

GKitTaglib

# Spreadsheet V2

(スプレッドシート バージョン2)  
マニュアル

20120823 版

# 目次

SpreadsheetV2 概要.....	5	header タグ.....	19
製品構成 .....	5	view タグ.....	20
インストール配布物 .....	5	● start アトリビュート.....	20
動作環境 .....	6	● length アトリビュート.....	20
セットアップ.....	6	record タグ.....	21
インストール方法 (Windows) .....	6	col タグ.....	21
インストール方法 (RedHat、Solaris) .....	8	● width アトリビュート.....	21
設定について.....	8	tr タグ.....	22
JSP タグ説明.....	10	● height アトリビュート.....	22
タグ構成 .....	10	● bgcolor アトリビュート.....	22
table タグ.....	11	● align アトリビュート.....	23
● border アトリビュート.....	11	● valign アトリビュート.....	23
● width アトリビュート.....	11	● fontcolor アトリビュート.....	24
● height アトリビュート.....	12	● fontsize アトリビュート.....	24
● bordercolor アトリビュート.....	12	● fontname アトリビュート.....	25
● bgcolor アトリビュート.....	13	● style アトリビュート.....	25
● cursorcolor アトリビュート.....	13	td タグ.....	26
● selectcolor アトリビュート.....	14	● bgcolor アトリビュート.....	26
● editcolor アトリビュート.....	14	● align アトリビュート.....	26
● align アトリビュート.....	15	● valign アトリビュート.....	27
● cellpadding アトリビュート.....	15	● colspan アトリビュート.....	27
● objectid アトリビュート.....	16	● rowspan アトリビュート.....	28
● objectscope アトリビュート.....	16	● fontcolor アトリビュート.....	28
● headercol アトリビュート.....	16	● fontsize アトリビュート.....	29
● fixedcol アトリビュート.....	17	● fontname アトリビュート.....	29
● checkname アトリビュート.....	17	● sort アトリビュート.....	30
● textname アトリビュート.....	17	● style アトリビュート.....	30
● selectmode アトリビュート.....	18	data タグ.....	31
● cursorhidden アトリビュート.....	18	● property アトリビュート.....	31
● changevh アトリビュート.....	18	● value アトリビュート.....	31
● pluralselect アトリビュート.....	18	● hidden アトリビュート.....	31
● contextmenu アトリビュート.....	18	● name アトリビュート.....	31
● editclearmode アトリビュート.....	19	text タグ.....	32
● caretcontrol アトリビュート.....	19	● property アトリビュート.....	32

● type アトリビュート .....	32	● hrefvalues アトリビュート .....	43
● name アトリビュート .....	32	● target アトリビュート .....	44
● restriction アトリビュート .....	32	記述方法 .....	45
● numericvalue アトリビュート .....	33	プロパティファイルの設定 .....	46
● imemode アトリビュート .....	33	配置場所 .....	46
● maxvalue アトリビュート .....	33	設定項目 .....	46
● minvalue アトリビュート .....	33	記述方法 .....	47
● maxlength アトリビュート .....	34	制限事項 .....	48
● minlength アトリビュート .....	34	メソッド .....	49
● value アトリビュート .....	34	メソッドの使い方 .....	49
● format アトリビュート .....	34	cursorChange メソッド .....	49
● onchange アトリビュート .....	36	setSelectRow メソッド .....	49
button タグ .....	37	setText メソッド .....	50
● property アトリビュート .....	37	getText メソッド .....	50
● type アトリビュート .....	37	getCellType メソッド .....	50
● name アトリビュート .....	37	isOpend メソッド .....	50
● onclick アトリビュート .....	37	setSelectOption メソッド .....	50
● value アトリビュート .....	37	clear メソッド .....	51
check タグ .....	38	close メソッド .....	51
● property アトリビュート .....	38	getRow メソッド .....	51
● type アトリビュート .....	38	getCol メソッド .....	51
● name アトリビュート .....	38	focus メソッド .....	52
● value アトリビュート .....	38	onSortASC イベント .....	52
● checked アトリビュート .....	38	onSortDESC イベント .....	53
● disabled アトリビュート .....	39	onChangeEvent イベント .....	54
● optionvalues アトリビュート .....	39	onClickEvent イベント .....	55
select タグ .....	39	onTabExitEvent イベント .....	56
● property アトリビュート .....	39	テーブルデータクラスの作成方法 .....	57
● value アトリビュート .....	40	機能説明 .....	58
● name アトリビュート .....	40	セル編集 .....	58
● onchange アトリビュート .....	41	コピー&ペースト .....	58
● optionvalue アトリビュート .....	41	内容の削除 .....	58
● optionvalues アトリビュート .....	41	セル選択 .....	58
● selected アトリビュート .....	42	スクロール .....	58
link タグ .....	43	行列のサイズ変更 .....	59
● value アトリビュート .....	43	ソート .....	59
● property アトリビュート .....	43	チュートリアル .....	60
● hrefvalue アトリビュート .....	43	HOW TO .....	63

行ごとの色を交互に設定したい .....	63
サブミットされたデータを取得したい .....	65
行選択されていた行の番号を取得したい .....	65
表示されるデータ件数について .....	65
ライセンス .....	66

サポートについて .....	66
制限事項 .....	67
付録 .....	68
タグ一覧 .....	68
ショートカットキー .....	73

## SpreadsheetV2 概要

Spreadsheet V2 (スプレッドシート バージョン2) は JSP で使用するカスタムタグライブラリです。Spreadsheet V2 を使用することでスクロールやソート機能、編集機能が付いた表を簡単に作成することができます。SpreadsheetV2 の記述は HTML のテーブルタグと同じような使い方で記述します。HTML を知っていれば簡単に結合セルなどの設定も可能です。

## 製品構成

インストールされる製品構成および動作環境を説明します。

### インストール配布物

以下のフォルダ構成でモジュールの配布が行われます。

#### [インストールフォルダ]

[jsp] ..... サンプルの JSP がはいています。

[scripts]

[GkitTlib]

spreadv2ext.htc ..... SpreadsheetV2 のスクリプトファイルです。

[WEB-INF]

[lib]

GkitTlibSpd2.jar ..... SpreadsheetV2 のライブラリです。

GkitTlibSpd2Lic.jar ..... SpreadsheetV2 のライセンスファイルです。

[src] ..... サンプルソースが入っています。

[tlds]

GkitTlibSpd2.tld ..... SpreadsheetV2 の TLD ファイルです。

SPREADV2DOC.pdf ..... 製品マニュアルです。

SampleForApcoordinator.war ..... Apcoordinator を使用したサンプルです。

SampleForStruts.war ..... Struts を使用したサンプルです。

README.txt ..... ソフトウェア説明書です。

SUPPORT.txt ..... 製品の問い合わせをする為の説明書です。

## 動作環境

下記の動作環境をサポートしています。

### 【サーバ】

- オペレーティングシステム
  - Microsoft Windows Server 2003 / 2003R2 / 2008 / 2008 R2
  - RedHat Linux Enterprise Edition version 3 / 4 / 5 / 6
  - Solaris 9 (SPARC 版) / Solaris10 (SPARC 版) / Solaris11 (SPARC 版)
- JSP コンテナ
  - Fujitsu Interstage Application Server V7.0 / V8.0 / V9.0 / V9.1 / V9.2 / V10.1
  - BEA WebLogic Server 8.1J / 9.2J / 10.0J
  - IBM WebSphere Application Server V5.0 / V6.1
  - Jakarta Tomcat 4.1 / 5.5 / 6.0 / 7.0
  - Servlet API2.2 JSP1.1 に準拠するコンテナ

### 【クライアント】

- オペレーティングシステム
    - Microsoft Windows XP
    - Microsoft Windows Vista
    - Microsoft Windows 7
  - ブラウザ
    - Microsoft Internet Explorer 7.0 / 8.0 / 9.0 ( )
- (※ Windows Vista / 7 のみ)

## セットアップ

ここでは各OSでのセットアップ手順について説明いたします。

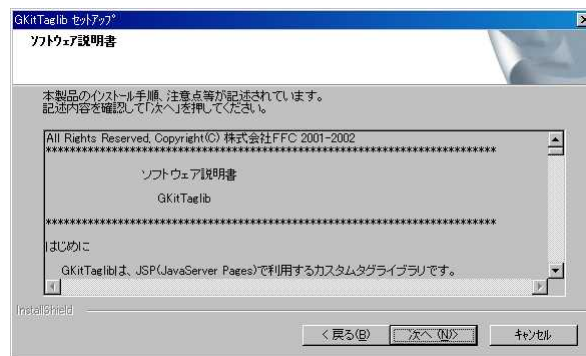
### インストール方法 (Windows)

インストールする端末にCDをいれるとインストーラが自動的に起動します。

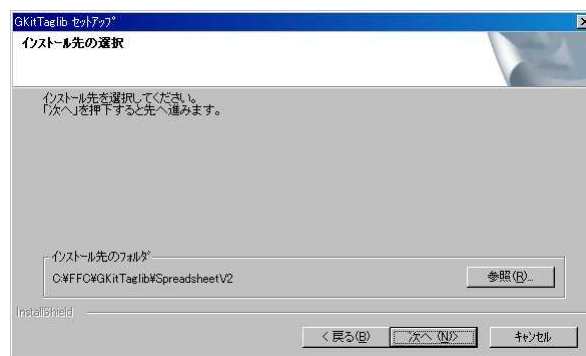
※自動的に起動しない場合はCD内のINSTALL.exe を実行してください。



ソフトウェア説明書が表示されるので一読してください。



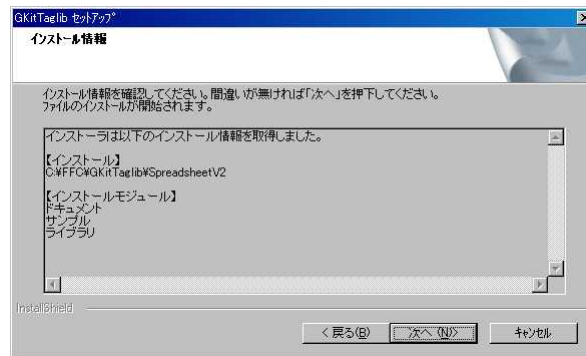
インストール先を選択してください。



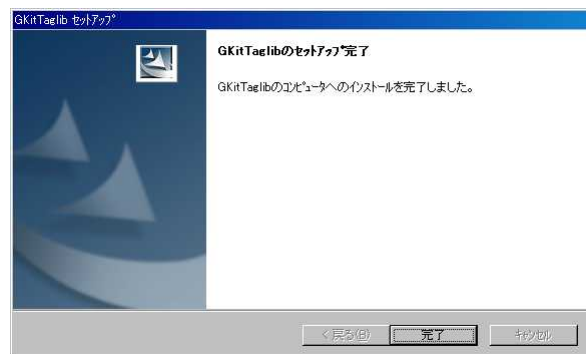
インストールするコンポーネントを選択してください。



インストール情報が表示されますので確認してください。  
次へを押下するとファイル転送が始まります。



完了画面が表示されるとインストールは終了です。



インストールフォルダを開きモジュールが配布されているか確認してください。

#### インストール方法（RedHat、Solaris）

RedHat、Solaris でインストールを行うために CD の tar フォルダに TAR ファイルを用意してあります。tar コマンドでファイルを展開してください。

※インストールされるものは Windows インストーラでインストールされるものと同一です。

#### 設定について

動作させる為に必要な設定について説明します。

- 1) GkitTlibSpd2.jar、GkitTlibSpd2Lic.jar にクラスパスを通す  
通常は [Web アプリケーション]/WEB-INF/lib に JAR ファイルを配置すれば問題ありません。
- 2) spreadv2ext.htc を配置する  
spreadv2ext.htc は[アプリケーションルート]/[scripts]/[GkitTlib]に配置しなければなりません。
- 3) GkitTlibSpd2.tld を配置する  
アプリケーション内に GkitTlibSpd2.tld を配置してください。
- 4) web.xml の設定  
web.xml にタグの設定を行ってください。



例

```
<taglib>  
    <taglib-uri>GkitTlibSpd2-taglib</taglib-uri>  
    <taglib-location>/WEB-INF/tlds/GkitTlibSpd2.tld</taglib-location>  
</taglib>
```

