1．繰り返しwhile　初級　解答手順

＜問題文＞

日本情報処理検定協会

HTML＋JavaScriptを学ぼう

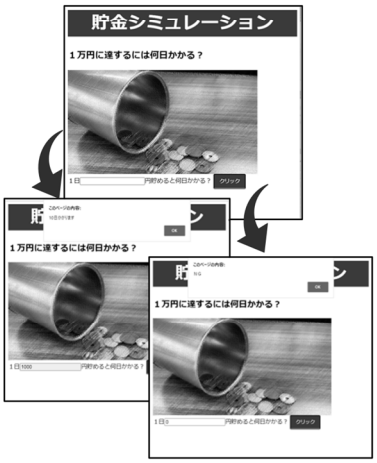
プログラミング問題　繰り返しwhile 初級

＜問　　題＞

1．【繰り返し while 初級】を開いて解答する。

2．［　　］内・網かけ部分は入力文字または値とする。

＜作成ページ例＞



＜処理条件＞

1．【ページ】のブロックセットを挿入しなさい。

2．<title>ブロックの中にページタイトルを入力しなさい。ページタイトルは下記のとおりとする。

［貯金をしよう］

3．<body>ブロックの中に下記の(1)から順にブロックを挿入し、処理をしなさい。

(1)　【見出し１】ブロックを挿入し、下記の文字を入力しなさい。

［貯金シミュレーション］

(2)　【見出し２】ブロックを挿入し、下記の文字を入力しなさい。

［１万円に達するには何日かかる？］

(3)　【画像】ブロックを挿入し、ファイル名を top. jpg にしなさい。

(4)　【フォーム】のブロックセット（ナンバー・ボタン）を挿入しなさい。

4．<head>ブロックの中の<link～>ブロックの下に、下記の指示通りブロックを挿入し、処理をしなさい。

［前提］入力された金額が１万円に達するまでの数（日数）をアラートで表示させる。

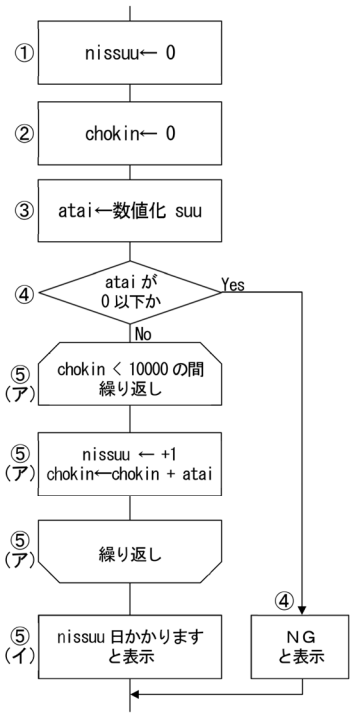
［機能］

・10000 を超えるまで指定された値（フォームのテキストに入力された値）の加算を繰り返し、その回数をカウントする。

・カウントした回数を表示する。

・フォームに０以下の値が入力された場合、「ＮＧ」と表示する。

＜フローチャート＞



(1)　【スクリプト・関数】ブロックを挿入しなさい。

(2)　【部品】内のブロックをすべて用いて、(1)の<function>ブロックの中に下記処理手順どおりに組み立てなさい。

■処理手順

①　日数「nissuu」を０と設定する。

②　貯金額「chokin」を０と設定する。

③　フォームのナンバー（suu）の値を数値にして「atai」に設定する。

④　もし「atai」が０以下の場合、「ＮＧ」の文字を表示する。

⑤　それ以外の場合

（ア）貯金額「chokin」が 10000 未満の間、下記の処理を繰り返す。

日数「nissuu」に１を足す。

貯金額「chokin」と「atai」を足して貯金額「chokin」に入れる。

（イ）日数「nissuu」の値と「日かかります」の文字を表示する。

5．プレビュー画面で確認しなさい。

教材サイトからアプリを開く

Javloc 教材サイトから[初級プロジェクト]の[繰り返し「while」初級]をクリック。

<処理条件>1～3. では、ページの HTML 部分を作成し、4. では JavaScript を作成します。

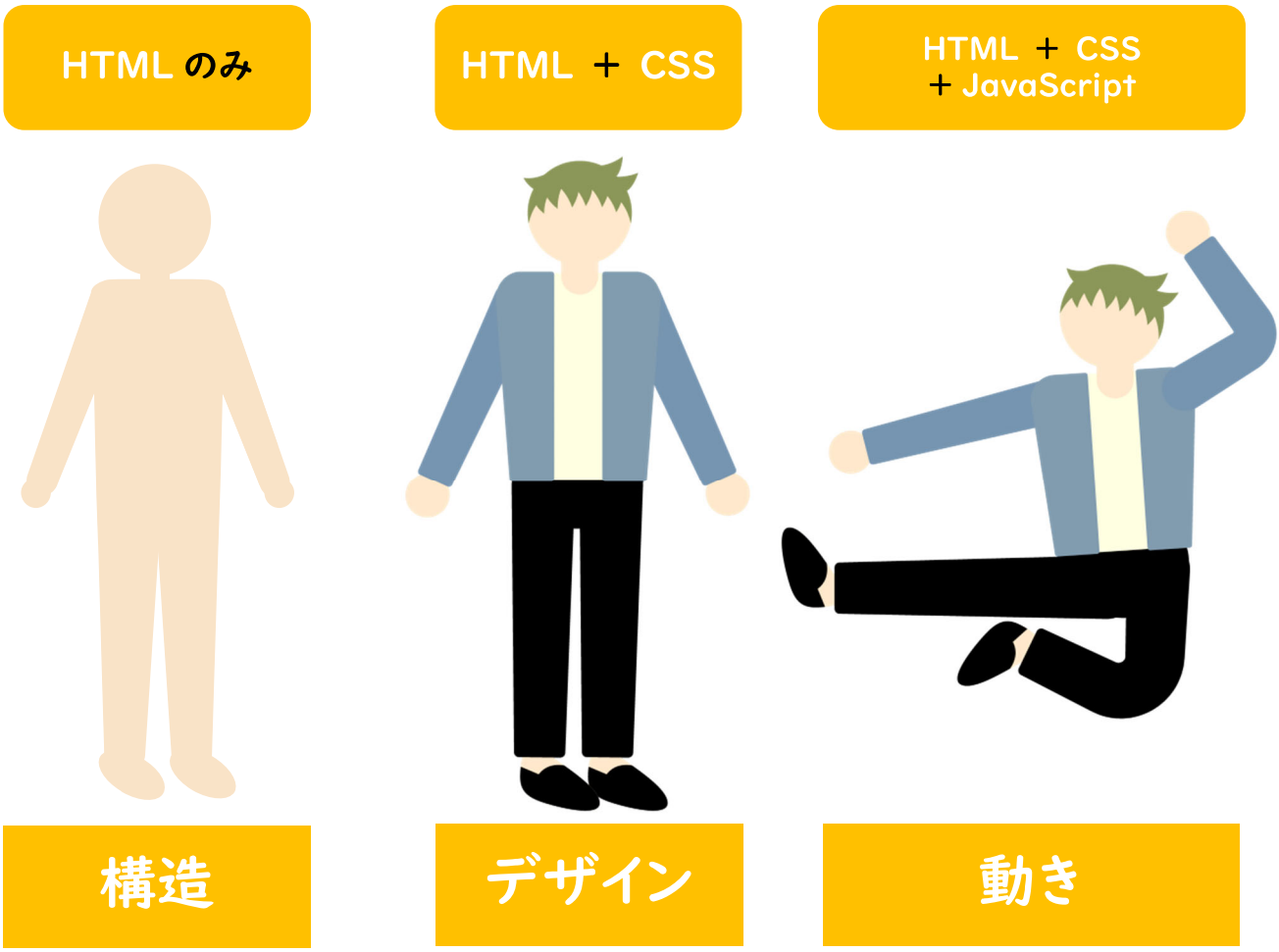
1～3

表示されている文字や画像、情報を入力する場所（入力欄やボタンなど）を HTML で設定します。

