

入力フォーム

1. 入力フォーム機能の概要	4
1-1. 入力フォーム機能の概要	4
1-2. Create!Form Screen が提供する主な機能	4
入力フォーム機能	4
アクション機能	4
1-3. Create!Form Cast での入力フォームの表示	4
2. 入力フォームの作成	6
2-1. 入力フォームを作る	6
入力フォームオブジェクトの定義方法	6
入力フォームオブジェクト属性ダイアログの構成	6
2-2. 入力フォームの外観、表示状態を設定する	7
フォントを設定する	7
背景色を設定する	8
境界線を設定する	8
入力フォームの表示 / 非表示を設定する	9
入力フォームを読取専用にする	9
入力フォームへの入力（選択）を必須にする	9
入力フォームプロパティ設定のエラーチェック	9
2-3. テキスト入力を行う入力フォームを作る - テキストボックス	10
初期値を指定する	10
プレースホルダーを設定する	10
テキストの配置を指定する	10
入力可能な最大文字数を指定する	11
複数行のテキスト入力を可能にする	11
テキストボックス内でスクロールを可能にする	11
パスワード入力を行うテキストボックスを作る	11
マス目に区切られたテキストボックスを作る (Cast のみ)	12
出力時にテキストボックスに出力するデータが空白の場合、初期値を出力する	12
2-4. 単一項目が選択可能なメニューを作る - コンボボックス	12
項目を追加（編集・削除）する	13
外部の CSV ファイルから項目をインポートする	13
初期選択項目を指定する	14
登録した項目を並べ替える	14
2-5. メニューリストを作る - リストボックス	15
複数の項目を選択可能にする	15
項目を追加（編集・削除）する	15
外部の CSV ファイルから項目をインポートする	15
初期選択項目を指定する	15
登録した項目を並べ替える	15
2-6. チェックボックスを作る - チェックボックス	15
チェックマークのスタイルを指定する	16

チェック時の値を指定する	16
非チェック時の値を指定する	16
デフォルトで選択状態にする	17
ラベルテキストを表示する	17
2-7. ラジオボタンを作る - ラジオボタン	17
チェックマークのスタイルを指定する	18
チェック時の値を指定する	18
デフォルトで選択状態にする	18
ラベルテキストを表示する	18
2-8. ボタンを作る - ボタン	18
ボタンに表示するテキストを指定する	18
2-9. 送信ボタンを作る - サブミットボタン	19
ボタンに表示するテキストを指定する	19
フォームデータを処理するサーバスクリプトの URL を指定する	19
2-10. クリアボタンを作る - リセットボタン	20
ボタンに表示するテキストを指定する	20
3. 入力フォームに特殊機能をつける	21
3-1. 入力したテキストの表示形式を指定する - フォーマット	21
入力したテキストを数値表記する	21
入力したテキストをパーセント表記する	23
入力したテキストを日付表記する	23
サブミット時の送信値	26
内部値の取得・設定	27
3-2. 他の入力フォーム間の値の計算結果を表示させる - 計算	28
入力フォームを指定して計算に使用する	28
任意の値を計算に使用する	29
3-3. 入力した（選択した）入力フォームの値を検証する - 検証	30
テキストボックスの値の範囲を数値で指定する	31
3-4. アクションを組み込む	31
入力フォームにアクションを組み込む	32
ページにアクションを組み込む	32
アクションを組み込む - JavaScript の実行	33
3-5. 入力チェック	33
入力チェックの種類	33
入力エラー発生時の挙動	34
入力チェックが発生するタイミング	34
4. 外部からのデータを反映させる	35
4-1. 初期データを外部データから指定する	35
テキストボックスに表示するテキストを外部データから指定する	35
選択状態にするメニュー項目を外部データから指定する	35
チェックマークを入れる項目を外部データから指定する	35
フォームデータの送信先 URL を外部データから指定する	35
4-2. 入力フォームオブジェクトにデータマッピングを行う	35
マッピング方法	36
マッピングデータ	36

4-3. データマッピングを行わない場合に反映される入力フォームの値	36
5. 値の同期	37
5-1. 同名オブジェクトの値の同期	37
5-2. ページインデックス	37
5-3. 複数ページ間で値が同期されるとき挙動	37
5-3-1. ページインデックスを付与しない場合の送信時の name 値	38
5-4. 複数ページ間で値を同期させたくない場合	38
5-4-1. ページインデックスを付与する場合の送信時の name 値	38
5-5. フォーム識別子	39
5-6. 各種オブジェクト毎の同期の挙動	39
5-6-1. 同期対象のオブジェクト	39
5-6-2. 各種オブジェクトの同期のルール	40
6. Tips	41
6-1. 出力時に入力フォームの外観、状態を切り替える - FormSwitch オプション	41
出力パターン別に入力フォームの外観設定を行う	41
出力パターン別にファイルの出力を行う	42
6-2. メニュー項目を外部データから取り込む	43
6-3. 指定したサーバスクリプトにフォームデータを送信する	44
送信されるフォームデータ	44
データ送信形式とその処理方法の例	44
送信時の文字コード	45
6-4. ツールチップを表示させる	45
6-5. Tab キーで移動する順序を指定する	46
選択した入力フォームのタブ順序を先頭に指定する	47
選択した入力フォームのタブ順序を最後尾に指定する	47
選択した入力フォームのタブ順序を1つ上げる	47
選択した入力フォームのタブ順序を1つ下げる	47
選択した入力フォームのタブ順序を任意の位置に指定する	47
選択した入力フォームをタブオーダーの対象外にする	47
タブオーダー対象外入力フォームを再度タブオーダーの対象にする	48
複数ページ時のタブオーダー	48
6-6. JavaScript を使用する	48
JavaScript をコーディングする	48
6-7. JavaScript を外部エディタを用いて編集する	49
任意の外部エディタを登録する	49
外部エディタを起動する	50
6-8. 外部の JavaScript / CSS を適用する	50
外部ファイルの指定方法	50
ファイル指定で追加できるファイル	51
6-9. セレクタを用いて特定のオブジェクトにアクセスする	51
CSS クラス名によるアクセス	51
name 属性によるアクセス	52

1. 入力フォーム機能の概要

1-1. 入力フォーム機能の概要

入力フォームはHTML ファイル上に設けられた入力フォームに対してテキスト入力や選択といったユーザとHTML ファイル間の対話を可能にするインタラクティブ（対話）機能です。Create!Form Screen では、入力フォーム機能を実装するためのオブジェクト（以下入力フォームオブジェクトと呼びます）を含むHTML ファイルの出力を行うことができます。入力フォーム機能はGoogle Chrome 等のブラウザ上で動作します。

1-2. Create!Form Screen が提供する主な機能

入力フォーム機能

Create!Form Screen では8種類の入力フォームオブジェクトをサポートしています。

テキストボックス	ボタン
コンボボックス	サブミットボタン
リストボックス	リセットボタン
チェックボックス	
ラジオボタン	

これらのオブジェクトの機能を用いて以下のような機能を含んだHTML ファイルを出力することができます。

- 入力フォームに対するテキスト入力
- 入力フォーム項目の選択
- サーバスクリプトへのフォームデータ送信
- 入力フォーム間の計算
- 入力したテキストの表示形式の変換
- 入力フォームデータの数値範囲の検証
- 入力フォーム操作によるアクションの実行

アクション機能

Create!Form Screen では、入力フォームオブジェクトを用いたインタラクティブなHTML 機能のほかに、JavaScript を組み込んでブラウザ上で任意の動作を行わせるアクション機能を組み込むことができます。

アクション機能の詳細については、「3-4. アクションを組み込む」をご覧ください。

1-3. Create!Form Cast での入力フォームの表示

入力フォームオブジェクトはCreate!Form Cast ランタイムでも出力できます。入力フォームオブジェクトをCast ランタイムで出力した場合、以下のような出力になります。

- ・PDF 出力時の初期データはScreen ランタイムで出力される初期データと原則同様となります。
- ・PDF 上で入力を行うことはできません。
- ・JavaScript やアクションは実行されません。
- ・フォントはScreen のフォントに設定された代替フォントが使用されます。

また、以下の機能は Cast でのみ有効になる機能です。これらの機能を使用した場合は Cast と Screen で異なる表示となります。

- ・ フォントサイズ自動
- ・ マス目区切り

2. 入力フォームの作成

2-1. 入力フォームを作る

入力フォームオブジェクトの定義方法

入力フォームオブジェクトは Create!Form Design 付属の Form エディタ上で定義します。Form エディタのオブジェクトツールバーから各入力フォームオブジェクトを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで定義されます。

図：オブジェクトツールバー



入力フォームオブジェクト属性ダイアログの構成

定義されたオブジェクトをダブルクリックすることで入力フォームオブジェクト属性ダイアログが表示されます。

図：入力フォームオブジェクト属性ダイアログ

入力フォームオブジェクト属性ダイアログは次のようなタブから構成されています。

[表示] タブ

入力フォームオブジェクトの外観を設定します。

[オプション] タブ

各入力フォーム特有の設定を行います。

[アクション] タブ

入力フォームオブジェクトに対する操作に応じて動作するアクションを設定します。

[フォーマット] タブ

入力フォームオブジェクトに表示されるテキスト文字列の表示形式を設定します。

※このタブはテキストボックスのみに含まれます。

[検証] タブ

入力フォームオブジェクトの値の範囲等を検証します。

※このタブはテキストボックスのみに含まれます。

[計算] タブ

他の入力フォームオブジェクト間の値の計算方法を設定します。

※このタブはテキストボックスのみに含まれます。

[位置] タブ

入力フォームオブジェクトの定義位置を設定します。

[HTML] タブ

入力フォームオブジェクトの連携情報を設定します。

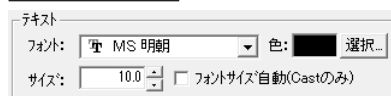
2-2. 入力フォームの外観、表示状態を設定する

各入力フォームオブジェクト属性ダイアログ上の [表示] タブでは、入力フォームの外観（見た目）に関する設定を行うことができます。

フォントを設定する

入力フォームオブジェクト属性ダイアログの [表示]-[テキスト] で入力フォームオブジェクト内で表示されるテキスト文字列のフォント設定を行うことができます。

図：テキスト設定

**[フォント]**

入力フォームオブジェクト上で表示されるテキスト文字列のフォントを指定します。

入力フォームオブジェクトで設定可能なフォント種別は Create!Form Screen で使用できるフォントと同様になります。Cast では Screen のフォントに設定された代替フォントが使用されます。

[サイズ]

入力フォームオブジェクト上で表示されるテキスト文字列のフォントサイズを指定します。フォントサイズには 2 ~ 300 までの値を指定することができます。

Create!Form Cast で出力された PDF では [フォントサイズ自動 (Cast のみ)] を指定すると、入力した文字列の長さによってフォントサイズが 4pt ~ 144pt の間で自動的に調整されるようになります。