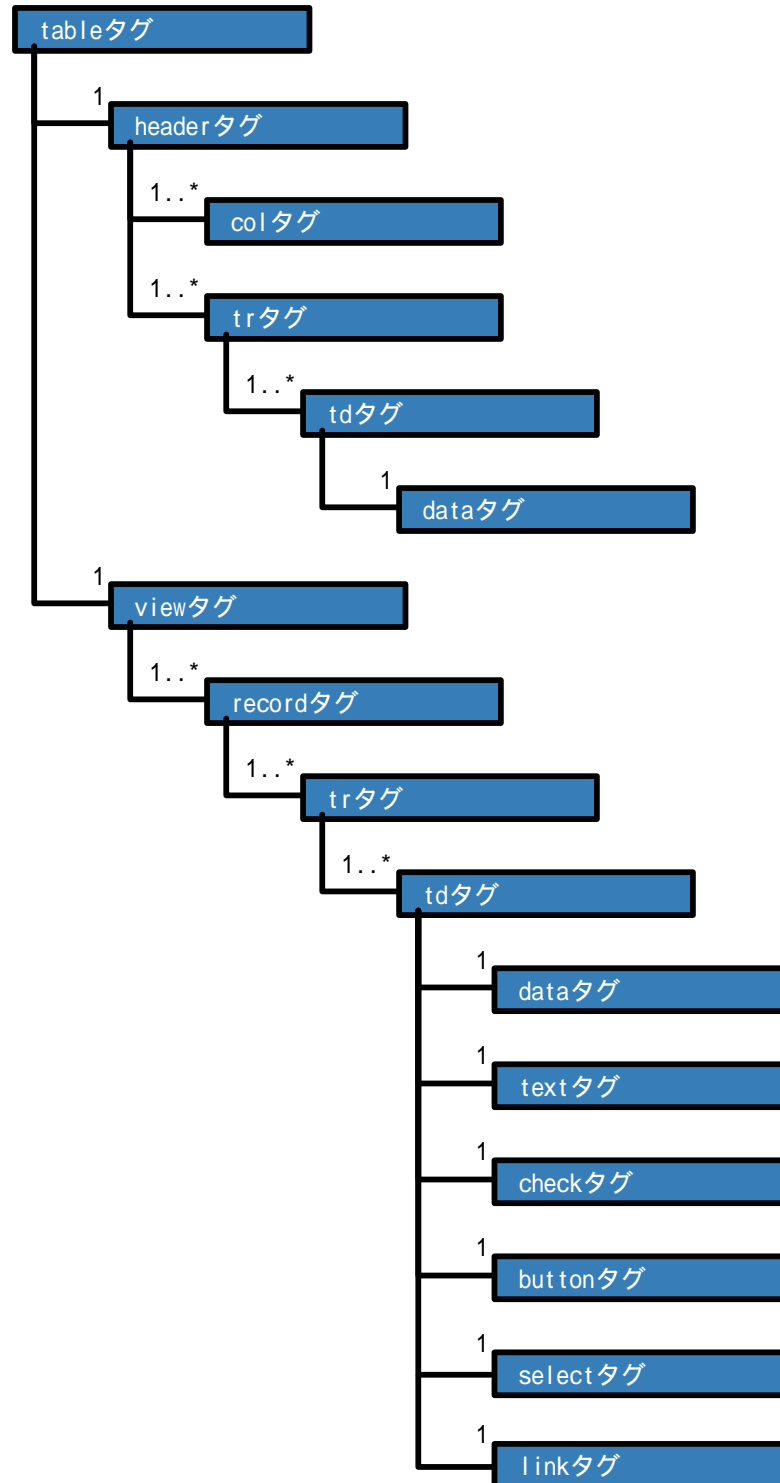
以上で環境設定は完了です。

## JSP タグ説明

ここでは SpreadsheetV2 の JSP タグについて説明します。

### タグ構成

次の図は SpreadsheetV2 のタグの構造図です。SpreadsheetV2 タグを記述する際は図のような構造で記述する必要があります。

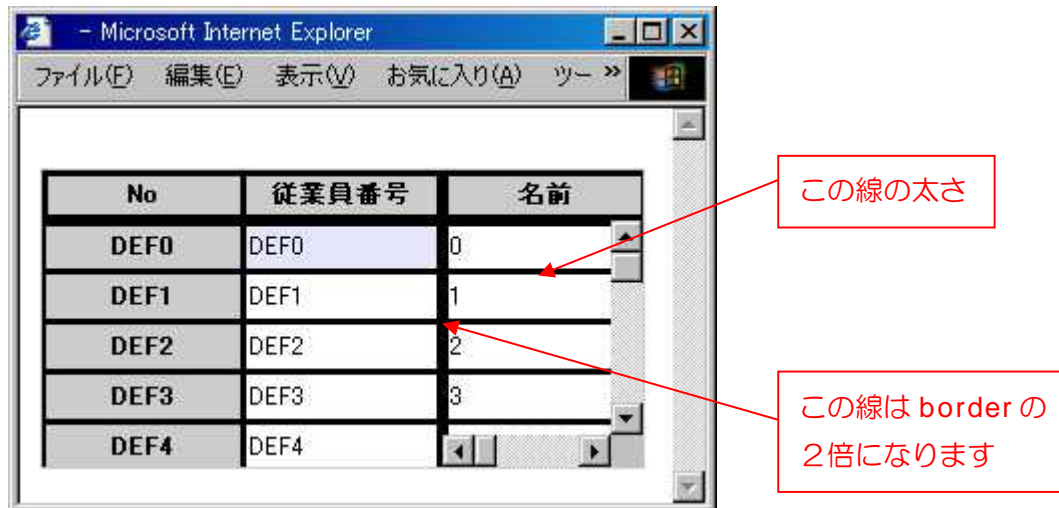


## table タグ

### ● border アトリビュート

border アトリビュートは表の罫線の太さを数値で設定します。

border="3"とすると太さが3px の罫線になります。



### ● width アトリビュート

width アトリビュートは表の幅を数値で設定します。

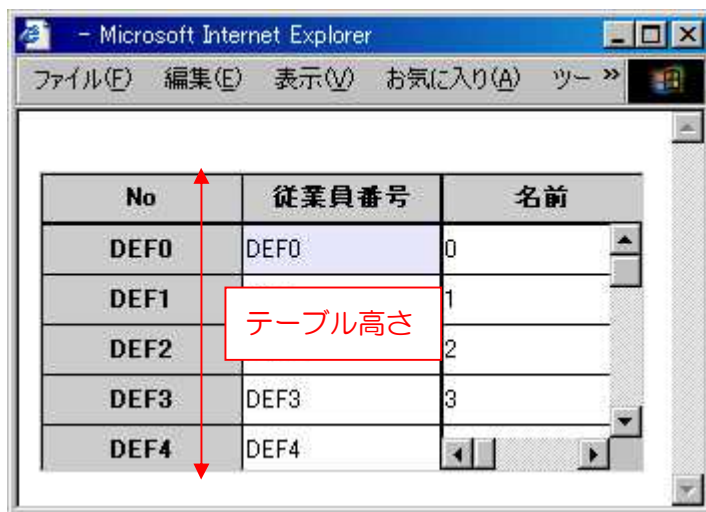
width="500"と設定すると幅が500px に設定されます。



- height アトリビュート

height アトリビュートは表の高さを数値で設定します。

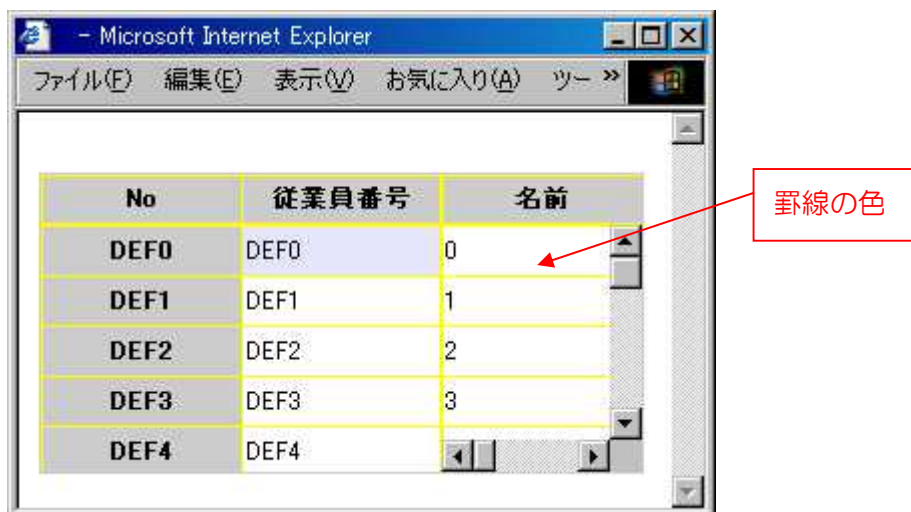
height="150"と設定すると高さが 150px に設定されます。



- bordercolor アトリビュート

bordercolor アトリビュートは罫線の色を設定します。

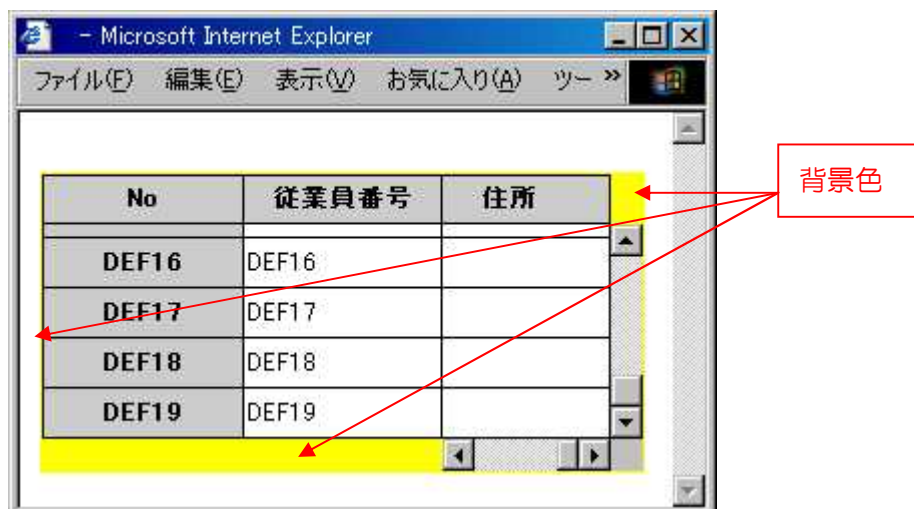
bordercolor="#FFFF00"と設定すると罫線が黄色に設定されます。