4

入力された情報を受け取り、結果を返すといったような表示上は見えない「動き」の仕組みを JavaScript で設定します。

HTML と CSS と JavaScript の役割 イメージ図

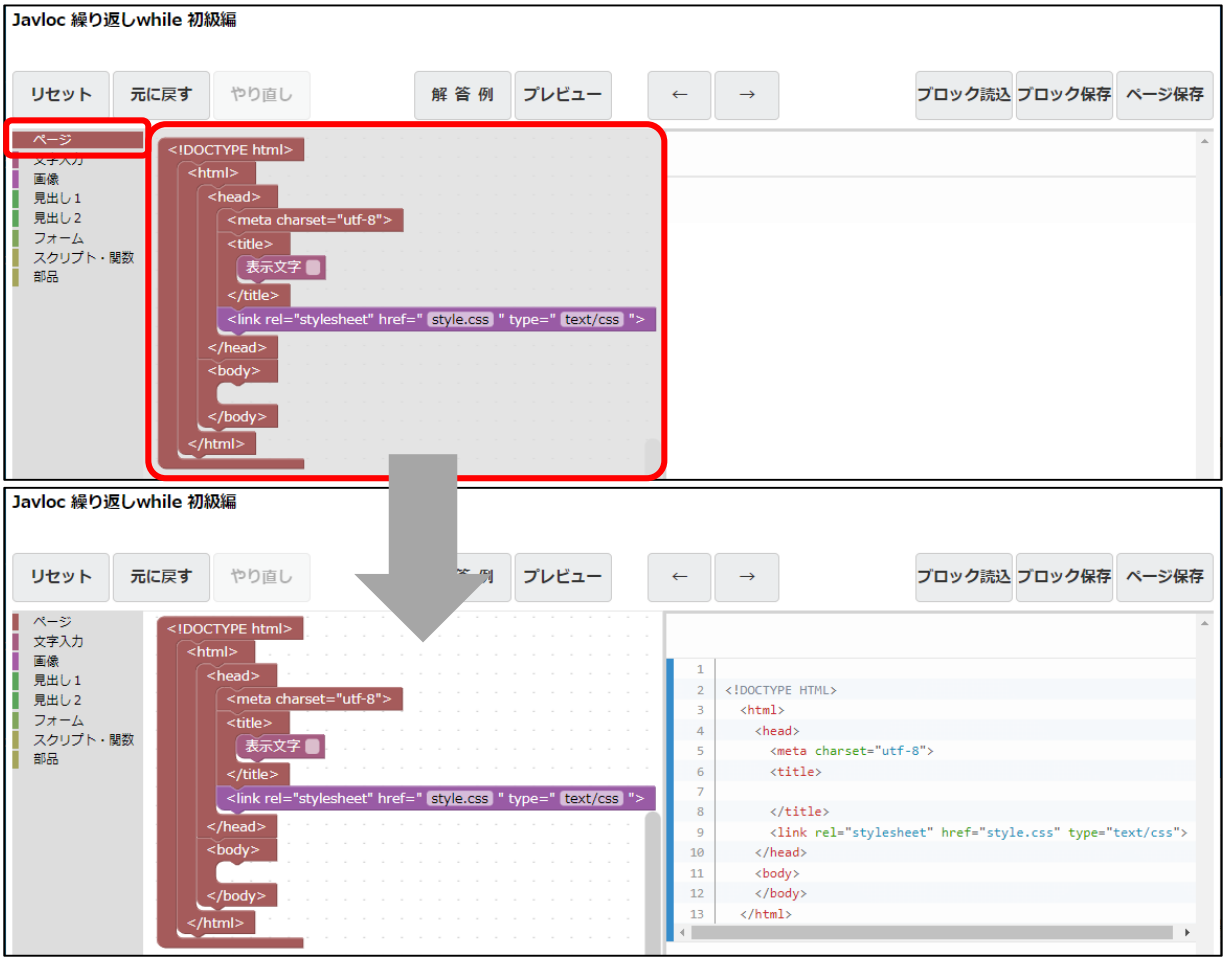


ページを構成（HTML 部分の作成）

1．【ページ】のブロックセットを挿入しなさい。

リスト内[ページ]をクリック。

枠内にあるブロックセットをドラッグし、編集エリアにドロップ。



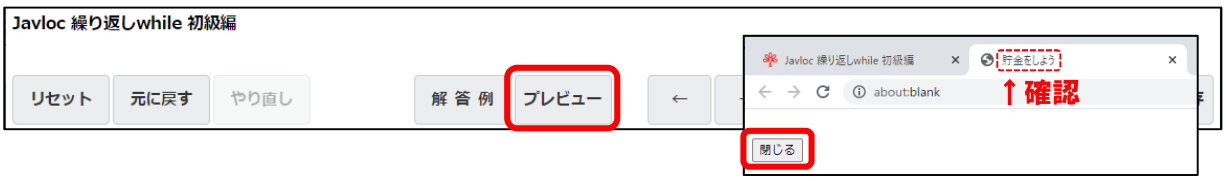
2．〈title〉ブロックの中にページタイトルを入力しなさい。ページタイトルは下記のとおりとする。

貯金をしよう

<title>ブロックの表示文字ブロックの空欄に「貯金をしよう」と入力。

入力後、ソース表示エリアでページタイトルが設定されていることを確認。

[プレビュー]ボタンをクリックして確認。[閉じる]ボタンをクリックしブロック操作画面へ戻る。



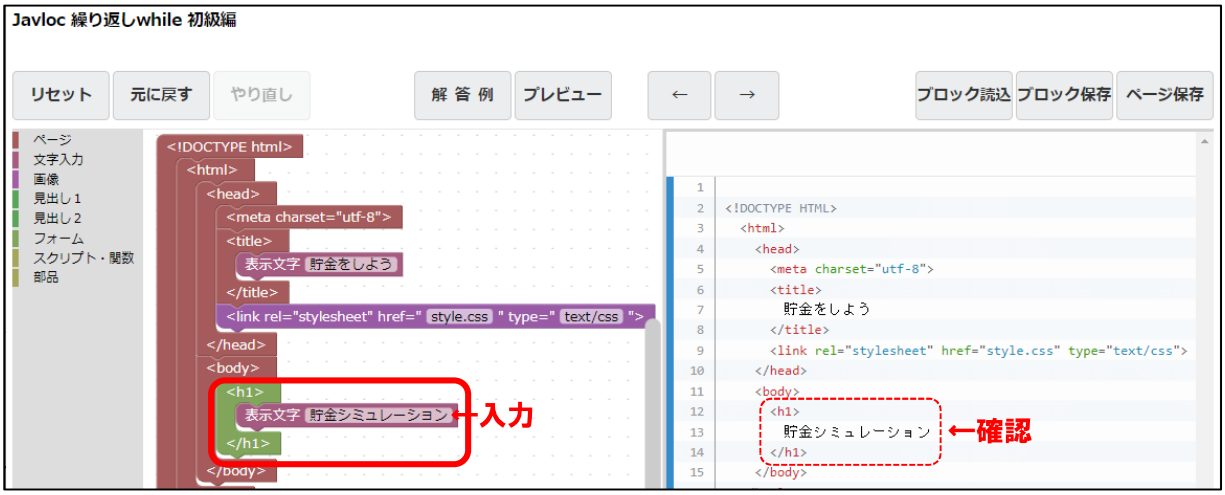
３．<body>ブロックの中に下記の(1)から順にブロックを挿入し、処理をしなさい。

(1) 【見出し１】ブロックを挿入し、下記の文字を入力しなさい。

貯金シミュレーション

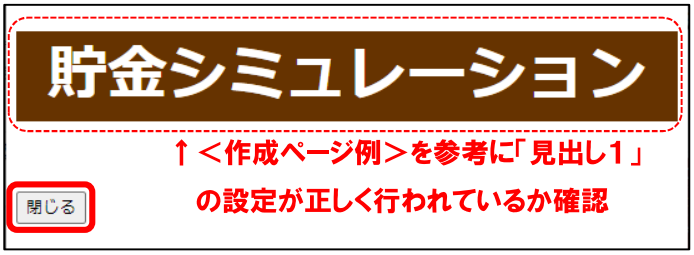
リスト内[見出し１]をクリック、枠内にある<h1>ブロックをドラッグし、<body>ブロックの中にドロップ。

表示文字ブロックの空欄に「貯金シミュレーション」と入力し、ソース表示エリアで見出し１が設定されていることを確認。

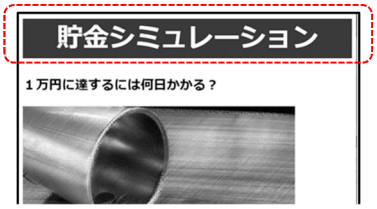


[プレビュー]ボタンをクリックし、正しく設定されているか確認。

[閉じる]ボタンをクリックし、ブロック操作画面へ戻る。



＜作成ページ例＞



見出し１の背景色が茶色で、文字色が白色になっているのは、CSS で既に設定しているためです。

(2) 【見出し２】ブロックを挿入し、下記の文字を入力しなさい。

１万円に達するには何日かかる？

リスト内[見出し２]をクリック、枠内にある<h2>ブロックをドラッグし、<h1>ブロックの下にドロップ。

表示文字ブロックの空欄に「１万円に達するには何日かかる？」と入力し、ソース表示エリアで見出し２が設定されていることを確認。

[プレビュー]ボタンをクリックし、正しく設定されているか確認。

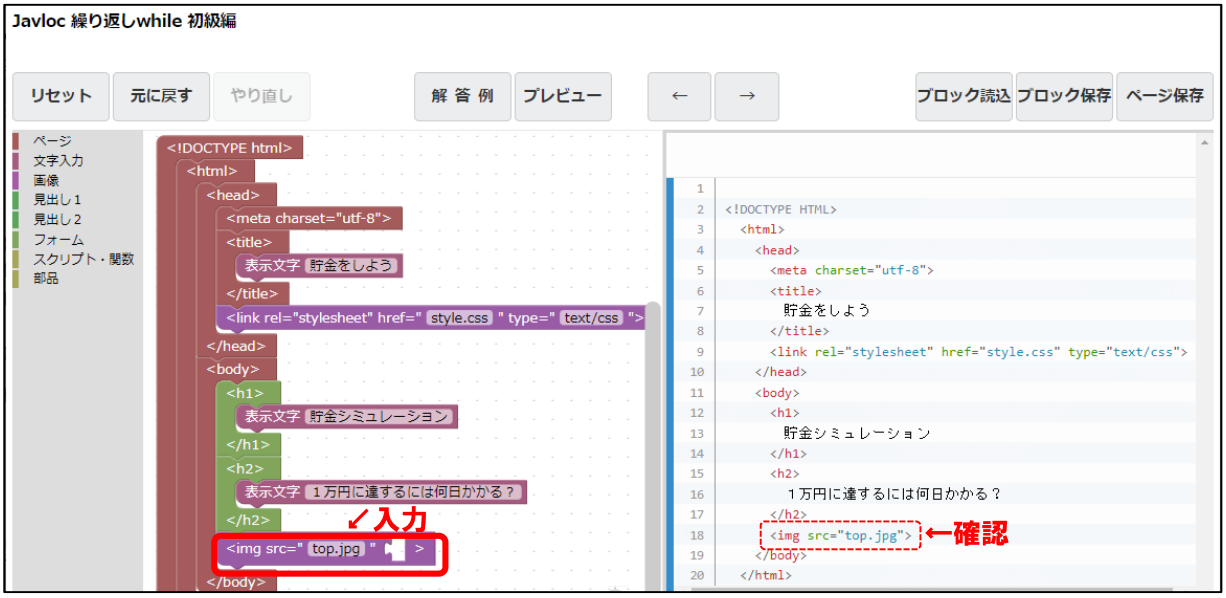
[閉じる]ボタンをクリックし、ブロック操作画面へ戻る。

＜作成ページ例＞

(3) 【画像】ブロックを挿入し、ファイル名を top. jpg にしなさい。

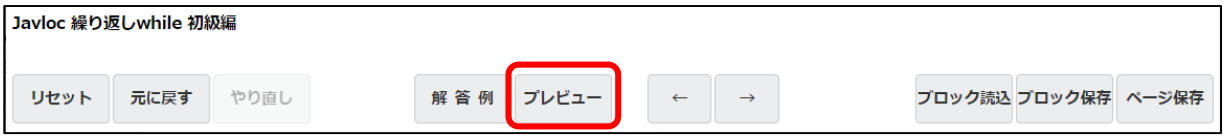
リスト内[画像]をクリック、枠内にある<img>ブロックをドラッグし<h2>ブロックの下にドロップ。 “”の空欄に「top.jpg」と入力し、ソース表示エリアで画像が設定されていることを確認。

