: <code>wk layout unit</code>の単位でのサイズ ◦ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 10,000pt。 <code>wk min height</code>の値は<code>wk height</code>属性を上書きします。 注: この属性は行、カラム、セルにおいてはサポートされていません。 要素の最小限の幅を設定します。これは<code>wk width</code>プロパティの値が<code>wk min width</code>プロパティより小さくなることを防ぎます。取り得る値は以下の通りです: • <code>wk auto</code> (デフォルト): 最小限の幅は要素の中身に基づいて決められます。 • 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズ: ◦ 実数: <code>wk layout unit</code>の単位でのサイズ ◦ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 10,00pt。 <code>wk min width</code>の値は<code>wk width</code>属性を上書きします。
wk min width	文字列	minWidth	
wk mixed	倍長整数	-2147483648	レンジまたはドキュメント内のある属性に対して異なる値がある場合に返されます。

定数	型	値	コメント
wk new line style sheet	文字列	newLineStyleSheet	段落に新たな行を追加する際のスタイルシートを指定します。取り得る値は以下の通りです: - 既存のスタイルシート名 <u>wk none</u> (デフォルト)
wk no repeat	倍長整数	3	背景画像は繰り返されません(<u>wk background repeat</u> 用の値)
wk none	倍長整数	0	
wk normal	倍長整数	0	標準のHTML コード
wk notes	文字列	notes	ドキュメントについてのコメントを指定します(文字列)
wk outset	倍長整数	9	3Dアウトセット境界線が使用されます(<u>wk border style</u> 用の値)
wk outside	文字列	Outside	選択されたエリアが複数の段落を含むとき、ある属性が対応する段落の外側のプロパティにのみ適用されることを指定します(内側には適用されません)。これは境界線、パディング、そしてマージン属性に対してのみ指定可能で、特定の属性に対して追加される必要があります。WP SET ATTRIBUTESコマンドの例2を参照して下さい。
wk owner	文字列	owner	(読み込み専用のレンジ属性)

定数	型	値	コメント
wk padding	文字列	padding	要素の全てのサイドのパディングを指定します。取り得る値は以下の通りです: - 数値あるいは文字列値を使用して表現されるサイズ: - 数値: wk layout unitの単位でのサイズ - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。 wk none (デフォルト): 特定のパディングを指定しません。
wk padding bottom	文字列	paddingBottom	要素の下部のパディングを指定します。取り得る値は以下の通りです: - 数値あるいは文字列値を使用して表現されるサイズ: - 数値: wk layout unitの単位でのサイズ - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。 wk none (デフォルト): 特定のパディングを指定しません。
wk padding box	倍長整数	1	背景はパディングボックスのサイズに切り取られます(wk background clip用の値)、あるいは、背景画像はパディングの左上端の際から開始されます(wk background origin用の値)。 要素の左側のパディングを指定します。取り得る値は以下の通りです:
wk padding left	文字列	paddingLeft	- 数値あるいは文字列値を使用して表現されるサイズ: - 数値: wk layout unitの単位でのサイズ - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。 wk none (デフォルト): 特定のパディングを指定しません。

定数	型	値	コメント
wk padding right	文字列	paddingRight	要素の右側のパディングを指定します。取り得る値は以下の通りです: - 数値あるいは文字列値を使用して表現されるサイズ: - 数値: <u>wk layout unit</u>の単位でのサイズ - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。 <u>wk none</u> (デフォルト): 特定のパディングを指定しません。
wk padding top	文字列	paddingTop	要素の上部のパディングを指定します。取り得る値は以下の通りです: - 数値あるいは文字列値を使用して表現されるサイズ: - 数値: <u>wk layout unit</u>の単位でのサイズ - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。 <u>wk none</u> (デフォルト): 特定のパディングを指定しません。
wk page break	倍長整数	2	改ページ: 新しいページを定義します。 Page height (in page mode) expressed using a real or string value (Read-only with section or sub-section). Possible values:
wk page height	文字列	pageHeight	- Real: Height in <u>wk layout unit</u>. - String: CSS string with value and unit concatenated (e.g.: "12pt" for 12 points, or "1.5cm" for 1.5 centimeters). Supported units: pt,cm,mm, inches.
wk paper box	倍長整数	3	背景は端まで描画される(<u>wk background clip</u>用の値)か、あるいは背景画像は端から描画される(<u>wk background origin</u>用の値)。ドキュメント用あるいはセクション用のみ。

定数	型	値	コメント
wk prepend	倍 長 整 数	1	コンテンツをレンジの始めに挿入する
wk repeat	倍 長 整 数	0	背景画像は垂直方向にも水平方向にも繰り返されます(wk background repeat 用の値)
wk repeat x	倍 長 整 数	1	背景画像は水平方向にのみ繰り返されます(wk background repeat 用の値)
wk repeat y	倍 長 整 数	2	背景画像は垂直方向にのみ繰り返されます(wk background repeat 用の値)
wk replace	倍 長 整 数	0	レンジのコンテンツを置き換える
wk ridge	倍 長 整 数	7	3D 隆起境界線が使用されます(wk border style 用の値)
wk right	倍 長 整 数	1	テキストまたはタブを右揃えにする(wk text align または wk tab stop types 用の値)、あるいは背景画像の開始位置を設定します(wk background position h 用の値)
wk right page	倍 長 整 数	3	
wk right to left	倍 長 整 数	1	右から左への方向が使用されます(wk direction 用の値)
wk row count	文 字 列	rowCount	(読み込みのみ属性)行の総数。値:倍長整数
wk section break	倍 長 整 数	1	セクションブレイク:新しいセクションを定義します。

定数	型	値	コメント
wk semi transparent	倍長整数	5	半透明の線が使用されます(<u>wk text linethrough style</u> または <u>wk text underline style</u> 用の値)
wk small uppercase	倍長整数	4	全ての文字を小さな大文字(スモールキャップス)に変換します(<u>wk text transform</u> 用の値)
wk solid	倍長整数	1	実線、あるいは境界線が使用されます(<u>wk border style</u> , <u>wk text linethrough style</u> または <u>wk text underline style</u> 用の値)
wk square	倍長整数	12	四角のマーカーが使用されます(<u>wk list style type</u> 用の値)
wk start	文字列	start	(読み込み専用のレンジ属性)
wk start text	倍長整数	1	ドキュメントの最初をテキストレンジの初めに設定します。
wk style sheet	文字列	styleSheet	選択された要素に対してカレントのスタイルシートを指定します。取り得る値は以下の通りです: • <u>wk none</u> (デフォルト) • 既存のスタイルシート名
wk subject	文字列	subject	ドキュメントの題名を指定します(文字列)
wk subscript	倍長整数	6	要素を下付きのテキストとして配置させます(<u>wk vertical align</u> 用の値)
wk superscript	倍長整数	5	要素を上付きのテキストとして配置させます(<u>wk vertical align</u> 用の値)

定数	型	値	コメント
			段落レンジにおけるTabの先頭文字を指定します。
			取り得る値:
			- テキスト配列 各Tabオフセット用の先頭文字が順番に並んだリスト(左マージンからスタート)。最後の値によって定義された文字がそれぞれの段落内で入力された追加のタブに対して繰り返し使用されます。 - テキスト 単一の文字が先頭文字として使用されます。二つ以上の文字が指定されていた場合には、最初のものだけが使用されます。
wk tab stop leadings	文字列	tabStopLeadings	<u>wk tab stop offsets</u> および <u>wk tab stop types</u> 配列と同期しています。配列のサイズが(空でなくて)合致しない場合、配列の最後の文字がTabのデフォルトとして使用されます。配列が空の場合、なしの文字が使用されます。
			注: 最後に使用された文字が常にTabのデフォルト文字となります。

定数

型 値

コメント

段落のタブストップを指定します。取り得る値は以下の通りです:

- タブ値の配列: 絶対値の順番付けされたりストで、左のマージンから始まります。最後の値によって定義されたタブオフセットは、段落内で入力される追加のタブ文字にて繰り返し使用されます。タブオフセットが段落の幅より大きい場合、テキストは次の行に移行し、最初のタブ値から開始されます。配列内の値が以前の値より小さい場合、それは無視されます。
- スカラー値(デフォルトは35.45pt): 段落全体のデフォルトのオフセットです。**WP GET ATTRIBUTES** コマンドは最後に使用されたオフセットを返します(最後の絶対オフセットよりあとのオフセットについては、デフォルトの相対オフセットになります)。

wk tab stop
offsets

文
字
列

tabStopOffsets

Note: 同じ呼び出しにおいて異なる属性に対して配列とスカラーを同時に使用することはできません。

値はCSS文字列(デフォルト)または`wk layout unit`での単位を使用する実数値で表現されます。最大値は10,000ptです。

`wk tab stop leadings` および `wk tab stop types` 配列と同期しています。

定数

型 値

コメント

段落でのタブストップタイプを指定します。取り得る値は以下の通りです:

wk tab stop
types 文
 字
 列 tabStopTypes

- タブストップ値の配列(タブストップが配列を通して定義されていた場合)
- wk left (デフォルト): テキストはタブストップから右側に拡張されます。
- wk right: タブスペースが埋まるまで、テキストはタブストップから左側に拡張されます。
- wk center: テキストはタブストップにて中央に配置されます。
- wk decimal: 小数点より前のテキストは左に拡張され、小数点より後のテキストは右に拡張されます。
- wk bar: 指定された位置に垂直の線がおかれます。

wk tab stop offsets および wk tab stop leadings 配列と同期しています。
テーブル要素の水平方向並びを指定します。取り得る値は以下の通り:

wk table
align 文
 字
 列 tableAlign

- wk left (デフォルト)
- wk right
- wk center

wk table ID 文
 字
 列 tableID

(読み込みのみ属性)親テーブルのID。値:文字列

wk text
align 文
 字
 列 textAlign

段落内でのテキストの行揃えを指定します。取り得る値は以下の通りです:

- wk left (デフォルト)
- wk right
- wk justify
- wk center

定数	型	値	コメント
wk text color	文字列	color	テキストのカラーを指定します。取り得る値は以下の通りです: - CSSカラー("#010101" や "#FFFFFF" または "red")。デフォルトは文字列の場合"#000000"です。 4Dカラー倍長整数値(OBJECT SET COLORコマンドを参照して下さい) R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列 段落の最初の行のインデントを指定します。取り得る値は以下の通りです: - 実数: <u>wk layout unit.</u>の単位でのサイズ。デフォルトは0です。 - 文字列: 値と単位が連結されたCSS文字列。 例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 10000pt テキストの打ち消し線のカラーを指定します。取り得る値は以下の通りです: - CSSカラー("#010101"や"#FFFFFF" または "red") 4Dカラー倍長整数値(OBJECT SET COLORコマンドを参照してください) R、G、B、コンポーネントそれぞれの要素(0-255)を格納した倍長整数配列 デフォルトは、文字列なら"currentColor"、あるいは倍長整数なら<u>wk default</u> です。
wk text indent	文字列	textIndent	
wk text linethrough color	文字列	textLinethroughColor	

定数	型	値	コメント
			テキストの打ち消し線のスタイルを(あれば)指定します。取り得る値は以下の通りです:
wk text linethrough style	文 字 列	textLinethroughStyle	• <u>wk none</u> (デフォルト): 打ち消し線エフェクトはありません。 • <u>wk solid</u>: 選択されたテキストに対して実線を描画します。 • <u>wk dotted</u>: 選択されたテキストに対して点線を描画します。 • <u>wk dashed</u>: 選択されたテキストに対して破線を描画します。 • <u>wk double</u>: 選択されたテキストに対して二重線を描画します。 • <u>wk semi transparent</u>: 選択されたテキストに対してグレーの線を描画します。他のスタイルと組み合わせる事が可能です。 • <u>wk word</u>: 単語にのみ線を描画します(空白のスペースは描画しません)。他のスタイルと組み合わせる事が可能です。
			選択されたテキストのシャドウカラーを指定します。取り得る値は以下の通りです:
wk text shadow color	文 字 列	textShadowColor	• CSSカラー("#010101" や "#FFFFFF" または "red") • 4Dカラー倍長整数値(OBJECT SET COLORコマンドを参照して下さい) • R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列 • <u>wk transparent</u> (デフォルト)
wk text shadow offset	文 字 列	textShadowOffset	シャドウエフェクトのオフセットを指定します。 取り得る値: • ポイント単位で指定されたサイズ。デフォルト値: 1pt

定数	型	値	コメント
wk text transform	文字列	textTransform	テキスト内での大文字と小文字を指定します。取り得る値は以下の通り: • <u>wk capitalize</u>: 最初の文字は大文字に設定されます。 • <u>wk lowercase</u>: 文字は小文字に設定されます。 • <u>wk uppercase</u>: 文字は大文字に設定されます。 • <u>wk small uppercase</u>: 文字はスモールキャップスに設定されます。 • <u>wk none</u> (デフォルト): 変換なし テキストの下線のカラーを指定します。取り得る値: • CSS カラー("#010101" や "#FFFFFF" または "red") • 4Dカラー倍長整数値(OBJECT SET COLORコマンドを参照して下さい) • R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列 デフォルトは、文字列なら"currentColor"、あるいは倍長整数なら <u>wk default</u> です。 テキストの下線のスタイルを(あれば)指定します。取り得る値は以下の通りです: • <u>wk none</u> (デフォルト): 下線エフェクトはありません。 • <u>wk solid</u>: 実線の下線を描画します。 • <u>wk dotted</u>: 点線の下線を描画します。 • <u>wk dashed</u>: 破線の下線を描画します。 • <u>wk double</u>: 二重線の下線を描画します。 • <u>wk semi transparent</u>: グレーの下線を描画します。他の線スタイルと組み合わせる事が可能です。 • <u>wk word</u>: 単語にのみ下線を描画します(空白のスペースは描画しません)。他の線スタイルと組み合わせる事が可能です。
wk text underline color	文字列	textUnderlineColor	
wk text underline style	文字列	textUnderlineStyle	
wk title	文字列	title	ドキュメントのタイトル(文字列)を指定します。デフォルトは"New 4D Write Pro Document"です。

定数	型	値	コメント
wk top	倍長整数	0	背景画像の位置を指定する(wk background position v 用の値)、あるいは要素を行上の最も高い要素の上に揃えます(wk vertical align 用の値)。
wk transparent	倍長整数	-1	透明色を指定します(wk background color あるいは wk text shadow color 用の値)
wk true	倍長整数	1	
wk type	文字列	type	(読み込みのみ可能なレンジ属性) 4D Write Pro オブジェクトの型。取りうる値: • wk type default: 未定義の型のレンジ • wk type paragraph: 段落型のレンジ • wk type image: 画像(アンカーされた画像およびインライン画像) • wk type image anchored: アンカーされた画像 • wk type image inline: インライン画像 • wk type container: 例えばヘッダーやフッターなど • wk type table: 表参照 • wk type table row: 表の行参照 • wk type table cell: 表のセル参照 • wk type table column: 表のカラム参照(カラムレンジに対してのみ) • wk type body: 本文参照
wk type body	倍長整数	8	本文参照(wk type 用の値)
wk type container	倍長整数	3	ヘッダーまたはフッター(wk type 用の値)
wk type default	倍長整数	0	デフォルトのレンジ型

定数	型	値	コメント
wk type image	倍 長 整 数	2	アンカーされた画像/インライン画像
wk type image anchored	倍 長 整 数	200	ドキュメントオブジェクトに対するアンカーされ た画像の参照
wk type image inline	倍 長 整 数	100	ドキュメントオブジェクトに対するインライン画 像の参照
wk type paragraph	倍 長 整 数	1	段落タイプレンジ
wk type table	倍 長 整 数	4	テーブル参照
wk type table cell	倍 長 整 数	6	テーブルのセル参照(<u>wk type</u> 用の値)
wk type table column	倍 長 整 数	7	カラム参照 - カラムレンジに対してのみ(<u>wk type</u> 用の値)
wk type table row	倍 長 整 数	5	テーブルの行参照
wk unit cm	文 字 列	cm	使用される単位はセンチメートルです(<u>wk layout unit</u> 用の値)
wk unit inch	文 字 列	in	使用される単位はインチです(<u>wk layout unit</u> 用の値)
wk unit mm	文 字 列	mm	使用される単位はミリメートルです(<u>wk layout unit</u> 用の値)
wk unit percent	文 字 列	%	使用される単位はパーセンテージです(<u>wk layout unit</u> 用の値)

定数	型	値	コメント
wk unit pt	文字列	pt	使用される単位はポイントです(wk layout unit 用の値)
wk unit px	文字列	px	使用される単位はピクセルです(wk layout unit 用の値)
wk upper latin	倍長整数	16	大文字のASCII文字が使用されます(wk list style type 用の値)
wk upper roman	倍長整数	17	大文字のローマ数字が使用されます(wk list style type 用の値)
wk uppercase	倍長整数	3	全ての文字を大文字に変換します(wk text transform 用の値)
wk value unit not percentage	倍長整数	-100000	WP GET ATTRIBUTES コマンドに対して、カレントの値の単位がパーセンテージではないのに文字列値を渡した場合に返されます(不正な結果です)。
wk value unit percentage	倍長整数	-100001	WP GET ATTRIBUTES コマンドに対して、カレントの値の単位がパーセンテージなのに数値を渡した場合に返されます(不正な結果です)。
wk version	文字列	version	ドキュメントの内部4DWPバージョンを返します(実数)。この数値は WP GET ATTRIBUTES を使用してのみ読み込み可能です。設定することはできません。

定数	型	値	コメント
wk vertical align	文字列	verticalAlign	要素の垂直揃えを設定します。文字、段落、ピクチャー、テーブル、テーブル行、およびテーブルのカラムとセルに対して使用可能です。取り得る値は以下の通りです: • <u>wk baseline</u> (デフォルト): 要素のベースラインを親要素のベースラインと揃えます。 • <u>wk top</u>: 要素の上を、行上で最も高い要素の上と揃えます。 • <u>wk bottom</u>: 要素の下を、行上で最も低い要素の下と揃えます。 • <u>wk middle</u>: 要素は親要素の中央に配置されます。 • <u>wk superscript</u>: 要素を上付き文字として配置します。 • <u>wk subscript</u>: 要素を下付き文字として配置します。 文字においては、<u>wk top</u>と<u>wk bottom</u>は<u>wk baseline</u>と同じ効果を持ちます。 段落においては、<u>wk baseline</u>、<u>wk superscript</u>と<u>wk subscript</u> は<u>wk top</u>と同じ効果を持ちます。 テーブル、テーブル行、およびテーブルのカラムとセルにおいては、<u>wk top</u>、<u>wk bottom</u> および <u>wk middle</u> のみがサポートされます。 .htm または .html 拡張子。このドキュメントは標準HTMLとして保存され、そのリソースは別に保存されます。4Dタグは除去され、式は値が計算されます。このフォーマットは特に4D Write Pro ドキュメントWeb ブラウザで表示したい場合に特に適しています。 4D Write ProドキュメントはHTMLとして保存され、4D独自のタグを含みます。それぞれの式はノンブレイクのスペースとして挿入されます。このフォーマットはロスレスであるため、テキストフィールドへの保存目的に適しています。
wk web page complete	倍長整数	2	
wk web page html 4D	倍長整数	3	