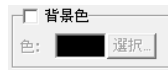
[色]

入力フォームオブジェクト上で表示されるテキスト文字列の色を設定します。

背景色を設定する

入力フォームオブジェクト属性ダイアログの [表示]-[背景色] で入力フォームオブジェクトの領域を塗りつぶす任意の背景色を指定することができます。

図：背景色設定



[背景色]

入力フォームオブジェクト領域内の背景色の有無を設定します。

[色]

入力フォームオブジェクト領域内を塗りつぶす色を設定します。

境界線を設定する

入力フォームオブジェクト属性ダイアログの [表示]-[境界線] で入力フォームオブジェクトの領域を囲む境界線の有無やスタイルを設定することができます。

図：境界線設定



[境界線]

入力フォームオブジェクトの領域を囲む境界線の描画の有無を設定します。

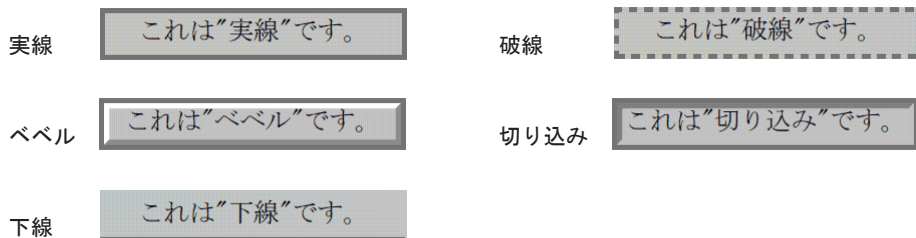
[スタイル]

入力フォームオブジェクトの領域を囲む境界線のスタイルを指定します。設定可能な境界線スタイルは、

実線 破線 ベベル 切り込み 下線

の5種類です。

図：境界線スタイルの出力例



[線幅]

入力フォームオブジェクトの領域を囲む境界線の線幅を " 細 "、" 標準 "、" 太 " から指定します。

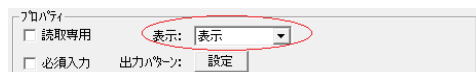
〔線色〕

入力フォームオブジェクトの領域を囲む境界線の色を指定します。

入力フォームの表示 / 非表示を設定する

入力フォームオブジェクト属性ダイアログの [表示]-[プロパティ]-[表示] では、出力された HTML ファイルをブラウザ上で表示した時の入力フォームオブジェクトの表示状態を指定することができます。

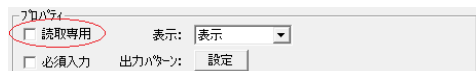
図： [表示と印刷] 設定



入力フォームを読取専用にする

入力フォームオブジェクト属性ダイアログの [表示]-[プロパティ]-[読取専用] を設定することで、ブラウザ上で入力フォームオブジェクトを操作することができなくなります。

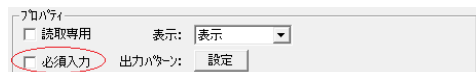
図：読取専用設定



入力フォームへの入力（選択）を必須にする

入力フォームオブジェクト属性ダイアログの [表示]-[プロパティ]-[必須入力] を設定することで、サブミットフォームで入力フォームデータをサーバスクリプトへ送信する際に、入力フォームの空白データ（データの有無）の検知が行われます。サブミットフォームについては、「6-3. 指定したサーバスクリプトにフォームデータを送信する」をご覧ください。

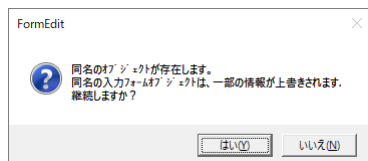
図：必須設定



入力フォームプロパティ設定のエラーチェック

同名のオブジェクトが存在した場合にメッセージボックスが表示されます。

図：メッセージボックス



メッセージボックスを表示させないためには、Form エディタのメニューから [ファイル]-[環境設定] を選択し、[オプション 2] タブの [入力フォームオブジェクト] で [同名のオブジェクト確認をする。] のチェックを外してください。

2-3. テキスト入力を行う入力フォームを作る - テキストボックス

テキストボックスは、ブラウザ上の HTML ファイルに対してテキスト入力を行うことができます。

定義方法

Form エディタのオブジェクトツールバーからテキストボックスを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで作成することができます。

このテキストボックスに対して文字数制限などの特有の設定を行うためには、入力フォームオブジェクト属性ダイアログ内の [オプション] タブで設定を行います。

図：テキストボックスオプション設定

各オプションの詳細については以下の項目を参照してください。

初期値を指定する

テキストボックスオプション設定の [初期値] では、リセットフォームが行われた場合にテキストボックス内に表示されるテキスト文字列を指定します。データマッピングについては、「4. 外部からのデータを反映させる」をご覧ください。

初期値には最大 9,999 文字の文字列を指定することができます。

また、[複数行を許可する] を指定すると、改行を含む複数行のテキストを初期値として指定することが可能です。

プレースホルダーを設定する

テキストボックスオプション設定の [プレースホルダー] では、未入力の時に表示する文字列を指定できます。入力内容の注意点やヒントを表示する場合に有効です。プレースホルダーには複数行のテキストを指定することはできません。

プレースホルダーには最大 128 文字の文字列を指定することができます。

図：プレースホルダー

メールアドレス

例: createform@example.com

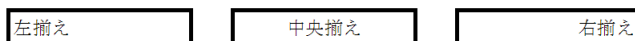
テキストの配置を指定する

ファイルの出力時、もしくはブラウザ上での入力時のテキストボックス領域内でのテキスト文字列の表示位置を指定します。

この設定を行うには、テキストボックスオプション設定の [配置] で右揃え、中央揃え、左揃えから選択します。初期状態では左寄せが指定されています。

なお [パスワード] が設定されている場合には、この設定を行うことはできません。この場合は左揃えが指定されます。

図：配置によるテキスト表示位置



入力可能な最大文字数を指定する

テキストボックス内に入力可能な最大文字数を指定します。

この設定を行うには、テキストボックスオプション設定で [最大文字数の指定] を選択して任意の最大文字数を指定します。この値はバイト数ではなく文字数となります。最大文字数には 0 ~ 10,240 文字を指定することができます。

複数行のテキスト入力を可能にする

テキストボックスに対して複数行のテキスト入力を可能にします。

この設定を行うには、テキストボックスオプション設定で [複数行を許可する] を選択します。

なお [パスワード]、[マス目で区切る] が設定されている場合には、この設定を行うことはできません。

テキストボックス内でスクロールを可能にする

テキストボックス内へ入力したテキストの長さが領域を越えてしまった場合に、スクロールバーを表示するようにします。

この設定を行うには、テキストボックスオプション設定で [スクロールバーを表示する] を選択します。

なお [マス目で区切る] が設定されている場合には、この設定を行うことはできません。また、[複数行を許可する] が設定されていない場合も、この設定を行うことはできません。

パスワード入力を行うテキストボックスを作る

テキストボックス内に入力した文字列を表記上マスクして表示させます。

この設定を行うには、テキストボックスオプション設定で [パスワード] を選択します。

図：[パスワード] 出力例



なお [複数行を許可する]、[マス目で区切る] のいずれかが設定されている場合には、この設定を行うことはできません。

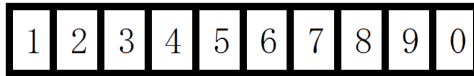
※このオプションは入力したテキストをブラウザ上ではマスク表示しますが、内部では入力された値がそのまま保持されます。

マス目に区切られたテキストボックスを作る (Cast のみ)

Create!Form Cast で出力した PDF 上で、出力されたテキストボックスの横領域を指定した文字数で分割し、マス目ごとに文字を表示することを可能にします。

この設定を行うには、テキストボックスオプション設定で [マス目で区切る (Cast のみ)] を選択し、[最大文字数の指定] で任意の文字数を指定します。この値はバイト数ではなく文字数となります。文字数は 0 ~ 999 文字を指定することができます。

図 : [マス目で区切る] 出力例 (Cast)



なお [複数行を許可する]、[スクロールバーを表示する]、[パスワード] が設定されている場合には、この設定を行うことはできません。

出力時にテキストボックスに出力するデータが空白の場合、初期値を出力する

出力時にデータマップによって取得した外部データが空白の場合やデータマッピングされていない場合に、初期値で指定したテキスト文字列を表示するかどうかを指定することができます。この設定を行うには、テキストボックスオプション設定で [空白時に初期値を使用 (出力時のみ)] を選択します。

※この設定は Create!Form Cast / Screen によってファイルを出力した初期状態でのみ有効です。出力したファイルをブラウザ上で編集した際には無効となりますのでご注意ください。

2-4. 単一項目が選択可能なメニューを作る - コンボボックス

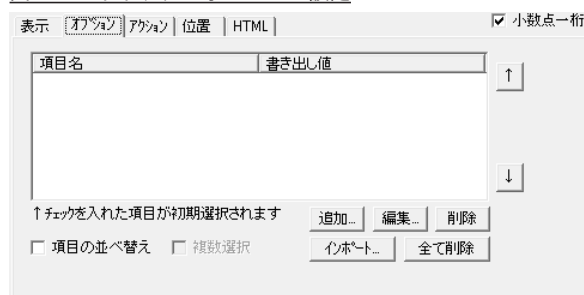
コンボボックスは HTML ファイル上に項目の一覧をポップアップ表示します。

定義方法

Form エディタのオブジェクトツールバーからコンボボックスを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで作成することができます。

このコンボボックスに対する特有の設定を行うためには、入力フォームオブジェクト属性ダイアログ内の [オプション] タブで設定を行います。

図 : コンボボックスオプション設定

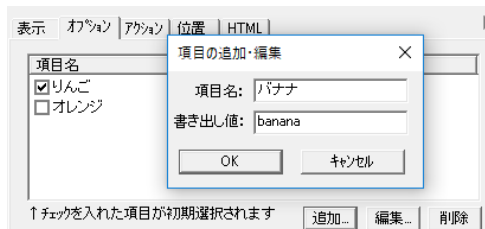


項目を追加（編集・削除）する

メニュー項目を追加する

メニューに表示する項目を登録するためには、[追加]ボタンを押下して表示される[項目の追加・編集]ダイアログを用いて行います。メニュー項目は256個まで登録することができます。

図：項目の追加・編集



[項目名]

実際にHTMLファイル上でメニューの選択肢として表示されるテキストです。

[書き出し値]