※必ず半角で入力しましょう。



[プレビュー]ボタンをクリックし、正しく設定されているか確認。

[閉じる]ボタンをクリックし、ブロック操作画面へ戻る。

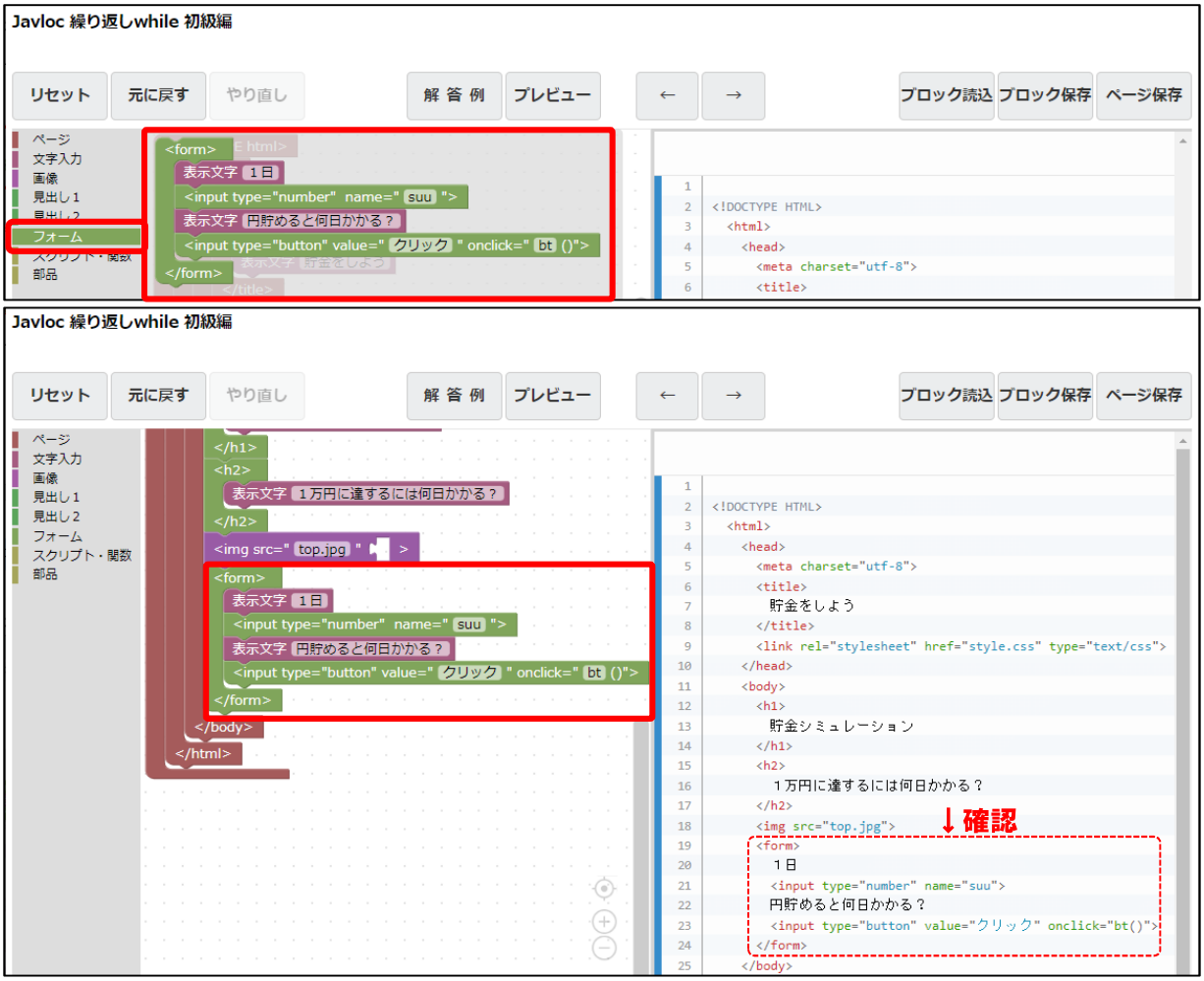


＜作成ページ例＞

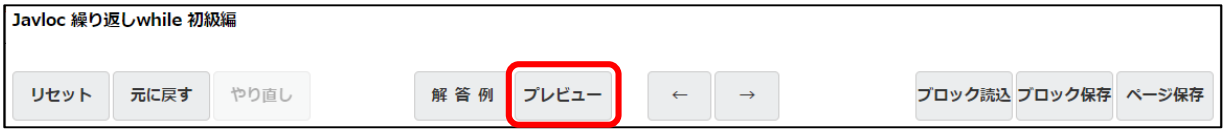
←＜作成ページ例＞を参考に「画像」の設定が正しく行われているか確認

(4) 【フォーム】のブロックセット（ナンバー・ボタン）を挿入しなさい。

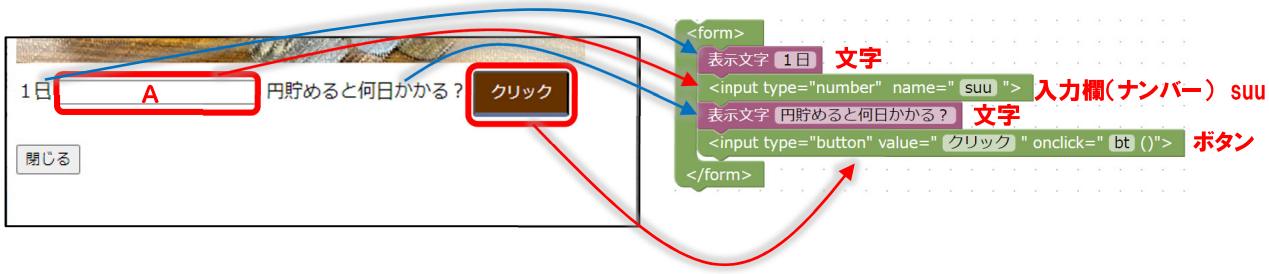
リスト内[フォーム]をクリック、枠内にある<form>ブロックセットをドラッグし<img>ブロックの下にドロップ。ソース表示エリアでフォームが設定されていることを確認。



[プレビュー]ボタンをクリックし、＜作成ページ例＞を参考に「フォーム」の設定が正しく行われているか確認。[閉じる]ボタンをクリックし、ブロック操作画面へ戻る。



まだ JavaScript が設定されていないため、A の入力欄（ナンバー）に数を入力し「クリック」ボタンを押しても、何も表示されません。この後設定する JavaScript によって、「クリック」ボタンを押すと入力された数を使って処理が実行され、結果が表示されるようになります。

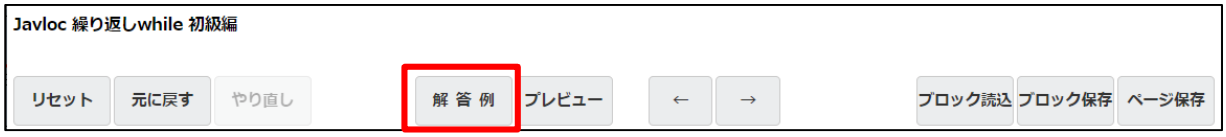


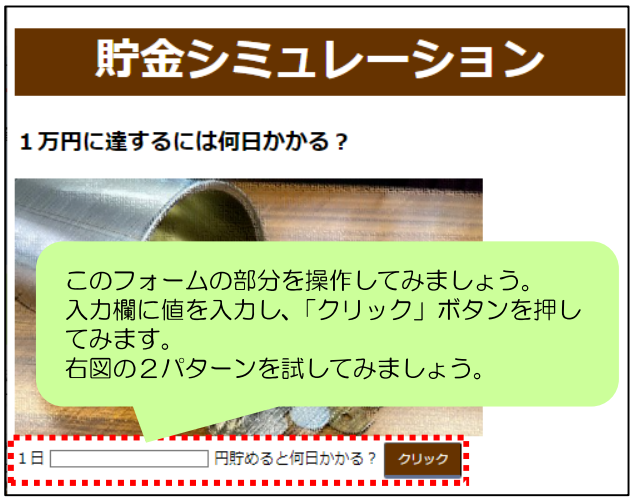
4．<head>ブロックの中の<link～>ブロックの下に、下記の指示通りブロックを挿入し、処理をしなさい。

［前提］入力された金額が１万円に達するまでの数（日数）をアラートで表示させる。
［機能］
・10000 を超えるまで指定された値（フォームのテキストに入力された値）の加算を繰り返し、その回数をカウントする。
・カウントした回数を表示する。
・フォームに 0 以下の値が入力された場合、「ＮＧ」と表示する。

まだこの時点では前提や機能の意味が分からなくても、解答例のページを確認し、「こんな動きのプログラムを作成するんだな」と想像するくらいで問題ありません。

まず、[前提]と[機能]を読み、「解答例」ボタンをクリックし、解答例ページの動きを確認します。





0 と入力

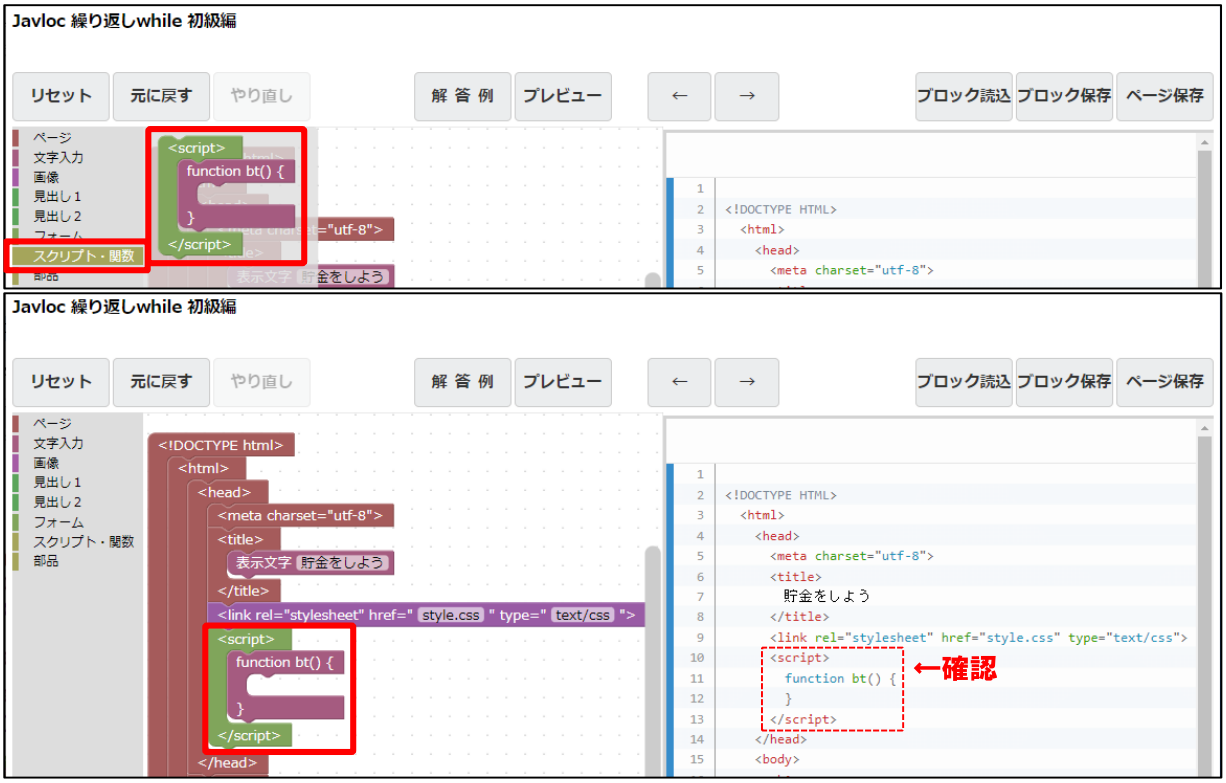
＜入力した値が 0 の場合＞
アラートメッセージに「ＮＧ」と出力する。

＜それ以外の場合＞
10000 を超えるまで
入力した値の加算を繰り返して
その回数をカウントし
アラートメッセージにカウントした回数を出力する。

1000 と入力

(1) 【スクリプト・関数】ブロックを挿入しなさい。

リスト内[スクリプト・関数]をクリック、枠内にある<script>ブロックセットをドラッグし、<link>ブロックの下にドロップ。ソース表示エリアでスクリプトが設定されていることを確認。



