



Gluegent Flow 小技集 (e)JavaScript を使う



目次

1. 概要	2
2. 具体例	3
2.1. 分岐する経路	3
2.2. 消費税計算	4
2.3. 課税/非課税計算	4
2.4. 「その他」の場合入力必須	5
2.5. 「その他」の場合のみ入力可能	6
2.6. 入力形式チェック	7
2.7. 年度の取得	8
2.8. 日付の大小チェック	9
2.9. 時刻の大小チェック	10

1. 概要

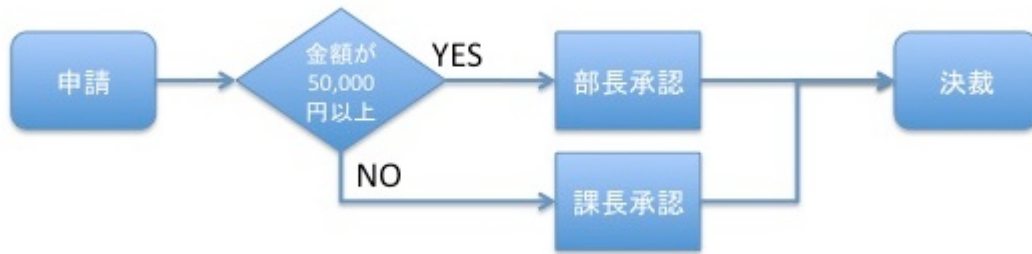
Gluegent Flow には分岐やカスタムラベルなどの入力フォーム、カスタムバリデーションなどの入力チェックを使った高機能なモデルを作成できます。これらの機能を使うには JavaScript を使う必要があります。これらの機能を使うときにどういった JavaScript のコードを書けばいいかを、具体例を挙げながら紹介していきます¹。

¹ JavaScript については、サポート対象外です。入門書やサイト等をご覧ください解決していただけますようお願いいたします。

2. 具体例

2.1. 分岐する経路

金額などの入力フォームの値によって、承認経路が変化するというモデルを作成したい。



この場合は、「経路」の「ボタン表示切替」を使います。JavaScript は以下のようなコードを記載します。

コード	解説
<pre> if (\${金額} >= 50000) { return true; } else { return false; } </pre>	<p>このコードのポイントは以下の 3 点です。</p> <ol style="list-style-type: none"> IF 条件式 この構文を使って入力値と指定の値を比較します。 if (条件式) { 条件式が True の時に実行される式 } else { 条件式が False の時に実行される式 } \${入力フォーム名} (プレースホルダー) \${ } で囲むことで入力フォームの値をこのスクリプト内で取得できます。 return true / return false true の場合はボタンが表示または使用可、false の場合はボタンが非表示または使用不可となります。 <p>つまり、このコードは入力フォーム「金額」の値が 50,000 以上の場合ボタンが押せる、50,000 未満の場合はボタンが押せないとなります。</p>

2.2. 消費税計算

単価・個数を入力すると税込価格が自動で表示されるといったモデルを作成したい。

単価	<input type="text" value="100"/> [数値]
個数	<input type="text" value="10"/> [数値]
税込金額	1080

この場合、タイプ「計算式」の入力フォームを使います。JavaScript は以下のようなコードを記載します。

コード	解説
return \${単価} * \${個数} * 1.08;	入力フォーム「単価」「個数」の積に税率(8%)をかけています。 算術演算子は加算(+)、減算(-)、乗算(*)、除算(/)です。

2.3. 課税/非課税計算

単価・個数を入力し、課税対象か非課税かで異なる計算結果が表示されるといったモデルを作成したい。

単価	<input type="text" value="100"/> [数値]
個数	<input type="text" value="10"/> [数値]
消費税	<input checked="" type="radio"/> 課税対象 <input type="radio"/> 非課税
金額	1000

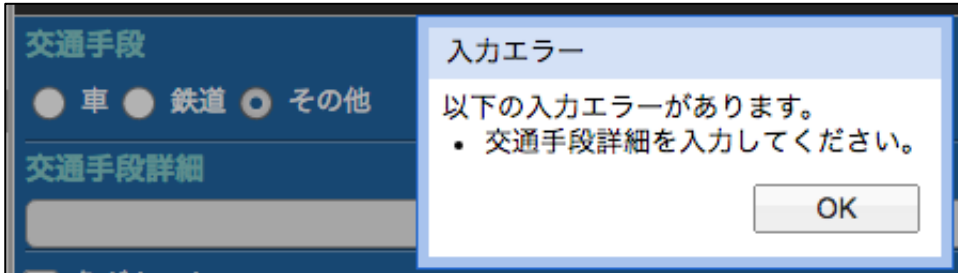
単価	<input type="text" value="100"/> [数値]
個数	<input type="text" value="10"/> [数値]
消費税	<input type="radio"/> 課税対象 <input checked="" type="radio"/> 非課税
金額	1080