フォームデータの送信時に選択された項目の値として出力される値です。この値を設定しない場合は、項目名が代替値として出力されます。

メニュー項目を変更する

メニュー項目の一覧で選択されている項目を編集するためには、[編集]ボタンを押下して表示される項目の追加・編集ダイアログを用いて変更します。

メニュー項目を削除する

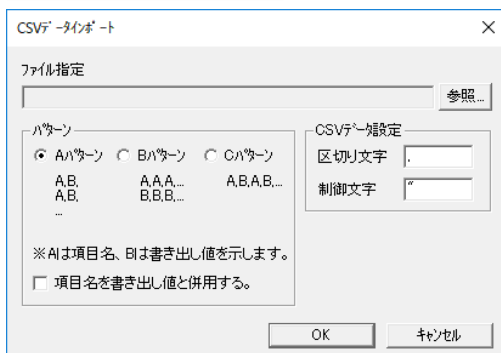
メニュー項目の一覧で選択されている項目を削除するためには、[削除]ボタンを押下します。また一覧の項目をすべて削除したい場合には、[全て削除]ボタンを押下します。

外部のCSVファイルから項目をインポートする

メニュー項目が多数存在する場合、一つ一つ手作業で登録する手間が発生します。この手間を省くために外部のCSVファイルにあらかじめメニュー項目を登録しておき、それをフォーム作成時にインポートすることで項目登録作業を簡略化することができます。

外部CSVファイルからメニュー項目をインポートするためには、[インポート]ボタンを押下して表示されるCSVデータインポートダイアログを使用します。

図：[CSVデータインポート]



[ファイル指定]

[参照] ボタンを押下してメニュー項目のインポートに使用する CSV ファイルを指定します。

[パターン]

メニュー項目のインポートに使用する CSV ファイルのデータパターンを選択します。

次の 3 パターンの CSV データからメニュー項目をインポートすることができます。

A パターン : 1 行に 1 項目の項目名と書き出し値を記述したデータ。

[例] name1, value1,
name2, value2,
name3, value3,
.....

B パターン : 項目名を 1 行目に、書き出し値を 2 行目にそれぞれ区切り文字で区切って繰り返したデータ。

[例] name1, name2, name3,
value1, value2, value3,

C パターン : 項目名と書き出し値を区切り文字ごとに繰り返したデータ。

[例] name1, value1, name2, value2, name3, value3,

※上記の例では区切り文字にカンマを使用しています。

また項目名のみを記述した CSV ファイルを使用する場合には、[項目名を書き出し値と併用する] を設定します。この場合、項目名のみがメニュー項目の一覧として登録されます。

[CSV データ設定]

CSV ファイル内で使用されている区切り文字と制御文字を任意に指定することができます。区切り文字、制御文字にはアスキーコード 0x21 ~ 0x3F 内の半角数字および半角記号を指定します。

初期の状態では区切り文字にカンマ (,)、制御文字にダブルクォーテーション (") が指定されています。

初期選択項目を指定する

登録項目の一覧上で項目名の左横にあるチェックボックスを選択することにより、このメニュー項目の初期選択項目を設定することができます。

リセットフォームが行われた場合やデータマッピングされていない場合には、ここで指定した項目が選択された状態になります。データマッピングについては、「4. 外部からのデータを反映させる」をご覧ください。

図 : 初期選択項目

項目名	書き出し値
<input checked="" type="checkbox"/> りんご	apple
<input type="checkbox"/> オレンジ	orange
<input type="checkbox"/> バナナ	banana

登録した項目を並べ替える

出力時に登録されたメニュー項目を数字、アルファベット順に並べ替えることができます。

この設定を行うには、コンボボックスオプション設定で [項目の並べ替え] を選択します。

※この設定は、出力時に外部データからメニュー項目をインポートする際にも有効となります。

2-5. メニューリストを作る - リストボックス

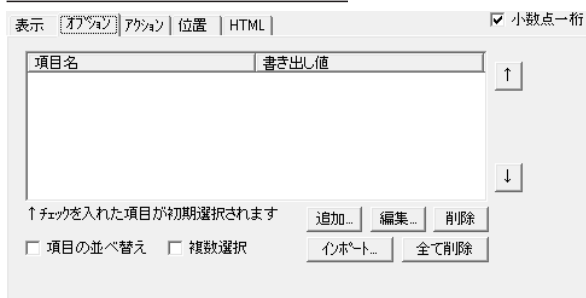
リストボックスはHTMLファイル上に項目の一覧を表示します。登録された項目が領域内に収まらない場合には、自動的にスクロールバーがつけられます。

定義方法

Formエディタのオブジェクトツールバーからリストボックスを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで定義されます。

このリストボックスに対する特有の設定を行うためには、入力フォームオブジェクト属性ダイアログ内の [オプション] タブで設定を行います。

図：リストボックスオプション設定



複数の項目を選択可能にする

リストボックスは通常単一の項目のみの選択となりますが、オプションを指定することにより複数の項目をブラウザ上で選択できるようになります。

この設定を行うには、リストボックスオプション設定で [複数選択] を選択します。

項目を追加（編集・削除）する

外部の CSV ファイルから項目をインポートする

初期選択項目を指定する

登録した項目を並べ替える

※前述の「2-4. 単一項目が選択可能なメニューを作る - コンボボックス」の項をご覧ください。

2-6. チェックボックスを作る - チェックボックス

チェックボックスは、該当する項目の可否を選択する形式のボタンです。

定義方法

Formエディタのオブジェクトツールバーからチェックボックスを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで作成することができます。

このチェックボックスに対する特有の設定を行うためには、入力フォームオブジェクト属性ダイアログ内の [オプション] タブで設定を行います。

図：チェックボックスオプション設定

チェックマークのスタイルを指定する

選択した際のチェックマークの表示形式を指定することができます。

この設定を行うには、チェックボックス [表示] タブの [チェックマーク種別] から選択します。チェックマーク種別は“HTML 標準”、“グラフィカル”から選択できます。

“HTML 標準”を選択した場合は、ブラウザ標準のチェックマークが表示されます。“グラフィカル”を選択した場合は種別を“チェック”、“ひし形”、“円形”、“四角形”、“十字形”、“星形”から選択できます。出力時の表示は以下のようになります。

図：[表示] タブ [チェックマーク種別]

図：“グラフィカル”チェックマーク種別の出力例



チェック時の値を指定する

フォームデータの送信の際に出力される入力フォームの実値データを設定することができます。

この設定を行うためには、チェックボックスオプション設定の [書き出し値]-[チェック時] に値を設定します。

非チェック時の値を指定する

チェックボックスでは、チェックがオフになっているときの送信データを設定することができます。

この設定を行うためには、チェックボックスオプション設定の [書き出し値]-[非チェック時] に値を設定します。空白が設定されている場合は値が送信されません。

また、[デフォルトでチェックする] が設定されている場合でも、非チェック時の値をデータから指定することでチェックをオフに変更することができます。データマッピングについては、「4. 外部からのデータを反映させる」をご覧ください。

デフォルトで選択状態にする

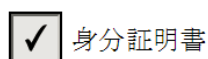
リセットフォームが行われた場合やデータマッピングされていない場合に選択状態にする初期設定を行うことができます。データマッピングについては、「4. 外部からのデータを反映させる」をご覧ください。

この設定を行うためには、チェックボックスオプション設定の [デフォルトでチェックする] を選択します。

ラベルテキストを表示する

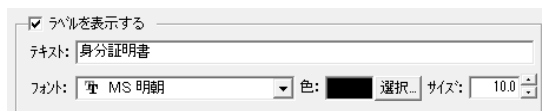
チェックマークの右側にラベルテキストを表示することができます。ラベルテキストはチェックマークと連動しており、ラベルテキストのクリックでもチェック状態を切り替えることができます。ラベルテキストには最大で 128 文字の文字列を設定することができます。

図：ラベルテキスト



ラベルテキストを設定するには、チェックボックスオプション設定の [ラベルを表示する] にチェックを入れ、内部の [テキスト] に表示する文字列を入力します。また、[フォント]、[色]、[サイズ] から表示するラベルテキストのフォント設定を変更できます。

図：オプション設定 [ラベルを表示する]



2-7. ラジオボタンを作る - ラジオボタン

ラジオボタンは複数の選択項目のうち 1 つだけ選択する形式のボタンです。

定義方法

Form エディタのオブジェクトツールバーからラジオボタンを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで作成することができます。

※共通の項目に対する選択肢としてグループ化するラジオボタンは、全て同じオブジェクト名、異なる書き出し値を設定する必要があります。

このラジオボタンに対する特有の設定を行うためには、入力フォームオブジェクト属性ダイアログ内の [オプション] タブで設定を行います。

図：ラジオボタンオプション設定

チェックマークのスタイルを指定する

チェック時の値を指定する

デフォルトで選択状態にする

ラベルテキストを表示する

※前述の「2-6. チェックボックスを作る - チェックボックス」の項をご覧ください。

2-8. ボタンを作る - ボタン

ボタンはアクションを動作させるトリガーの役割として利用することができます。

定義方法

Form エディタのオブジェクトツールバーからボタンを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで作成することができます。

このボタンに対する特有の設定を行うためには、入力フォームオブジェクト属性ダイアログ内の [オプション] タブで設定を行います。

図：ボタンオプション設定

ボタンに表示するテキストを指定する

[スタイル]

ボタンを押下した際のボタンの表示方法を指定することができます。

押下時のスタイルは、

なし：外観は変わりません。

プッシュ：押下時にボタンが押し込まれたような外観に変わります。

から選択することができます。

[ラベル設定]

ボタンに表示されるテキスト文字列を設定します。

2-9. 送信ボタンを作る - サブミットボタン

サブミットボタンは、Web サーバへフォームデータを送信する目的に特化したボタンです。データの送信については、「6-3. 指定したサーバスクリプトにフォームデータを送信する」をご覧ください。

定義方法

Form エディタのオブジェクトツールバーからサブミットボタンを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで作成することができます。

このサブミットボタンに対する特有の設定を行うためには、入力フォームオブジェクト属性ダイアログ内の [オプション] タブで設定を行います。

図：サブミットボタンオプション設定

表示	オプション	アクション	位置	HTML	<input checked="" type="checkbox"/> 小数点一桁
表示するラベル: <input type="text" value="OK"/>					
送信先URL: <input type="text" value="http://"/>					