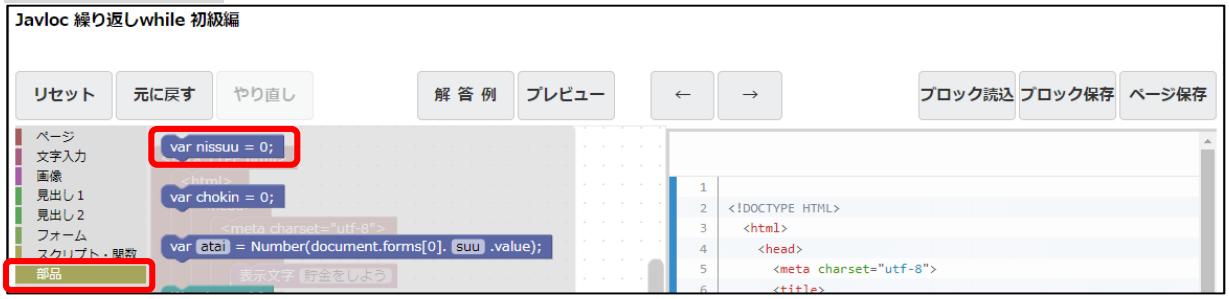
<script> ～ </script> とは、JavaScript などのスクリプトを文書内に埋め込んだり外部スクリプトを読み込んだりするために使用するものです。

function □（ ）とは、ユーザーが作成する関数を定義する JavaScript です。□部分に関数名を記述します。これは、さまざまな処理を一つにまとめて、名前を付けることができるものです。

(2) 【部品】内のブロックをすべて用いて、(1)の<function>ブロックの中に下記処理手順どおりに組み立てなさい。

■処理手順
① 日数「nissuu」を０と設定する。

リスト内[部品]をクリック、枠内にある var nissuu = 0; ブロックを選択し、ドラッグして function bt (){～}ブロックの中にドロップ。



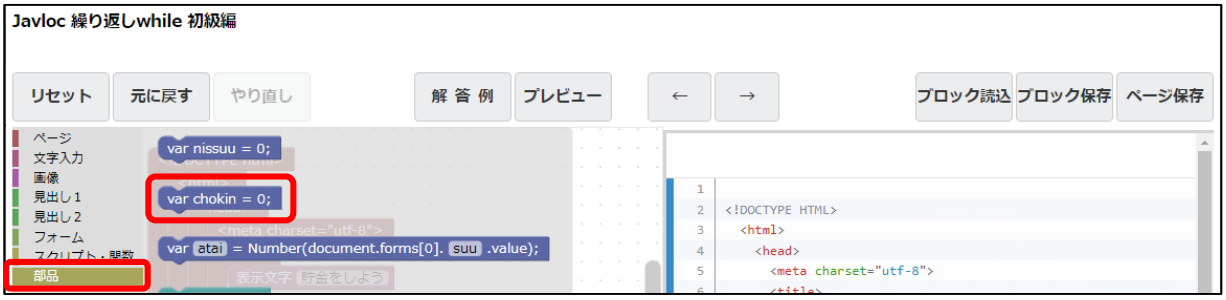
「var」というのは、変数を新規に作りますという意味です。
変数とは、値や文字列を保管しておく箱のようなものです。
「＝」は**代入演算子**と呼ばれ、右側に書かれる値や文字列をこの箱に入れるという意味になります。

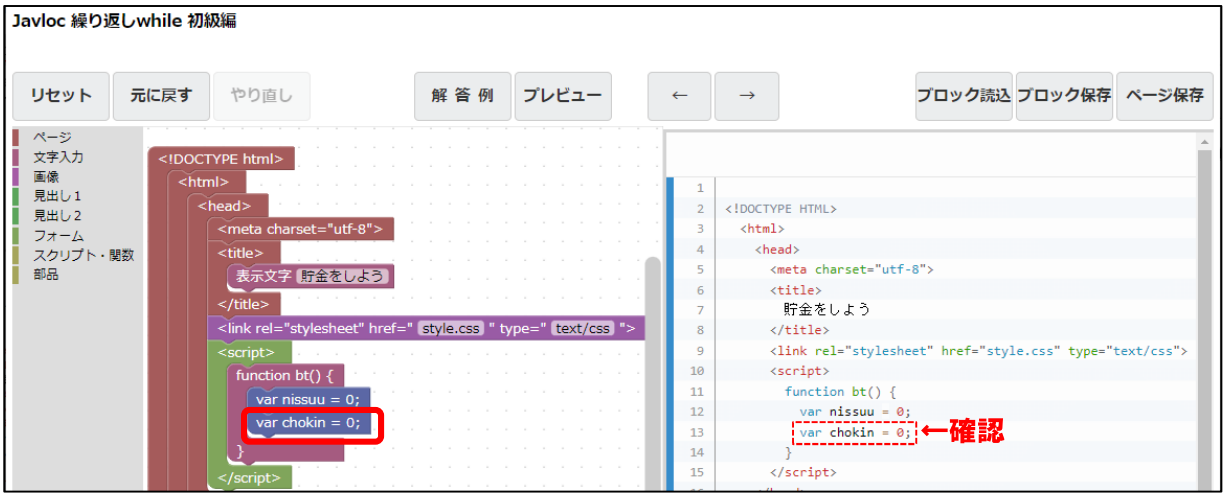
比較演算子「＝＝」と間違えやすいので注意しましょう。

ここでは、変数「nissuu」を新規に作成し、そこに「0」という値を入れています。

② 貯金額「chokin」を０と設定する。

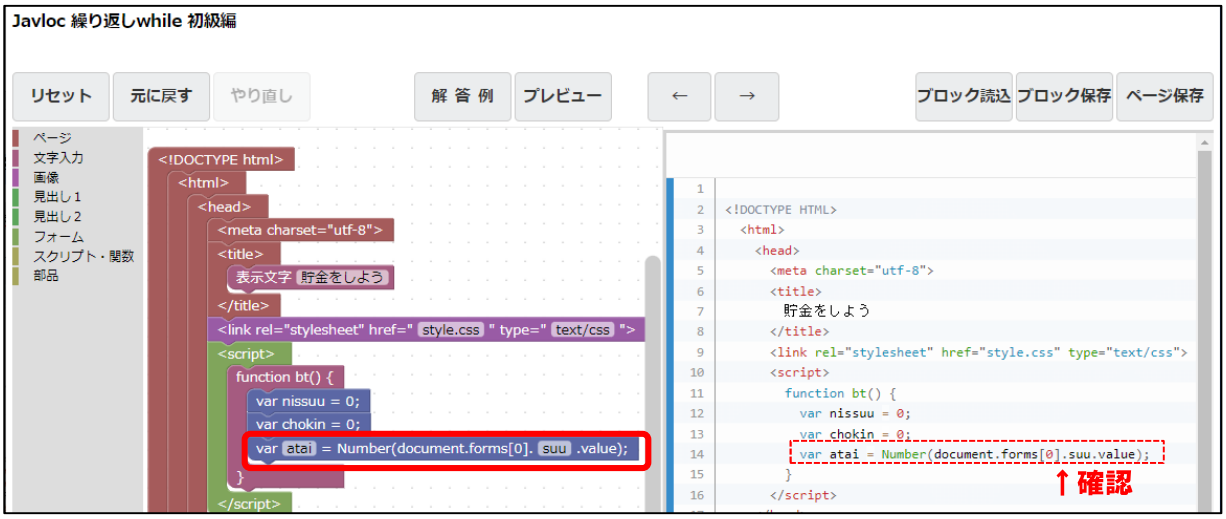
リスト内[部品]をクリック、枠内にある `var chokin = 0;` ブロックを選択し、ドラッグして①で設定したブロックの下にドロップ。





③ フォームのナンバー（suu）の値を数値にして「atai」に設定する。

リスト内[部品]をクリック、枠内にある `var atai = Number(document.forms[0].suu.value);` ブロックを選択し、ドラッグして②で設定したブロックの下にドロップ。



「Number(～)」は、（～）内にある値を**数値に変換**します。
「document.forms[0]」は、今作成している Web ページに最初に登場するフォームのことを指しています。（この問題では、処理条件３－(4)で挿入した<form>ブロックセットのことです。）