- bgcolor アトリビュート

bgcolor は表の背景色を設定します。

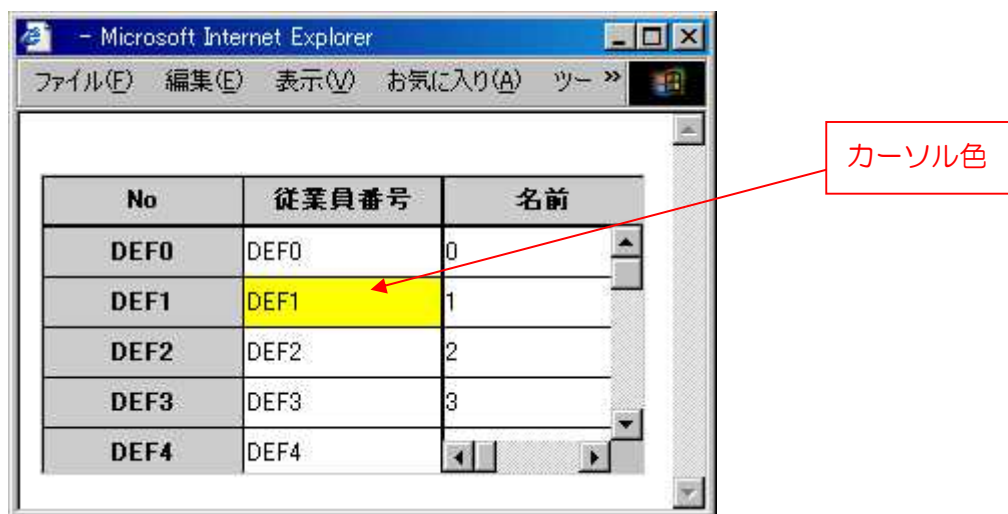
bgcolor="#FFFF00"と設定すると背景が黄色に設定されます。



- cursorcolor アトリビュート

cursorcolor アトリビュートはカーソルの色をを設定します。

cursorcolor="#FFFF00"と設定すると黄色に設定されます。



- selectcolor アトリビュート

selectcolor は選択色を設定します。

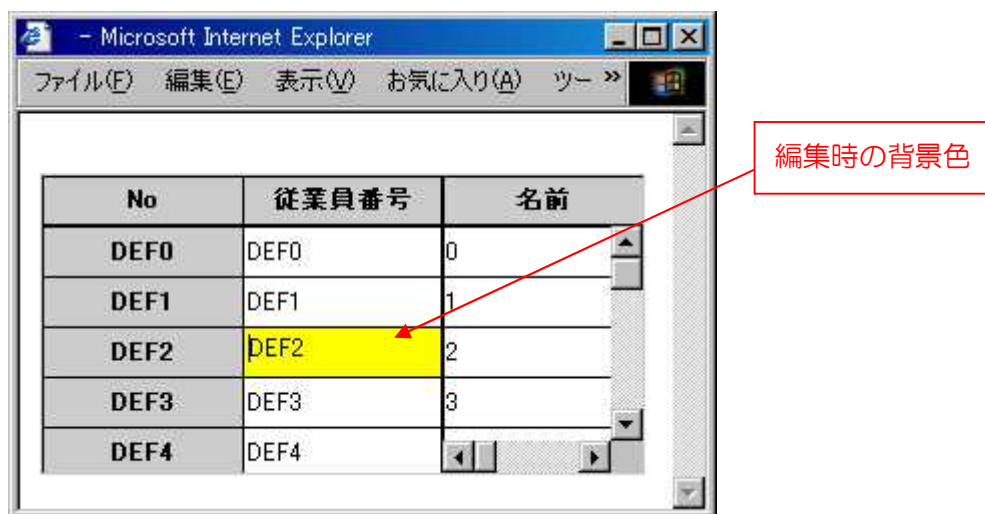
selectcolor="#FFFF00"を設定すると黄色に設定されます。



- editcolor アトリビュート

editcolor アトリビュートは編集時の背景色を設定します。

editcolor="#FFFF00"を設定すると編集時の背景色が黄色に設定されます。



- align アトリビュート

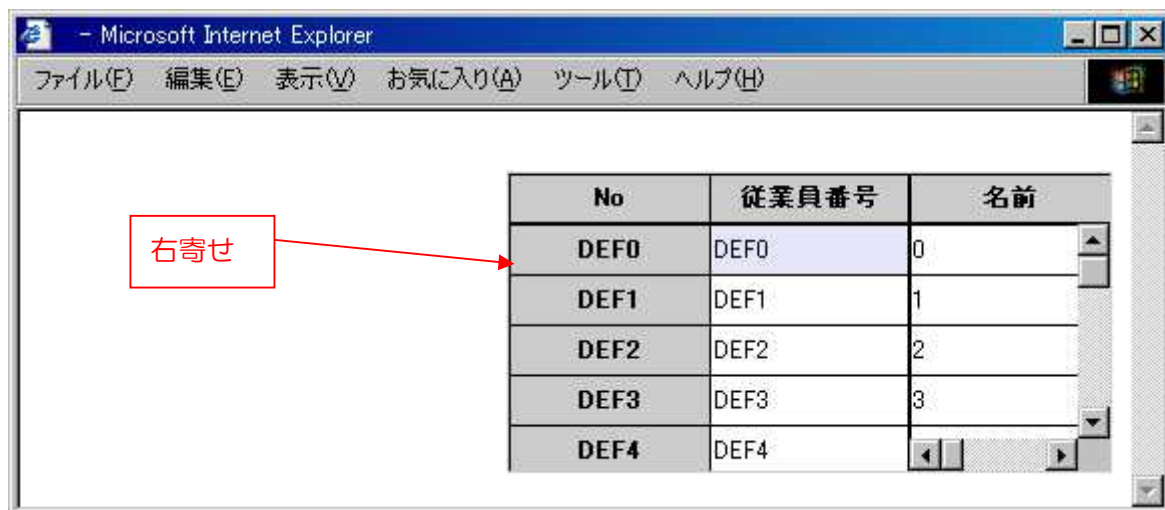
align は表の位置を設定します。以下の値を設定できます。

left：左寄せになります。

center：中央寄せになります。

right：右寄せになります。

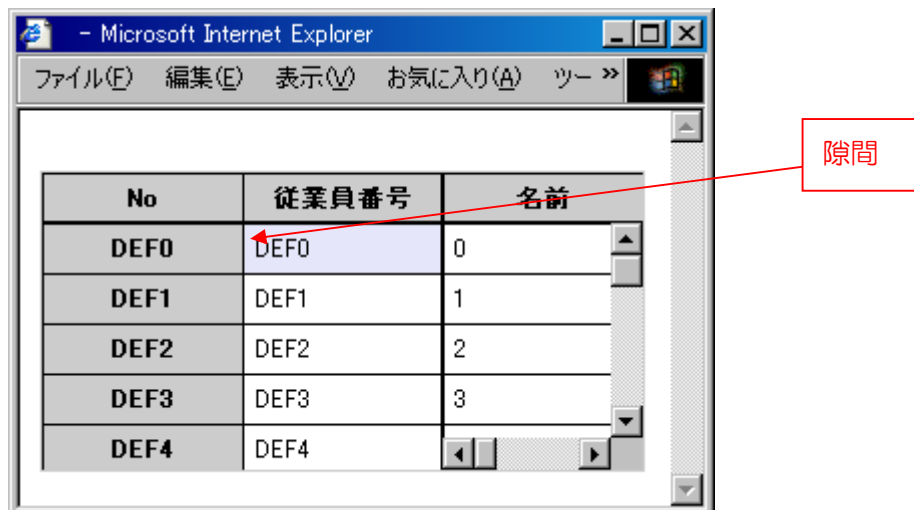
align="right"と設定すると右寄せになります。



- cellpadding アトリビュート

cellpadding アトリビュートはセルの内側の左側、上側の隙間を数値で設定します。

cellpadding="5"と設定すると隙間は 5px に設定されます。



- objectid アトリビュート

objectid アトリビュートはテーブルデータが含まれているオブジェクト名を指定します。

サブレット側でテーブルデータを下記のようにセットした場合、

```
request.setAttribute("sample01", data);
```

objectid="sample01"と設定することで各セルにデータを展開することができるようになります。

- objectscope アトリビュート

objectscope アトリビュートはテーブルデータが含まれているスコープを設定します。

以下の設定値があります。

request：リクエストからテーブルデータを取得します。

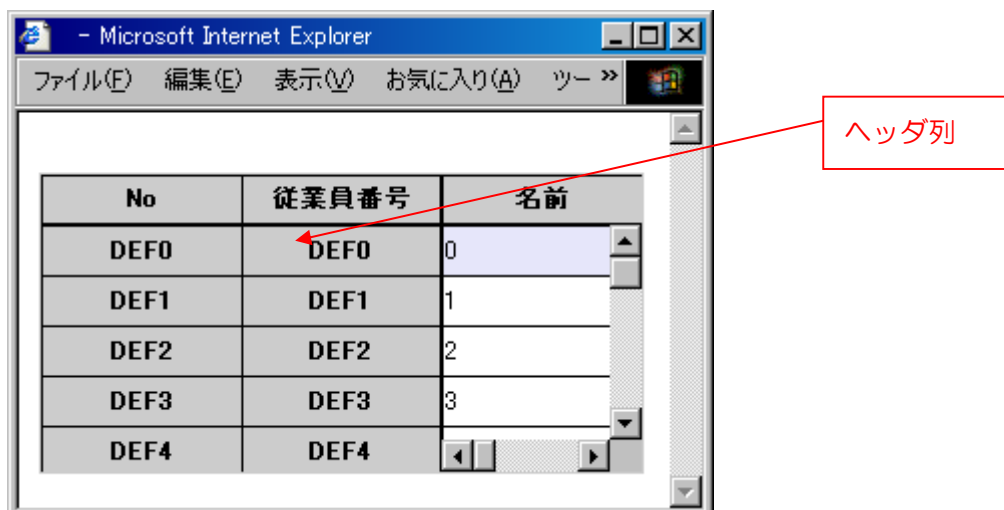
session：セッションからテーブルデータを取得します。

objectscope="request"とした場合はリクエストから objectid アトリビュートで設定したオブジェクトを取得します。

- headercol アトリビュート

headercol アトリビュートはヘッダ列数を数値で設定します。

headercol="2"とした場合はヘッダ列は2列になります。



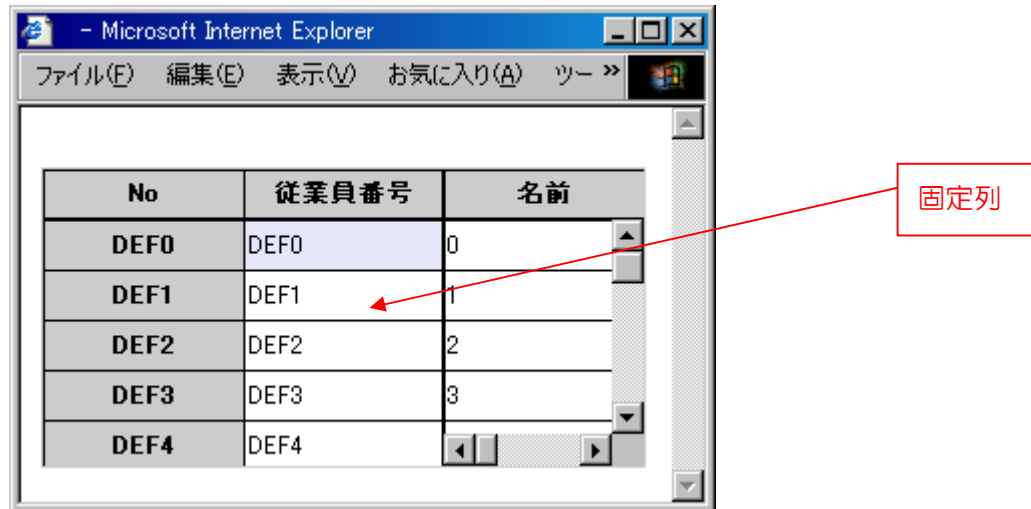
No	従業員番号	名前
DEF0	DEF0	0
DEF1	DEF1	1
DEF2	DEF2	2
DEF3	DEF3	3
DEF4	DEF4	

※ヘッダはデフォルトでは背景色は灰色、フォントは太字、中央揃えになります。



- fixedcol アトリビュート

fixedcol アトリビュートは固定列数を数値で設定します。  
fixedcol="1"とした場合は固定列数は1列になります。



- checkname アトリビュート

checkname アトリビュートはサーバー側で選択レコード番号を取得する際のキーを設定します。デフォルトでは"jp\_co\_ffc\_spread2\_check\_0"となります。2つ目以降のテーブルは"jp\_co\_ffc\_spread2\_check\_1"、"jp\_co\_ffc\_spread2\_check\_2"・・・とキー設定されます。

【サーバーでの取得例】

```
String[] check = request.getParameterValues("jp_co_ffc_spread2_check_0");
```

JSP に2つテーブルが存在し、1つ目のテーブルに checkname アトリビュートの設定がされていて2つ目は設定されていない場合、2つ目のテーブルの選択レコード番号を取得するためのキーは "jp\_co\_ffc\_spread2\_check\_1" となります。

- textname アトリビュート

textname アトリビュートは初期表示時の各レコード番号をサーバーで取得する場合のキーを設定します。デフォルトでは "jp\_co\_ffc\_gkit\_spread2\_text\_0"となります。2つ目以降のテーブルは"jp\_co\_ffc\_spread2\_text\_1"、"jp\_co\_ffc\_spread2\_text\_2"・・・とキー設定されます。選択されたレコードの初期表示時のレコード番号を取得する際に使用します。

【サーバーでの取得例】

//選択した一番上のレコードの初期表示時のレコード番号を取得します。

```
String[] check = request.getParameterValues("jp_co_ffc_spread2_check_0");
```

```
String[] text = request.getParameterValues("jp_co_ffc_spread2_text_0");
```

```
String selectRowNum = text[ Integer.parseInt( check[0] ) ];
```

JSP に2つテーブルが存在し、1つ目のテーブルに `textname` アトリビュートの設定がされていて2つ目は設定されていない場合、2つ目のテーブルの選択レコード番号を取得するためのキーは `"jp_co_ffc_spread2_text_1"` となります。

- `selectmode` アトリビュート

`selectmode` アトリビュートは選択モードを数値で設定することができます。

0：セル選択（デフォルト）

1：複数行選択モード

マウスクリックで行選択。選択行をクリックで選択解除となります。複数行選択を行うことができます。

2：単数行選択モード

マウスクリックで行選択。他の行をクリックすると前の選択行は解除されクリックした行が選択行になります。複数行選択は行うことができません。

3：選択なしモード

スクロールのみのモードになります。選択・編集・ソートは行うことができません。チェックボックスやセレクト、ボタン等の操作は可能です。

- `cursorhidden` アトリビュート

`cursorhidden` アトリビュートはカーソル表示を行わないように設定することができます。

true     ：カーソル表示を行わない

false    ：カーソル表示を行う（デフォルト）

- `changevh` アトリビュート

`changevh` アトリビュートはマウスによる列幅・行高さの変更を禁止することができます。

true     ：マウスによる列幅・行高さの変更が可能です。（デフォルト）

false    ：マウスによる列幅・行高さの変更ができません。

- `pluralselect` アトリビュート

`pluralselect` アトリビュートは `SHIFT`+クリックによる行選択を制御することができます。

true     ：`SHIFT`+クリックによる複数行範囲選択が可能です。（デフォルト）

false    ：複数行の範囲選択はできません。

`selectmode=1`（複数行選択モード）と一緒に設定することで、クリックイベントを使用し選択された行番号の順番を取得し易くなります。

- `contextmenu` アトリビュート

`contextmenu` アトリビュートは選択された領域を右クリックした際にコピー／貼り付けのメ

メニューを表示させることができます。

true : メニューを表示させます。

false : メニューを表示させません。(デフォルト)

#### ● editclearmode アトリビュート

editclearmode アトリビュートは、キー入力（F2、INSERT キーは除く）によりセルが編集状態に遷移した場合に、セル内の文字列を削除することができます。

true : セル内の文字列を削除します。

false : セル内の文字列を削除しません。(デフォルト)

#### ● caretcontrol アトリビュート

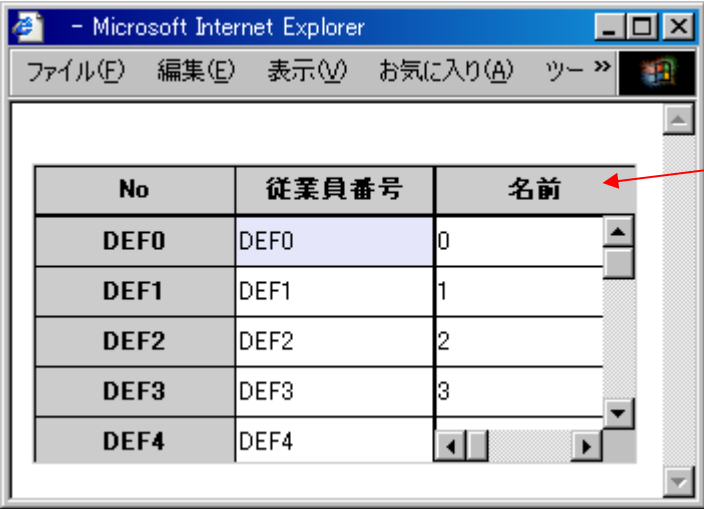
caretcontrol アトリビュートは、F2、INSERT キー押下によりセルが編集状態に遷移した場合に、文字列の右端にキャレットを移動させることができます。

true : F2、INSERT キー押下時にセルの右端にキャレットを移動します。

false : F2、INSERT キー押下時にセルの左端にキャレットを移動します。(デフォルト)

### header タグ

header タグはテーブルの上側のヘッダ部の設定を行います。header タグに記述された tr タグや td タグは上側のヘッダ部の設定となります。

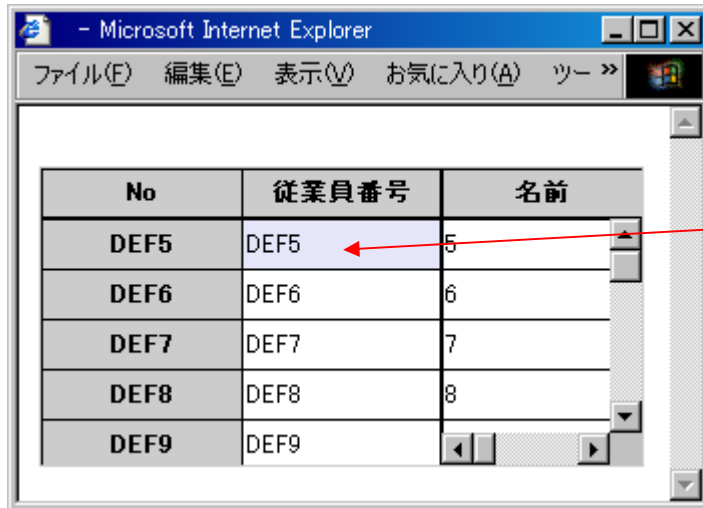


No	従業員番号	名前
DEF0	DEF0	0
DEF1	DEF1	1
DEF2	DEF2	2
DEF3	DEF3	3
DEF4	DEF4	

上部ヘッダ

### ●start アトリビュート

start アトリビュートは table タグ objectid で指定したテーブルデータの配列の開始番号を設定します。start="5"とした場合は配列の 5 番目からデータを表示します。



No	従業員番号	名前
DEF5	DEF5	5
DEF6	DEF6	6
DEF7	DEF7	7
DEF8	DEF8	8
DEF9	DEF9	

配列の開始番号

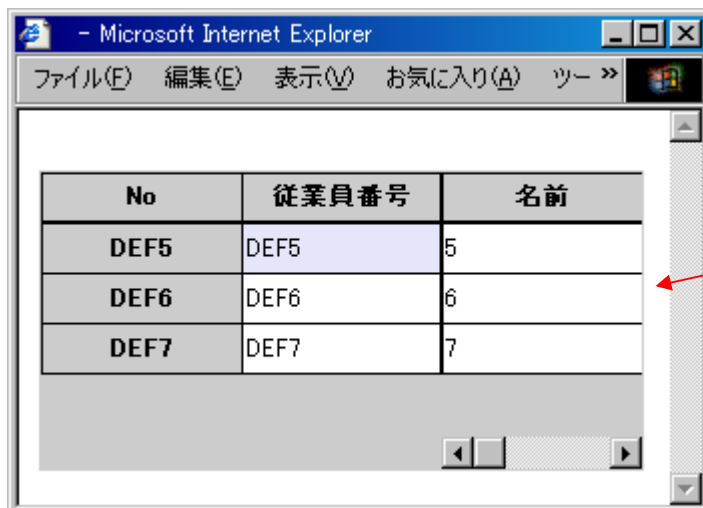
※配列が下記のようにになっている場合に start="2"とすると表の1レコード目には 7 と表示されます。

```
int[] l = {2, 5, 7, 4, 5, 8, 9};
```

### ●length アトリビュート

length アトリビュートは表に表示させるレコード数を設定します。

length="3"とした場合は3レコード表示されます。



No	従業員番号	名前
DEF5	DEF5	5
DEF6	DEF6	6
DEF7	DEF7	7

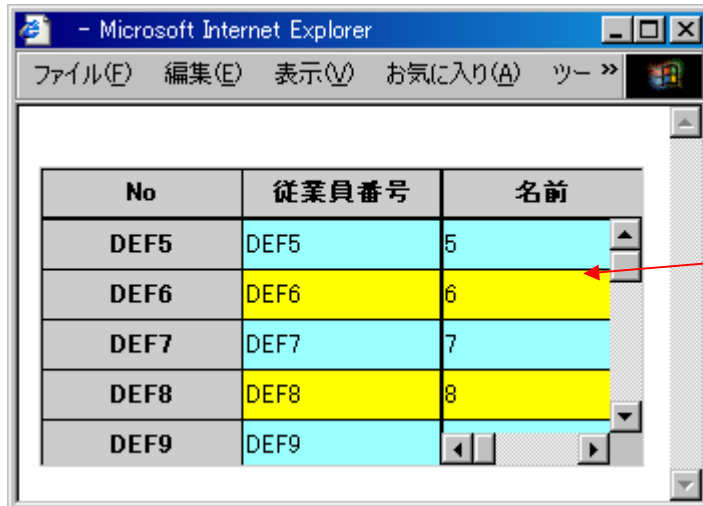
length 設定

## record タグ

record タグはレコードを設定するために使用します。

record タグで囲まれた部分を1レコードとして扱うことができます。

record タグを使用することで偶数レコードの背景色を黄色、奇数レコードの背景色を青色とすることもできます。



No	従業員番号	名前
DEF5	DEF5	5
DEF6	DEF6	6
DEF7	DEF7	7
DEF8	DEF8	8
DEF9	DEF9	

record タグ

record タグを複数使用する場合は下記の制限があります。

レコード毎に tr タグや td タグの構造を変更することはできません。

例えば奇数レコードは3行、偶数レコードは2行などは設定しないようにしてください。

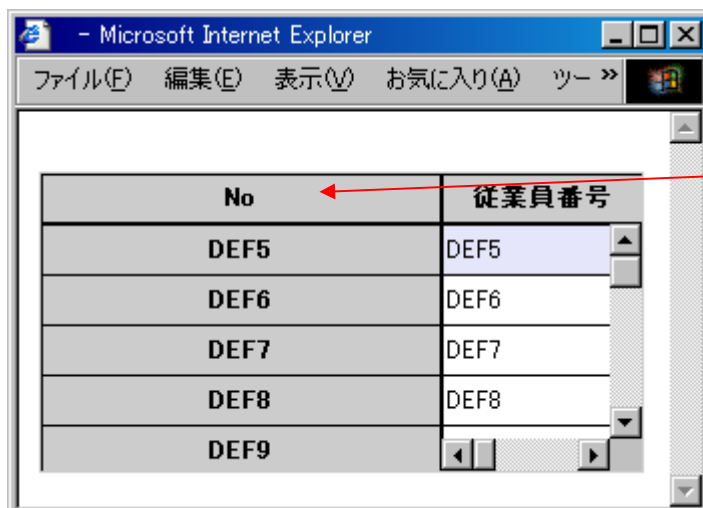
td タグの rowspan、colspan もレコード毎に設定しないでください。