定数

型 値

コメント

要素の幅を設定します。取り得る値は以下の通りです:

wk width 文
 字 width
 列

- wk auto (デフォルト): 幅は、要素の中身に基づいて設定されます。
- 定義されたサイズ: サイズは実数値あるいは文字列値を使用して表現されたサイズになります:
 - 実数: wk layout unitの単位でのサイズ。
 - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 10,000pt

wk width 属性は、wk min widthが定義されていればそれによって上書きされます。

wk word 倍
 長
 整
 数 6

単語のみに下線を引きます(空白スペースには引きません)(wk text linethrough style あるいは wk text underline style用の値)

4D Write Pro属性を使用するとドキュメント内に保存されているテキストと画像の全てのグラフィックな要素を管理する事ができます。これらの属性は以下のコマンドによって管理されます:

- WP SET ATTRIBUTES
- WP GET ATTRIBUTES
- WP RESET ATTRIBUTES

注: 4D Write Pro エリアの属性は、4D オブジェクト記法あるいは一般的なコマンドを使用しても管理する事が可能です:

- オブジェクト記法(*) - 例えば、以下の二つの宣言は非常に似ています:

```
$bcol:=$range[wk background color]
$bcol:=$range.backgroundColor //同じ結果を得る
```

- OB SET および OB Get コマンドを使用します。例:

```
$bcol:=OB Get($range;wk background color)
```

(*) オブジェクト記法がデータベース内で有効化されている必要があります。詳細な情報については、オブジェクト記法の使用 の章を参照してください。

背景

背景属性を使用すると、ドキュメント内の背景画像エフェクトを定義する事ができます。これの適用対応表は以下のようになります:

ドキュメント	段落	文字	ピクチャー	テーブル	行	カラム/セル	ヘッダー/フッター/ボディ
0	0		0	0	0	0	0

定数

値

wk
background clip backgroundClip

wk
background color backgroundColor

コメント

背景の塗りエリアを指定します。取り得る値:

- wk border box (デフォルト): 背景は境界線の外側の端まで塗られます。
- wk content box: 背景はコンテンツボックスの内部まで塗られます。
- wk padding box: 背景はパディングの端の外側まで塗られます(もしくはあった場合には境界線の内側まで)。
- wk paper box: 背景は端まで塗られます(ドキュメントあるいはセクションのみ)

要素の背景色を指定します。取り得る値:

- CSS カラー("#010101" または "#FFFFFF" または "red")
- 4D カラーの倍長整数値(**OBJECT SET COLOR**コマンドを参照して下さい)
- R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列

ドキュメントのデフォルトは"#FFFFFF" と wk transparentで、段落と画像に対しては"transparent"がデフォルトです。

定数

値

wk

background height backgroundHeight

wk

background image backgroundImage

コメント

背景画像の垂直方向のサイズを指定します。取り得る値は以下の通りです:

- wk auto (デフォルト): 背景画像はその高さを保ちます。
- wk contain: 背景画像のアスペクト比を保ったまま、画像全体が表示される範囲で最大のサイズへと拡大/縮小します。このオプションは他のサイズ属性の値も変更します。
- wk cover: 背景画像のアスペクト比を保ったまま、背景全体を画像でカバーできる大きさまで拡大/縮小します。このとき背景画像の一部は見切れる可能性があります。このオプションは他のサイズ属性の値も変更します。
- 定義されたサイズ: 背景画像の垂直方向のサイズを実数値あるいは文字列値を使用して表現します:
 - 実数: サイズはwk layout unitで指定します。
 - 文字列: 連結された値と単位のCSS文字列。例: 12ptは12ポイント、1.5cmは1.5センチを意味します。
最小値: 0pt、最大値: 1000pt。相対値(パーセンテージ、%)はサポートされています。

背景画像参照を指定します。4Dピクチャー変数あるいは式などの有効な画像であれば可能です。

- 返された値(**WP GET ATTRIBUTES**): 画像がネットワークURLを通して定義された場合には、その画像がすでに読み込まれている場合にはターゲット画像が返されます。そうでない場合には空の画像が返されます。

ピクチャーをURLあるいはローカルのURIを通して扱いたい場合にはwk background image url を使用してください。

定数

値

wk

background image url backgroundImageURL

wk

background origin backgroundOrigin

wk

background position horizontal backgroundPositionHorizontal

コメント

URL(文字列)を通して定義された背景画像です。
ネットワークURL あるいはデータURI あるいはローカルファイルURLです(絶対パスあるいはストラクチャーからの相対パス)。

- 返された値(**WP GET ATTRIBUTES**): ネットワークURL あるいはデータURI。ネットワークURLで参照されたものではない画像に関しては、最初のURLと同じではないことがあります(ネットワークURL のみが保存されます)。ローカルファイルのURLに関しては、画像ストリームそのものはドキュメント内に保管されていて、そのため返されたURL はデータURIと、base64にエンコードされた画像ストリームです。

背景画像をピクチャー式として扱いたい場合には、wk background image を使用してください。

背景画像どのような位置におかれるかを指定します。取りうる値は以下の通りです:

- wk padding box (デフォルト): 背景画像はパディング(または内側の境界線の端)からスタートします。
- wk border box: 背景画像は境界線(外側の境界線の端)の四角形からスタートします。
- wk content box: 背景画像はコンテンツ四角形からスタートします。
- wk paper box: 背景画像は端からスタートします(ドキュメントあるいはセクションのみ)。

背景画像の水平方向のスタート位置を指定します。取り得る値は以下の通りです:

- wk left (デフォルト): 背景画像は水平方向に対して要素の左側からスタートします。
- wk center: 背景画像は水平方向に対して要素の中央からスタートします。
- wk right: 背景画像は水平方向に対して要素の右側からスタートします。

定数	値	コメント
wk background position vertical	backgroundPositionVertical	背景画像の垂直方向のスタート位置を指定します。取り得る値は以下の通りです: • <u>wk top</u> (デフォルト): 背景画像は垂直方向に対して要素の上側からスタートします。 • <u>wk middle</u>: 背景画像は垂直方向に対して要素の中央からスタートします。 • <u>wk bottom</u>: 背景画像は垂直方向に対して要素の下側からスタートします。 背景画像をどのように繰り返すか(あるいは繰り返さないか)を指定します。取り得る値は以下の通りです:
wk background repeat	backgroundRepeat	• <u>wk repeat</u> (デフォルト): 背景画像は垂直方向にも水平方向にも繰り返されます。 • <u>wk no repeat</u>: 背景画像は繰り返されません。 • <u>wk repeat x</u>: 背景画像は水平方向にのみ繰り返されます。 • <u>wk repeat y</u>: 背景画像は垂直方向にのみ繰り返されます。

定数	値	コメント
		背景画像の水平方向のサイズを指定します。取り得る値は以下の通りです:
		• <u>wk auto</u> (デフォルト): 背景画像はその幅を保ちます。 • <u>wk contain</u>: 背景画像のアスペクト比を保ったまま、画像全体が表示される範囲で最大のサイズへと拡大/縮小します。このオプションは他のサイズ属性の値も変更します。 • <u>wk cover</u>: 背景画像のアスペクト比を保ったまま、背景全体を画像でカバーできる大きさまで拡大/縮小します。このとき背景画像の一部は見切れる可能性があります。このオプションは他のサイズ属性の値も変更します。 • 定義されたサイズ: 背景画像の水平方向のサイズは実数値あるいは文字列値を使用して表現されます: ◦ 実数: サイズは<u>wk layout unit</u>で指定します。 ◦ 文字列: 連結された値と単位のCSS文字列。例: 12ptは12ポイント、1.5cmは1.5センチを意味します。 最小値: 0pt、最大値: 1000pt。相対値(パーセンテージ、%)はサポートされています。
wk background width	backgroundWidth	

境界線

境界線属性を使用すると、要素の境界線のスタイル、幅、カラーを指定できます。この属性の適用対応表は以下のようになっています:

ドキュメント	段落	文字	ピクチャー	テーブル	行	カラム/セル	ヘッダー/フッター/ボディ
0	0		0	0	0	0	0

定数 コメント

境界線4つともにカラーを設定します。取りうる値は以下の通りです:

wk
border
color

- CSSカラー("#010101"または"#FFFFFF"または"red")。
- 4Dカラーの倍長整数値(**OBJECT SET COLOR**コマンドを参照して下さい)
- R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列

デフォルト値は"#000000"です(文字列値の場合)。複数のカラーがある場合、**WP GET ATTRIBUTES**は空の文字列を返します。

下の境界線のカラーを設定します。取りうる値は以下の通りです:

wk
border
color
bottom

- CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。
- 4Dカラーの倍長整数値(**OBJECT SET COLOR**コマンドを参照して下さい)
- R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列

左の境界線のカラーを設定します。取りうる値は以下の通りです:

wk
border
color
left

- CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。
- 4Dカラーの倍長整数値(**OBJECT SET COLOR**コマンドを参照して下さい)
- R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列

右の境界線のカラーを設定します。取りうる値は以下の通りです:

wk
border
color
right

- CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。
- 4Dカラーの倍長整数値(**OBJECT SET COLOR**コマンドを参照して下さい)
- R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列

上の境界線のカラーを設定します。取りうる値は以下の通りです:

wk
border
color
top

- CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。
- 4Dカラーの倍長整数値(**OBJECT SET COLOR**コマンドを参照して下さい)
- R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列

角の丸い境界線を指定します。取り得る値は以下の通りです:

wk
border
radius

- wk none (デフォルト): 境界線は丸い角を持ちません。
- 数値または文字列値を使用して表現された半径:
 - 数値: wk layout unitでの半径
 - 文字列値: 値と単位が連結されたCSS文字列。例: 12ptは12ポイント、1.5cmは1.5センチを意味します。

定数 コメント

境界線の4辺全てのスタイルを指定します。取り得る値は以下の通りです:

wk
border
style

- wk none (デフォルト): 境界線なし
- wk hidden: wk noneと同様、ただし境界線の競合解決時を除く
- wk solid: 実線の境界線
- wk dotted: 点線の境界線
- wk dashed: 破線の境界線
- wk double: 二重境界線
- wk groove: 3D くぼみ境界線(実際の効果は境界線カラーによります)
- wk ridge: 3D 隆起境界線(実際の効果は境界線カラーによります)
- wk inset: 3D インセット境界線(実際の効果は境界線カラーによります)
- wk outset: 3D アウトセット境界線(実際の効果は境界線カラーによります)