ブロックの中身は

`var atai = Number(document.forms[0].suu.value);`

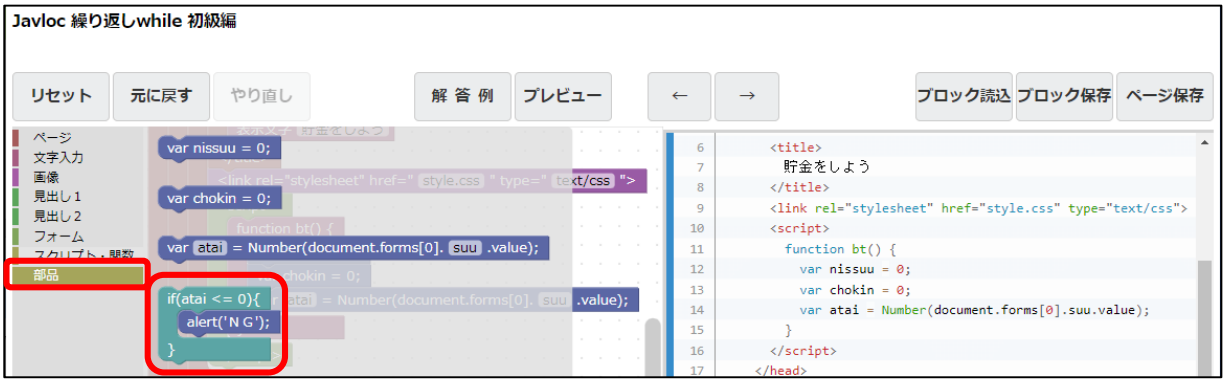
なので、これは何をしているかというと、

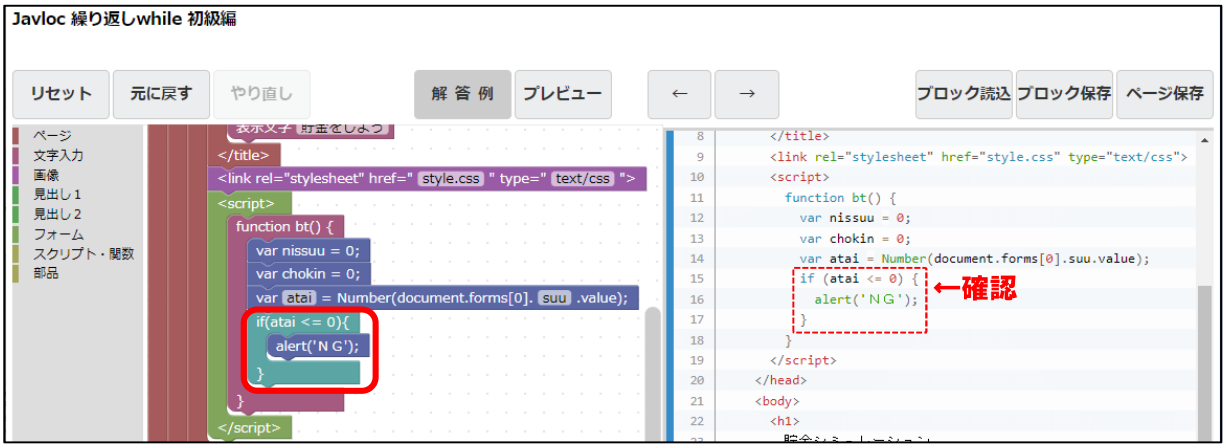
フォームの「suu」の値を、数値に変換して、「atai」という箱（変数）に入れておく。

ということになります。

④ もし「atai」が０以下の場合、「ＮＧ」の文字を表示する。

リスト内[部品]をクリック、枠内にある if (atai <= 0) {～} ブロックを選択し、ドラッグして③で設定したブロックの下にドロップ。





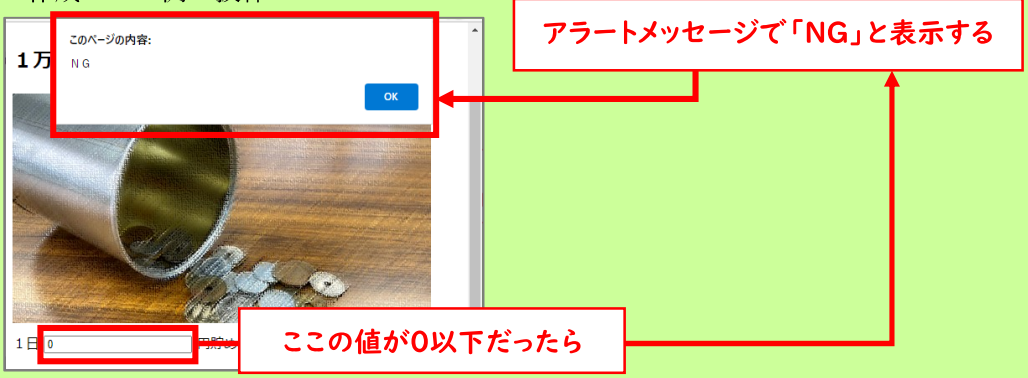
ブロックの中身を見てみましょう。

```
if (atai <= 0) {
  alert('NG');
}
```

これを日本語で読み下すと

```
もし（ 変数 atai が 0 以下だったら）{
    アラートメッセージで表示（「NG」）
}
```

＜作成ページ例　抜粋＞

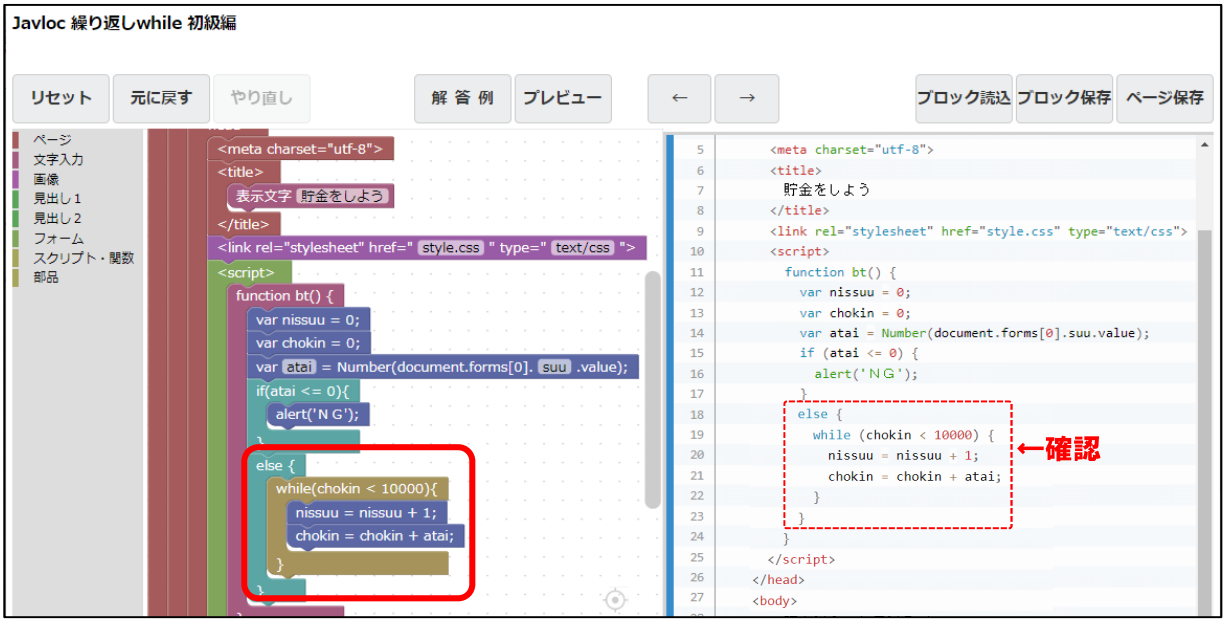


上記のように「もし（A）だったら、{B} する」という条件式は「if(A) {B}」と記述します。

⑤ それ以外の場合

（ア）貯金額「chokin」が 10000 未満の間、下記の処理を繰り返す。
日数「nissuu」に 1 を足す。
貯金額「chokin」と「atai」を足して貯金額「chokin」に入れる。

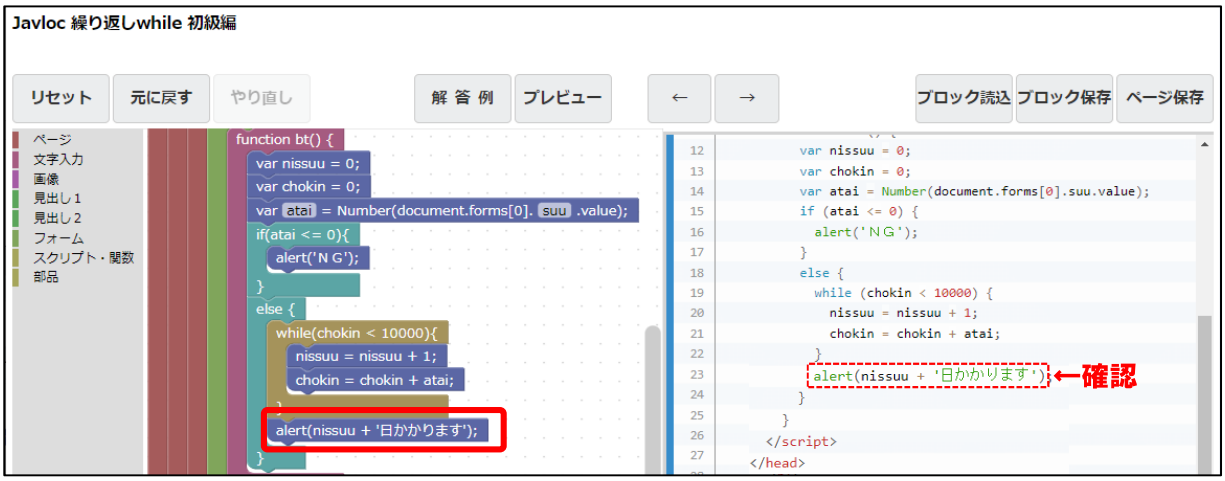
リスト内[部品]をクリック、枠内にある else {～} ブロックセットを選択し、ドラッグして④で設定したブロックの下にドロップ。



（イ）日数「nissuu」の値と「日かかります」の文字を表示する。

リスト内[部品]をクリック、枠内にある alert(nissuu + '日かかります'); ブロックを選択し、ドラッグ。

（ア）で設定した<else>ブロックの中、<while>ブロックの下にドロップ。



ブロックの中身を見てみましょう。

```
else {
  while (chokin < 10000) {
      nissuu = nissuu + 1;
      chokin = chokin + atai;
  }
  alert (nissuu + '日かかります');
}
```

これを日本語で読み下すと

```
それ以外の場合 {
  繰り返す（変数 chokin が 10000 より小さい間） {
    変数 nissuu に 1 を足す ←― 日数をカウントする箱
    変数 chokin に変数 atai を足す ←― 貯金箱
  }
  アラートメッセージで表示する（変数 nissuu の値と「日かかります」の文字）
}
```

今回は if と else を分解して説明していますが、これはひとかたまりの条件分岐です。

もし 0 以下が入力されたら

→「NG」とアラートメッセージを表示する。

それ以外だったら

→貯金箱の中身が 10000 円より小さい間

・日数を数える箱に 1 日足す

・貯金箱に 1 日分のお金を入れる

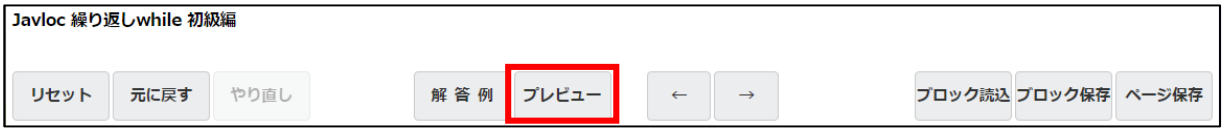
を繰り返し、10000 円貯まるまでに何日かかるか（変数 nissuu の中身）を求める。