## col タグ

### ● width アトリビュート

width アトリビュートは各列の幅を数値で設定します。

width=200 と設定すると設定された列は 200px となります。



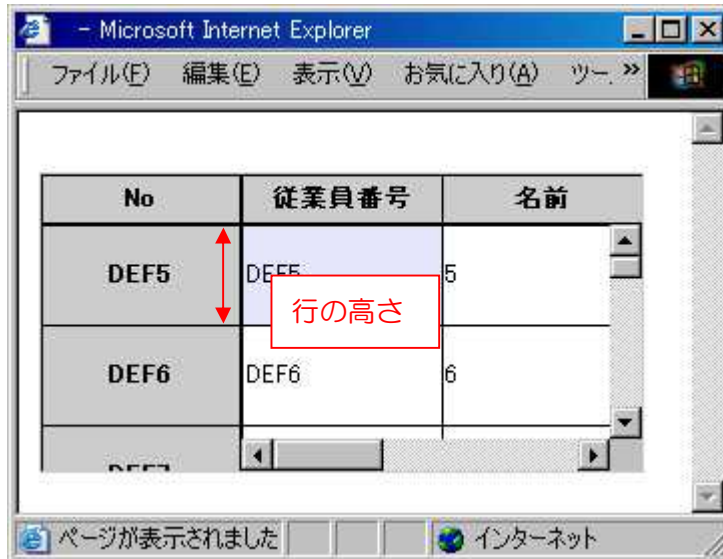
No	従業員番号
DEF5	DEF5
DEF6	DEF6
DEF7	DEF7
DEF8	DEF8
DEF9	