ボタンに表示するテキストを指定する

サブミットボタンに表示されるテキスト文字列を設定することができます。

ラベル文字列を設定するには、サブミットボタンオプション設定の [表示するラベル] にテキスト文字列を指定します。この文字列はボタン上で中央揃えで表示されます。

ラベル文字列には最大 99 文字のテキスト文字列を指定できます。

フォームデータを処理するサーバスクリプトの URL を指定する

フォームデータ送信先のサーバスクリプトの URL を指定することができます。

サーバスクリプトでは送信されたフォームデータを処理する仕組みを持っている必要があります。

送信先 URL を設定するには、サブミットボタンオプション設定の [送信先 URL] に URL 文字列を指定します。

2-10. クリアボタンを作る - リセットボタン

リセットボタンは、ブラウザ上で入力、選択されたフォームデータを各入力フォームの初期値にリセットする機能に特化したボタンです。

定義方法

Form エディタのオブジェクトツールバーからリセットボタンを選択して、オブジェクト領域の開始点から終了点までマウスをドラッグすることで作成することができます。

このリセットボタンに対する特有の設定を行うためには、入力フォームオブジェクト属性ダイアログ内の [オプション] タブで設定を行います。

図：リセットボタンオプション設定

表示 オプション | アクション | 位置 | HTML | 小数点一桁

表示するラベル: CANCEL

送信先URL:

ボタンに表示するテキストを指定する

※「2-9. 送信ボタンを作る - サブミットボタン」をご覧ください。

3. 入力フォームに特殊機能をつける

3-1. 入力したテキストの表示形式を指定する - フォーマット

テキストボックスの属性ダイアログに属する [フォーマット] タブでは、テキストボックスに表示されるデータ形式を設定することができます。

フォーマットでは“数値”、“パーセント”、“日付”からデータ形式を指定することができます。いずれかのフォーマットを選択することにより以下の項目で説明する詳細なフォーマット設定を行うことができます。

入力したテキストを数値表記する

[フォーマット] タブの [分類] から“数値”を選択することでテキストボックス内に入力することができる値を数値のみに設定することができます。

また、[小数点以下の桁数]、[0埋めする] [有効桁以下の処理]、[通貨記号]、[桁区切の表記方法]、[負数の表記方法]といった設定を行うことにより入力したテキストに対して表示形式を指定することができます。

図：数値フォーマット設定

[小数点以下の桁数]

入力されたデータのテキストボックスでの小数点以下の表示桁数を指定します。設定可能な桁数は0～3桁までとなります。

図：小数点以下の桁数 - 「1234.5678」を入力した場合

桁数	出力例	桁数	出力例
0桁	1,235	2桁	1,234.57
1桁	1,234.6	3桁	1,234.568

[0埋めする]

[小数点以下の桁数] で設定した桁数まで0を表示します。

図：0埋めする - 「1234.5678」を入力した場合

桁数	出力例	
	チェックあり	チェックなし
0桁	123	123
1桁	123.0	123
2桁	123.00	123
3桁	123.000	123

[有効桁以下の処理]

[小数点以下の桁数] で設定した桁数以下の数値についての処理を設定します。「四捨五入」と「切捨て」から選択できます。

図：小数点以下の桁数 - 「1234.5678」を入力した場合

■有効桁以下の処理「四捨五入」の場合

桁数	入力値	出力例
0	123.4	123
	123.5	124
3	123.4444	123.444
	123.4445	123.445

■有効桁以下の処理「切捨て」の場合

桁数	入力値	出力例
0	123.4	123
	123.5	123
3	123.4444	123.444
	123.4445	123.444

[通貨記号]

入力されたデータに対して指定した通貨記号をつけた形式でテキストボックス上に表示します。

設定可能な通貨記号は、

円 (¥) ドル (\$) ポンド (£)
 フラン (F) ユーロ (EUR) マルク (DM)

となります。

図：通貨記号 - 「1234」を入力した場合

通貨記号	出力例	通貨記号	出力例
円 (¥)	¥1,234	フラン (F)	1,234 F
ドル (\$)	\$1,234	ユーロ (EUR)	EUR1,234
ポンド (£)	£1,234	マルク (DM)	1,234 DM

[桁区切の表記方法]

入力されたデータを指定の桁区切り方法を用いてテキストボックス上に表示します。

図：桁区切の表記方法 - 「1234.567(1234,567)」を入力した場合

種別	出力例
1,234.56	1,234.567
1234.56	1234.567

[負数の表記方法]

入力されたデータが負数の場合、指定した表記方法でテキストボックス上に表示します。

図：負数の表記方法 - 「-1234」を入力した場合

種別	出力例	種別	出力例
マイナス記号	-1,234	括弧囲み表記	(1,234)
朱記表記	1,234	カッコ囲み 朱記表記	(1,234)

入力したテキストをパーセント表記する

[フォーマット] タブの [分類] から “パーセント” を選択することでテキストボックス内に入力することができる値を数値のみに設定することができます。パーセントフォーマットを設定した場合、テキストボックス上の表示は “入力した数値データ × 100” の値にパーセントマーク (%) がついた形式となります。例えば、テキストボックスに “0.5” と入力した場合にはテキストボックス上では “50%” と表示されます。

図：パーセントフォーマット設定

[小数点以下の桁数]**[0 埋めする]****[有効桁以下の処理]****[桁区切の表記方法]**

※前述の『入力したテキストを数値表記する』をご覧ください。

入力したテキストを日付表記する

[フォーマット] タブの [分類] から “日付” を選択することでテキストボックス内に入力した日付データを指定した日付形式で表示することができます。

図：日付フォーマット設定

〔日付の表記方法〕

入力された日付を表すデータを指定した日付形式でテキストボックス上に表示します。
日付の表記方法は以下の 38 種類から選択できます。

yyyy 年 m 月 d 日	yyyy 年 mm 月 dd 日	yy 年 m 月 d 日
yy 年 mm 月 dd 日	yyyy/mm/dd	yyyy-mm-dd
yyyy. mm. dd	m/d	m/d/yy
m/d/yyyy	mm/dd/yy	mm/dd/yyyy
mm/yy	mm/yyyy	d-mmm
d-mmm-yy	d-mmm-yyyy	dd-mmm-yy
dd-mmm-yyyy	yy-mm-dd	mmm-yy
mmm-yyyy	mmmm-yy	mmmm-yyyy
mmm d, yyyy	mmmm d, yyyy	ge 年 m 月 d 日
gge 年 m 月 d 日	ggge 年 m 月 d 日	gee 年 mm 月 dd 日
ggee 年 mm 月 dd 日	gggee 年 mm 月 dd 日	ge. m. d
gge. m. d	ggge. m. d	gee. mm. dd
ggee. mm. dd	gggee. mm. dd	

※記号の意味

yyyy：西暦年（4桁）

ggg：元号（漢字2文字）

g：元号（アルファベット1文字）

ee：和暦年（数字2桁、0埋め有）

mmmm：月（英語）

mm：月（数字2桁、0埋め有）

dd：日（数字2桁、0埋め有）

yy：西暦年（下2桁）

gg：元号（漢字1文字）

e：和暦年（数字最大2桁、0埋め無）

mmm：月（英語略記）

d：日（数字最大2桁、0埋め無）

図：日付の表記方法 - 「2005/9/1」を表す場合

種別	入力例	出力例
yyyy年mm月dd日	2005/9/1	2005年09月01日
yyyy-mm-dd	2005/9/1	2005-09-01
mm/dd/yyyy	9/1/2005	09/01/2005
mmmm d, yyyy	9/1/2005	September 1, 2005

※元号表示で「令和」以降の元号を表示する場合、以下の場所に元号設定ファイル「era_

jp.dat」ファイルを作成、または修正する必要があります。

Windows 環境 : ユーザ設定（共通）フォルダ

Linux/UNIX 環境 : ランタイム製品導入ディレクトリ

「era_jp.dat」ファイルの詳細に関してはマネージャのメニュー [ヘルプ]-[オンラインマニュアル] から「1. インストール」-「1.6 導入環境の設定」-「1-6-9 元号対応」をご覧ください。

「era_jp.dat」ファイルを用いた元号の設定例

ex1)

era_jp.dat ファイルの記述

2019/05/01, 令和, 令, R

入力テキストの設定

分類 : 日付

日付の表記方法 : gggee 年 mm 月 dd 日

変換元 : 2019/05/01

変換後 : 令和元年 05 月 01 日

ex2)

era_jp.dat ファイルの記述

2019/05/01, 令和, 令, R

元号を追加しました

2021/09/01, 久里, 久, C

入力テキストの設定

分類 : 日付

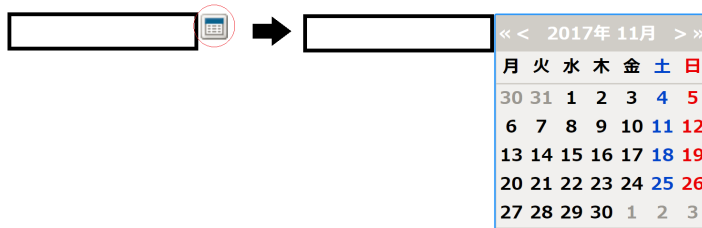
日付の表記方法 : ge. m. d

変換元 : 2023/10/01

変換後 : C3. 10. 1

[日付] フォーマットが設定されたテキストボックスでは、カレンダーを使用した日付入力を行えます。[日付] フォーマットが設定されたテキストボックスにフォーカスが当たるとカレンダーアイコンが表示されます。アイコンをクリックするとカレンダーが表示され、カレンダーから日付を選択できます。

図：カレンダーアイコン / カレンダー



<< 注意 >>

オブジェクトに読取専用が設定されている場合は、カレンダーは表示されません。

【直接入力を許可する】

ユーザによる直接入力を許可するかどうかを設定できます。設定されていない場合、カレンダーからの入力のみが有効になります。設定されている場合、直接入力が可能となります。

<< 注意 >>

〔日付〕フォーマットのテキストボックスに対してユーザの直接入力により想定外の形式の文字列が入力された場合は、入力されたデータがそのまま送信されます。

サブミット時の送信値

〔日付〕フォーマットが設定された入力テキストオブジェクトを送信した場合、「YYYY（西暦年4桁）」、「DD（月2桁）」、「MM（日2桁）」とスラッシュ（/）を組み合わせたデータが送信されます。組み合わせの形式は「日付の表記方法」設定により変わります。（表記方法に使用されている値の種類と順番に依存します。）

以下が送信される組み合わせの形式と、それに対応する日付の表記方法になります。

・「YYYY/MM/DD」形式

yyyy 年 m 月 d 日	yyyy 年 mm 月 dd 日	yy 年 m 月 d 日
yy 年 mm 月 dd 日	yyyy/mm/dd	yyyy-mm-dd
yyyy. mm. dd	yy-mm-dd	ge 年 m 月 d 日
gge 年 m 月 d 日	ggge 年 m 月 d 日	gee 年 mm 月 dd 日
ggee 年 mm 月 dd 日	gggee 年 mm 月 dd 日	ge. m. d
gge. m. d	ggge. m. d	gee. mm. dd
ggee. mm. dd	gggee. mm. dd	