下部の境界線を指定します。取り得る値は以下の通りです:

wk
border
style
bottom

- wk none (デフォルト): 下部境界線なし
- wk hidden: wk noneと同様、ただし境界線の競合解決時を除く
- wk solid: 実線の下部境界線
- wk dotted: 点線の下部境界線
- wk dashed: 破線の下部境界線
- wk double: 二重下部境界線
- wk groove: 3D くぼみ下部境界線(実際の効果は下部境界線カラーによります)
- wk ridge: 3D 隆起下部境界線(実際の効果は下部境界線カラーによります)
- wk inset: 3D インセット下部境界線(実際の効果は下部境界線カラーによります)
- wk outset: 3D アウトセット下部境界線(実際の効果は下部境界線カラーによります)

左の境界線を指定します。取り得る値は以下の通りです:

wk
border
style
left

- wk none (デフォルト): 下部境界線なし
- wk hidden: wk noneと同様、ただし境界線の競合解決時を除く
- wk solid: 実線の左境界線
- wk dotted: 点線の左境界線
- wk dashed: 破線の左境界線
- wk double: 二重左境界線
- wk groove: 3D くぼみ左境界線(実際の効果は左境界線カラーによります)
- wk ridge: 3D 隆起左境界線(実際の効果は左境界線カラーによります)
- wk inset: 3D インセット左境界線(実際の効果は左境界線カラーによります)
- wk outset: 3D アウトセット左境界線(実際の効果は左境界線カラーによります)

定数 コメント

右の境界線を指定します。取り得る値は以下の通りです:

wk
border
style
right

- wk none (デフォルト): 右境界線なし
- wk hidden: wk noneと同様(境界線の競合解決時を除く)
- wk solid: 実線の右境界線
- wk dotted: 点線の右境界線
- wk dashed: 破線の右境界線
- wk double: 二重右境界線
- wk groove: 3D くぼみ右境界線(実際の効果は右境界線カラーによります)
- wk ridge: 3D 隆起右境界線(実際の効果は右境界線カラーによります)
- wk inset: 3D インセット右境界線(実際の効果は右境界線カラーによります)
- wk outset: 3D アウトセット右境界線(実際の効果は右境界線カラーによります)

上部境界線を指定します。取り得る値は以下の通りです:

wk
border
style
top

- wk none (デフォルト): 上部境界線なし
- wk hidden: wk noneと同様(境界線の競合解決時を除く)
- wk solid: 実線の上部境界線
- wk dotted: 点線の上部境界線
- wk dashed: 破線の上部境界線
- wk double: 二重上部境界線
- wk groove: 3D くぼみ上部境界線(実際の効果は上部境界線カラーによります)
- wk ridge: 3D 隆起上部境界線(実際の効果は上部境界線カラーによります)
- wk inset: 3D インセット上部境界線(実際の効果は上部境界線カラーによります)
- wk outset: 3D アウトセット上部境界線(実際の効果は上部境界線カラーによります)

四辺全ての境界線の幅を指定します。境界線幅を指定する前に境界線スタイルを指定する必要があります。取り得る値は以下の通りです:

wk
border
width

- 数値あるいは文字列値を使用して表現された幅:
 - 数値: wk layout unitでの幅
 - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。
- デフォルト値: 2pt

下部境界線の幅を指定します。取り得る値は以下の通りです:

wk
border
width
bottom

- 数値あるいは文字列値を使用して表現された幅:
 - 数値: wk layout unitでの幅
 - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。
- デフォルト値: 2pt

定数 コメント

左境界線の幅を指定します。取り得る値は以下の通りです:

- wk
border
width
left
 - 数値あるいは文字列値を使用して表現された幅:
 - 数値: `wk layout unit`での幅
 - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。
 - デフォルト値: 2pt

右境界線の幅を指定します。取り得る値は以下の通りです:

- wk
border
width
right
 - 数値あるいは文字列値を使用して表現された幅:
 - 数値: `wk layout unit`での幅
 - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。
 - デフォルト値: 2pt

上部境界線の幅を指定します。取り得る値は以下の通りです:

- wk
border
width
top
 - 数値あるいは文字列値を使用して表現された幅:
 - 数値: `wk layout unit`での幅
 - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。
 - デフォルト値: 2pt

wk
inside

選択されたエリアに複数の段落が含まれる場合、属性は、対応して被っている段落のプロパティに対してのみ適用することを指定します(範囲の外側は含まれません)。境界線、パディング、マージン属性に対してのみ適用可能で、指定された属性に追加されなければなりません。 **WP SET ATTRIBUTES** コマンドの例2を参照して下さい。

wk
outside

選択されたエリアが複数の段落を含むとき、ある属性が対応する段落の外側のプロパティにのみ適用されることを指定します(内側には適用されません)。これは境界線、パディング、そしてマージン属性に対してのみ指定可能で、特定の属性に対して追加される必要があります。 **WP SET ATTRIBUTES** コマンドの例2を参照して下さい。

ドキュメント

ドキュメント属性を使用すると、ドキュメントの題名、執筆者名、会社名、注記といった標準のドキュメント情報を取得・設定することができます。この適用対応表は以下のようになっています:

ドキュメント	段落	文字	ピクチャー	テーブル	行	カラム/セル	ヘッダー/フッター/ボディ
○			○				

定数 コメント

wk
author ドキュメントの著者名を指定(文字列)

wk
column
count (読み込みのみ属性)レンジ内にあるカラムの総数。値:倍長整数

垂直方向のカラムルーラーのカラー。取りうる値は以下の通り:

wk
column
rule
color

- CSSカラー("#010101"あるいは"#FFFFFF"あるいは"red")。デフォルトは"#000000"(black)です。
- 4D カラー倍超整数値(**OBJECT SET COLOR**コマンド参照のこと)
- RGBコンポーネントの値をそれぞれ格納している倍超整数配列(0-255)

垂直方向のカラムルーラーのスタイル。取りうる値は以下の通り:

wk
column
rule
style

- wk none (デフォルト): ルーラーなし
- wk hidden: wk none と同じですが、ルーラー干渉の解決の場合を除く
- wk solid: 実線のルーラー
- wk dotted: 点線のルーラー
- wk dashed: 破線のルーラー
- wk double: 二重線のルーラー
- wk groove: 3Dルーラー(実際のエフェクトはルーラーのカラーによります)
- wk ridge: 3Dくぼみルーラー(実際のエフェクトはルーラーのカラーによります)
- wk inset: 3Dインセットルーラー(実際のエフェクトはルーラーのカラーによります)

垂直方向のルーラーの幅。取りうる値は以下の通り:

wk
column
rule
width

- 実数: wk layout unit で指定した単位での幅
- 文字列: 連結された幅の値と単位(例.: "12pt"で12ポイント、"1.5cm"で1.5cmを指定)

デフォルト値="2.5pt"

(ドキュメントあるいはセクションのみ)二つのカラム間の間隔。取り得る値は以下のとおり:

wk
column
spacing

- 実数: wk layout unit 定数で指定した単位での幅
- 文字列: 連結された幅の値と単位(例.: "12pt"で12 ポイント、あるいは"1.5cm"で1.5cmを指定)

デフォルト値="12pt"

定数	コメント
wk column width	(ドキュメントとセクションのみ)読み込みのみ属性。各カラムのカレントの幅、つまり実際のページ幅、ページマージン、カラム数およびカラム間隔に基づいて計算された幅です。 ドキュメントの場合、デフォルトのセクションカラム幅を使用するので、セクション内で一部属性が上書きされていた場合にはセクションのカラム幅と異なる場合があります。 取りうる値: • 実数: <u>wk layout unit</u> で指定された単位での幅 • 文字列: 連結された幅の値と単位(例: "12pt"で12ポイントを、"1.5cm"で1.5cmを指定します)
wk company	ドキュメントに関連づけられる会社を指定します(文字列)
wk date creation	ドキュメントの作成日(日付)を返します。この値は読み込みのみで、設定する事はできません。
wk date modified	ドキュメントの最終更新日(日付)を返します。この値は読み込みのみであり、設定する事はできません。
wk dpi	内部でのピクセル <-> ポイント変換に使用されるDPI(整数)。常に96です(読み込みのみ)変更あるいは読み出し可能であるドキュメントのカレントのビューのDPIを変更あるいは読み出す"dpi"標準アクションと、この内部属性を混同しないようにしてください。 値が整数あるいは実数として設定・取得された際の一部の属性の次元の単位を指定します。取り得る値は以下の通り: • <u>wk unit cm</u> (デフォルト): センチメートル • <u>wk unit pt</u>: ポイント • <u>wk unit px</u>: ピクセル • <u>wk unit percent</u> (<u>wk line height</u>、<u>wk background size h</u> / <u>wk background size v</u>に対してのみ使用可能) • <u>wk unit mm</u>: ミリメートル • <u>wk unit inch</u>: インチ
wk layout unit	
wk notes	ドキュメントについてのコメントを指定します(文字列)
wk subject	ドキュメントの題名を指定します(文字列)
wk title	ドキュメントのタイトル(文字列)を指定します。デフォルトは"New 4D Write Pro Document"です。
wk version	ドキュメントの内部4DWPバージョンを返します(実数)。この数値は WP GET ATTRIBUTES を使用してのみ読み込み可能です。設定することはできません。

フォントとテキスト

これらの属性を使用するとテキストのフォントファミリー名、サイズ、スタイルを定義する事ができます。適応対応表は以下の通りです:

ドキュメン ト	段 落	文 字	ピクチャ ー	テーブ ル	行	カラム/セ ル	ヘッダー/フッター/ボディ
		0		0*	0*	0*	0*

* セル内の文字に適用

定数

コメント

wk font

FONT STYLE LIST コマンドによって返される、完全なフォント名とスタイル名を指定します。無効なフォント名を設定した場合、コマンドは何もしません。デフォルト値: "Times New Roman"

テキストの太さを指定します(利用可能なフォントスタイルによります)。取り得る値は以下の通りです:

wk font
bold

- wk true を指定すると、選択された文字を太字フォントスタイルへと設定します。**WP GET ATTRIBUTES** コマンドを使用するとき、選択された文字のうち少なくとも1文字が太字フォントスタイルをサポートする場合、wk true が返されます。
- wk false (デフォルト) を指定すると、選択された文字から太字フォントスタイルを除去します。**WP GET ATTRIBUTES** コマンドを使用するとき、選択された文字がどれも太字フォントスタイルをサポートしない場合、wk false が返されます。

wk font
family

wk font で定義されたフォントファミリー名を指定します。デフォルトの値は "Times New Roman" です。

選択された文字に異なるフォントファミリープロパティが含まれる場合、**WP GET ATTRIBUTES** コマンドは空の文字列を返します。

テキストの、イタリックスタイルを指定します(利用可能なフォントスタイルによります)。取り得る値は以下の通りです:

wk font
italic

- wk true は選択された文字をイタリック体あるいは斜体スタイルに設定します。**WP GET ATTRIBUTES** コマンドは選択された文字のうち少なくとも一つ以上の文字がイタリック体あるいは斜体フォントスタイルをサポートする場合、wk true が返されます。
- wk false (デフォルト) は選択された文字のうちイタリック体あるいは斜体フォントスタイルがある場合にはそれらを除去します。コマンドは選択された文字のうちイタリック体あるいは斜体フォントスタイルをサポートする文字がない場合、wk false が返されます。

wk font
size

テキストのフォントサイズを指定します。取り得る値は以下の通りです(単位はポイントのみ):

- 実際の値(デフォルト = 12)
- 値と単位が連結されたCSS文字列。例: 12pt (12 ポイントを意味する)

wk text
color

テキストのカラーを指定します。取り得る値は以下の通りです:

- CSSカラー("#010101" や "#FFFFFF" または "red")。デフォルトは文字列の場合 "#000000" です。
- 4Dカラー倍長整数値(**OBJECT SET COLOR** コマンドを参照して下さい)
- R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列

定数

コメント

テキストの打ち消し線のカラーを指定します。取り得る値は以下の通りです:

- | | |
|---------------------------------|---|
| wk text
linethrough
color | <ul style="list-style-type: none">• CSSカラー("#010101"や"#FFFFFF" または "red")• 4Dカラー倍長整数値(OBJECT SET COLORコマンドを参照してください)• R、G、B、コンポーネントそれぞれの要素(0-255)を格納した倍長整数配列 |
|---------------------------------|---|

デフォルトは、文字列なら"currentColor"、あるいは倍長整数なら`wk default` です。
テキストの打ち消し線のスタイルを(あれば)指定します。取り得る値は以下の通りです:

- | | |
|---------------------------------|---|
| wk text
linethrough
style | <ul style="list-style-type: none">• <code>wk none</code> (デフォルト): 打ち消し線エフェクトはありません。• <code>wk solid</code>: 選択されたテキストに対して実線を描画します。• <code>wk dotted</code>: 選択されたテキストに対して点線を描画します。• <code>wk dashed</code>: 選択されたテキストに対して破線を描画します。• <code>wk double</code>: 選択されたテキストに対して二重線を描画します。• <code>wk semi transparent</code>: 選択されたテキストに対してグレーの線を描画します。他のスタイルと組み合わせる事が可能です。• <code>wk word</code>: 単語にのみ線を描画します(空白のスペースは描画しません)。他のスタイルと組み合わせる事が可能です。 |
|---------------------------------|---|

選択されたテキストのシャドウカラーを指定します。取り得る値は以下の通りです:

- | | |
|----------------------------|---|
| wk text
shadow
color | <ul style="list-style-type: none">• CSSカラー("#010101" や "#FFFFFF" または "red")• 4Dカラー倍長整数値(OBJECT SET COLORコマンドを参照して下さい)• R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列• <code>wk transparent</code> (デフォルト) |
|----------------------------|---|

- | | |
|-----------------------------|---|
| wk text
shadow
offset | <p>シャドウエフェクトのオフセットを指定します。取り得る値:</p> <ul style="list-style-type: none">• ポイント単位で指定されたサイズ。デフォルト値: 1pt |
|-----------------------------|---|

テキスト内での大文字と小文字を指定します。取り得る値は以下の通り:

- | | |
|----------------------|--|
| wk text
transform | <ul style="list-style-type: none">• <code>wk capitalize</code>: 最初の文字は大文字に設定されます。• <code>wk lowercase</code>: 文字は小文字に設定されます。• <code>wk uppercase</code>: 文字は大文字に設定されます。• <code>wk small uppercase</code>: 文字はスモールキャップスに設定されます。• <code>wk none</code> (デフォルト): 変換なし |
|----------------------|--|

テキストの下線のカラーを指定します。取り得る値:

- | | |
|-------------------------------|--|
| wk text
underline
color | <ul style="list-style-type: none">• CSS カラー("#010101" や "#FFFFFF" または "red")• 4Dカラー倍長整数値(OBJECT SET COLORコマンドを参照して下さい)• R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列 |
|-------------------------------|--|

デフォルトは、文字列なら"currentColor"、あるいは倍長整数なら `wk default` です。

定数

コメント

テキストの下線のスタイルを(あれば)指定します。取り得る値は以下の通りです:

wk text
underline
style

- wk none (デフォルト): 下線エフェクトはありません。
- wk solid: 実線の下線を描画します。
- wk dotted: 点線の下線を描画します。
- wk dashed: 破線の下線を描画します。
- wk double: 二重線の下線を描画します。
- wk semi transparent: グレーの下線を描画します。他の線スタイルと組み合わせる事が可能です。
- wk word: 単語にのみ下線を描画します(空白のスペースは描画しません)。他の線スタイルと組み合わせる事が可能です。

要素の垂直揃えを設定します。文字、段落、ピクチャーに対して使用可能です。取り得る値は以下の通りです:

wk vertical
align

- wk baseline (デフォルト): 要素のベースラインを親要素のベースラインと揃えます。
- wk top: 要素の上を、行上で最も高い要素の上と揃えます。
- wk bottom: 要素の下を、行上で最も低い要素の下と揃えます。
- wk middle: 要素は親要素の中央に配置されます。
- wk superscript: 要素を上付き文字として配置します。
- wk subscript: 要素を下付き文字として配置します。

文字においては、wk topとwk bottomはwk baselineと同じ効果を持ちます。
段落においては、wk baseline、wk superscriptとwk subscript はwk topと同じ効果を持ちます。

高さ/幅

高さ/幅属性を使用すると、段落や画像の高さと幅を設定することができます。適用の対応表は以下の通りです:

ドキュメント	段落	文字	ピクチャー	テーブル	行	カラム/セル	ヘッダー/フッター/ボディ
	O[tab/]	O		O	O*	O**	

* セルに適用
**ヘッダーとフッターに対しては読み込みのみ(自動的に設定されます)

定数 コメント

要素の高さを設定します。高さのプロパティには、パディング、境界線、マージンは含まれません。このプロパティは要素のパディング、境界線、マージンの内側のエリアの高さを設定します。取り得る値は以下の通りです:

wk
height

- wk auto (デフォルト): 高さは要素の中身に基づいて決められます。
- 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズです:
 - 実数: wk layout unitで設定された単位でのサイズ
 - 文字列: 値と単位が結合されたCSS文字列。例: 12ptは12ポイントを意味し、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 10,000pt

wk height属性は(定義されていれば)wk min heightによって上書きされます。

要素の最小限の高さを設定します。これはwk heightプロパティの値がwk min heightプロパティより小さくなることを防ぎます。取り得る値は以下の通りです:

wk
min
height

- wk auto (デフォルト): 最小限の高さは要素の中身に基づいて決められます。
- 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズ:
 - 実数: wk layout unitの単位でのサイズ
 - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 10,000pt。

wk min heightの値はwk height属性を上書きします。

注: この属性は行、カラム、セルにおいてはサポートされていません。

要素の最小限の幅を設定します。これはwk widthプロパティの値がwk min widthプロパティより小さくなることを防ぎます。取り得る値は以下の通りです:

wk
min
width

- wk auto (デフォルト): 最小限の幅は要素の中身に基づいて決められます。
- 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズ:
 - 実数: wk layout unitの単位でのサイズ
 - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 10,000pt。

wk min widthの値はwk width属性を上書きします。

要素の幅を設定します。取り得る値は以下の通りです:

wk
width

- wk auto (デフォルト): 幅は、要素の中身に基づいて設定されます。
- 定義されたサイズ: サイズは実数値あるいは文字列値を使用して表現されたサイズになります:
 - 実数: wk layout unitの単位でのサイズ。
 - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 10,000pt

wk width 属性は、wk min widthが定義されていればそれによって上書きされます。

画像

画像属性を使用すると、エリア内に挿入あるいは追加されたピクチャーを管理する事ができます。適用の対応表は以下の通りです:

ドキュメン ト	段 落	文 字	ピクチャ ー	テーブ ル	行	カラム/セ ル	ヘッダー/フッター/ボディ
			0	0*	0*	0*	

* セル内の画像に適用(インライン画像のみ)

リマインダ: ピクチャーを管理する のセクションで説明されているように、4D Write Pro は2種類のピクチャーをサポートしています:

- **WP INSERT PICTURE** あるいは **ST INSERT EXPRESSION** コマンドを使用してインラインに挿入されたピクチャー
- **WP Add picture** コマンドを使用してページにアンカーされたピクチャー