列幅

- height アトリビュート

height アトリビュートは行の高さを数値で設定します。

height="50"と設定すると設定された行の高さは 50px となります。



- bgcolor アトリビュート

bgcolor アトリビュートは行の背景色を設定します。

bgcolor="#FFFF00"と設定すると行の背景色は黄色に設定されます。



- align アトリビュート

align アトリビュートは行の各セルの横方向の表示位置を設定します。

left：左寄せになります。

center：中央寄せになります。

right：右寄せになります。

align="right"と設定すると各セルの表示位置は右寄せになります。



No	従業員番号	名前
DEF5	DEF5	
DEF6	DEF6	
DEF7	DEF7	
DEF8	DEF8	
DEF9		

- valign アトリビュート

valign アトリビュートは行の各セルの縦方向の表示位置を設定します。

top：上寄せになります。

middle：中央寄せになります。

bottom：下寄せになります。

valign="bottom"とすると各セルの表示位置は下寄せになります。

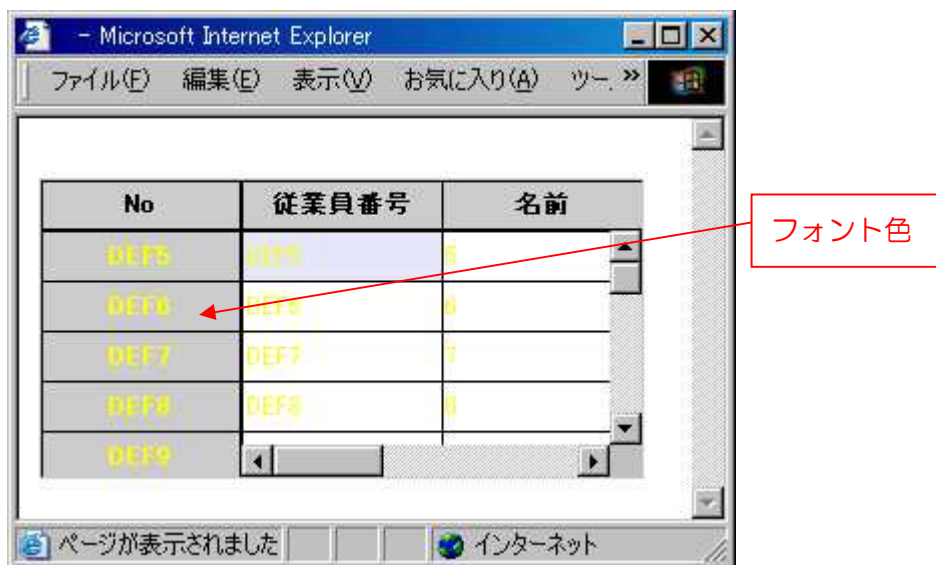


No	従業員番号	名前
DEF5	DEF5	5
DEF6	DEF6	6

- fontcolor アトリビュート

fontcolor アトリビュートは行の各セルのフォント色を設定します。

fontcolor="#FFFF00"とした場合はフォント色は黄色に設定されます。



- fontsize アトリビュート

fontsize アトリビュートは行の各セルのフォントサイズを設定します。

fontsize="20"とした場合はフォントサイズ 20px を設定されます。

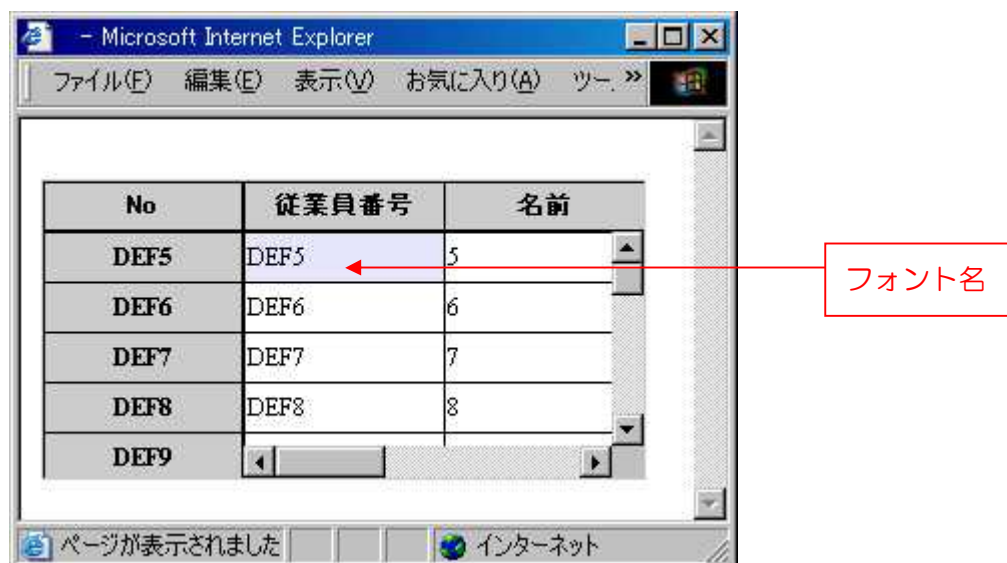




- fontname アトリビュート

fontname アトリビュートは行の各セルのフォント名を設定します。

fontname="HG 丸ゴシック M-PRO"とした場合、フォント名は HG 丸ゴシック M-PRO と設定されます。



- style アトリビュート

style アトリビュートは行の各セルの style を設定することができます。

style を設定することでアトリビュートに存在しないスタイルを設定することができます。

style アトリビュートは影響が非常に大きい為、使用は推奨されません。ご使用になる場合は影響がないよう注意してご使用ください。

- bgcolor アトリビュート

bgcolor アトリビュートはセルの背景色を設定します。

bgcolor="#FFFF00"と設定した場合、セルの背景色は黄色に設定されます。



- align アトリビュート

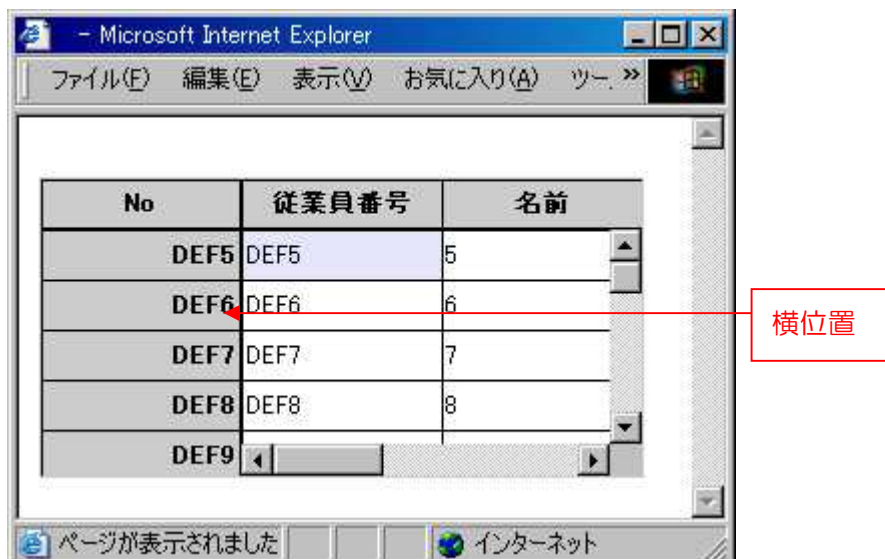
align アトリビュートはセルの横方向の表示位置を設定します。

left：左寄せになります。

center：中央寄せになります。

right：右寄せになります。

align="right"と設定すると各セルの表示位置は右寄せになります。



- valign アトリビュート

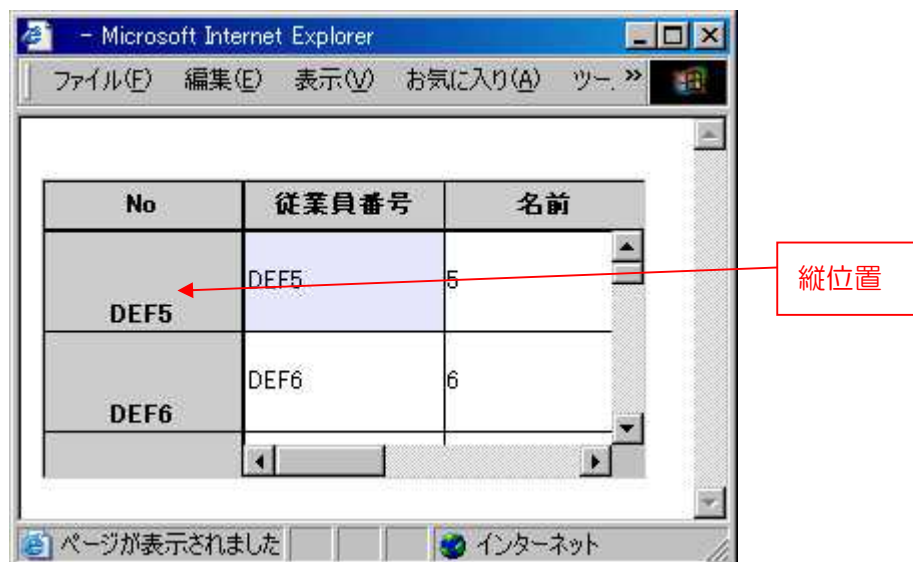
valign アトリビュートは行の各セルの縦方向の表示位置を設定します。

top：上寄せになります。

middle：中央寄せになります。

bottom：下寄せになります。

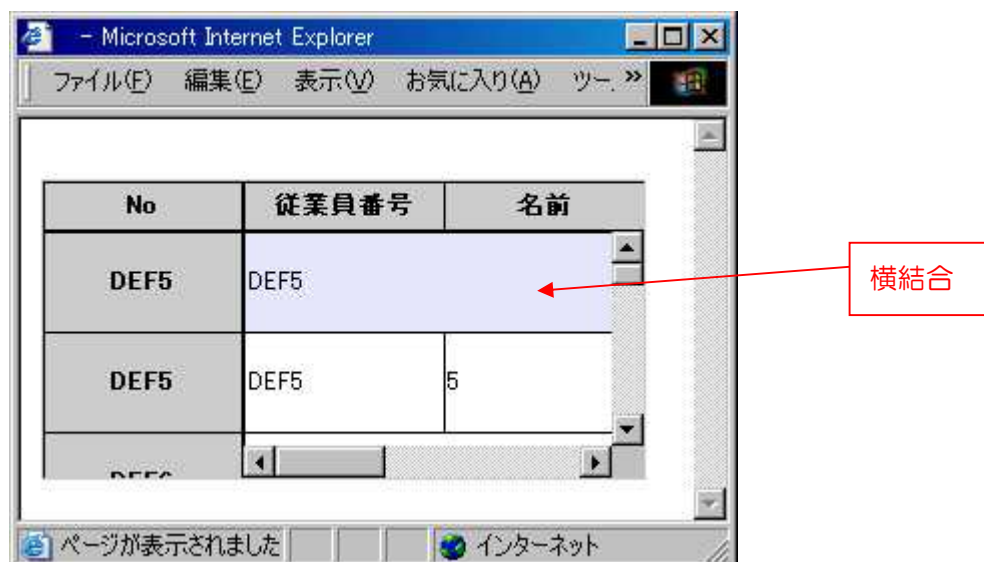
valign="bottom"とすると各セルの表示位置は下寄せになります。



- colspan アトリビュート

colspan アトリビュートでセルの横結合を行うことができます。

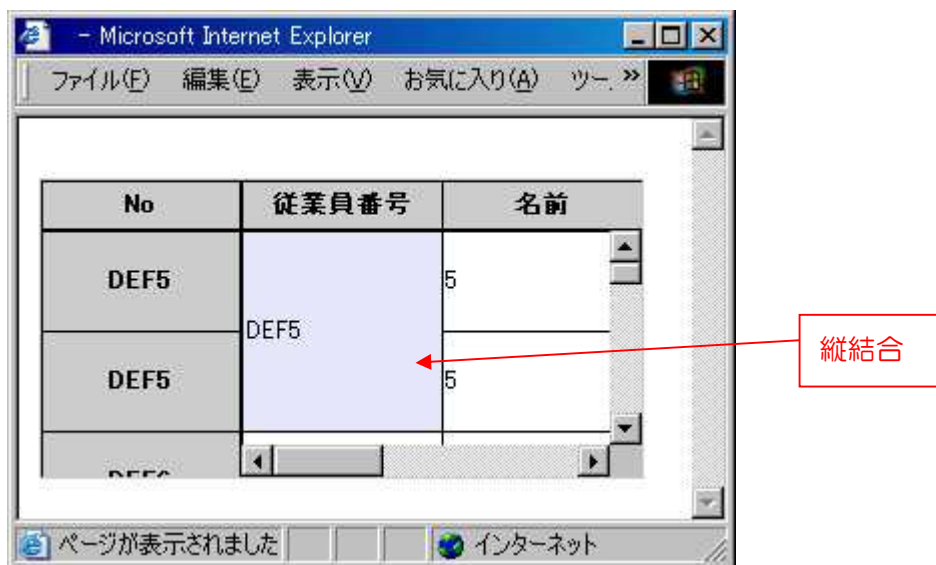
colspan="2"とした場合、2つのセルが横結合されます。



- rowspan アトリビュート

rowspan アトリビュートはセルの縦結合を行うことができます。

rowspan="2"とした場合、2つのセルが縦結合されます。

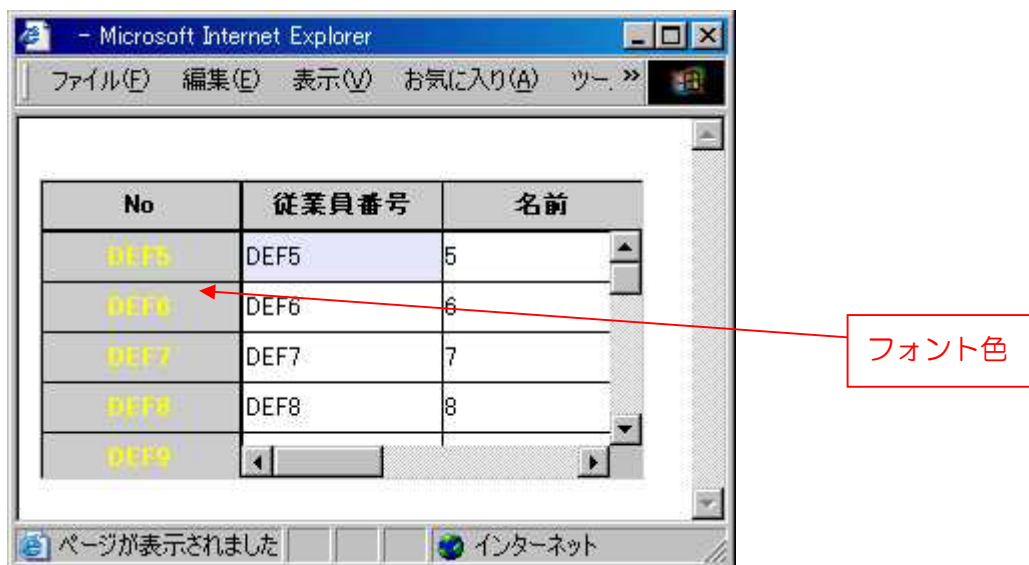


No	従業員番号	名前
DEF5	DEF5	5
DEF5		5
DEF6		

- fontcolor アトリビュート

fontcolor アトリビュートはセルのフォント色を設定します。

fontcolor="#FFFF00"とした場合はフォント色は黄色に設定されます。

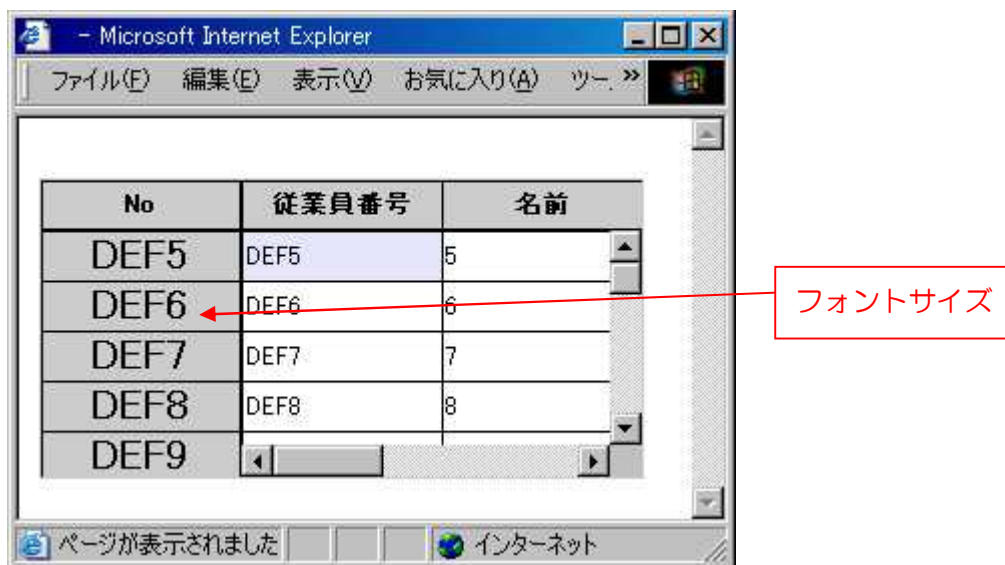


No	従業員番号	名前
DEF5	DEF5	5
DEF6	DEF6	6
DEF7	DEF7	7
DEF8	DEF8	8
DEF9		

- fontsize アトリビュート

fontsize アトリビュートはセルのフォントサイズを設定します。

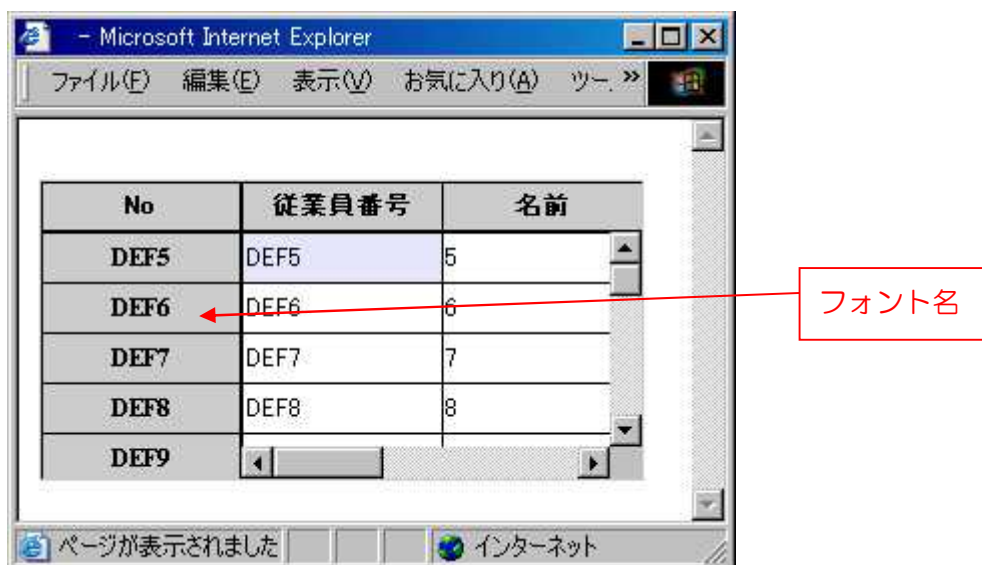
fontsize="20"とした場合はフォントサイズ 20px を設定されます。



- fontname アトリビュート

fontname アトリビュートはセルのフォント名を設定します。

fontname="HG 丸ゴシック M-PRO"とした場合、フォント名は HG 丸ゴシック M-PRO と設定されます。



### ● sort アトリビュート

sort アトリビュートはソートメニューを表示するために使用します。

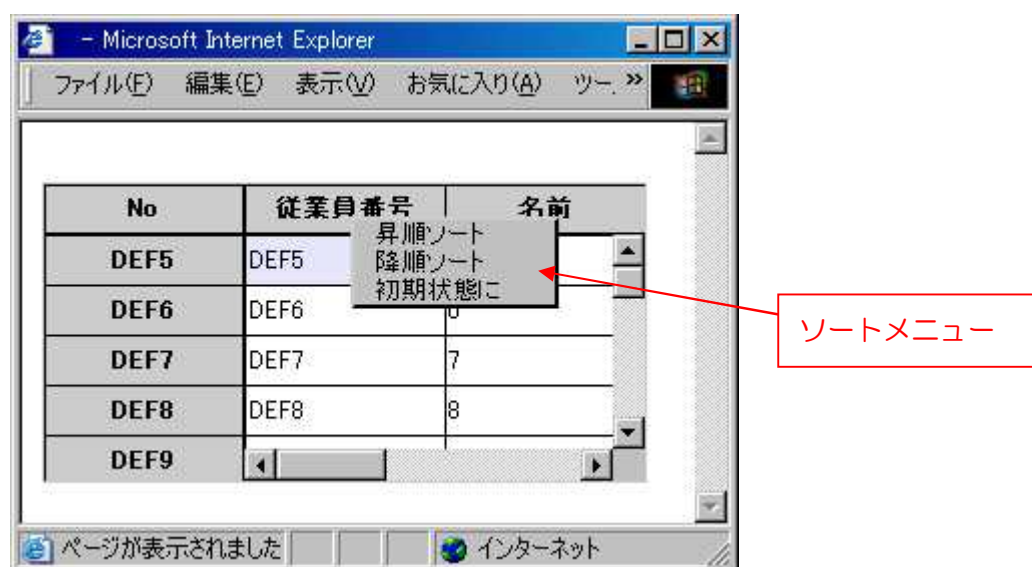
sort="行番号,列番号"と数字をカンマで区切った値を設定することで該当のセルをレコードでソートできるようになります。

例えば record タグで囲まれたセルの構造が下記のようになっていたとき

(0,0)	(0,1)	(0,3)
	(1,1)	(1,2)

赤いセルでソートを実行したい場合は sort="1,2"と設定します。

sort アトリビュートは header タグ内でのみ有効になります。



### ● style アトリビュート

style アトリビュートはセルの style を設定することができます。

style を設定することでアトリビュートに存在しないスタイルを設定することができます。

style アトリビュートは影響が非常に大きい為、使用は推奨されません。ご使用になる場合は影響がないよう注意してご使用ください。

## data タグ

data タグは td タグ内で使用します。data タグを使用するとテーブルデータのデータを順番に出力することができます。data タグを使用しているセルは編集はできません。

### ● property アトリビュート

property アトリビュートはセルに表示させたいテーブルデータクラスのメソッドにアクセスする為の値を設定します。

例えばセルに表示させたいデータが下記のメソッドで取得できる場合は

```
public String[] getName();
```

property="name"と設定します。

### ● value アトリビュート

value アトリビュートはセルに表示させる文字列を設定します。設定した文字列は以下の場合に出力されます。

- property アトリビュートが設定されていないとき
- property アトリビュートで設定された値でデータが取得できないとき
- property アトリビュートが表示されている行数より少なく、セルに表示させるデータが足りなくなるとき

### ● hidden アトリビュート

hidden アトリビュートは data タグで表示されるデータを hidden 項目としてデータを持つかどうかを設定します。hidden 項目で持つとサーバーにデータを送信することができます。

true           : hidden でデータを持ちます。

false          : 持ちません（デフォルト）

hidden 項目の name は name アトリビュートで設定した値、name アトリビュートの設定がない場合は property で設定した値になります。name アトリビュートおよび property アトリビュートの両方が設定されない場合は hidden 項目は出力されません。

### ● name アトリビュート

name アトリビュートは hidden=true とした場合に hidden 項目の name を設定します。name を設定しない場合は property アトリビュートで設定した値が name として設定されます。



## text タグ

text タグは td タグ内で使用します。text タグを使用することで編集可能なセルを設定することができます。

### ● property アトリビュート

property アトリビュートはセルに表示させたいテーブルデータクラスのメソッドにアクセスする為の値を設定します。

例えばセルに表示させたいデータが下記のメソッドで取得できる場合は

```
public String[] getName();
```

property="name"と設定します。

### ● type アトリビュート

text タグの type アトリビュートは下記の設定を行うことができます。

text：編集可能なセルを設定します。入力した文字は確認することができます。

password：入力した文字が「\*」で伏せられる編集可能なセルを設定します。

### ● name アトリビュート

サーバにデータをサブミットするときに使用する name を設定します。name="score"と設定した場合はサーバ側で下記のようにデータを取得することができます。

```
String[] score = request.getParameterValues ("score");
```

name アトリビュートを設定していない場合は property アトリビュートで設定した値がそのまま name として設定されます。

### ● restriction アトリビュート

restriction アトリビュートは入力制限を行うことができます。下記の設定を行うことができます。入力にエラーがある場合はセルの編集を確定しようとしたときにエラー表示されます。

number：数字のみ入力できます。

money：数字のみか format アトリビュートで設定したフォーマットに合っている入力を行うことができます。セルの編集を確定する際、設定したフォーマットに合わされます。

halfsize：半角英字のみ入力できます。

date：数字のみか format アトリビュートで設定したフォーマットに合っている入力をおこなうことができます。存在しない日付を入力した場合はエラーとなります。セルの編集を確定する際、設定したフォーマットに合わされます。

regex：正規表現を使用することができます。正規表現の設定は format アトリビュートに行います。



### ● numericvalue アトリビュート

numericvalue アトリビュートはデータを数値で持つためのアトリビュートです。restriction アトリビュートが money の場合のみ有効となります。また、表示データ配列を数値で持つことによりフォーマットされたデータを表示することができます。

true       : データを数値で持ちます。

false       : 文字列でデータを持ちます。(デフォルト)

#### • <ffc:text restriction="money" numericvalue="false">の場合

表示       : ￥ 999,999

編集時     : ￥ 999,999

送信データ: ￥ 999,999

となります。

#### • <ffc:text restriction="money" numericvalue="true">の場合

表示       : ￥ 999,999

編集時     : 999999

送信データ: 999999

となります。

### ● imemode アトリビュート

imemode アトリビュートはセルが開いたときの IME 状態を設定します。下記の設定を行うことができます。

auto: 設定は自動になります。IME が ON なら ON で OFF なら OFF でセルが開きます。

active: 初期値が日本語入力モードになります。

inactive: 初期値が英数字入力モードになります。

disabled: 英数字入力モードになります。

### ● maxvalue アトリビュート

restriction アトリビュートで number か money を設定した場合に使用することができます。最大値を整数で設定することができます。maxvalue アトリビュートで設定した値は入力可能となります。

### ● minvalue アトリビュート

restriction アトリビュートで number か money を設定した場合に使用することができます。最小値を整数で設定することができます。minvalue アトリビュートで設定した値は入力可能となります。

● **maxlength** アトリビュート

入力可能な文字の最大文字数を設定します。maxlength アトリビュートで設定した文字数まで入力可能です。超えた入力をした場合はエラーとなります。

● **minlength** アトリビュート

入力される最小文字数を設定します。minlength アトリビュートで設定した文字数に足りないときはエラーとなります。

● **value** アトリビュート

value アトリビュートはセルに表示させる文字列を設定します。設定した文字列は以下の場合に出力されます。

- property アトリビュートが設定されていないとき
- property アトリビュートで設定された値でデータが取得できないとき
- property アトリビュートが表示されている行数より少なく、セルに表示させるデータが足りなくなったとき

● **format** アトリビュート

restriction アトリビュートで "money"、"date"、"regex" を設定した場合に有効になります。

• **money**

money では “#” を使用して金額型のフォーマットを設定することができます。

フォーマットの “.” 以下は # の数が小数点以下桁数を設定できます。

フォーマットに初めの # と終わりの # の間に 1 つの “.” 以外を設定することはできません。

以下に設定例を示します。

フォーマット	入力文字	判定
#.# #	123,456.45	OK
	¥ 123,456.45	NG
	123,456.456	123,456.45 にフォーマットされる
	123,456	123,456.00 にフォーマットされる
¥ #.#	123,456.4	¥ 123,456.4 にフォーマットされる
	¥ 123,456.4	OK
	\$ 123,456.4	NG
#.# 円	123,456.4 円	OK
	123456	123,456.0 円にフォーマットされる
	¥ 123,456.0	NG

## • date

date では日付型のフォーマットを設定することができます。

date で使用できるメタ文字は以下の通りです。

メタ文字	説明
yyyy	4桁の年（補完あり）
yy	2桁の年（補完あり）
mm	0補完する月
m	0補完しない月
dd	0補完する月
d	0補完しない月

※「0補完する」とは「1月」の場合「01月」のように0を加えて表示することです。

※和暦には対応していません。

※年の yyyy や yy を同じフォーマットで同時に使用することはできません。月・日に関しても同様です。

[フォーマット設定例]

yyyy/mm/dd

dd/mm/yy

西暦 yyyy 年 mm 月 dd 日

## • regex

regex は正規表現を使用して入力可能な文字パターンを設定することができます。

※オプションの設定はできません

[設定例]

パターン	説明
^[A-Z]*[A-Z]\$	大文字英字のみ入力可能
[0-9]{3}-[0-9]{4}\$	333-4444 のような郵便番号の記述のみ入力可能
^[A-Z]{2}[0-9]{4}\$	初めの2文字は大文字英字、次の4文字が数字のみ入力可能

### ● onchange アトリビュート

テキストが変更されたときにイベントを発生させるかどうか設定します。

true : イベントを発生させる

false : イベントを発生させない (デフォルト)

onchange="true"とした時はさらにイベントを登録する必要があります。

```
<script language="javascript">
function doChangeEvent(row, col, value, type){
    document.all.onchange_row.value = row;
    document.all.onchange_col.value = col;
    document.all.onchange_value.value = value;
    document.all.onchange_type.value = type;
}
//window.onloadのあとにdoloadを呼び出す
window.attachEvent('onload',doload);
function doload(){
    //イベントを登録する
    var faceobj = document.all.jp_co_ffc_gkit_spread2_div_0.face;
    faceobj.onChangeEvent = doChangeEvent;
}
</script>
```