そして、その変数 nissuu の値と「日かかります」をアラートメッセージで表示する。

問題文のフローチャートと一緒に確認してみましょう。

５．プレビュー画面で確認しなさい。

[プレビュー]ボタンをクリックし、動作を確認。



４．で解答例を基に確認した内容と同様の動作になっているか見てみましょう。

2．繰り返しwhile　中級　解答手順

＜問題文＞

日本情報処理検定協会

HTML＋JavaScriptを学ぼう

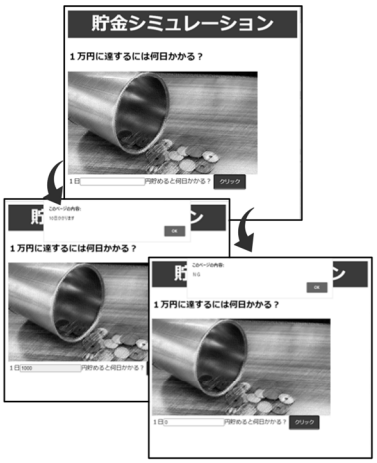
プログラミング問題　繰り返しwhile 中級

＜問　　題＞

1．【繰り返しwhile 中級】を開いて解答する。

2．［　　］内・網かけ部分は入力文字または値とする。

＜作成ページ例＞



＜処理条件＞

1．【ページ】のブロックセットを挿入しなさい。

2．<title>ブロックの中にページタイトルを入力しなさい。ページタイトルは下記のとおりとする。

［貯金をしよう］

3．<body>ブロックの中に下記の(1)から順にブロックを挿入し、処理をしなさい。

(1)　【見出し１】ブロックを挿入し、下記の文字を入力しなさい。

［貯金シミュレーション］

(2)　【見出し２】ブロックを挿入し、下記の文字を入力しなさい。

［１万円に達するには何日かかる？］

(3)　【画像】ブロックを挿入し、ファイル名をtop.jpgにしなさい。

(4)　【フォーム】から<form>ブロックを選択・挿入し、その中に下記のブロックを挿入し、処理をしなさい。

①　【文字入力】ブロックを挿入し、下記の文字を入力しなさい。

［１日］

②　【フォーム】からナンバーのブロックを選択・挿入しなさい。nameはsuu。

③　【文字入力】ブロックを挿入し、下記の文字を入力しなさい。

［円貯めると何日かかる？］

④　【フォーム】からボタンのブロックを選択・挿入しなさい。valueはクリック、onclickは関数名bt。

4．<head>ブロックの中の<link～>ブロックの下に、下記の指示通りブロックを挿入し、処理をしなさい。

［前提］入力された金額が１万円に達するまでの数（日数）をアラートで表示させる。

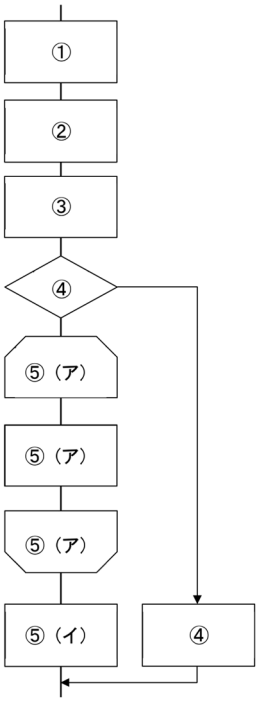
［機能］

・10000を超えるまで指定された値（フォームのテキストに入力された値）の加算を繰り返し、その回数をカウントする。

・カウントした回数を表示する。

・フォームのナンバーに０以下の値が入力された場合、「ＮＧ」と表示する。

＜フローチャート＞



(1)　【スクリプト・関数】ブロックを挿入し、３－(4)－④で指定した関数名を入力しなさい。

(2)　【部品】内から正しいブロックを選択し、(1)の関数ブロックの中に下記処理手順どおりに組み立てなさい。

■処理手順

①　変数nissuuを宣言し、0を代入

②　変数chokinを宣言し、0を代入

③　変数ataiを宣言し、フォームのナンバー（suu）の値を数値に変換し代入

④　変数ataiが０以下の場合、「ＮＧ」と表示

⑤　それ以外の場合

（ア）chokinが10000未満の間、下記の処理を繰り返す

nissuuに１を加算

chokinにataiを加算し、chokinに代入

（イ）「〇日かかります」と表示　※〇はnissuuの値

5．プレビュー画面で確認しなさい。

教材サイトからアプリを開く

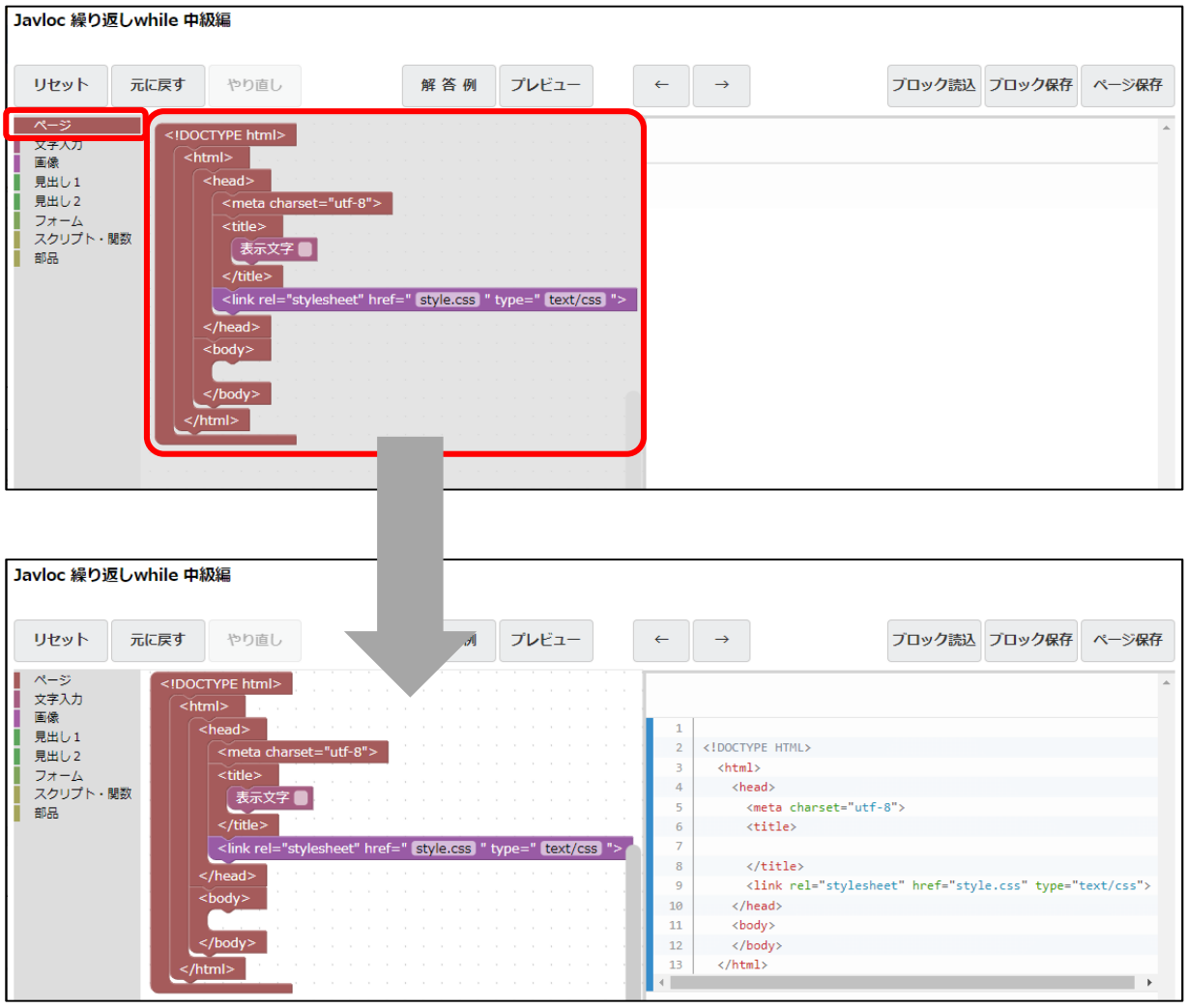
Javloc 教材サイトから[中級プロジェクト]の[繰り返し「while」中級]をクリック。

ページを構成（HTML 部分の作成）

1．【ページ】のブロックセットを挿入しなさい。

リスト内[ページ]をクリック。

枠内にあるブロックセットをドラッグし、編集エリアにドロップ。



2．<title>ブロックの中にページタイトルを入力しなさい。ページタイトルは下記のとおりとする。

貯金をしよう

<title>ブロックの表示文字ブロックの空欄に「貯金をしよう」と入力。

入力後、ソース表示エリア・プレビューでページタイトルが設定されていることを確認。

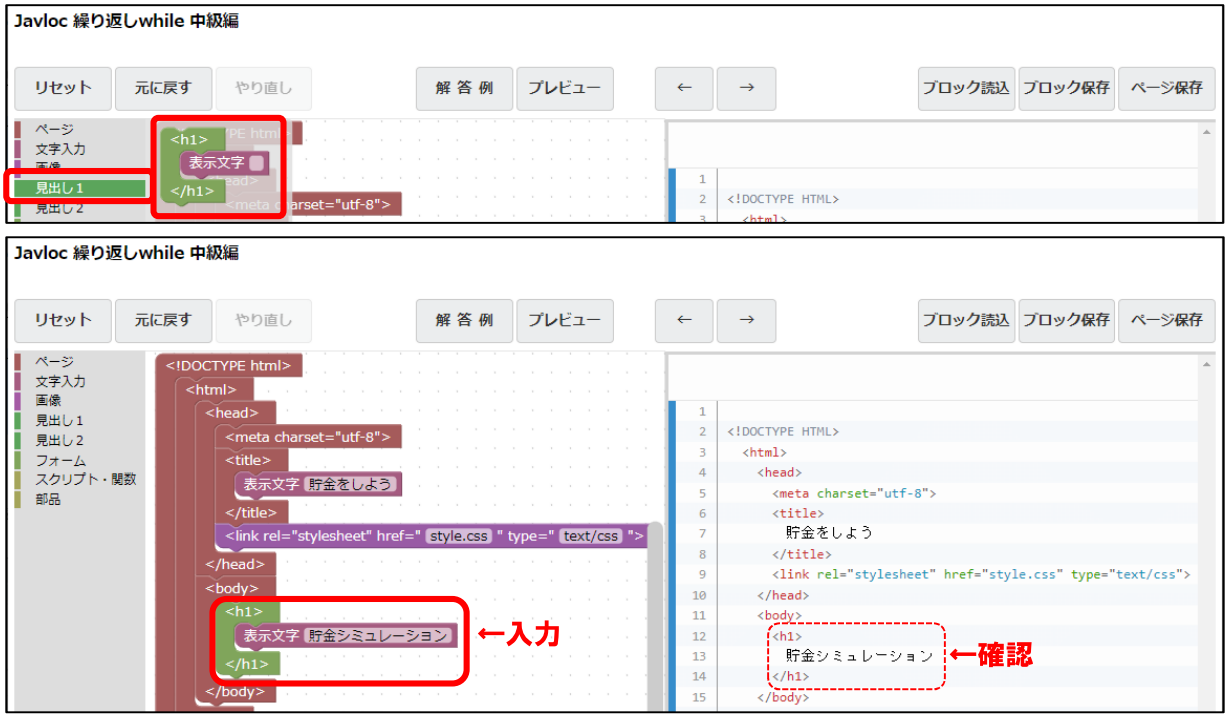
3．<body>ブロックの中に下記の(1)から順にブロックを挿入し、処理をしなさい。