この場合、タイプ「カスタムラベル」の入力フォームを使います。JavaScript は以下のようなコードを記載します。

コード	解説
<pre>if (\${消費税} == "課税対象") { return \${単価} * \${個数} * 1.08; } return \${単価} * \${個数};</pre>	<p>return 文の後は処理されないので else を省略できます。 数値に“円”を含めて表示したい場合は return (\${単価} * \${個数} * 1.08) + “円”; と記述します。</p>

2.4. 「その他」の場合入力必須

交通手段で「その他」が選択された場合は、交通手段詳細の入力が必須となるモデルを作成したい。



この場合、入力チェックのカスタムバリデーションを使います。JavaScript は以下のようなコードを記載します。

コード	解説
<pre>if (\${交通手段} == "その他"){ if (\${交通手段詳細} == undefined) { return false; } } return true;</pre>	<p>return false の場合はエラーとして扱われます。</p> <p>IF 文が 2 つ使われています。</p> <p>1 つ目の IF 文は交通手段がその他かどうかを聞いています。</p> <p>「その他」でない(車、鉄道)場合は処理を抜け true となります。</p> <p>「その他」の場合、2 つ目の IF 文で未入力かどうかを聞いています。</p> <p>未入力の場合「undefined」という値になります。</p>

2.5. 「その他」の場合のみ入力可能

種別で「その他」が選択された場合は、入力フォーム「種別その他」が表示されるような申請モデルを作成したい。

この場合、入力チェックの項目表示制御を使います。JavaScript は以下のようなコードを記載します。

コード	解説
<pre>if (\${種別} != "その他") { return "種別その他"; } return "";</pre>	<p>return “種別その他”; とは「種別その他」を非表示にするという意味になります。 非表示にしたい項目が複数ある場合は return [“項目 1”, “項目 2”]; のように配列として記述します。 逆に表示したい項目がない場合は return “”; となります。 != は「等しくない」です。</p>

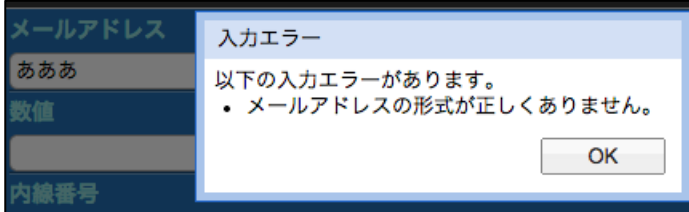
※ タイプが単一チェックの入力フォームの場合、項目表示切替があるため、この方法は不要です。

ただし 2 つ以上の単一チェックで項目表示切替を設定した場合や項目表示切替と入力チェックの項目表示制御の両方を設定している場合は正しく動作しません。

その場合は入力チェックの項目表示制御ですべて設定してください。

2.6. 入力形式チェック

対象の入力フォームの値がメールアドレスの形式となっていなかった場合はエラーとなるようなモデルを作成したい。



この場合、入力チェックのカスタムバリデーションを使います。JavaScript は以下のようなコードを記載します。

コード	解説
<pre>// 未入力チェック if (\${メールアドレス} == undefined) { // 未入力の場合は以下のチェックは行わず、 正常とする return true; } // 形式チェック if (!\${メールアドレス}.match(/^[A-Za-z0-9]+[\w-]+@[w\.-]+\w{2,}\$/)){ // !!は否定(~でない)。この場合、 match(xxx)で xxx に該当するものでないとなる。 return false; } return true;</pre>	<p>//はコメントです。以降に書かれた文字はスクリプトとして処理されません。</p> <p>未入力チェック部分は「メールアドレス」の入力値が空の場合、形式チェックは行われません。</p> <p>形式チェックは[文字列].match(正規表現)で表現します。</p> <p>入力値が正規表現の形式に合致する場合、true が返されます。</p> <p>IF 文を使わず、return \${メールアドレス}.match(xxx);とする方法もあります。</p>