テキストを変更したときには登録した  
イベント doChangeEvent が  
呼び出される。

## button タグ

button タグは td タグ内で使用します。button タグを使用するとセル内にボタンを配置することができます。

- property アトリビュート

property アトリビュートはボタンに表示させたいテーブルデータクラスのメソッドにアクセスする為の値を設定します。

例えばボタンに表示させたいデータが下記のメソッドで取得できる場合は

```
public String[] getName();
```

property="name"と設定します。

- type アトリビュート

ボタンのタイプを設定します。下記の設定を行うことができます。

button : 通常のボタンを配置します。

submit : サブミットボタンを配置します。

- name アトリビュート

サーバにデータをサブミットするときに使用する name を設定します。name="score"と設定した場合はサーバ側で下記のようにデータを取得することができます。

```
String[] score = request.getParameterValues ("score");
```

nameアトリビュートを設定していない場合はpropertyアトリビュートで設定した値がそのまま name として設定されます。

- onclick アトリビュート

ボタンを押下した時に発生するイベントを記述します。Javascript のメソッドを呼び出すことができます。

設定は下記のように行います。

onclick="alert();"

- value アトリビュート

value アトリビュートはボタンに表示させる文字列を設定します。設定した文字列は以下の場合に出力されます。

- property アトリビュートが設定されていないとき
- property アトリビュートで設定された値でデータが取得できないとき
- property アトリビュートが表示されている行数より少なく、ボタンに表示させるデータが足りなくなったとき

## check タグ

check タグは td タグ内で使用します。check タグはセルにチェックボックスを配置します。

- property アトリビュート

property アトリビュートはチェックボックスの横に表示させたいテーブルデータクラスのメソッドにアクセスする為の値を設定します。

例えばチェックボックスの横に表示させたいデータが下記のメソッドで取得できる場合は

```
public String[] getName();
```

property="name"と設定します。

- type アトリビュート

type アトリビュートには下記の値を設定できます。

checkbox：セルにチェックボックスを配置します。

radio：セルにラジオボタンを配置します。

- name アトリビュート

サーバにデータをサブミットするときに使用する name を設定します。name="score"と設定した場合はサーバ側で下記のようにデータを取得することができます。

```
String[] score = request.getParameterValues ("score");
```

name アトリビュートを設定していない場合は property アトリビュートで設定した値がそのまま name として設定されます。

- value アトリビュート

value アトリビュートはチェックボックスに表示させる文字列を設定します。設定した文字列は以下の場合に出力されます。

- property アトリビュートが設定されていないとき
- property アトリビュートで設定された値でデータが取得できないとき
- property アトリビュートが表示されている行数より少なく、ボタンに表示させるデータが足りなくなったとき

- checked アトリビュート

checked アトリビュートはチェックボックスの初期状態を設定します。チェックボックスの初期状態を boolean 配列で用意し、テーブルデータクラスのメソッドにアクセスする為の値を設定します。

配列は以下のように用意します。

```
boolean[] check = {true, false, true};
```

チェックボックスの初期状態の取得メソッドが以下のように設定されている場合

```
public boolean[] getCheck( ){ }
```

checked アトリビュートには「check」と設定します。

- disabled アトリビュート

disabled アトリビュートはチェックボックス使用不可にすることができるようになります。設定された check タグで定義されたチェックボックスすべてに対して設定を行います。

true       : 使用不可になります。  
false      : 使用可能です。(デフォルト)

- optionvalues アトリビュート

optionvalues アトリビュートは各レコードのチェックボックスのデータを設定することができます。optionvalues アトリビュートを使用することで表示されている値とは異なる値をサーバで取得することができるようになります。

テーブルデータクラスのメソッドを指定します。

**[テーブルデータクラス]**

```
private String[] values = {"0", "1", "2"};  
public String[] getValues(){  
    return values;  
}
```

**[JSP]**

```
<check optionvalues="values">
```

**[出力]**

1 行目     : <input type="checkbox" value="0">  
2 行目     : <input type="checkbox" value="1">  
3 行目     : <input type="checkbox" value="2">

**select タグ**

select タグは td タグ内で使用します。select タグを使用することでセルの中にセレクトを配置することができます。

- property アトリビュート

property アトリビュートはセレクトに表示させたいテーブルデータクラスのメソッドにアクセスする為の値を設定します。

例えばセレクトに表示させたいデータが下記のメソッドで取得できる場合は

```
public String[] getName();  
property="name"と設定します。
```

各リストは；（セミコロン）で区切ります。

- value アトリビュート

value アトリビュートはセレクトに表示させる文字列を設定します。設定した文字列は以下の場合に出力されます。

- property アトリビュートが設定されていないとき
- property アトリビュートで設定された値でデータが取得できないとき
- property アトリビュートが表示されている行数より少なく、ボタンに表示させるデータが足りなくなったとき

セレクトで使用する文字列は；（セミコロン）で区切ってリストを作成します。

例えば下記のようなリストを表示させたい場合は value="A型;B型;AB型"と設定します。

A型
B型
AB型

- name アトリビュート

サーバにデータをサブミットするときに使用する name を設定します。name="score"と設定した場合はサーバ側で下記のようにデータを取得することができます。

```
String[] score = request.getParameterValues ("score");
```

name アトリビュートを設定していない場合は property アトリビュートで設定した値がそのまま name として設定されます。



- onchange アトリビュート

select が変更されたときにイベントを発生させるかどうか設定します。

true : イベントを発生させる

false : イベントを発生させない (デフォルト)

onchange="true"とした時はさらにイベントを登録する必要があります。

```
<script language="javascript">
function doChangeEvent(row, col, value, type){
    document.all.onchange_row.value = row;
    document.all.onchange_col.value = col;
    document.all.onchange_value.value = value;
    document.all.onchange_type.value = type;
}
//window.onloadのあとにdoloadを呼び出す
window.attachEvent('onload',doload);
function doload(){
    //イベントを登録する
    var faceobj = document.all.jp_co_ffc_gkit_spread2_div_0.face;
    faceobj.onChangeEvent = doChangeEvent;
}
</script>
```

セレクトを変更したときには登録したイベント doChangeEvent が呼び出される。

- optionvalue アトリビュート

optionvalue アトリビュートはセレクトのオプションの value を;(セミコロン)区切りで設定します。

optionvalue = “0;1;2”と設定した場合、出力は以下のようになります。

```
<select name=" " style='width:100%;' >
    <option value='0'>A</option>
    <option value='1'>B</option>
    <option value='2'>C</option>
</select>
```

- optionvalues アトリビュート

optionvalues アトリビュートはテーブルデータクラスからセレクトのオプションの value を取得するメソッドを設定します。オプションの value は;(セミコロン)で区切ります。

[テーブルデータクラス]

```
String[] option = {"0;1;2", "A;B;C"};
public String[] getOption(){
    return option;
}
```

#### [select タグ]

```
<ff:select optionvalues = "option"/>
```

#### [出力内容]

- 1 レコード目

```
<select name=" style='width:100%;' >
  <option value='0'>A</option>
  <option value='1'>B</option>
  <option value='2'>C</option>
</select>
```

- 2 レコード目

```
<select name=" style='width:100%;' >
  <option value='A'>A</option>
  <option value='B'>B</option>
  <option value='C'>C</option>
</select>
```

#### ● selected アトリビュート

selected アトリビュートは初期表示で表示されるオプションの番号配列をデータクラスより取得するための設定を行います。データクラスには表示させるオプションを 0 からの int 値配列で用意します

#### [テーブルデータクラス]

```
int[] selecteoption = { 2, 3, 1, 0 };
public int[] getSelectoption(){
    return selecteoption;
}
```

#### [select タグ]

```
<select selected="selectoption" value="A型;B型;AB型;O型"/>
```

#### [初期表示]

- 1 行目：「AB型」が表示されます。
- 2 行目：「O型」が表示されます。
- 3 行目：「B型」が表示されます。
- 4 行目：「A型」が表示されます。

## link タグ

link タグは td タグ内で使用します。link タグを使用するとセル内にハイパーリンクを配置することができます。

### ● value アトリビュート

value アトリビュートはハイパーリンクに表示させる文字列を設定します。設定した文字列は以下の場合に出力されます。

- property アトリビュートが設定されていないとき
- property アトリビュートで設定された値でデータが取得できないとき
- property アトリビュートが表示されている行数より少なく、ハイパーリンクに表示させるデータが足りなくなったとき

### ● property アトリビュート

property アトリビュートはハイパーリンクに表示させたいテーブルデータクラスのメソッドにアクセスする為の値を設定します。

例えばボタンに表示させたいデータが下記のメソッドで取得できる場合は

```
public String[] getLink();
```

property="link"と設定します。

### ● hrefvalue アトリビュート

hrefvalue アトリビュートはリンク先の URL を文字列で設定します。設定した文字列は以下の場合に出力されます。

- hrefvalues アトリビュートが設定されていないとき
- hrefvalues アトリビュートで設定された値でデータが取得できないとき
- hrefvalues アトリビュートが表示されている行数より少なく、URL に使用するデータが足りなくなったとき

### ● hrefvalues アトリビュート

hrefvalues アトリビュートはリンク先の URL の文字列に使用するテーブルデータクラスのメソッドにアクセスする為の値を設定します。

例えばボタンに表示させたいデータが下記のメソッドで取得できる場合は

```
public String[] getHrefvalues ();
```

hrefvalues="hrefvalues"と設定します。

● target アトリビュート

リンク先のページを読み込む、表示先のウィンドウ(フレーム)を設定します。

設定は下記のように行います。

例えば、新たにウィンドウを立ち上げて表示する場合は、

`target="_blank"`

と設定します。

`_blank` : 新たにウィンドウを立ち上げて表示します。

`_top` : 最大化してのウィンドウに表示します。

`_self` : 同じウィンドウ(フレーム)に表示します。

`_parent` : 親フレームに表示します。

任意の名称 : 任意のウィンドウ(フレームに)表示します。

## 記述方法

タグの記述は HTML の TABLE タグの記述と似ていますが、header の中には上側部分、view の中には下側部分の構造を記述していきます。下記のようなテーブルを表示する場合は



※灰色の部分は上側部分（ヘッダ）となります。

下記のような記述を行います。

`<ffc:table>`

`<ffc:header>`……………header タグの中は上側部分（ヘッダ）の構造を記述する。

`<ffc:col/>`……………col は header の中に記述します。

`<ffc:col/>`

`<ffc:col/>`

`<ffc:tr>`……………tr は行毎の区切りを表します。行毎の設定も行います。

`<ffc:td colspan="2"></ffc:td>`…………td に個々のセルに関する設定を行います。

`<ffc:td></ffc:td>`

`</ffc:tr>`

`</ffc:header>`

`<ffc:view>`……………view タグは下側部分（データ）の構造を記述する。

`<ffc:record>`……………record はレコードの区切りを表します。record の中に複数行記述することで1レコードに対して複数行を設定することができます。

`<tr>`

`<td></td>`

`<td></td>`

`<td></td>`

`</tr>`

`</ffc:record>`

`</ffc:view>`

`</ffc:table>`

## プロパティファイルの設定

text タグでは、下表の関連アトリビュートを指定した場合、入力時にフォーマット違反、あるいは設定した文字数、値を超えた際にエラーメッセージを表示します。

このとき、「spreadv2msgkey.properties」ファイル（以下、プロパティファイル）に各エラー時のメッセージを指定することで任意のエラーメッセージを表示させることができます。

### 配置場所

「spreadv2msgkey.properties」は、クラスパス配下の下記のフォルダパスに配置してください。

### フォルダパス

[クラスパス] /jp/co/ffc/gkit/spread2/property

### 設定項目

設定可能な 10 種のメッセージと、text タグの関連アトリビュートを下表に記載します。

キー名	発生条件	関連アトリビュート
jp_co_ffc_gkit_spread2_format_money	金額型フォーマットに違反した場合	restriction アトリビュート • money format アトリビュート • money
jp_co_ffc_gkit_spread2_format_date	日付型フォーマットに違反した場合	restriction アトリビュート • date format アトリビュート • money
jp_co_ffc_gkit_spread2_format_regex	正規表現に違反した場合	restriction アトリビュート • regex format アトリビュート • regex
jp_co_ffc_gkit_spread2_format_minlength	設定した文字数よりも短い文字列を入力した場合	minlength アトリビュート
jp_co_ffc_gkit_spread2_format_maxlength	設定した文字数よりも長い文字列を入力した場合	maxlength アトリビュート

jp_co_ffc_gkit_spread2_format_minvalue	設定した値よりも小さい値を入力した場合	minvalue アトリビュート
jp_co_ffc_gkit_spread2_format_maxvalue	設定した値よりも大きい値を入力した場合	maxvalue アトリビュート
jp_co_ffc_gkit_spread2_format_date_error	存在しない日付を入力した場合	restriction アトリビュート • date
jp_co_ffc_gkit_spread2_format_number	半角数字以外を入力した場合	restriction アトリビュート • number
jp_co_ffc_gkit_spread2_format_halfsize	半角英数時以外を入力した場合	restriction アトリビュート • halfsize

## 記述方法

「spreadv2msgkey.properties」ファイルに、「キー名=メッセージ」を設定することで、任意のエラーメッセージを表示させることができます。

下記例のように 1 メッセージは 1 行で設定し、次のキー名（メッセージ）を設定する場合は、必ず改行してください。

### 「spreadv2msgkey.properties」ファイル設定例

```
jp_co_ffc_gkit_spread2_format_money =フォーマット[{0}]に合っていません
jp_co_ffc_gkit_spread2_format_date=日付型フォーマット[{0}]に合っていません
jp_co_ffc_gkit_spread2_format_number =数字のみ入力してください
jp_co_ffc_gkit_spread2_format_halfsize=半角数値以外の文字列が含まれています
...(略)
```

上記のように、パラメータ（※1）として、{0}を指定することで、format アトリビュートで指定したフォーマット、あるいは、境界値となる文字数、値を表示させることができます。

1 パラメータは、1 つのメッセージに対して、1 つのみ設定できます。2 つ以上指定した場合はフォーマットエラーとなります。

（フォーマットが存在しない “date\_error\_format”、“number\_format”、“halfsize\_format”は、パラメータを指定することはできません。）

また、下記条件に該当する場合、デフォルトメッセージを表示します。

- 指定したメッセージのフォーマットが正しくない場合（フォーマットエラー時）
- キー名に対し、メッセージが指定されていない場合（メッセージに空文字を指定した場合）
- プロパティファイルが存在しない、あるいはファイルが破損していて読み込めない場合

キー名	デフォルトメッセージ
jp_co_ffc_gkit_spread2_format_money	フォーマット[{0}]に合っていません
jp_co_ffc_gkit_spread2_format_date	日付型フォーマット[{0}]に合っていません
jp_co_ffc_gkit_spread2_format_regex	設定した正規表現[{0}]に合っていません
jp_co_ffc_gkit_spread2_format_minlength	{0}文字以上入力してください
jp_co_ffc_gkit_spread2_format_maxlength	文字数が{0}文字以上入力されています
jp_co_ffc_gkit_spread2_format_minvalue	{0}以上の数値を入力してください
jp_co_ffc_gkit_spread2_format_maxvalue	{0}より大きい数値が入力されています
jp_co_ffc_gkit_spread2_format_date_error	存在する日付を入力してください
jp_co_ffc_gkit_spread2_format_number	数字のみ入力してください
jp_co_ffc_gkit_spread2_format_halfsize	半角アルファベットまたは半角数値以外の文字列が含まれています