・「MM/DD/YYYY」形式

m/d/yy	m/d/yyyy	mm/dd/yy
mm/dd/yyyy	mmm d, yyyy	mmmm d, yyyy

・「DD/MM/YYYY」形式

d-mmm-yy	d-mmm-yyyy	dd-mmm-yy
dd-mmm-yyyy		

・「MM/YYYY」形式

mm/yy	mm/yyyy	mmm-yy
mmm-yyyy	mmmm-yy	mmmm-yyyy

- ・「MM/DD」形式

m/d

- ・「DD/MM」形式

d-mmm

<< 注意 >>

年が和暦の表記方法の場合でも、サブミット時は西暦 4 桁が送信されます。

月日が m 形式や d 形式などの 0 埋めされない表記方法の場合でも、サブミット時は 0 埋めされた値が送信されます。

内部値の取得・設定

フォーマットが設定されたオブジェクトの値を直接 JavaScript で取得 / 設定すると表示上のデータが取得されます。そのため、内部データを直接取得 / 設定する関数を用意しています。関数は、要素の DOM オブジェクトからアクセスできます。内部データを取得 / 設定したい要素の DOM オブジェクトを取得し、以下の関数を実行してください。

内部データの取得

内部データを取得する場合は `getInnerValue` 関数を使用します。

```
element.getInnerValue()
```

引数

なし

戻り値

オブジェクトのフォーマット適用前内部データ

内部データの設定

内部データを設定する場合は `setInnerValue` 関数を使用します。

```
element.setInnerValue(value)
```

引数

value : オブジェクトのフォーマット適用前内部データ

戻り値

なし

`setInnerValue` 関数では、内部値を更新後、表示値を更新します。同名入力フォームオブジェクトが配置されている場合は値の同期も行われます。値の同期の詳細については「5. 値の同期」を参照してください。

[例] 変数名「NAME」のオブジェクトについて、内部値を 1 加算する処理を記述する場合、例えば以下のような JavaScript ソースを記述することで対応できます。

```
// DOM オブジェクト取得
var element = document.getElementsByName("NAME");
// 内部値を取得
var before = parseInt(element[0].getInnerValue());
// 1 加算した値を内部値に設定
element[0].setInnerValue(before + 1);
```

3-2. 他に入力フォーム間の値の計算結果を表示させる - 計算

テキストボックスの属性ダイアログに属する [計算] タブでは、他に入力フォームの値を用いた計算設定を行うことができます。この設定は、フォーマット設定において “ 数値 ”、“ パーセント ” フォーマットが設定されている場合のみ有効となります。

図：計算設定

入力フォームを指定して計算に使用する

テキストボックスには他に入力フォームの値を用いた計算結果を表示することができます。計算の方法、使用する入力フォームの指定は [計算] タブの項目で設定します。

[計算の種類]

計算方法を設定するには、[計算の種類] で、

和 差 積 商 平均 最小 最大

の中から選択します。

[全ページ集計を行う]

複数ページを出力したときに全ページの「計算に使うカラム」を集計対象にすることができます。[全ページ集計を行う] は、[計算の種類] が以下に設定されている場合にのみ設定することができます。

和 平均 最小 最大

<< 注意 >>

全ページ集計では同じフォームのオブジェクトが集計対象になります。別のフォーム内に配置されている同名変数は集計の対象外となります。

[計算に使うカラム]

[計算の種類] で選択した計算方法で計算させる入力フォームを指定します。

[編集] を押下すると計算に使用する入力フォームの選択ダイアログが表示されます。

図：計算に使用する入力フォームの選択ダイアログ



左側には計算に使用することができる入力フォームオブジェクト名が一覧になっています。計算に使用した入力フォームオブジェクト名を選択して [追加] をクリックすることで右側の計算対象入力フォームの一覧に追加されます。また、計算対象から入力フォームを削除したい場合は、右側の一覧から対象の入力フォームオブジェクト名を選択して [削除] をクリックすることで、左側の一覧に入力フォームを戻します。

※ [計算に使うカラム] で選択した入力フォームの順番は計算の順序に影響します。例えば、[計算の種類] で “ 差 ” を選択した場合、[計算に使用するカラム] において “VALUE1 (値 =5)”、“VALUE2 (値 =3)”、“VALUE3 (値 =1)” の順で追加した場合には計算は次のように行われます。

$$\text{VALUE1 (5)} - \text{VALUE2 (3)} - \text{VALUE3 (1)} = 1$$

この追加の順番を逆から行くと、

$$\text{VALUE3 (1)} - \text{VALUE2 (3)} - \text{VALUE1 (5)} = -7$$

となります。つまり “ 差 ”、“ 商 ” といった計算方法が指定されている場合には、入力フォームの追加する順番によって計算結果が異なる可能性がありますのでご注意ください。

また、フォーム上に計算が設定された入力フォームが複数定義されている場合には、計算対象となるオブジェクトに設定されている計算が先に行われます。

[全てのカラムが空白の時]

[計算に使うカラム] が全て空白のときに、計算結果に 0 を表示するか表示なしにするかを選択できます。

例えば、[計算の種類] で “ 和 ” を選択し、[計算に使用するカラム] において “VALUE1 (値 = 空白)”、“VALUE2 (値 = 空白)” が追加されている場合、「全てのカラムが空白の時」に [0 を表示] が選択されている場合は計算結果として “0” を表示します。[表示なし] が選択されている場合は、計算結果は表示されません。

<< 注意 >>

1 つでも値が空白でないカラムがある場合は、「全てのカラムが空白の時」の設定にかかわらず計算値が表示されます。このとき、空白のカラムは数値が “0” として計算されます。

任意の値を計算に使用する

前述の入力フォーム間の計算結果に対して、固定の値を用いた計算を設定することができます。

固定値を用いた計算方法は [固定値]-[計算の種類] で、

和 差 積 商

から選択することができます。

固定値を用いた計算は、次のような消費税込みの金額を算出したい場合などで有効です。

計算例：VALUE1、VALUE2 の値の合計に 1.08 を掛けた消費税込みの金額を VALUE3 に表示する

※この計算設定は、計算結果を表示する VALUE3 で行います。

① [計算の種類] で “和” を選択する。

計算の種類:

② [計算で使用するカラム] で “VALUE1”、“VALUE2” を選択する。

計算に使うカラム:

③ [固定値] の [計算の種類] で “積” を選択する。

固定値
計算の種類:

④ [固定値] で “1.08” を指定する。

固定値:

以上の設定を行うことで VALUE3 には VALUE1、VALUE2 の値の合計に消費税を含んだ金額が表示されます。

VALUE1	VALUE2	VALUE3
¥1,000	+	¥500
		=
		¥1,620

※固定値を用いた計算は、入力フォーム間の計算結果に対して指定の値を計算させるものです。指定の値に対して入力フォーム間の計算結果の値を計算させることはできません。このような計算を行いたい場合には、後述の「カスタム JavaScript を使用して独自の計算方法を指定する」をご覧ください。

3-3. 入力した（選択した）入力フォームの値を検証する - 検証

テキストボックスの [検証] タブではテキストボックスに入力できる値を制限するための設定を行うことができます。この設定により、テキストボックス上に適切なデータのみを入力させることができるようになります。

図：検証設定

表示 | オプション | アクション | フォーマット | **検証** | 計算 | 位置 | HTML | 小数点一桁

値の範囲

最小値:

最大値:

テキストボックスの値の範囲を数値で指定する

数値、パーセントフォーマットが設定されている場合、テキストボックス内に入力可能な最小値、最大値を指定することができます。

最小値を検証する場合には、[最小値]の左のチェックボックスを選択して任意の最小値を指定します。同様に、最大値を検証する場合には、[最大値]の左のチェックボックスを選択して任意の最大値を指定します。

図：値の範囲設定

設定可能な数値の範囲は、-99,999,999,999.999 ~ 99,999,999,999.999 までの値となります。

3-4. アクションを組み込む

アクションとは、HTML ファイルの静的な表現に加え、ページが切り替った際の動作や入力フォームに対する操作が行われた際の動作などを設定することができる機能です。このアクションを利用することで HTML ファイルに対して動きや対話性を高めることができます。

Create!Form Screen では、以下の項目に対してアクションを設定することができます。以下のアクションを設定することができます。それぞれのアクションは JavaScript イベントに対応しています。

アクション	対応する JavaScript イベント
ページのアクション	
ページを開いた時	onload
入力フォームオブジェクトのアクション	
キーボードのキーを押したとき	onkeydown
キーボードのキーを入力したとき	onkeypress
キーボードのキーを放したとき	onkeyup
クリックしたとき	onclick
ドラッグ&ドロップで選択したとき	onselect
フォーカスを外したとき	onblur
フォーカスを合わせたとき	onfocus
ポインタを範囲外に出したとき	onmouseout
ポインタを範囲内に合わせたとき	onmouseover
内容を変更したとき	onchange

これらの項目はアクションを起こす“きっかけ”という意味から、以降『トリガー』と呼ぶこととします。

これらのトリガーに対して JavaScript を設定することができます。

入力フォームにアクションを組み込む

ブラウザ上で入力フォームに対してマウス操作や Tab キーによる移動などの操作が行われた際に動作するアクションを組み込むことができます。

入力フォームにアクションを組み込むには、入力フォームオブジェクト属性ダイアログ内の [アクション] タブで設定することができます。

図：入力フォームのアクション設定



[トリガー]

アクションを設定するトリガーを選択します。

[JavaScript イベント名]

選択したトリガーに対応する JavaScript イベントが表示されます。

[編集]

選択したトリガーにアクションを登録します。

登録するアクションの詳細な設定方法は、以降の内容をご覧ください。

アクションを設定できるトリガーは以下となります。

- キーボードのキーを押したとき
- キーボードのキーを入力したとき
- キーボードのキーを放したとき
- クリックしたとき
- ドラッグ&ドロップで選択したとき
- フォーカスを外したとき
- フォーカスを合わせたとき
- ポインタを範囲外に出したとき
- ポインタを範囲内に合わせたとき
- 内容を変更したとき

<< 注意 >>

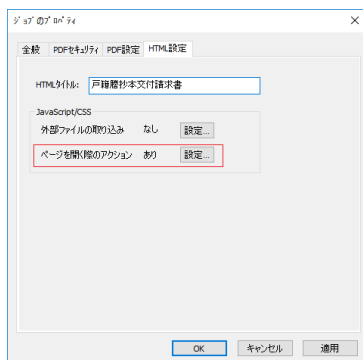
オブジェクトにより設定できるトリガーは異なります。各オブジェクトのプロパティダイアログの [アクション] タブ内で表示されているトリガーのみが設定可能なトリガーとなります。