以下の属性は、インライン画像・アンカーされた画像の両方に対して使用可能です:

定数

コメント

画像参照を指定します。4Dピクチャー変数あるいは式などの有効な画像であれば可能です。

wk
image

- 返された値(**WP GET ATTRIBUTES**): 画像がネットワークURLを通して定義された場合には、その画像がすでに読み込まれている場合にはターゲット画像が返されます。そうでない場合には空の画像が返されます。

ピクチャーをURLあるいはローカルのURIを通して扱いたい場合には wk image url を使用してください。

wk
image
alternate
text

画像が表示できない際の、画像の代替テキストを指定します。

URL(文字列)を通して定義された背景画像です。ネットワークURL あるいはデータURI (絶対パスあるいはストラクチャーからの相対パス)です。

wk
image
url

- 返された値(**WP GET ATTRIBUTES**): ネットワークURL あるいはデータURI。ネットワークURLで参照されたものではない画像に関しては、最初のURLと同じではないことがあります(ネットワークURL のみが保存されます)。ローカルファイルのURLに関しては、画像ストリームそのものはドキュメント内に保管されていて、そのため返されたURLはデータURIと、base64にエンコードされた画像ストリームです。

画像をピクチャー式として扱いたい場合には、 wk image を使用してください。

wk
owner

(読み込み専用のレンジ属性)

(読み込みのみ可能なレンジ属性) 4D Write Pro オブジェクトの型。取りうる値:

wk type

- wk type default: 未定義の型のレンジ
- wk type paragraph: 段落型のレンジ
- wk type image: 画像(アンカーされた画像およびインライン画像)
- wk type image anchored: アンカーされた画像
- wk type image inline: インライン画像
- wk type container: 例えばヘッダーやフッターなど
- wk type table: 表参照
- wk type table row: 表の行参照
- wk type table cell: 表のセル参照
- wk type table column: 表のカラム参照(カラムレンジに対してのみ)
- wk type body: 本文参照

以下の属性は、インライン画像に対してのみ使用可能です:

定数

コメント

wk end

(読み込み専用のレンジ属性)

wk start

(読み込み専用のレンジ属性)

要素の垂直揃えを設定します。文字、段落、ピクチャーに対して使用可能です。取り得る値は以下の通りです:

wk vertical
align

- wk baseline (デフォルト): 要素のベースラインを親要素のベースラインと揃えます。
- wk top: 要素の上を、行上で最も高い要素の上と揃えます。
- wk bottom: 要素の下を、行上で最も低い要素の下と揃えます。
- wk middle: 要素は親要素の中央に配置されます。
- wk superscript: 要素を上付き文字として配置します。
- wk subscript: 要素を下付き文字として配置します。

文字においては、wk topとwk bottomはwk baselineと同じ効果を持ちます。

段落においては、wk baseline、wk superscriptとwk subscript はwk topと同じ効果を持ちます。

以下の画像はアンカーされた画像に対してのみ使用可能です:

定数

コメント

画像の、原点からの相対的な水平方向揃えを定義します([wk anchor origin](#) 参照)。取り得る値は以下の通りです:

wk

anchor

horizontal

align

- [wk left](#) - 左揃え
- [wk center](#) - 中央揃え(*HTML*では互換性がなく、*Web*では画像が表示されません)
- [wk right](#) - 右揃え

[wk layout unit](#) からの画像の水平方向のオフセットを、CSS位置文字列あるいは倍長整数(cmまたはptあるいはpixel単位)で表現された長さで定義します。取り得る値:

wk

anchor

horizontal

offset

- [wk anchor horizontal align](#) に相対的なページの左端、あるいは右端
- 埋め込みモードの本文の左端、あるいは右端([wk anchor section](#) = [wk anchor embedded](#)の場合)

デフォルトの値 = 0

画像のレイアウト位置を、ページのテキストに対しての相対位置で定義します。取り得る値:

wk

anchor

layout

- [wk behind text](#) - 画像はテキストの背後にアンカーされている
- [wk in front of text](#) - 画像はテキストの前面にアンカーされている
- [wk inline with text](#) - 画像はテキストのインライン画像になっている(**WP INSERT PICTURE** で挿入された画像に対してのデフォルト)。読み込みのみ属性(インライン画像はプログラミングによってアンカー画像へと変換することはできません)。

画像がページ、ヘッダー、フッターのどれに対してアンカーされているかを定義します。取り得る値:

wk

anchor

origin

- [wk paper box](#) (デフォルト) - 画像はページの端に対してアンカーされています。
- [wk header box](#) - 画像はドキュメントのヘッダーに対してアンカーされています。ヘッダーが非表示の場合には画像は表示されません。
- [wk footer box](#) - 画像はドキュメントのフッターに対してアンカーされています。フッターが非表示の場合には画像は表示されません。

このセレクターは埋め込みモードでは無視されます。

定数

コメント

画像がアンカーされているページインデックス、あるいはページのタイプを定義します。取り得る値:

wk
anchor
page

- wk anchor all - 画像を、wk anchor section で定義されたセクションの全てのページにアンカーします。
- wk anchor embedded - 画像を、埋め込みモードのドキュメント本文にアンカーします(画像は埋め込みモードでのみ表示されます)。
- wk anchor first page - 画像を、wk anchor section で定義されたセクションの最初のページにアンカーします。
- wk anchor left page - 画像を、wk anchor section で定義されたセクションの左ページにアンカーします。
- wk anchor right page - 画像を、wk anchor section で定義されたセクションの右ページにアンカーします。
- 画像をアンカーするページを示す番号(0以上の倍長整数)。この場合、wk anchor section = wk anchor allとなります。セクションへのアンカーは、画像が単一のページにアンカーされている時は無視されます。

注: ページモードでの画像はブラウザでは表示されません。

画像がアンカーされているセクションインデックスあるいはセクションのタイプを定義します。取り得る値:

wk
anchor
section

- wk anchor all (デフォルト) - 画像をドキュメント内の全てのセクションにアンカーします(画像はページモードでのみ表示されます)。
- wk anchor embedded - 画像を埋め込みモードのドキュメント本文にアンカーします(画像は埋め込みモードでのみ表示されます)。
- 画像をアンカーするセクションを示す番号(1以上の倍長整数、この場合画像はページモードでのみ表示されます)。

注: ページモードでの画像はブラウザでは表示されません。

画像の、原点からの相対的な垂直方向揃えを定義します(wk anchor origin 参照)。取り得る値は以下の通りです:

wk
anchor
vertical
align

- wk top - 上揃え
- wk center - 中央揃え(*HTMLでは互換性がなく、ブラウザでは画像が表示されません*)
- wk bottom - 下揃え

wk layout unit からの画像の垂直方向のオフセットを、CSS位置文字列あるいは倍長整数(cmまたはptあるいはpixel単位)で表現された長さで定義します。取り得る値:

wk
anchor
vertical
offset

- ページの上、中央、下方向の限界(wk anchor origin 参照)
- 埋め込みモードの本文のページの上、中央、下方向の限界(wk anchor section = wk anchor embedded の場合)

デフォルト値 = 0

wk id

(読み込みのみ属性)要素のID。値:文字列

リンク

リンク属性を使ってレンジに対して設定したリンクを管理することができます。適用の対応表は以下の通りです:

ドキュメント	段落	文字	ピクチャー	テーブル	行	カラム/セル	ヘッダー/フッター/ボディ
	0	0	0*	0	0	0	

定数	コメント
	レンジに設定されたハイパーリンクです。取りうる値は以下のとおり:
wk link url	- 絶対URL。例: "http://www.4d.com/" - 相対URL。例: "/test/page.html" (データベースストラクチャーファイルを起点とします) - 空文字列 = リンクなし

*インライン画像のみ

リスト

4D Write Proは主に二つのタイプのリストをサポートしています:

- 順番付けのないリスト: リスト項目が中点と記されているリスト
- 順番付けされているリスト: リスト項目が番号あるいは文字で記されているリスト

リスト属性を使用するとリストを設定し、異なるリスト項目フォントあるいはマーカーを設定する事ができます。適用の対応表は以下の通りです:

ドキュメント	段落	文字	ピクチャー	テーブル	行	カラム/セル	ヘッダー/フッター/ボディ
	0		—	0*	0*	0*	

* セル内の段落に適用

定数 コメント

FONT STYLE LIST コマンドで返される、完全なフォント名を指定してリスト項目マーカ―を表示するようにします(しかし段落のテキストは表示しません)。システムがフォント名を認識しない場合、その代理を使用します。無効なフォント名を設定した場合、コマンドは何もしません。デフォルト値: "Times"

wk list font family wk list fontによって定義されたフォントファミリー名を使用してリスト項目マーカ―を表示します(しかし段落のテキストは表示しません)。デフォルト値は"Times New Roman"です。

並べられたリストの始まる値を設定します。取り得る値は以下の通りです:

- wk list start number**
- wk auto (デフォルト): 始まる値を0に設定します。
 - 整数値: 始まる値

左から右への段落方向二対してのリスト項目マーカ―文字列フォーマット。定義されると、リストのデフォルトのリスト項目マーカ―文字列を上書きします。

- wk list string format LTR**
- 順番に並んでいないリスト: 文字列がリスト項目マーカ―として使用されます(通常は単一文字による文字列、例: "-")
 - 順番に並んでいるリスト: "#"文字を含んだ文字列。 "#"は計算された数字や文字のプレースホルダーになります。デフォルトは"#."なので、例えばカレントのリスト項目番号が15でリストスタイルタイプが小数だった場合、リスト項目マーカ―文字列は"15."になります。

右から左への段落方向に対してのリスト項目マーカ―文字列フォーマット。定義されると、リストのデフォルトのリスト項目マーカ―文字列を上書きします。

- wk list string format RTL**
- 順番に並んでいないリスト: 文字列がリスト項目マーカ―として使用されます(通常は単一文字による文字列、例: "-")
 - 順番に並んでいるリスト: "#"文字を含んだ文字列。 "#"は計算された数字や文字のプレースホルダーになります。デフォルトは"#."なので、例えばカレントのリスト項目番号が 15でリストスタイルタイプが小数だった場合、リスト項目マーカ―文字列は"15."になります。