#### 制限事項

メッセージの指定には下記制限があります。

- メッセージの途中で改行することはできません。
- メッセージとして半角の「<（小なり）」、「>（大なり）」の文字列を使用することはできません。（指定したい場合は、全角の「<（小なり）」、「>（大なり）」をご利用ください。）
- プロパティファイルの編集はプロパティエディタをご利用ください。
- 新たにキー名を追加することはできません。
- 表毎にエラーメッセージを定義することはできません。（1 プロジェクトで設定可能なメッセージは 1 種類のみになります。）



## メソッド

SpreadsheetV2 は Javascript で操作できるメソッドを用意しています。ここでは SpreadsheetV2 で使用可能なメソッドについて説明いたします。

### メソッドの使い方

メソッドを使用するためにはそれぞれの SpreadsheetV2 のメソッドのオブジェクトを取得する必要があります。下記のコード例のようにメソッドオブジェクトを取得してください。

```
//メソッドオブジェクトの取得
var faceobj = document.all.jp_co_ffc_gkit_spread2_div_0.face;
//メソッドの実行
faceobj.cursorChange(rownum, colnum);
```

画面上の一つ目の SpreadsheetV2 は「jp\_co\_ffc\_gkit\_spread2\_div\_0.face」でメソッドオブジェクトを取得することができます。2つ目は「jp\_co\_ffc\_gkit\_spread2\_div\_1.face」となります。3、4、5、6..... つ目のテーブルは数字部分がそれぞれ 2、3、4、5..... となっていくます。

メソッドオブジェクトには SpreadsheetV2 を操作できるメソッドが用意されています。メソッドオブジェクトに続いて . (ピリオド) を記述し、メソッド使用してください。

### cursorChange メソッド

cursorChange メソッドはカーソル位置の設定を行うことができます。行・列番号を指定して実行します。マウスクリック時と同じような動作をします。selectmode により動作が異なります。

```
cursorChange(row, col)
    row      : 移動先の行番号を渡します。
    col      : 移動先の列番号を渡します。
```

### setSelectRow メソッド

setSelectRow メソッドは選択行を設定することができます。選択する行番号を指定して実行します。レコード単位での選択となります。行ヘッダをクリックした時と同じような動作をします。selectmode により動作が異なります。

```
setSelectRow(row)
    row      : 選択行の行番号を設定します。
```

### setText メソッド

setText メソッドはテキスト形式のセルの内容を設定することができます。変更する行番号、列番号、文字列を指定して実行します。

setText(row, col, str)

row           : 行番号を設定します。  
col           : 列番号を設定します。  
str           : 変更する文字列を設定します。

### getText メソッド

getText メソッドは該当セルの内容を文字列で返します。行番号、列番号を指定して実行します。

getText(row, col)

row           : 行番号を設定します。  
col           : 列番号を設定します。

### getCellType メソッド

getCellType メソッドはセルのタイプを返します。行番号、列番号を指定して実行します。

getCellType(row, col)

row           : 行番号を設定します。  
col           : 列番号を設定します。

### isOpened メソッド

isOpened メソッドは現在セルが編集中かどうか返します。

isOpened()

[戻り値]

true           : 編集中です。  
false           : 編集中ではありません。

### setSelectOption メソッド

setSelectOption メソッドはセレクトのオプションを変更するメソッドです。行番号、列番号、オプション内容を指定して実行します。

setSelectOption(row, col, str)

row : 行番号を指定します。

col : 列番号を指定します。

str : オプションの内容を指定します。各オプションは ”;”(セミコロン)で区切られます。

#### clear メソッド

clear メソッドはテーブルの選択状態を解除するメソッドです。

clear()

#### close メソッド

close メソッドは編集集中のセルを閉じるメソッドです。

close()

[戻り値]

true : 正常にセルを閉じました。

false : 入力値にエラーがある為、セルを閉じられませんでした。

#### getRow メソッド

getRow メソッドはカーソルの行番号を取得するメソッドです。

getRow()

[戻り値]

行番号

#### getCol メソッド

getCol メソッドはカーソルの列番号を取得するメソッドです。

getCol()

[戻り値]

列番号

## focus メソッド

focus メソッドはテーブルヘフォーカスを当てるメソッドです。

focus()

## onSortASC イベント

昇順ソートを行う際に値を比較するために呼び出されます。戻り値によりソート順番を入れ替えることができます。

onSortASC( a, b, row, col )

a           : 比較対象の1つ目の文字列  
b           : 比較対象の2つ目の文字列  
row         : ソート対象の行番号  
col         : ソート対象の列番号

[戻り値]

a > b の場合       : 正の値を返す  
a = b の場合       : 0を返す  
a < b の場合       : 負の値を返す

- 数字として昇順ソートさせる

```
<script language="javascript">
window.attachEvent('onload',doload);
function doload(){
    //昇順ソートの登録
    document.all.jp_co_ffc_gkit_spread2_div_0.face.onSortASC = doSortASC;
}
function doSortASC(a, b, row, col){
    var ai = parseInt(a);
    var bi = parseInt(b);
    if( isNaN(ai) || isNaN(bi) ){
        return 0
    }
    return (ai - bi);
}
</script>
```

昇順ソートを行ったときに  
doSortASC が呼び出されるように  
登録する

## onSortDESC イベント

降順ソートを行う際に値を比較するために呼び出されます。戻り値によりソート順番を入れ替えることができます。

onSortDESC( a, b, row, col )

a           : 比較対象の1つ目の文字列  
b           : 比較対象の2つ目の文字列  
row         : ソート対象の行番号  
col         : ソート対象の列番号

[戻り値]

a > b の場合       : 負の値を返す  
a = b の場合       : 0を返す  
a < b の場合       : 正の値を返す

- 数字として降順ソートさせる

```
<script language="javascript">
window.attachEvent('onload',doLoad);
function doLoad(){
    //降順ソートの登録
    document.all.jp_co_ffc_gkit_spread2_div_0.face.onSortDESC = doSortDESC;
}
function doSortDESC(a, b, row, col){
    var ai = parseInt(a);
    var bi = parseInt(b);
    if( isNaN(ai) || isNaN(bi) ){
        return 0
    }
    return (bi - ai);
}
</script>
```

降順ソートを行ったときに  
doSortDESC が呼び出されるように  
登録する

## onChangeEvent イベント

onChangeEventはonchangeアトリビュートをtrueにした場合にオブジェクトの値が変更されたときに発生するイベントの登録を行います。

onChangeEvent(row, col, value, type)

row : 変更が発生したセルの行番号  
col : 変更が発生したセルの列番号  
value : 変更後の値  
type : セルのタイプ

```
<script language="javascript">
function doChangeEvent(row, col, value, type){
    // イベント発火時の処理を記述する
}
// window.onloadのあとにdoloadを呼び出す
window.attachEvent('onload',doload);
function doload(){
    // イベントを登録する
    document.all.jp_co_ffc_gkit_spread2_div_0.face.onChangeEvent = doChangeEvent;
}
</script>
```

変更したときには登録したイベント doChangeEvent が呼び出される。

The diagram consists of a rectangular box containing Japanese text. Two arrows originate from this box. The first arrow points to the doChangeEvent function name in the first line of the code block. The second arrow points to the doChangeEvent variable in the assignment statement within the doload function.

## onClickEvent イベント

onClickEvent はセルをクリックした際に発生するイベントです。イベントの引数として取得できる値は行・列番号です。下記の場合にはイベントは発生しません。

[イベントが発生しない条件]

- ヘッダ（行ヘッダ・列ヘッダ）をクリックした場合
- table#selectmode = “3”（選択なしモード）の場合
- 編集集中のセルにエラーがある場合

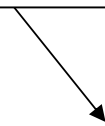
onClickEvent(row, col)

row           : クリックされたセルの行番号  
col           : クリックされたセルの列番号

```
<script language="javascript">
function doClickEvent(row, col){
    //イベント発火時の処理を記述する
}
//window.onloadのあとにdoLoadを呼び出す
window.attachEvent('onload',doLoad);
function doLoad(){
    //イベントを登録する
    document.all.jp_co_ffc_gkit_spread2_div_0.face.onClickEvent = doClickEvent;
}

```

onClickEvent が発生する時に  
doClickEvent が呼ばれるようにする



## onTabExitEvent イベント

onTabExitEvent イベントは SpreadsheetV2 内から右下セルで Tab キー、または、左上セルで SHIFT+Tab キーを押下した際に発生するイベントです。

onTabExitEvent(shift)

shift : 0 Tab キーが押下、1 SHIFT+Tab キーが押下


[戻り値]

SpreadsheetV2 独自のフォーカス処理を継続させる場合は true を返します。

```
<script language="javascript">
function doTabExitEvent(shift){
    //イベント発火時の処理を記述する
}
//window.onloadのあとにdoloadを呼び出す
window.attachEvent('onload',doload);
function doload(){
    //イベントを登録する
    document.all.jp_co_ffc_gkit_spread2_div_0.face.onTabExitEvent = doTabExitEvent;
}

```

イベントの登録を行う。





## テーブルデータクラスの作成方法

テーブルデータクラスの作成について説明します。

テーブルデータクラスを作成するには各セルのデータは配列で持つようにしてください。また配列データを取得するメソッドを public で持つ必要があります。下記に簡単な例を示します。

```
class TableData{
    //配列でデータを持つ
    private String[] name;
    //データ配列を返す get メソッドを用意する
    public String[] getName(){
        return name;
    }
}
```

このようにクラスを作成した場合、name のデータ配列をセルに表示させるには property アトリビュートに”name”と設定する必要があります。

get メソッドの命名規則は下記に従います。

- get + 大文字 + 小文字

property の設定値は大文字を小文字にしたものを設定してください。

メソッドが「getName」なら設定値は「name」

メソッドが「getAddress」なら設定値は「address」

メソッドが「get 名前」なら設定値は「名前」

## 機能説明

### セル編集

text タグを使用したセルは編集することができます。編集する為には text のセルをダブルクリックまたはキーボードで文字を入力することで編集可能状態に変わります。編集から抜ける場合は ENTER キーを押下またはマウスで他のセルをクリックすることで編集を抜けることができます。セルを閉じるまではセルの値は確定されません。

### コピー&ペースト

「CTRL + C」でコピー。「CTRL + V」でペースト。「CTRL + X」でカットを行うことができます。ただし、セルの文字列を変更できるのは text のセルのみとなります。

text : セルに表示されている文字列のコピー、ペースト、カットが可能です。ただし、入力がエラーとなる場合はペーストすることはできません。また、password の場合はコピーできません。

button : ボタンに表示されている文字列をコピーすることができます。ペーストすることはできません。

check : セルに表示されている文字列をコピーすることができます。ペーストすることはできません。

select : 表示されているリストの文字列をコピーすることができます。ペーストを行うことはできません。

data : セルに表示されている文字列をコピーすることができます。ペーストを行うことはできません。

### 内容の削除

セルの型が text の場合は DELETE キーで内容を削除することができます。その他の型では DELETE キーでセルの文字列を削除することはできません。

### セル選択

セルを選択するにはマウสดラッグと「SHIFT + マウスクリック」、「SHIFT + 矢印」で行う方法があります。セルの選択範囲は四角形に拡張されます。選択範囲を解除するには SHIFT を押さずに移動、SHIFT を押さずにセルクリックで解除することができます。

### スクロール

スクロールはマウスのホイールボタンおよびセルの移動などで行うことができます。ヘッダ部のみの場合はスクロールすることはできません。

## 行列のサイズ変更

マウスをヘッダ部のセルとセルの境目に持っていき、マウスクリックしてそのままドラッグすると行の高さ、列の幅を変更することができます。ただし、固定列、ヘッダ列はサイズ変更することはできません。

## ソート

td タグに sort アトリビュートを設定したセルは右クリックすることでソートを行うことができます。ソートメニューは昇順、降順、初期状態に戻すがあります。ソートはすべて文字列ソートで行われます。

## チュートリアル

SpreadsheetV2 を使用した Web アプリケーションの作成手順を説明いたします。

ここで作成するサンプルは下記のようなテーブル構造をもったものです。

番号	名前	連絡先		年齢	職種	血液型	選択	ボタン
		住所	電子メール					
ヘッダ	固定列	text	text	text	text	select	check	button

※灰色部分は上側ヘッダ部分になります。

このテーブルにデータ用のクラスを作成します。DataClass というクラスで下記のようなフィールドを用意します。

```
/** 番号を格納 */
private String[] num;
/** 名前を格納 */
private String[] name;
/** 住所を格納 */
private String[] address;
/** 電子メールアドレスを格納 */
private String[] email;
/** 年齢を格納 */
private int[] age;
/** 職種を格納 */
private String[] employment;
/** 血液型を格納 */
private String[] blood;
```

これらのフィールドには下記のような値が格納されます。

```
num = { "A138328", "A138383" }
name = { "鈴木 浩二", "柳田 敏郎" }
address = { "千葉県千葉市", "北海道函館市" }
email = { "suzuki@gkit.domain", "yanagida@gkit.domain" }
age = { 48, 73 }
employment = { "製造業", "無職" }
blood = { "A型;B型;AB型;O型", "A型;B型;AB型;O型" }
```

それぞれのフィールドに対応した get メソッドを用意します。

```
public String[] getNum() { return num; }
public String[] getName() { return name; }
```

```

public String[] getAddress() { return address; }
public String[] getEmail() { return email; }
public int[] getAge() { return age; }
public String[] getEmployment() { return employment; }
public String[] getBlood() { return blood; }

```

サブレッドでは DataClass クラスを sample01 という名前でリクエストに格納し、表示する JSP に飛ばします。

```

/* データを格納するクラスのインスタンスを生成する */
DataClass data = new DataClass();

/* 属性を設定してデータを格納する */
request.setAttribute("sample01", data);

/* 要求をリソースへ転送するためのクラスを取得する */
RequestDispatcher rd = request.getRequestDispatcher("/jsp/Sample01.jsp");

/* 要求を別のリソース(Sample01.jsp)に転送する */
rd.forward(request, response);