2.7. 年度の取得

日付項目の入力日の年度を取得するような請モデルを作成したい。

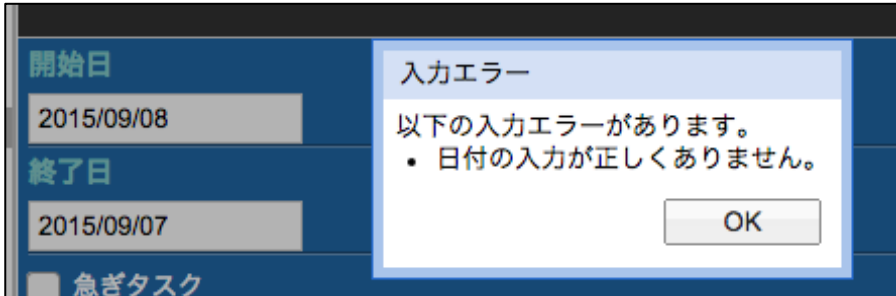
日付
2014/01/03
年度
2013

この場合、カスタムラベルを使います。JavaScript は以下のようなコードを記載します。

コード	解説
<pre>var dt = \${日付}.split("/"); var y = Number(dt[0]); var m = Number(dt[1]); if ((m >= 4) && (m <= 12)) { return y; } else { return y-1; }</pre>	<p>var xxx で変数を定義します。</p> <p>split で日付項目の“yyyy/mm/dd”の“/”で分割し、dt に配列として格納します。</p> <p>dt[0]に年、dt[1]に月、dt[2]に日の値が文字列型で格納されます。</p> <p>Number()で文字列型から数値型に変換しています。</p>

2.8. 日付の大小チェック

タイプが日付の入力フォームで開始日が終了日より後になった場合はエラーとしたい。

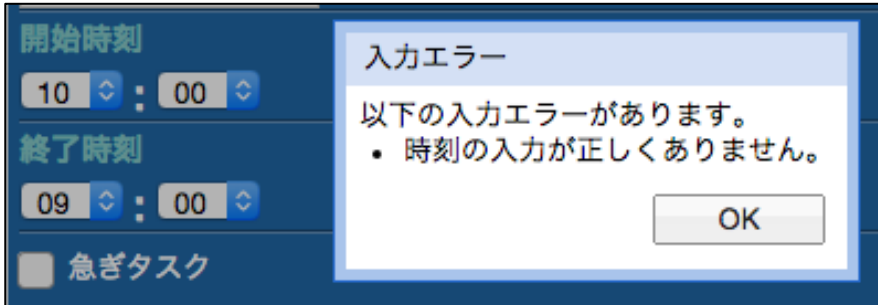


この場合、入力チェックのカスタムバリデーションを使います。JavaScript は以下のようなコードを記載します。

コード	解説
<pre>// 未入力チェック if ((\${開始日} == undefined) (\${終了日} == undefined)) { // 開始日・終了日に未入力のものがあった場合はチェックしない return true; } var from = \${開始日}.split("/"); var to = \${終了日}.split("/"); var fromDate = new Date(Number(from[0]), Number(from[1]) - 1, Number(from[2])); var toDate = new Date(Number(to[0]), Number(to[1]) - 1, Number(to[2])); if (fromDate.getTime() > toDate.getTime()) { return false; } return true;</pre>	<p>IF 文が 2 つ使われています。</p> <p>1 つ目の IF 文は未入力チェックを行っています。未入力の場合「undefined」という値になります。未入力の場合は比較できませんのでチェックは行いません。</p> <p>開始日、終了日(yyyy/MM/dd)を年・月・日に分けます。</p> <p>日付型に変換します。</p> <p>2 つ目の IF 文で大小比較をします。日付型変数の時間を取得して比較を行います</p> <p>return false の場合はエラーとして扱われます。</p>

2.9. 時刻の大小チェック

タイプが時刻の入力フォームで開始時刻が終了時刻より後になった場合はエラーとしたい。



この場合、入力チェックのカスタムバリデーションを使います。JavaScript は以下のようなコードを記載します。

コード	解説
<pre>// 未入力チェック if ((\${開始時刻} == undefined) (\${終了時刻} == undefined)) { // 開始時刻・終了時刻に未入力のものがあった場合はチ // ャックしない return true; } var from = \${開始時刻}.split(":"); var to = \${終了時刻}.split(":"); var fromDate = new Date(1999, 0, 1, from[0], from[1], 0, 0); var toDate = new Date(1999, 0, 1, to[0], to[1], 0, 0); if (fromDate.getTime() > toDate.getTime()) { return false; } return true;</pre>	<p>IF 文が 2 つ使われています。</p> <p>1 つ目の IF 文は未入力チェックを行っています。未入力の場合「undefined」という値になります。未入力の場合は比較できませんのでチェックは行いません。</p> <p>開始時刻、終了時刻(hh:mm)を時・分に分けます。日付型に変換します。1999/1/1 の〇時〇分として比較します。</p> <p>2 つ目の IF 文で大小比較をします。日付型変数の時間を取得して比較を行います</p> <p>return false の場合はエラーとして扱われます。</p> <p>※日付の項目と併せて比較する場合は、1999/1/1 の部分を入力された日付にします。</p>



Gluegent Flow 小技集

2020年10月1日
サイオステクノロジー株式会社

※本書にて記述した製品名および会社名は、各社の商標または登録商標です。
※本書の内容の無断転載および改変を禁止します。