番号なしリストにおいて、画像をリスト項目マーカ―として指定します。取り得る値は以下の通りです:

- wk list style image**
- wk none (デフォルト): リスト項目マーカ―は画像によっては定義されていません。
 - 4Dピクチャー変数あるいは式などの有効な画像
 - 返された値(**WP GET ATTRIBUTES**): 画像がネットワークURLを通して定義された場合には、その画像がすでに読み込まれている場合にはターゲット画像が返されます。そうでない場合には空の画像が返されます。

ピクチャーをURLあるいはローカルのURIを通して扱いたい場合にはwk list style image url を使用してください。

定数

コメント

リスト項目マーカーとして使用される画像の高さを設定します。取り得る値は以下の通りです:

wk list
style
image
height

- wk auto (デフォルト): 高さは画像のサイズに基づいて決められます。
- 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズ:
 - 実数: wk layout unitの単位でのサイズ
 - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 1000pt。

URL(文字列)を通して定義された、番号なしリストのリスト項目マーカーの画像。取り得る値:

wk list
style
image
url

- wk none (デフォルト): リスト項目マーカーは画像によって定義されていません。
- ネットワークURL あるいはデータURI(絶対パスあるいはストラクチャーからの相対パス)
- 返された値(**WP GET ATTRIBUTES**): ネットワークURL あるいはデータURI。ネットワークURLで参照されたものではない画像に関しては、最初のURLと同じではないことがあります(ネットワークURL のみが保存されます)。ローカルファイルのURLに関しては、画像ストリームそのものはドキュメント内に保管されていて、そのため返されたURLはデータURIと、base64にエンコードされた画像ストリームです。

リスト項目マーカー画像をピクチャー式として扱いたい場合には、wk list style image を使用してください。

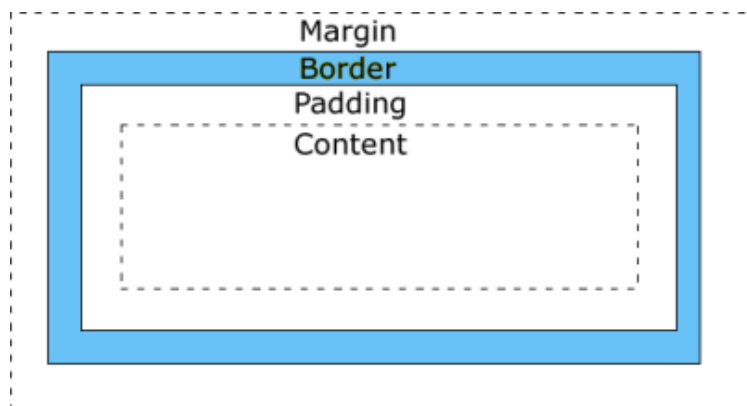
定数 コメント

順番付けされた、あるいは順番付けされていないリスト項目マーカのタイプを指定します。取り得る値は以下の通りです：

- | | |
|--------------------|---|
| wk list style type | <ul style="list-style-type: none">• wk disc (デフォルト)• wk circle• wk square• wk decimal: 1 2 3• wk decimal leading zero: 01 02 03• wk lower latin: a b c• wk lower roman: i ii iii iv• wk upper latin: A B C• wk upper roman: I II III IV• wk lower greek: α、β、γ、等• wk armenian• wk georgian• wk hebrew• wk hiragana• wk katakana• wk cjk ideographic• wk hollow square• wk diamond• wk club• wk decimal greek• wk custom: 順番付けされていないリストで "-" をデフォルトのリスト項目マーカとして使用します。これは標準のリスト項目マーカを変更する事なく wk list string format LTR または wk list string format RTL のリスト項目マーカをカスタマイズするのに便利なスタイルです。• wk none |
|--------------------|---|

マージン

マージンとは、要素の境界線の外側にあるエリアのことで、透明なものです。以下の図は、"ボックス"の要素に対して設定可能な様々な要素の関係性を表しています：



マージン属性は、以下のように適用できます：

ドキュメント	段落	文字	ピクチャー	テーブル	行	カラム/セル	ヘッダー/フッター/ボディ
0	0		0	0			0

定数 コメント

wk inside 選択されたエリアに複数の段落が含まれる場合、属性は、対応して被っている段落のプロパティに対してのみ適用することを指定します(範囲の外側は含まれません)。境界線、パディング、マージン属性に対してのみ適用可能で、指定された属性に追加されなければなりません。 **WP SET ATTRIBUTES** コマンドの例2を参照して下さい。

要素の全てのマージンのサイズを指定します。取り得る値は以下の通りです:

- 数値あるいは文字列値を使用して表現されたサイズ:
 - 数値: wk layout unitの単位でのサイズ
 - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。
- wk none (デフォルト): 特にマージンを指定しません。

要素の下部のマージンのサイズを指定します。取り得る値は以下の通りです:

- 数値あるいは文字列値を使用して表現されたサイズ:
 - 数値: wk layout unitの単位でのサイズ
 - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。
- wk none (デフォルト): 特にマージンを指定しません。

要素の左のマージンのサイズを指定します。取り得る値は以下の通りです:

- 数値あるいは文字列値を使用して表現されたサイズ:
 - 数値: wk layout unitの単位でのサイズ
 - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。
- wk none (デフォルト): 特にマージンを指定しません。

要素の右のマージンのサイズを指定します。取り得る値は以下の通りです:

- 数値あるいは文字列値を使用して表現されたサイズ:
 - 数値: wk layout unitの単位でのサイズ
 - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。
- wk none (デフォルト): 特にマージンを指定しません。

要素の上部のマージンのサイズを指定します。取り得る値は以下の通りです:

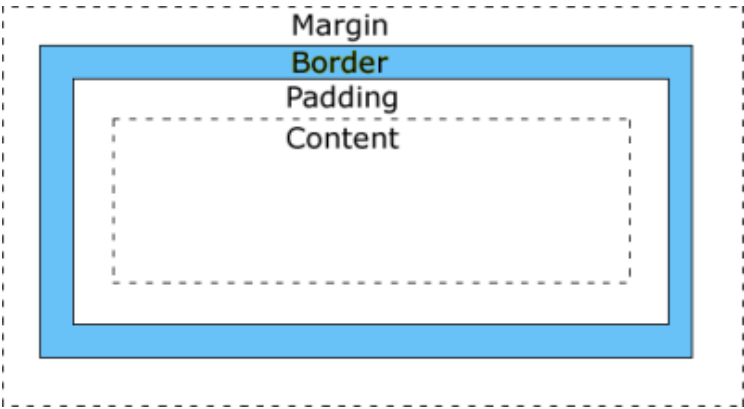
- 数値あるいは文字列値を使用して表現されたサイズ:
 - 数値: wk layout unitの単位でのサイズ
 - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。
- wk none (デフォルト): 特にマージンを指定しません。

wk outside 選択されたエリアが複数の段落を含むとき、ある属性が対応する段落の外側のプロパティにのみ適用されることを指定します(内側には適用されません)。これは境界線、パディング、そしてマージン属性に対してのみ指定可能で、特定の属性に対して追加される必要があります。 **WP SET ATTRIBUTES** コマンドの例2を参照して下さい。

パディング

パディングとは、要素の中身と要素の境界線の間の空白のことを指します。パディングは要素の背景色に影響されます。

以下の図は、"ボックス"の要素に対して設定可能な様々な要素の関係性を表しています：



パディング属性は、以下のように適用できます：

ドキュメント	段落	文字	ピクチャー	テーブル	行	カラム/セル	ヘッダー/フッター/ボディ
0	0		0	0*	0*	0	0

* セルに適用

定数 コメント

wk inside 選択されたエリアに複数の段落が含まれる場合、属性は、対応して被っている段落のプロパティに対してのみ適用することを指定します(範囲の外側は含まれません)。境界線、パディング、マージン属性に対してのみ適用可能で、指定された属性に追加されなければなりません。**WP SET ATTRIBUTES**コマンドの例2を参照して下さい。

wk outside 選択されたエリアが複数の段落を含むとき、ある属性が対応する段落の外側のプロパティにのみ適用されることを指定します(内側には適用されません)。これは境界線、パディング、そしてマージン属性に対してのみ指定可能で、特定の属性に対して追加される必要があります。**WP SET ATTRIBUTES**コマンドの例2を参照して下さい。

要素の全てのサイドのパディングを指定します。取り得る値は以下の通りです:

- 数値あるいは文字列値を使用して表現されるサイズ:
 - 数値: wk layout unitの単位でのサイズ
 - 文字列: 値と単位が連結されたCSS文字列。例:12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。
- wk none (デフォルト): 特定のパディングを指定しません。

要素の下部のパディングを指定します。取り得る値は以下の通りです:

- 数値あるいは文字列値を使用して表現されるサイズ:
 - 数値: wk layout unitの単位でのサイズ
 - 文字列: 値と単位が連結されたCSS文字列。例:12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。
- wk none (デフォルト): 特定のパディングを指定しません。

要素の左側のパディングを指定します。取り得る値は以下の通りです:

- 数値あるいは文字列値を使用して表現されるサイズ:
 - 数値: wk layout unitの単位でのサイズ
 - 文字列: 値と単位が連結されたCSS文字列。例:12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。
- wk none (デフォルト): 特定のパディングを指定しません。

要素の右側のパディングを指定します。取り得る値は以下の通りです:

- 数値あるいは文字列値を使用して表現されるサイズ:
 - 数値: wk layout unitの単位でのサイズ
 - 文字列: 値と単位が連結されたCSS文字列。例:12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。
- wk none (デフォルト): 特定のパディングを指定しません。