ページにアクションを組み込む

ページを開いたタイミングで動作するアクションを組み込むことができます。

ページにアクションは、マネージャのジョブのプロパティダイアログから [HTML 設定] タブの「ページを開く際のアクション」から設定することができます。

図：ジョブプロパティダイアログ [HTML 設定] タブ



アクションを組み込む - JavaScript の実行

以下の操作を行う際に [JavaScript の編集] ダイアログが表示されます。

1. Form エディタ 入力フォームオブジェクトプロパティダイアログ [アクション] タブの [編集] ボタンを押下したとき
2. マネージャ ジョブプロパティダイアログ [HTML 設定] タブの「ページを開く際のアクション」の [設定] ボタンを押下した時

トリガーによって任意の動作を行わせたい場合には、その動作内容を JavaScript でプログラムし、アクションに組み込むことで実現することができます。

[JavaScript の編集] ダイアログについては、「6-6. JavaScript を使用する」をご覧ください。

3-5. 入力チェック

Create!Form Screen から出力された HTML では、ブラウザ上で入力されたデータが、オブジェクトの入力ルールと合致しない場合は警告メッセージが表示されます。

入力チェックの種類

入力チェックは以下の種類があります。

数値チェック

入力された値が数値であるかどうかをチェックします。フォーマットに [数値] または [パーセント] が設定されている場合にチェック処理が行われます。

数値範囲チェック

入力された値が、最大値 / 最小値の設定の範囲内であるかどうかをチェックします。[検証] タブの [値の範囲] が設定されている場合にチェック処理が行われます。

最大文字数チェック

入力された値が設定された最大文字数を超過していないかをチェックします。テキストボックスオブジェクトの[オプション]タブで[最大文字数の指定]が設定されている場合にチェック処理が行われます。

必須入力チェック

必須入力設定されている入力フォームに入力が行われているかをチェックします。[表示]タブで「必須入力」が設定されている場合にチェック処理が行われます。

入力エラー発生時の挙動

入力エラーが発生した場合、以下の3つの挙動によりユーザへ入力エラーが発生したことを通知します。

1. メッセージボックスを表示します。
2. エラーアイコンが表示されます。
3. 背景色を赤く表示します。(テキストボックスのみ)

図：エラーアイコン / 背景色赤表示



入力チェックが発生するタイミング

入力チェックは入力時とサブミットボタン押下時に行われます。

入力時

対象のオブジェクトの入力値が変更されたときに、入力が行われた入力フォームに対してチェックが行われます。ただし、「必須入力チェック」は行われません。

サブミットボタン押下時

Create!Form Screenで配置されるサブミットボタンを押下したときには、全ての入力フォームに対してチェックが行われます。入力エラーが発生した場合は、サブミット処理は中断されます。

4. 外部からのデータを反映させる

4-1. 初期データを外部データから指定する

テキストボックスに表示するテキストを外部データから指定する

Datamap エディタ上で対象のテキストボックスに表示、出力したい文字列の記述位置をマッピングします。

テキストボックスに対してフォーマットが設定されている場合には、設定に対応したデータが用意されている必要があります。例えば、数値フォーマットが設定されている入力フォームに“あいうえお”というデータがマッピングされている場合には、警告レベルのエラーとなり入力フォームにデータが出力されません。

同様に検証設定（数値の範囲のみ）が行われている場合においても、設定に対応したデータが用意されている必要があります。

選択状態にするメニュー項目を外部データから指定する

Form エディタで設定したコンボボックス（もしくはリストボックス）に登録されているメニュー項目の一覧から選択状態にしたい項目の“書き出し値”（書き出し値の指定がない場合には“項目名”）をマッピングデータとして用意する必要があります。

コンボボックス、リストボックスでは、通常 Form エディタ上で登録するメニュー項目の一覧をマッピングデータから取り込むこともできます。この詳細については、「6-2. メニュー項目を外部データから取り込む」をご覧ください。

チェックマークを入れる項目を外部データから指定する

ラジオボタン、もしくはチェックボックスを外部のデータを反映させて選択状態にするには、Form エディタ上のオブジェクトオプションの [書き出し値]-[チェック時] で指定されている値に対応したデータが用意されている必要があります。一致しない場合には、出力時に選択状態が反映されません。

反対に、[デフォルトでチェックする] が指定されたチェックボックスを外部のデータを反映させて非選択状態にすることもできます。この場合、Form エディタ上のオブジェクトオプションの [書き出し値]-[非チェック時] で指定されている値に対応したデータが用意されている必要があります。

フォームデータの送信先 URL を外部データから指定する

サブミットボタンの送信先 URL を外部のデータから指定するには、送信先の URL を示すテキストデータに対してデータマッピングを行います。

Form エディタ上のオブジェクトオプションの [送信先 URL] で送信先が指定されている場合でも、データマッピングが行われている場合にはマッピングデータが優先的に反映されます。

4-2. 入力フォームオブジェクトにデータマッピングを行う

Create!Form Cast / Screen では、入力フォームオブジェクトに対してデータマッピング（データの割り当て）を行うことで、出力時に入力フォームにデータを埋め込んだ状態でファイルを出力することができます。

データマッピングの詳細な説明は、Design マネージャのメニュー [ヘルプ] - [オンラインマニュアル] から「3. 機能リファレンス」-「3.2 エディタの操作」-「Datamap エディタ」をご覧ください。

マッピング方法

入力フォームオブジェクトのデータマッピングは他の変数のデータマッピングと同様に Datamap エディタ上で行います。マッピングの方法も他の変数オブジェクト同様に行うことができます。

※リセットボタン、ボタンは、データマッピングを行うことができません。

マッピングデータ

マッピングされるデータとその出力結果は入力フォームオブジェクトごと、またはその設定内容によって異なります。以下の項では、入力フォームオブジェクトごとのマッピングデータについて説明します。

4-3. データマッピングを行わない場合に反映される入力フォームの値

入力フォームオブジェクトは、他の変数と異なりデータマッピングを行わなくてもファイルに出力されます。この場合、出力には各入力フォームオブジェクトの初期設定内容が反映されません。

テキストボックス

[オプション] タブ内の「空白時に初期値を使用」が設定されている場合は、Form エディタ上のオブジェクトオプションの [初期値] に指定された文字列が出力されます。

コンボボックス

リストボックス

Form エディタ上のオブジェクトオプションの登録されたメニュー項目一覧で初期選択として指定された項目が選択された状態として出力されます。

チェックボックス

ラジオボタン

Form エディタ上のオブジェクトオプションの [デフォルトでチェックする] が設定されているオブジェクトが選択された状態で出力されます。設定されていない場合には、未選択状態で出力されます。

サブミットボタン

Form エディタ上のオブジェクトオプションの [送信先 URL] で指定された URL 文字列がフォームデータの送信先として設定されます。

5. 値の同期

5-1. 同名オブジェクトの値の同期

入力フォームオブジェクトでは、同名変数名で配置された複数の入力フォームオブジェクト間で値が同期されます。また、ページインデックスが付与されていない入力フォームオブジェクトに関しても、設定により値を同期させる事ができます。

値の同期は以下のタイミングで行われます。

1. ランタイム実行時

ランタイムが実行されたタイミングで、該当オブジェクトの初期データは同期されます。複数ページ間の同期が有効で、かつ複数ページ間で別々の値がマッピングされている場合、最初に決定した値に基づいて値の同期が行われます。

2. HTML 上でのフォーム入力値変更時

ユーザが HTML 上の入力フォームの入力値を変更したときに、該当オブジェクトの入力値は同期されます。

5-2. ページインデックス

「ページインデックスを付与」の設定により、複数ページ間の同名オブジェクトで値の同期を行うかどうかを切り替えることができます。

「ページインデックスを付与」の設定有無により、以下のような挙動になります。

- ・ 設定なし：値の同期が行われます。
- ・ 設定あり：値の同期は行われません。（※）

※同ページ内の同名オブジェクトについては値が同期されます。

「ページインデックスを付与」の設定は入力フォームオブジェクトのプロパティダイアログの [HTML] タブから設定できます。

図：[HTML] タブ



5-3. 複数ページ間で値が同期される時の挙動

ページインデックスが付与されていない入力フォームオブジェクトでは、複数ページ間で値が同期されます。

①改ページ出力

例えば、[NAME] というテキストボックスオブジェクトが定義されているフォームを用いて出力された 3 ページの HTML ファイル上で、1 ページ目の [NAME] テキストボックスに入力を行った場合、2、3 ページ目は以下のような内容になります。

図：改ページ時のテキストボックスの挙動

■ 1 ページ目に「帳票太郎」と入力した場合

2 ページ目

名前：

3 ページ目

名前：

■ 3 ページ目に「帳票花子」と入力した場合

1 ページ目

名前：

2 ページ目

名前：

②マルチフォーム・セット出力

マルチフォーム、セット出力の場合も 1 フォーム内に同名入力フォームオブジェクトが繰り返し使用されている場合、もしくは異なるフォーム内で同名入力フォームオブジェクトが複数定義されている場合には、改ページ出力と同様に HTML ファイル内の同名入力フォームオブジェクトは値が同期されます。

5-3-1. ページインデックスを付与しない場合の送信時の name 値

「ページインデックスを付与」が設定されていないオブジェクトをサブミットした時の name 値は、入力フォームオブジェクトに設定された変数名と同値になります。

5-4. 複数ページ間で値を同期させたくない場合

複数ページ間で値を同期させないようにするためには、「ページインデックスを付与」の設定を行います。

「ページインデックスを付与」が設定されたオブジェクトでは他のページの同名入力フォームオブジェクトと値を同期しなくなります。

5-4-1. ページインデックスを付与する場合の送信時の name 値

「ページインデックスを付与」が設定されているオブジェクトをサブミットしたときの name 値は以下のルールに従って決定されます。

変数名_フォーム識別子(※)_同一フォームの出力回数インデックス(0～)

※フォーム識別子に関しては後述します。

[例]

フォーム識別子がそれぞれ formA、formB に設定されている 2 つのフォームが登録されているマルチフォームがあり、全てのページに「ページインデックスを付与する」の設定が行われた変数名 [NAME] のテキストボックスオブジェクトが配置されているとします。

出力順と各ページの入力値を以下とします。

- ・出力順 : formA → formB → formA
- ・入力値 1 ページ目 : 「ABC」 / 2 ページ目 : 「DEF」 / 3 ページ目 : 「GHI」

このとき、サブミットボタンを押下したときに送信される name 値と送信値の組み合わせは以下となります。

- ・NAME_formA_0 : ABC
- ・NAME_formB_0 : DEF
- ・NAME_formA_1 : GHI

5-5. フォーム識別子

フォーム識別子は、各フォーム毎に設定可能なフォームを区別するための文字列です。フォーム識別子は、入力フォームオブジェクトにページインデックスを付与した場合に、name 値の一部として使用されます。フォーム識別子は Form エディタの [メニュー]-[フォーム設定]-[Screen 設定] タブで設定できます。