```

JSP では下記のようなタグを記述します。objectid にはリクエストにセットした名前を設定しそれぞれの property には取得したいデータのメソッドにアクセスできる値を設定します。

```

<ffc:table objectid="sample01" fixedcol="1">
  <ffc:header>
    <ffc:col width="100"/>
    <ffc:col width="100"/>
    <ffc:col width="200"/>
    <ffc:col width="200"/>
    <ffc:col width="50"/>
    <ffc:col width="150"/>
    <ffc:col width="100"/>
    <ffc:col width="100"/>
    <ffc:col width="100"/>
  <ffc:tr>
    <ffc:td rowspan="2">番号</ffc:td>
    <ffc:td rowspan="2">名前</ffc:td>
    <ffc:td colspan="2">連絡先</ffc:td>
    <ffc:td rowspan="2">年齢</ffc:td>

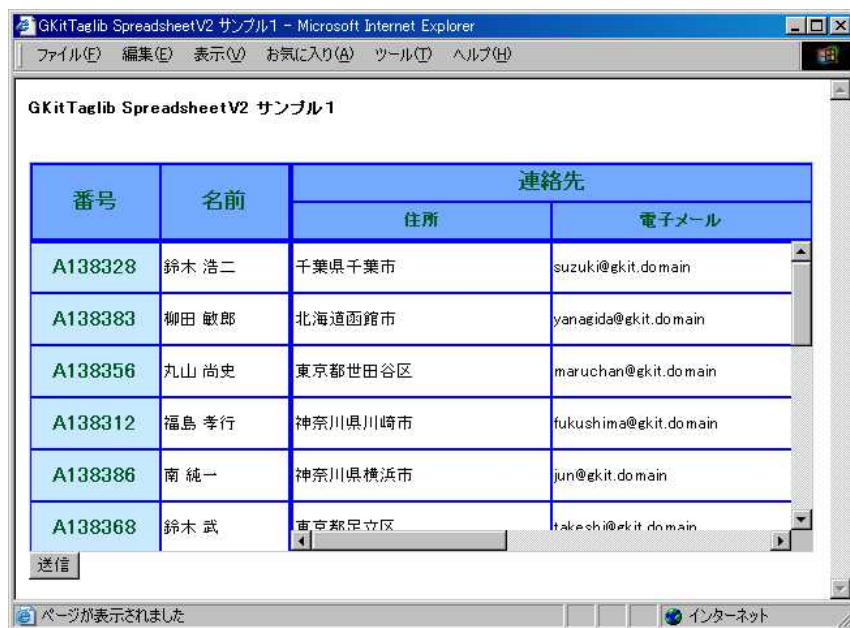
```

```

        <ffc:td rowspan="2">職種</ffc:td>
        <ffc:td rowspan="2">血液型</ffc:td>
        <ffc:td rowspan="2">選択</ffc:td>
        <ffc:td rowspan="2">ボタン</ffc:td>
    </ffc:tr>
    <ffc:tr>
        <ffc:td>住所</ffc:td>
        <ffc:td>電子メール</ffc:td>
    </ffc:tr>
</ffc:header>
<ffc:view>
    <ffc:record>
        <ffc:tr>
            <ffc:td><ffc:data property="num"/></ffc:td>
            <ffc:td><ffc:text property="name"/></ffc:td>
            <ffc:td><ffc:text property="address"/></ffc:td>
            <ffc:td><ffc:text property="email"/></ffc:td>
            <ffc:td><ffc:text property="age"/></ffc:td>
            <ffc:td><ffc:text property="employment"/></ffc:td>
            <ffc:td><ffc:select property="blood"/></ffc:td>
            <ffc:td><ffc:check property="num"/></ffc:td>
            <ffc:td><ffc:button property="num"/></ffc:td>
        </ffc:tr>
    </ffc:record>
</ffc:view>
</ffc:table>

```

以上の設定で動作させると下記の表示を行うことができます。



※チュートリアル中のサンプルコードは一部省略されています。  
さらに詳しい作り方は製品についているサンプルを参考にしてください。

## HOW TO

行ごとの色を交互に設定したい

下記のようなサンプルのように行ごとに交互に色設定を行うサンプルを示します。



record タグを2つ用意しそれぞれ別な設定をすることで交互に色を変えることができます。

```
<ffc:table>
```

```
<ffc:header>
```

```

</ffc:tr>
    <ffc:td>No</ffc:td>
    <ffc:td>従業員番号</ffc:td>
    <ffc:td>名前</ffc:td>
    <ffc:td>電話番号</ffc:td>
    <ffc:td>郵便番号</ffc:td>
    <ffc:td>住所</ffc:td>
</ffc:tr>
</ffc:header>
<ffc:view>
    <ffc:record>
        <ffc:tr>
            <ffc:td><ffc:text property="string02"/></ffc:td>
            <ffc:td><ffc:text property="string02"/></ffc:td>
            <ffc:td><ffc:data property="num01"/></ffc:td>
            <ffc:td><ffc:data/></ffc:td>
            <ffc:td><ffc:data/></ffc:td>
            <ffc:td><ffc:data/></ffc:td>
        </ffc:tr>
    </ffc:record>
    <ffc:record>
        <ffc:tr bgcolor="#FFFF00">
            <ffc:td><ffc:text property="string02"/></ffc:td>
            <ffc:td><ffc:text property="string02"/></ffc:td>
            <ffc:td><ffc:data property="num01"/></ffc:td>
            <ffc:td><ffc:data/></ffc:td>
            <ffc:td><ffc:data/></ffc:td>
            <ffc:td><ffc:data/></ffc:td>
        </ffc:tr>
    </ffc:record>
</ffc:view>
</ffc:table>

```



#### サブミットされたデータを取得したい

サブミットされたデータを取得するときは

```
String[] values = request.getParameterValues([name の値]);
```

で取得することができます。

取得した配列はソートにより表示直後の順番に並んでいない可能性があります。

リクエストから”jp\_co\_ffc\_gkit\_spread2\_text\_0”の配列を取得することで表示直後の行番号を取得することができます。

```
String[] gyoubango = request.getParameterValues(“jp_co_ffc_gkit_spread2_text_0”);
```

gyoubango 配列の 0 番目の値が3だった場合、values の 0 番目の値は表示直後3行目であったことを示します。

jp\_co\_ffc\_gkit\_spread2\_text\_0 の最後の数字は1つ目のテーブルは 0 となり2つ目のテーブルでは 1 となります。テーブルが複数ある場合は該当のテーブルの番号で表示直後の行番号を取得してください。

#### 行選択されていた行の番号を取得したい

ヘッダ列をクリックするとレコードを選択することができます。「CTRL + ヘッダ列クリック」を行うと行選択の追加が行われ、「SHIFT + ヘッダ列クリック」を行うとクリックした場所までの行の範囲選択が行われます。そのままサブミットすると jp\_co\_ffc\_gkit\_spread2\_check\_0 の名前で選択行番号を取得することができます。jp\_co\_ffc\_gkit\_spread2\_check\_0 で取得できる番号はサブミット直前の行番号になります。jp\_co\_ffc\_gkit\_spread2\_text\_0 とあわせて使用することで選択された行の表示直後の行番号を取得することも可能です。

jp\_co\_ffc\_gkit\_spread2\_check\_0 の最後の数字は1つ目のテーブルは 0 となり2つ目のテーブルでは 1 となります。

#### 表示されるデータ件数について

表示されるデータ件数は view の length とテーブルデータから取り出す配列のサイズによって変わります。

length が設定されているときは length で設定した行数分、行が表示されます。取得した配列が length より短い場合は配列を超えた分は value の値、value が設定されていないときはセルのデータは空として行が作られます。

objectid を設定していない、length を設定していないときは 20 行表示されます。

## ライセンス

GKitTaglib SpreadsheetV2 は CPU ライセンスとなっております。運用されるサーバの CPU 1 個に対し 1 つのライセンスが必要になります。開発ライセンスはフリーとなっております。

## サポートについて

製品の問い合わせは基本的に無償となっております。サポートのやり取りはホームページ上のお問い合わせフォームにてやり取りをさせていただいております。ご質問内容や状況によっては早急に回答できないことがあります。予めご了承ください。

お問い合わせフォーム：<https://www-s.fujitsu.com/jp/group/fae/contact/gkit.html>

お問合せの際は以下の内容にてお問合せいただきますようお願いいたします。

- ・ **タイトル**：お問い合わせの内容が判断できるような簡潔なものとしてください。
- ・ **連絡先**：お客様の会社名、氏名、メールアドレス、電話番号、FAX 番号
- ・ **製品名**：お問い合わせの製品名を記載してください。
- ・ **開発環境**：OS、AS、JDKの製品名およびバージョン
- ・ **運用環境**：OS、AS、JDKの製品名およびバージョン
- ・ **内容**：お問い合わせの内容を詳細に記載してください。

障害と思われる現象についてはどのような操作をしたときに発生するのか、また発生頻度（常に発生するのか、特定の条件でのみ発生するのかなど）も必ず記載してください。

可能であれば、スタンドアロンでの現象が確認できる再現プログラムをお送りください。再現確認の時間が短縮されることによって、より早い解決に繋がります。

## 制限事項

record を複数記述する場合レコードの構造はすべての record で同じにする必要があります。セルの型や構造を同じにしてください。

エクセルからのペーストでダブルクォーテーションまたは改行が入ったセルをコピーし SpreadsheetV2 にペースト処理を行うとたたくペーストが行われないことがあります。

text タグを使用している場合、同じフォーム内にリセットボタンを使用しないようにしてください。

text タグを使用している場合、ブラウザの戻るボタンで SpreadsheetV2 を使用している画面に移動した場合、表示内容と実際のデータの内容が異なることがあります。

印刷には対応していません。

IE5.5sp2 で CTRL + A で全選択を行った場合、画面全体が選択状態になります。

TAB キーで SpreadsheetV2 から他のオブジェクトに移動できないことがあります。

上側ヘッダをクリックして選択されるのはクリックしたセルの一番左の列となります。

bgcolor、type、restriction、format、imemode には全角の"＋（プラス）"、"｜（パイプ）"、"、（カンマ）"を使用することはできません。

バージョン2はバージョン1とは互換性はありません。

閉じたセルに対し、日本語入力が ON の状態で入力を行った場合、1 文字目は半角英数字の入力となります。

## 付録

### タグ一覧

SpreadsheetV2 のタグとアトリビュートの一覧です。

table タグ	アトリビュート	説明	設定値	デフォルト
	border	罫線の太さ	数値	1
	bordercolor	罫線の色	文字列	#000000
	bgcolor	背景色	文字列	#CCCCCC
	align	テーブルの表示位置	文字列	なし
	cellpadding	セルの隙間のピクセル値	数値	1
	height	テーブル全体の高さ	数値	400
	width	テーブル全体の幅	数値	600
	objectid	オブジェクト名	文字列	なし
	objectscope	オブジェクトスコープ request: リクエストからデータ取得 session: セッションからデータ取得	文字列	request
	headercol	ヘッダ列数	数値	1
	selectcolor	選択色	文字列	#AFEEEE
	cursorcolor	キャレットセル色	文字列	#E6E6FA
	fixedcol	固定列数	数値	0
	editcolor	編集時の背景色	文字列	#CCCCCC
	textname	サブミットされる初期表示時のレコード番号を取得する為のキー	文字列	jp_co_ffc_gkit_spread2_text_0
	checkname	サブミット直前に選択されていたレコード番号を取得する為のキー	文字列	jp_co_ffc_gkit_spread2_check_0
	selectmode	選択モード 0: 通常モード 1: 複数行選択モード 2: 単数行選択モード 3: 選択なしモード	数値	0
	cursorhidden	カーソル非表示 true: 非表示 false: 表示	boolean	false
	changevh	マウスによる列幅・行高さ変更 true: 可 false: 不可	boolean	true

pluralselect	SHIFT+クリックによる行範囲選択 true: 可 false: 不可	boolean	true
contextmenu	マウス右クリックでコピー / 貼り付けのメニューを表示 true: 表示 false: 非表示	boolean	false
editclearmode	キー入力 (F2、INSERT キーは除く) によりセルが編集状態に遷移した場合に、セル内の文字列を削除 true: セル内の文字列を削除する false: セル内の文字列を削除しない	boolean	false
caretcontrol	F2、INSERT キー押下によりセルが編集状態に遷移した場合にキャレット位置を移動 true: 右端に移動 false: 左端に移動	boolean	false

header タグ	アトリビュート	説明	設定値	デフォルト
		上部ヘッダ部分の指定する		

view タグ	アトリビュート	説明	設定値	デフォルト
	start	データ配列の開始番号	数値	0
	length	表示する行の長さ -1 の時はすべてのデータが出力されます。	数値	-1

record タグ	アトリビュート	説明	設定値	デフォルト
		1 レコードの区切りとして使用する		

col タグ	アトリビュート	説明	設定値	デフォルト
	width	列幅	数値	100

tr タグ	アトリビュート	説明	設定値	デフォルト
	height	行の高さ	数値	25
	bgcolor	背景色	文字列	なし
	align	横方向の位置	文字列	なし
	valign	縦方向の位置	文字列	なし
	style	スタイル	文字列	なし

fontcolor	文字色	文字列	なし
fontsize	フォントサイズ	数値	なし
fontname	フォント名	文字列	なし

td タグ	アトリビュート	説明	設定値	デフォルト
	bgcolor	背景色	文字列	なし
	align	横方向の位置	文字列	なし
	valign	縦方向の位置	文字列	なし
	colspan	横方向のセル結合	数値	なし
	rowspan	縦方向のセル結合	数値	なし
	style	スタイル	文字列	なし
	fontcolor	文字色	文字列	なし
	fontsize	フォントサイズ	数値	なし
	fontname	フォント名	文字列	なし
	sort	ソート項目設定 "行番号,列番号"で設定する。 例) "2,3"	文字列	なし

data タグ	アトリビュート	説明	設定値	デフォルト
	property	データ項目	文字列	なし
	value	出力文字列	文字列	なし
	hidden	hidden 項目の出力設定	boolean	false
	name	hidden 項目の name 設定	文字列	なし

text タグ	アトリビュート	説明	設定値	デフォルト
	property	データ項目	文字列	なし
	type	タイプ設定 text, password	文字列	text
	name	name 設定	文字列	property で設定した値

restriction	入力制限 number: 数値入力 money: 金額入力 halfsize: 半角入力 date: 日付入力 regex: 正規表現入力	文字列	なし
numericvalue	データを数値でもつ	boolean	false
format	フォーマットの設定	文字列	money: ¥# date: yyyy/mm/dd
imemode	IME モード auto: active: 初期値が日本語入力モード inactive: 初期値が英数字入力モード disabled: 英数字入力モード	文字列	なし
maxvalue	最大値	数値	なし
minvalue	最小値	数値	なし
maxlength	最大文字数	数値	なし
minlength	最小文字数	数値	なし
value	出力文字列	文字列	なし
onchange	テキストに変更があったときにイベントを発生させるかどうか	boolean	false

button タグ	アトリビュート	説明	設定値	デフォルト
	property	データ項目	文字列	なし
	type	タイプ設定 button、submit	文字列	button
	name	name 設定	文字列	property で設定した値
	onclick	onclick 時の処理	文字列	なし
	value	出力文字列	文字列	なし

check タグ	アトリビュート	説明	設定値	デフォルト
	property	データ項目	文字列	なし
	type	タイプ設定 checkbox、radio	文字列	checkbox
	name	name 設定	文字列	property で設定した値と同じ

value	出力文字列	文字列	なし
checked	チェックボックスの初期状態の設定	文字列	なし
disabled	チェックボックスの使用不可設定	boolean	false
optionvalues	チェックボックスの value をデータクラスから取得する設定	文字列	なし

select タグ	アトリビュート	説明	設定値	デフォルト
	property	データ項目	文字列	なし
	value	出力文字列 セミコロン(;)で区切ってリストを設定 例) A 型;B 型;AB 型;O 型	文字列	なし
	name	name 設定	文字列	property で設定した値と同じ
	onchange	セレクトに変更があったときにイベントを発生させるかどうか	boolean	false
	optionvalue	オプションの value をセミコロン(;)で区切って設定	文字列	なし
	optionvalues	オプションの value をデータクラスが取得する設定	文字列	なし
	selected	初期表示のオプションを取得する	文字列	なし

link タグ	アトリビュート	説明	設定値	デフォルト
	value	表示文字列	文字列	なし
	property	表示文字列のデータ	文字列	なし
	hrefvalue	リンク先の URL	文字列	なし
	Hrefvalues	リンク先の URL のデータ	文字列	なし
	target	ページを読み込むウィンドウ(フレーム)	文字列	なし



## ショートカットキー

SpreadsheetV2 で使用されるショートカットキーです。

キー	内容
CTRL + C	セルの内容をクリップボードにコピー
CTRL + V	クリップボードの内容をセルにペースト
CTRL + X	セルの内容をクリップボードにコピーしてセルの内容を削除
CTRL + A	セルを全選択
SHIFT + 矢印	選択範囲の拡大
SHIFT + ENTER	上に移動
矢印	セルの移動
END	最終行のセルに移動
HOME	1行目のセルに移動
ENTER	下に移動 セルの内容の確定
F2	セルを開く
INSERT	セルを開く
DELETE	セルの内容を削除
TAB	次のセルに移動
SPACE	スペースの挿入 ボタンクリックイベント発火 チェックボックス・ラジオボタンのチェック