(1) 【見出し１】ブロックを挿入し、下記の文字を入力しなさい。

貯金シミュレーション

リスト内[見出し１]をクリック、枠内にある<h1>ブロックをドラッグし、<body>ブロックの中にドロップ。

表示文字ブロックの空欄に「貯金シミュレーション」と入力し、ソース表示エリア・プレビューで見出し１が設定されていることを確認。



(2) 【見出し２】ブロックを挿入し、下記の文字を入力しなさい。

１万円に達するには何日かかる？

リスト内[見出し２]をクリック、枠内にある<h2>ブロックをドラッグし<h1>ブロックの下にドロップ。

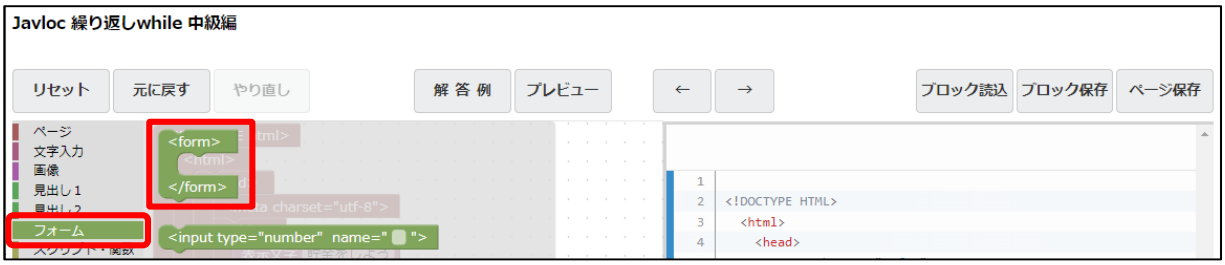
表示文字ブロックの空欄に「１万円に達するには何日かかる？」と入力し、ソース表示エリア・プレビューで見出し２が設定されていることを確認。

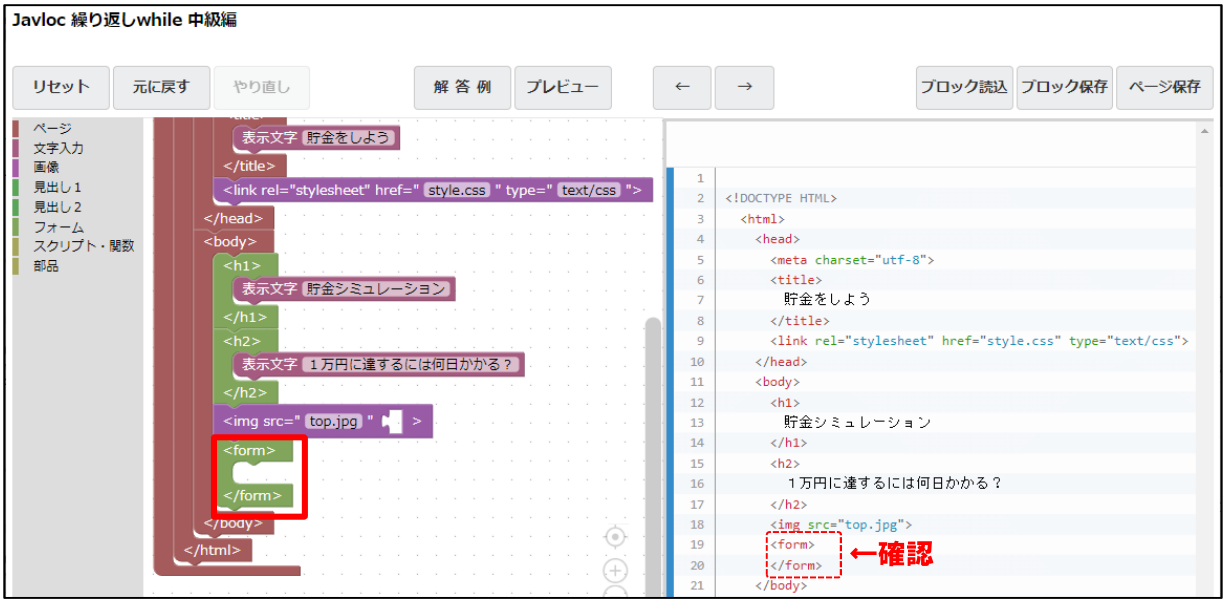
(3) 【画像】ブロックを挿入し、ファイル名を top. jpg にしなさい。

リスト内[画像]をクリック、枠内にある<img>ブロックをドラッグし<h2>ブロックの下にドロップ。 “”の空欄に「top.jpg」と入力し、ソース表示エリア・プレビューで画像が設定されていることを確認。※必ず半角で入力しましょう。

(4) 【フォーム】から<form>ブロックを選択・挿入し、その中に下記のブロックを挿入し、処理をしなさい。

リスト内[フォーム]をクリック、枠内にある<form>ブロックをドラッグし<img>ブロックの下にドロップ。ソース表示エリアでフォームが設定されていることを確認。





<form>タグは、**入力欄や送信ボタンを作成する際に使用する要素**です。ここで入力された情報はプログラムを使用して Web サーバーへ送信することができるので、アンケートや会員登録などのページで利用されています。

本教材では、ここに入力された情報を JavaScript でコントロールします。

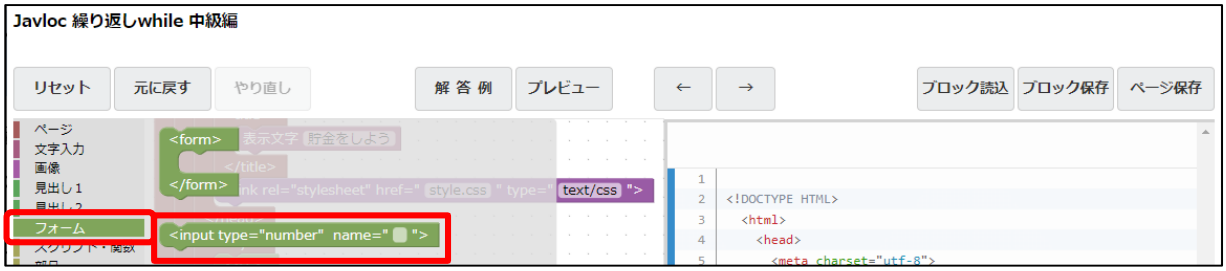
手順通りに完成した後に動きを確認すると分かりやすいので、今の時点では、この(4)で組み立てたブロックが「情報を入力する場所」になるというイメージを持っていれば大丈夫です。

① 【文字入力】ブロックを挿入し、下記の文字を入力しなさい。
１日

リスト内[文字入力]をクリック、枠内にある表示文字ブロックをドラッグし、<form>ブロックの中にドロップ。空欄に「枚」と入力し、ソース表示エリア・プレビューで文字が設定されていることを確認。
※表示される文字の設定なので全角入力でも問題ありません。

② 【フォーム】からナンバーのブロックを選択・挿入しなさい。name は suu。

リスト内[フォーム]をクリック、枠内にある<input type="number" ～>ブロックをドラッグし、表示文字ブロックの下にドロップ。空欄に「suu」と入力し、ソース表示エリア・プレビューでボタンが設定されていることを確認。※必ず半角で入力しましょう。



③　【文字入力】ブロックを挿入し、下記の文字を入力しなさい。
円貯めると何日かかる？

リスト内[文字入力]をクリック、枠内にある表示文字ブロックをドラッグし、
<input type="number"　～>ブロックの下にドロップ。空欄に「円貯めると何日かかる？」と入力し、
ソース表示エリア・プレビューで文字が設定されていることを確認。

④　【フォーム】からボタンのブロックを選択・挿入しなさい。value はクリック、onclick は関数名 bt。

リスト内[フォーム]をクリック、枠内にある<input type="button" ～>ブロックをドラッグし、表示文字ブロックの下にドロップ。value の“”空欄に「クリック」、onclick の“”空欄に「bt」と入力し、ソース表示エリア・プレビューでボタンが設定されていることを確認。

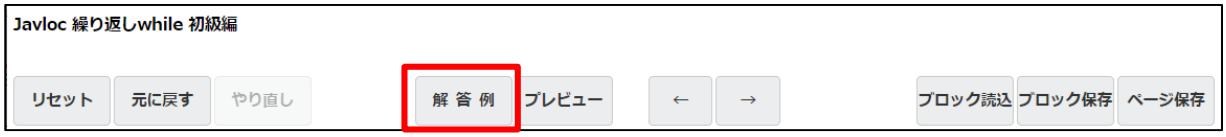
※bt は必ず半角で入力しましょう。

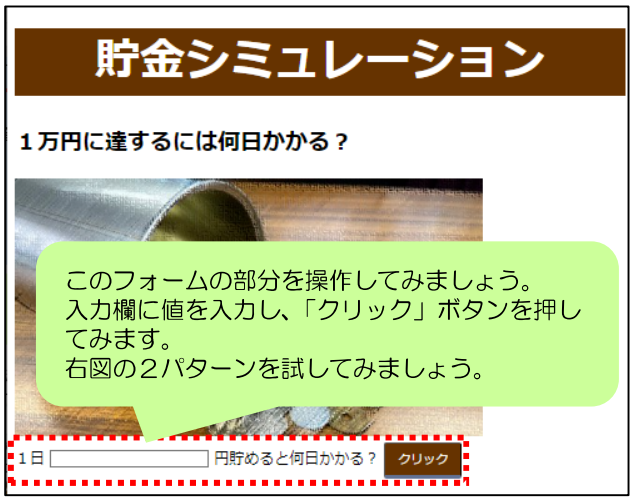
4．<head>ブロックの中の<link～>ブロックの下に、下記の指示通りブロックを挿入し、処理をしなさい。