図 : [Screen 設定] タブ



5-6. 各種オブジェクト毎の同期の挙動

ここでは、値の同期が行われる場合のオブジェクト毎の挙動について説明します。

5-6-1. 同期対象のオブジェクト

以下のオブジェクトは同期対象のオブジェクトとなります。

- ・テキストボックスオブジェクト
- ・コンボボックスオブジェクト
- ・リストボックスオブジェクト
- ・チェックボックスオブジェクト
- ・サブミットボタンオブジェクト

5-6-2. 各種オブジェクトの同期のルール

同期される値やルールはオブジェクト種別ごとに異なります。

テキストボックス

テキストボックスでは入力された値が同名オブジェクト間で同期されます。

コンボボックス/リストボックス

コンボボックスやリストボックスでは、選択肢が同名オブジェクト間で同期されます。

チェックボックスオブジェクト

チェックボックスでは、「変数名」と「チェック時の書き出し値」の両方が一致するオブジェクト間でチェック状態が同期されます。変数名だけが一致しているオブジェクト間ではチェック状態は同期されません。

<< 注意 >>

非チェック時の書き出し値は同期のルールには影響を与えません。

6. Tips

6-1. 出力時に入力フォームの外観、状態を切り替える - FormSwitch オプション

Create!Form Cast / Screen では、出力パターンごとに入力フォームの外観等の設定を行うことができ、また出力時に FormSwitch オプション (“-fsN”) を実行コマンドラインに指定することでパターン別に入力フォームを出力することができます。この機能により同一の資源ファイルを使用して複数パターンを入力フォームを出力することができます。

出力パターン別に入力フォームの外観設定を行う

出力パターンごとの入力フォームの外観設定は、各入力フォームオブジェクトの属性ダイアログの [表示] タブで行います。

出力パターン別に設定できる項目は以下のとおりです。各項目の詳細については、「2-2. 入力フォームの外観、表示状態を設定する」をご覧ください。

境界線の有無	境界線種
境界線幅	境界線色
背景色の有無	背景色
読取専用	表示

入力フォームオブジェクト属性ダイアログの [表示]-[プロパティ]-[出力パターン] 横の [設定] ボタンを押下すると、[出力パターン一覧] ダイアログが表示されます。

図：[出力パターン一覧]



[No.]

出力パターン番号を示します。

[線種]

境界線種を表示します。境界線の描画を行わない設定の場合には“設定なし”と表示されます。

[線幅]

境界線幅を表示します。境界線の描画を行わない設定の場合には“設定なし”と表示されます。

[線色]

境界線色が表示されます。境界線の描画を行わない設定の場合には“x”と表示されます。

[背景]

背景色が表示されます。背景色を設定しない場合には“×”と表示されます。

[読取専用]

読取専用の設定の有無が表示されます。設定されている場合には“○”、設定されていない場合には“×”が表示されます。

[表示]

表示の設定内容が表示されます。

設定したい出力パターン項目を選択して [編集] ボタンを押下（もしくは項目をダブルクリック）すると [出力パターン] ダイアログが表示されます。ここでパターンに対応した入力フォームの外観設定を行います。

図： [出力パターン]

初期状態では、[表示] 設定での値を反映するようになっています。パターン別の設定を行うには [パターン設定] を選択し、各項目の設定を行います。各項目の設定方法、内容は前述の項目をご覧ください。

出力パターン別にファイルの出力を行う

アプリケーションから Create!Form Cast / Screen を呼び出してファイルを生成する際に前項までに設定した出力パターンごとの入力フォームの外観設定を反映させるには、実行コマンドラインに FormSwitch オプション (“-fsN”) を指定して Create!Form Cast / Screen を呼び出し実行します。

FormSwitch オプション (“-fsN”) の “N” には 1 ～ 10 の出力パターン番号を指定します。指定されない場合 (“-fs” のみ) には、“-fs1” と同じ出力が行われます。

図：通常の実行例

図：FormSwitch オプションを指定した実行例

自動車総合保険申込書 < 保険申込書代理店控 >
 兼保険料徴収書写し

郵便番号	160-0023	電話番号	03-3360-6691
申込人住所	東京都新宿区西新宿7-5-25		
保名	帳票 太郎		

THE ITM
 株式会社ITM
 〒160-0023 東京都新宿区西新宿7-5-25
 TEL:03-3360-6691 FAX:03-3360-6692

6-2. メニュー項目を外部データから取り込む

コンボボックスやリストボックスのメニュー項目は通常、Form エディタ上でオブジェクトのオプション設定で登録を行いますが、Create!Form Cast / Screen ではデータマッピングを行うことで出力時に外部のデータからメニュー項目を登録することができます。

この場合、マッピングデータとして定型のデータを用意する必要があります。

定型データ (TXT/CSV/XML/DB 共通)

```
selected_value;item_name1:item_value1;item_name2:item_value2;...
```

説明

selected_value	選択状態を示す項目の書き出し値
item_name	メニュー項目リストに追加する項目名
item_value	項目名に対応する書き出し値

データの先頭には、通常のコボボックス / リストボックスのマッピングデータである選択状態とする項目の書き出し値を記述します。そのデータのあとにセミコロン（';'）で区切って項目リストデータを追加していきます。1つの項目データは item_name と item_value をコロン（':'）で区切ったセット（※）で構成されます。複数の項目を登録する場合には、セミコロン（';'）で区切って項目のセットを追加していきます。

※追加する項目データは item_name のみでも可能です。この場合、入力フォームオブジェクト内では項目名が書き出し値を兼ねることになります。

メニュー項目を外部データから取り込んだ場合には、入力フォーム作成時に登録したメニュー項目は無効となります。

図：実行例

■コンボボックス (リストボックス) 入力フォームオプション設定

項目名	書き出し値
<input checked="" type="checkbox"/> りんご	apple
<input type="checkbox"/> オレンジ	orange
<input type="checkbox"/> バナナ	banana

■データマッピングとマッピングデータ

変数名	種別	属性	マッピングデータ
COMBO	フィールド	コンボ	grape;いちご:strawberry;ぶどう:grape;キウイ:kiwi;

■ 出力結果

ぶどう ▼
いちご
ぶどう
キウイ

6-3. 指定したサーバスクリプトにフォームデータを送信する

サブミットフォームアクションを実行することによって、入力、もしくは選択した入力フォームのデータを Web サーバに送信することができます。サーバアプリケーション側では送信されたフォームデータの形式に合わせて、スクリプトなどデータを処理する仕組みを備えている必要があります。

送信されるフォームデータ

各入力フォームの送信されるデータは以下のとおりです。

テキストボックス

テキストボックスに入力されたデータが送信されます。[フォーマット] 設定が行われている場合には、テキストボックス上で表示されている形式ではなく、フォーマット変換前の形式のまま送信されます。

コンボボックス / リストボックス

選択されたメニュー項目に設定されている書き出し値が送信されます。メニュー項目に書き出し値が設定されていない場合には項目名が送信されます。

コンボボックスで [フォーマット] 設定が行われている場合には、表示されているデータの形式ではなく、メニュー項目に登録されている書き出し値の形式で送信されます。

リストボックスで [複数選択] が設定されている場合、サブミットボタンで送信される値は選択されている項目の書き出し値をカンマ区切りで連結したデータとなります。

[例] 項目名、書き出し値が以下の設定されているとします。

```
name1, value1
name2, value2
name3, value3
```

このとき、name1 と name2 が選択されているときの送信データは「value1,value2」となります。

チェックボックス / ラジオボタン

選択された入力フォームオブジェクトに設定されている書き出し値が送信されます。

データ送信形式とその処理方法の例

フォームデータを Web サーバに送信する際のデータ形式と、データを処理する方法例は以下となります。

[説明]

フォームデータを POST リクエストとして送信します。

[処理例]

送信されたデータを Web サーバ側で処理するためには、HTTP POST メソッドに対応した各スクリプト言語のフォームデータ取得メソッド（例えば Java Servlet の HttpServletRequest インタフェースで提供されている getParameter (String) 等）を利用することでフォームデータを取得することができます。

送信時の文字コード

フォームデータを Web サーバに送信する際の文字コードを指定できます。送信時文字コードは以下から選択できます。

UTF-8
Shift-JIS
EUC-JP
iso-2022-jp

Windows 環境の送信時文字コード設定方法

送信時文字コードはマネージャの環境設定の [Screen 設定] タブから設定できます。

図 : [Screen 設定] タブ

**Linux 環境の送信時文字コード設定方法**

以下に説明する送信時文字コード設定用の環境変数を設定して、送信時の文字コードを変更してください。

文字コード設定用の環境変数

CREATE_SUBMIT_CHARSET= 値 (値 : UTF8、SJIS、EUC、ISO2022)

6-4. ツールチップを表示させる

ブラウザ上でマウスが入力フォームの上にある場合、ツールチップを表示することができます。

設定は、入力フォームオブジェクトの属性ダイアログ上で行います。

図：ツールチップの設定

入力フォームテキストボックス

オブジェクト番号: 084 記述: ITTEXT

オブジェクト名: NAME

ツールチップ: 名前

[ツールチップ]

ツールチップ機能の使用の有無を設定します。横のテキストボックスにはツールチップとして表示するテキスト文字列を指定します。

図：ツールチップの表示例



6-5. Tab キーで移動する順序を指定する

Create!Form Screen で出力した HTML ファイルをブラウザで開いた際に、Tab キーを押して移動する入力フォームの順序を指定することができます。

この設定は、Form エディタのメニューから [オプション] - [入力フォーム] - [タブオーダー] を選択して表示される [タブオーダーの設定] ダイアログで行います。

図：[タブオーダーの設定]

タブNo.	オブジェクトNo.	入力フォーム名	記述
1	101	SUBMIT1	OK
2	102	SUBMIT2	OK
3	103	B_CREATE	ICLR
4	98	B_DETAIL	PBUTTON
5	104	TODAY	ITEXT
6	99	HONSEKI	本籍
7	100	HITTOUSYA	筆頭者
8	24	CHECK_KIND1	戸籍
9	25	CHECK_KIND2	除籍
10	26	CHECK_KIND3	改製原戸籍
11	27	CHECK_KIND4	戸籍附票
12	28	CHECK_KIND5	PCHECK
13	31	TOUHON_NUM	届本請求数
14	29	SYOUHON_NAME	届本対象者氏名
15	106	SYOUHON_DATE	ITEXT
16	32	SYOUHON_NUM	抄本請求数
17	30	SYOUMESYO_NAME	身分証明書対象者氏名
18	105	SYOUMESYO_DATE	ITEXT
19	33	SYOUMESYO_NUM	証明書請求数
20	69	USE1	続柄
21	70	USE2	続柄

指定番号へ移動

OK キャンセル

[タブ No.]