要素の上部のパディングを指定します。取り得る値は以下の通りです:

- 数値あるいは文字列値を使用して表現されるサイズ:
 - 数値: wk layout unitの単位でのサイズ
 - 文字列: 値と単位が連結されたCSS文字列。例:12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。
- wk none (デフォルト): 特定のパディングを指定しません。

段落

段落属性を使用すると段落内でのテキストの配置のためのプロパティを定義する事ができます。適用の対応表は以下の通りです：

ドキュメント	段落	文字	ピクチャー	テーブル	行	カラム/セル	ヘッダー/フッター/ボディ
	0			0*	0*	0*	0*

* セル内の段落に適用

定数 コメント

段落のテキストの方向を指定します。取り得る値は以下の通りです:

wk
direction

- [wk left to right](#) (デフォルト)
- [wk right to left](#)

行間スペースを指定します。取り得る値は以下の通り:

wk line
height

- [wk normal](#) (デフォルト): テキストサイズに基づいた値を使用
- 数値あるいは文字列値で表現された高さを使用:
 - 数値: [wk layout unit](#) の単位での高さ
 - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。相対値(パーセンテージ%)はサポートされています。

段落レンジにおけるTabの先頭文字を指定します。

取り得る値:

wk tab
stop
leadings

- テキスト配列
各Tabオフセット用の先頭文字が順番に並んだリスト(左マージンからスタート)。最後の値によって定義された文字がそれぞれの段落内で入力された追加のタブに対して繰り返し使用されます。
- テキスト
単一の文字が先頭文字として使用されます。二つ以上の文字が指定されていた場合には、最初のものだけが使用されます。

[wk tab stop offsets](#) および [wk tab stop types](#) 配列と同期しています。配列のサイズが(空でなくて)合致しない場合、配列の最後の文字がTabのデフォルトとして使用されます。配列が空の場合、なしの文字が使用されます。

注: 最後に使用された文字が常にTabのデフォルト文字となります。

段落のタブストップを指定します。取り得る値は以下の通りです:

wk tab
stop
offsets

- タブ値の配列: 絶対値の順番付けされたリストで、左のマージンから始まります。最後の値によって定義されたタブオフセットは、段落内で入力される追加のタブ文字にて繰り返し使用されます。タブオフセットが段落の幅より大きい場合、テキストは次の行に移行し、最初のタブ値から開始されます。配列内の値が以前の値より小さい場合、それは無視されます。
- スカラー値(デフォルトは35.45pt): 段落全体のデフォルトのオフセットです。**WP GET ATTRIBUTES** コマンドは最後に使用されたオフセットを返します(最後の絶対オフセットよりあとのオフセットについては、デフォルトの相対オフセットになります)。

Note: 同じ呼び出しにおいて異なる属性に対して配列とスカラーを同時に使用することはできません。

値はCSS文字列(デフォルト)または[wk layout unit](#)での単位を使用する実数値で表現されます。最大値は10,000ptです。

[wk tab stop leadings](#) および [wk tab stop types](#) 配列と同期しています。

定数 コメント

段落でのタブストップタイプを指定します。取り得る値は以下の通りです:

wk tab
stop
types

- タブストップ値の配列(タブストップが配列を通して定義されていた場合)
- wk left (デフォルト): テキストはタブストップから右側に拡張されます。
- wk right: タブスペースが埋まるまで、テキストはタブストップから左側に拡張されます。
- wk center: テキストはタブストップにて中央に配置されます。
- wk decimal: 小数点より前のテキストは左に拡張され、小数点より後のテキストは右に拡張されます。
- wk bar: 指定された位置に垂直の線がおかれます。

wk tab stop offsets および wk tab stop leadings 配列と同期しています。

段落内でのテキストの行揃えを指定します。取り得る値は以下の通りです:

wk text
align

- wk left (デフォルト)
- wk right
- wk justify
- wk center

段落の最初の行のインデントを指定します。取り得る値は以下の通りです:

wk text
indent

- 実数: wk layout unit.の単位でのサイズ。デフォルトは0です。
- 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 10000pt

要素の垂直揃えを設定します。文字、段落、ピクチャーに対して使用可能です。取り得る値は以下の通りです:

wk
vertical
align

- wk baseline (デフォルト): 要素のベースラインを親要素のベースラインと揃えます。
- wk top: 要素の上を、行上で最も高い要素の上と揃えます。
- wk bottom: 要素の下を、行上で最も低い要素の下と揃えます。
- wk middle: 要素は親要素の中央に配置されます。
- wk superscript: 要素を上付き文字として配置します。
- wk subscript: 要素を下付き文字として配置します。

文字においては、wk topとwk bottomはwk baselineと同じ効果を持ちます。

段落においては、wk baseline、wk superscriptとwk subscript はwk topと同じ効果を持ちます。

スタイルシート

スタイルシート属性を使用すると指定されたスタイルシートを管理する事ができます。適用の対応表は以下の通りです:

ドキュメント	段落	文字	ピクチャー	テーブル	行	カラム/セル	ヘッダー/フッター/ボディ
	O		O**				O*

*ヘッダーとフッターに対しては読み込みのみ。これらは4D Write Proで管理されます(セクションヘッダーあるいはフッターには一つスタイルシートがあります)

**インライン画像のみ。

定数

コメント

段落に新たな行を追加する際のスタイルシートを指定します。取り得る値は以下の通りです:

wk new line style
sheet

- 既存のスタイルシート名
- wk none (デフォルト)

選択された要素に対してカレントのスタイルシートを指定します。取り得る値は以下の通りです:

wk style sheet

- wk none (デフォルト)
- 既存のスタイルシート名

レンジ

レンジ属性は(wk id を除いて)読み込みのみであり、レンジプロパティについての情報を提供します。レンジ属性はレンジオブジェクトに対してのみ適用可能です。

定数

コメント

wk end

(読み込み専用のレンジ属性)

wk id

(読み込みのみ属性)要素のID。値:文字列

wk owner

(読み込み専用のレンジ属性)

wk start

(読み込み専用のレンジ属性)

(読み込みのみ可能なレンジ属性) 4D Write Pro オブジェクトの型。取りうる値:

wk type

- wk type default: 未定義の型のレンジ
- wk type paragraph: 段落型のレンジ
- wk type image: 画像(アンカーされた画像およびインライン画像)
- wk type image anchored: アンカーされた画像
- wk type image inline: インライン画像
- wk type container: 例えばヘッダーやフッターなど
- wk type table: 表参照
- wk type table row: 表の行参照
- wk type table cell: 表のセル参照
- wk type table column: 表のカラム参照(カラムレンジに対してのみ)
- wk type body: 本文参照

テーブル

テーブル属性はテーブルレンジに対応しています。適用の対応表は以下の通りです：

ドキュメント 段落 文字 ピクチャー テーブル 行 カラム/セル ヘッダー/フッター/ボディ

0

定数	コメント
wk cell count	行内にあるセルの総数。値の型: 倍超整数(<u>wk type table row</u> 用の値)
wk column count	(読み込みのみ属性)レンジ内にあるカラムの総数。値:倍長整数
wk first column	(読み込みのみ属性)テーブル内にある最初のカラムの数。値:倍長整数
wk first row	(読み込みのみ属性)テーブル内の最初の行の数。値:倍長整数
wk row count	(読み込みのみ属性)行の総数。値:倍長整数
	テーブル要素の水平方向並びを指定します。取り得る値は以下の通り:
wk table align	• <u>wk left</u> (デフォルト) • <u>wk right</u> • <u>wk center</u>
wk table ID	(読み込みのみ属性)親テーブルのID。値:文字列

表の中で連結された要素に対して異なる属性が適用された場合、以下の順番で適用されてレンダリングされます：

1. 表が最初にレンダリングされます。
2. 行がレンダリングされます(このとき表の属性を上書きします)
3. セル/列がレンダリングされます(このとき行の属性を上書きします)

4D Write Proリファレンス - 新着

	17.0	16.2	16
⚙ WP Create table range	New 17.0		
⚙ WP Get element by ID	New 17.0		
⚙ WP Get elements	New 17.0		
⚙ WP Get position	New 17.0		
⚙ WP Get text	New 17.0		
⚙ WP INSERT BREAK	Updated 17.0		
⚙ WP SET TEXT	New 17.0		
⚙ WP Table get cells	Updated 17.0		
⚙ WP Table get columns	Updated 17.0		
⚙ WP Table get rows	Updated 17.0		
📄 4D Write Pro エリアを使用する	Updated 17.0		
📄 4D Write ドキュメントの読み込み	Updated 17.0		
📄 4D Write Pro標準アクションの使用	Updated 17.0		
☰ 4D Write Pro属性	Updated 17.0		

4D Write Proリファレンス - コマンドリスト (文字順)

W

- ⚙ WP CREATE BOOKMARK
- ⚙ WP Create paragraph range
- ⚙ WP Create picture range
- ⚙ WP Create range
- ⚙ WP DELETE BOOKMARK
- ⚙ WP EXPORT DOCUMENT
- ⚙ WP EXPORT VARIABLE
- ⚙ WP GET ATTRIBUTES
- ⚙ WP Get body
- ⚙ WP Get bookmark range
- ⚙ WP GET BOOKMARKS
- ⚙ WP Get footer
- ⚙ WP Get frame
- ⚙ WP Get header
- ⚙ WP Get page count
- ⚙ WP Get position New 17.0
- ⚙ WP Import document
- ⚙ WP INSERT DOCUMENT
- ⚙ WP INSERT PICTURE
- ⚙ WP Insert table
- ⚙ WP New
- ⚙ WP PRINT
- ⚙ WP SET ATTRIBUTES
- ⚙ WP SET FRAME
- ⚙ WP Table append row
- ⚙ WP Table get cells Updated 17.0
- ⚙ WP Table get rows Updated 17.0
- ⚙ WP USE PAGE SETUP