[入力フォーム名] で示される入力フォームの現在のタブ順序番号が表示されます。

[オブジェクト No.]

[入力フォーム名] で示される入力フォームのオブジェクト生成番号が表示されます。

[入力フォーム名]

フォーム上で定義されている入力フォーム名が全て表示されます。同じ名前が付けられている入力フォームが複数フォーム上に存在する場合、ここには複数の同名の入力フォーム名が表示されます。この場合には[オブジェクトNo.]で入力フォームを識別するか、フォーム上の入力フォームをクリックすることでリスト内の対応する入力フォーム項目を選択状態にすることができます。

[記述]

[入力フォーム名]で示される入力フォームを作成した際に属性ダイアログ上の[記述]で指定した文字列が表示されます。

選択した入力フォームのタブ順序を先頭に指定する

タブ順序を変更したい入力フォームを選択し、[最上位へ]ボタンをクリックします。

選択した入力フォームのタブ順序を最後尾に指定する

タブ順序を変更したい入力フォームを選択し、[最下位へ]ボタンをクリックします。

選択した入力フォームのタブ順序を1つ上げる

タブ順序を変更したい入力フォームを選択し、[上へ]ボタンをクリックします。

選択した入力フォームのタブ順序を1つ下げる

タブ順序を変更したい入力フォームを選択し、[下へ]ボタンをクリックします。

選択した入力フォームのタブ順序を任意の位置に指定する

タブ順序を変更したい入力フォームを選択し、[指定番号へ移動...]ボタンをクリックすると[番号の指定]ダイアログが表示されます。

図：タブ順序 - [番号の指定]



任意のタブ順序番号を入力し、[OK]をクリックしてください。ここで、定義されている入力フォーム数よりも大きな番号を指定した場合には最後尾にタブ順序が変更されます。

選択した入力フォームをタブオーダーの対象外にする

タブ順序を変更したい入力フォームを選択し、[無効]ボタンをクリックします。

タブオーダー対象外入力フォームを再度タブオーダーの対象にする

タブ順序を変更したい入力フォームを選択し、[有効] ボタンをクリックします。

<< 注意 >>

「チェックマーク種別」に“グラフィカル”が設定されたチェックボックス / ラジオボタンはタブオーダーの対象外となります。

複数ページ時のタブオーダー

タブオーダー対象のオブジェクトが複数ページに配置されている場合、ページ内の全てのタブオーダーオブジェクトに遷移した後に次ページの先頭オブジェクトにタブが遷移します。

6-6. JavaScript を使用する

アクションで JavaScript を用いることでそれぞれに対応した任意の動作を行わせることができます。

JavaScript をコーディングする

アクションで設定する JavaScript は、[JavaScript の編集] ダイアログでコーディングを行います。

図： [JavaScript の編集]



[コメント]

この JavaScript アクションに対する任意のコメントを記述します。

[コード]

JavaScript コードを記述します。

このダイアログ上では、30,000 バイトまでのコードを編集することができます。

[外部エディタ]

外部エディタを使用して JavaScript コードを記述します。

外部エディタについては、「6-7. JavaScript を外部エディタを用いて編集する」をご覧ください。

[OK] をクリックすることで、JavaScript コードの登録が終了します。

6-7. JavaScript を外部エディタを用いて編集する

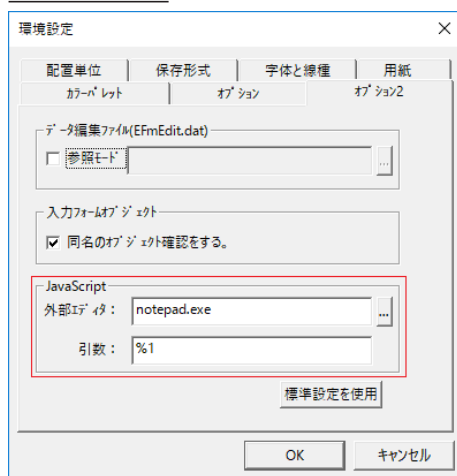
Form エディタ上で JavaScript を作成、編集する際にあらかじめ登録されていた外部エディタを使用することができます。初期状態では、Windows に付属されているメモ帳が外部エディタとして登録されています。

任意の外部エディタを登録する

JavaScript の登録、編集時に使用する外部エディタを登録するには、Form エディタの環境設定で指定することができます。

Form エディタのメニューから [ファイル]-[環境設定] を選択し、[オプション2] タブの [JavaScript] で任意の外部エディタを指定してください。

図：[環境設定]



[外部エディタ]

使用する外部エディタのパスを指定します。[...] ボタンを押下するとファイル指定ダイアログを利用することができます。

[引数]

外部エディタを起動する際の引数を指定します。

初期値で設定されている "%1" は外部エディタを使用して開くファイルパスを示しています。この前後に使用する外部エディタのオプション引数を指定してください。起動時の引数に関する詳細は、各エディタのヘルプを参照してください。

外部エディタを起動する

JavaScript アクションや JavaScript 関数を登録、編集する際に使用する [JavaScript の編集] ダイアログから [外部エディタ] ボタンをクリックすることで登録されている外部エディタを起動させることができます。

編集の場合には、既存 JavaScript コードの内容が外部エディタに反映されます。起動した外部エディタ上で JavaScript コードを記述した後、ファイルを保存して外部エディタを閉じると呼び出し側のダイアログ上に編集内容が反映されます。

6-8. 外部の JavaScript / CSS を適用する

Create!Form Screen では外部の JavaScript や CSS を用いて、ページ上の動作や装飾を定義することができます。

外部ファイルの指定方法

外部 JavaScript/CSS はファイル指定と絶対パス指定の 2 つの方法で指定できます。

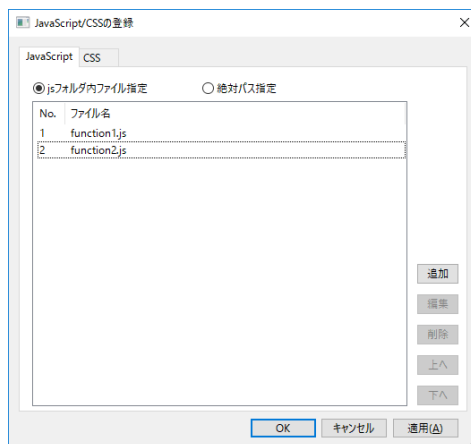
ファイル指定

ファイル指定は JavaScript URL / CSS URL に設定されたパス内でのファイルを指定する方法です。JavaScript ファイルや CSS ファイルの置き場が決まっているときに有効です。

絶対パス指定

絶対パス指定は JavaScript や CSS の URL をフルパスで指定する取り込み方法です。ファイル指定で対応できない場合や、ホスティングされた JavaScript / CSS を参照する場合に有効です。ファイル名指定、絶対パス指定の JavaScript/CSS は [JavaScript/CSS の登録] ダイアログから設定できます。

図：[JavaScript/CSS の登録] ダイアログ



[JS/CSS フォルダ内ファイル指定 / 絶対パス指定]

前述のファイル指定、絶対パス指定の選択を行います。

[ファイル / URL リスト]

設定されたファイルまたは URL のリストを表示します。ファイル指定の場合は、後述の [ファイル指定で追加できるファイル] の中からコンボボックスで選択します。絶対パス指定の場合は指定する URL をテキストで直接記述します。

[追加]

ファイル / URL を追加します。

[編集]

選択中のファイル / URL を編集します。

[削除]

選択中のファイル / URL を削除します。

[上へ]、[下へ]

選択中のファイル / URL の順序を変更します。順序を変更することにより、ファイル / URL の取り込み順序を変更できます。

取り込み JavaScript / CSS が複数設定されている場合、以下の順番で取り込まれます。ファイル間に依存関係がある場合はご注意ください。

絶対パス指定 JavaScript / CSS No 昇順 → ファイル指定 JavaScript / CSS No 昇順

ファイル指定で追加できるファイル

ファイル指定による URL の指定では、以下の JavaScript フォルダ / CSS フォルダに配置されたファイルから選択して追加できます。取り込む必要がある JavaScript / CSS ファイルは、以下のフォルダにコピーして配置してください。

- ・ JavaScript フォルダ
ユーザ設定（共通）フォルダ（※）直下の js フォルダ
- ・ CSS フォルダ
ユーザ設定（共通）フォルダ（※）直下の css フォルダ

※ユーザ設定（共通）フォルダはマネージャの [ヘルプ]-[バージョン情報]-[バージョン情報詳細] で確認できます。

JavaScript フォルダ / CSS フォルダはテスト実行の際に参照されます。そのため、本番環境で設定される URL が有効ではない場合でもファイル指定の JavaScript / CSS を使用してテスト実行ができます。

6-9. セレクタを用いて特定のオブジェクトにアクセスする

JavaScript / CSS から特定のオブジェクトを参照する場合、セレクタを用いてアクセスします。Create!Form Screen では、CSS クラス名と name 属性によるアクセスをサポートしています。

CSS クラス名によるアクセス

オブジェクトプロパティの [HTML] タブで設定できる [CSS クラス名] を用いてアクセスすることができます。[CSS クラス名] は半角英数字 / ハイフン / アンダーバーからなる 256 文字以内の文字列を指定できます。

図：[HTML] タブ

[例] CSS クラス名に “abcde” を指定した場合、JavaScript では以下のようにしてオブジェクトを取得できます。

```
// クラス名 “abcde” が設定されたオブジェクトのリストを取得
var objects = document.getElementsByClassName(“abcde”);
for (var i=0; i<objects.length; i++) {
  // 処理を記述 (例：値を “12345” に書き換える)
  objects[i].value = “12345”;
}
```

name 属性によるアクセス

オブジェクトの name 属性はオブジェクトに設定されている変数名に依存して決定されます。オブジェクトの name 属性値はページインデックス設定の有無によって変化します。name 属性値の詳細につきましては「5-3-1. ページインデックスを付与しない場合の送信時の name 値」、「5-4-1. ページインデックスを付与する場合の送信時の name 値」をご参照ください。

[例] 変数名に “TEXT” を指定したページインデックスを付与しないオブジェクトの場合、JavaScript では以下のようにしてオブジェクトを取得できます。

```
// name 値に “TEXT” が設定されたオブジェクトのリストを取得
var objects = document.getElementsByName(“TEXT”);
for (var i=0; i<objects.length; i++) {
  // 処理を記述 (例：値を “12345” に書き換える)
  objects[i].value = “12345”;
}
```

Create!Form 11

入力フォーム 第5版

発行日	2020年7月
発行者	インフォテック株式会社 〒160-0023 東京都新宿区西新宿 7-5-25