［前提］入力された金額が１万円に達するまでの数（日数）をアラートで表示させる。
［機能］
・10000 を超えるまで指定された値（フォームのテキストに入力された値）の加算を繰り返し、その回数をカウントする。
・カウントした回数を表示する。
・フォームに 0 以下の値が入力された場合、「ＮＧ」と表示する。

まだこの時点では前提や機能の意味が分からなくても、解答例のページを確認し、「こんな動きのプログラムを作成するんだな」と想像するくらいで問題ありません。

まず、[前提]と[機能]を読み、「解答例」ボタンをクリックし、解答例ページの動きを確認します。





0 と入力

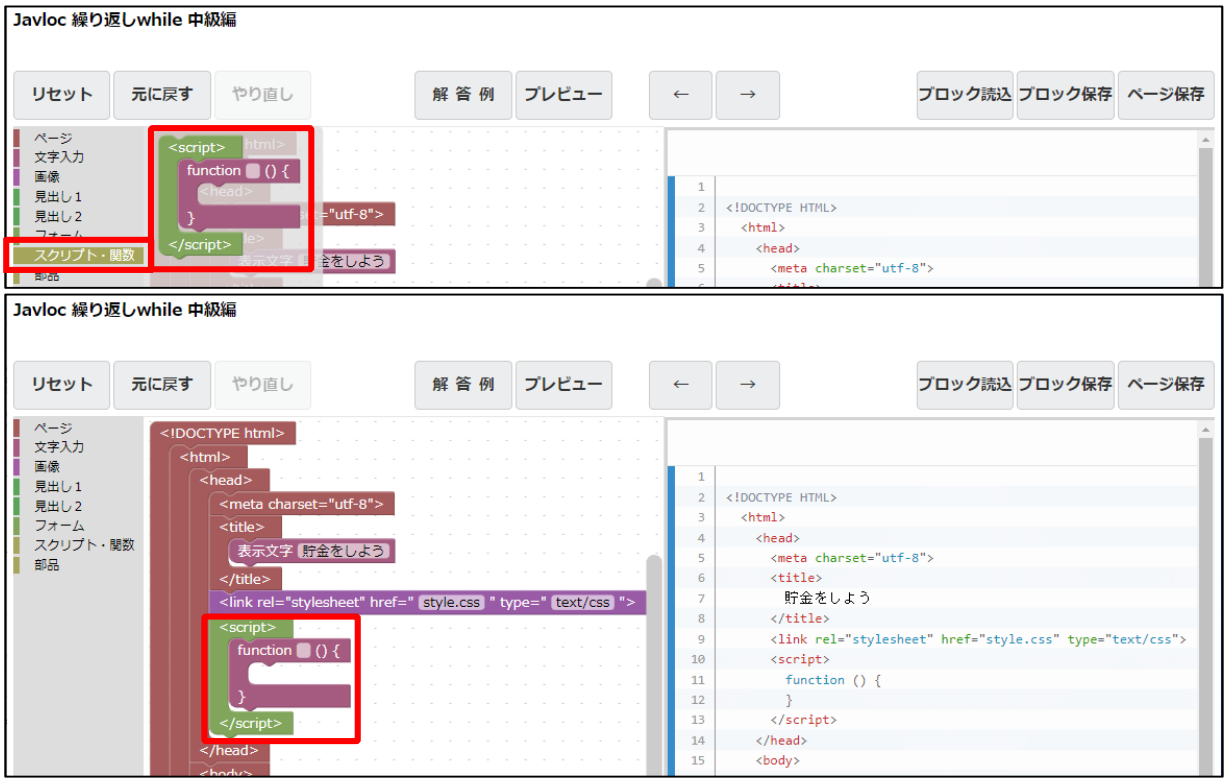
＜入力した値が 0 の場合＞
アラートメッセージに「ＮＧ」と出力する。

＜それ以外の場合＞
10000 を超えるまで
入力した値の加算を繰り返して
その回数をカウントし
アラートメッセージにカウントした回数を出力する。

1000 と入力

(1) 【スクリプト・関数】ブロックを挿入し、3－(4)－④で指定した関数名を入力しなさい。

リスト内[スクリプト・関数]をクリック、枠内にある<script>ブロックセットをドラッグし、<link>ブロックの下にドロップ。



3－(4)－④で挿入・入力したボタンのブロック <input type="button" value="クリック" onclick="bt()"> にある **onclick="()"** を参照し、**関数名**が「**bt**」だと確認します。

関数名「bt」を空欄に入力。※必ず半角で入力しましょう。

ソース表示エリアでスクリプト・関数が設定されていることを確認。

function □( )とは、関数を定義する JavaScript です。□部分に関数名を記述します。これは、さまざまな処理を一つにまとめて、名前を付けることができるものです。

「function bt (){～}」となるように入力します。

これで、クリックボタンが押されたら（onclick）、関数（bt）が処理されるという仕組みになります。

これから、その関数内の処理を組み立てます。

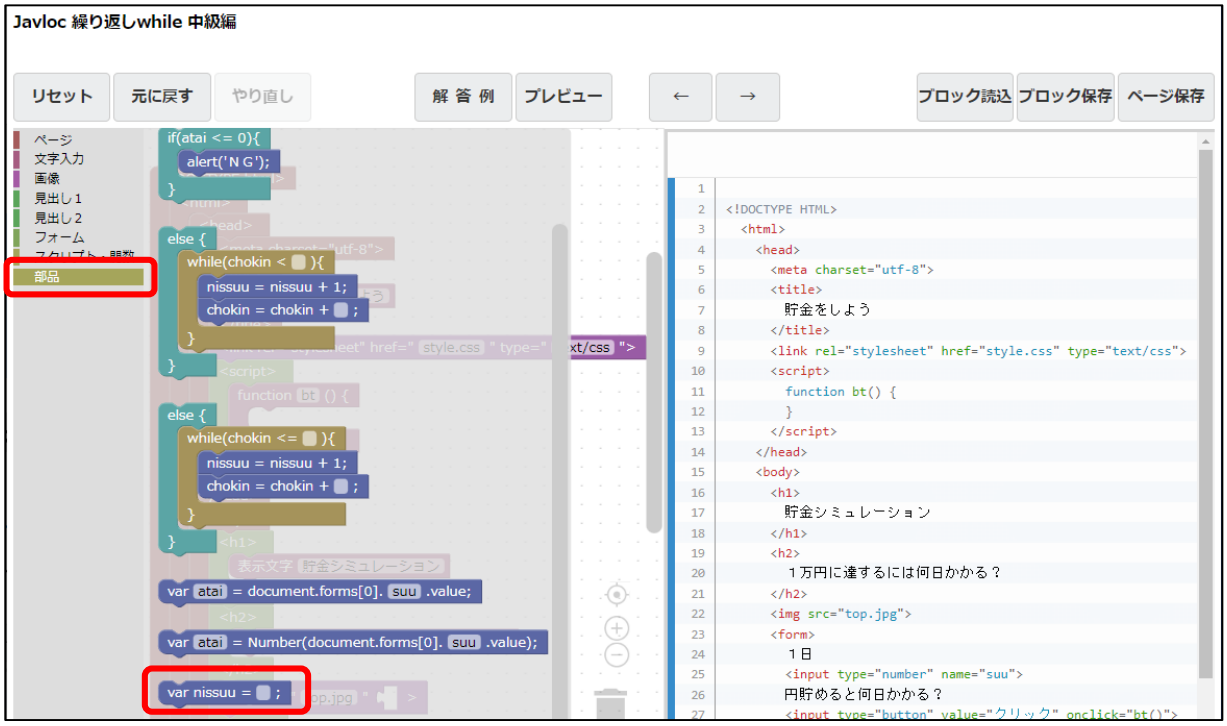
(2)　【部品】内から正しいブロックを選択し、(1)の関数ブロックの中に下記処理手順どおりに組み立てなさい。

■処理手順

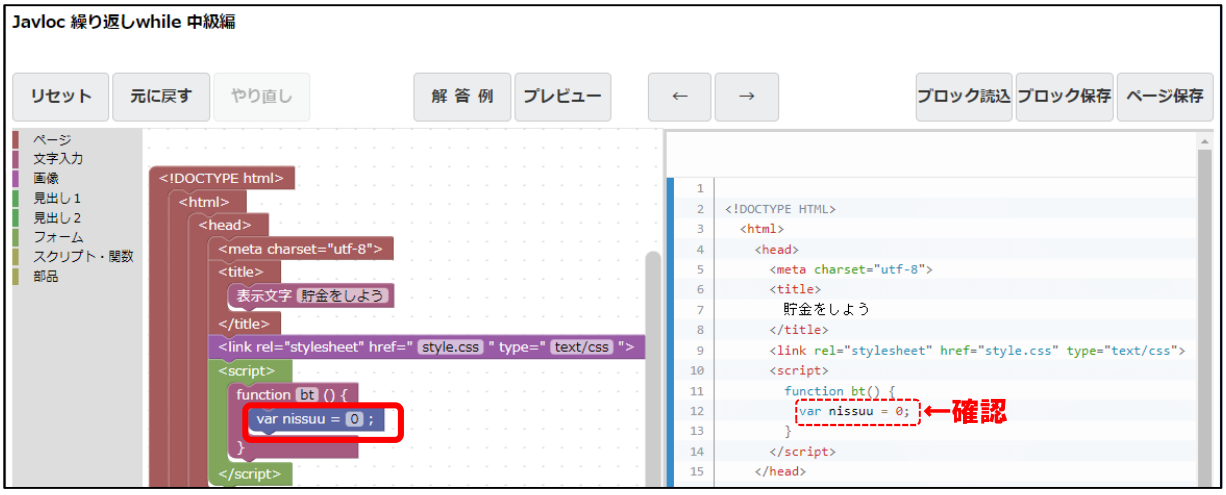
①　変数 nissuu を宣言し、0 を代入

リスト内[部品]をクリック、枠内にある var nissuu =　; ブロックを選択し、ドラッグして function bt (){～}ブロックの中にドロップ。

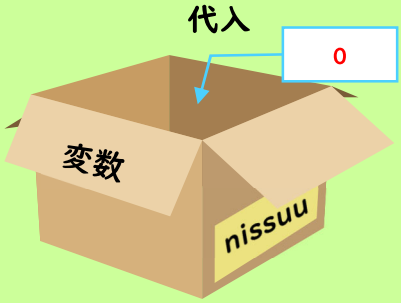
ブロック選択のヒント　問題文に「変数」とあったら「var」を含むブロックを選びましょう。



空欄に「0」と入力。※必ず半角で入力しましょう。



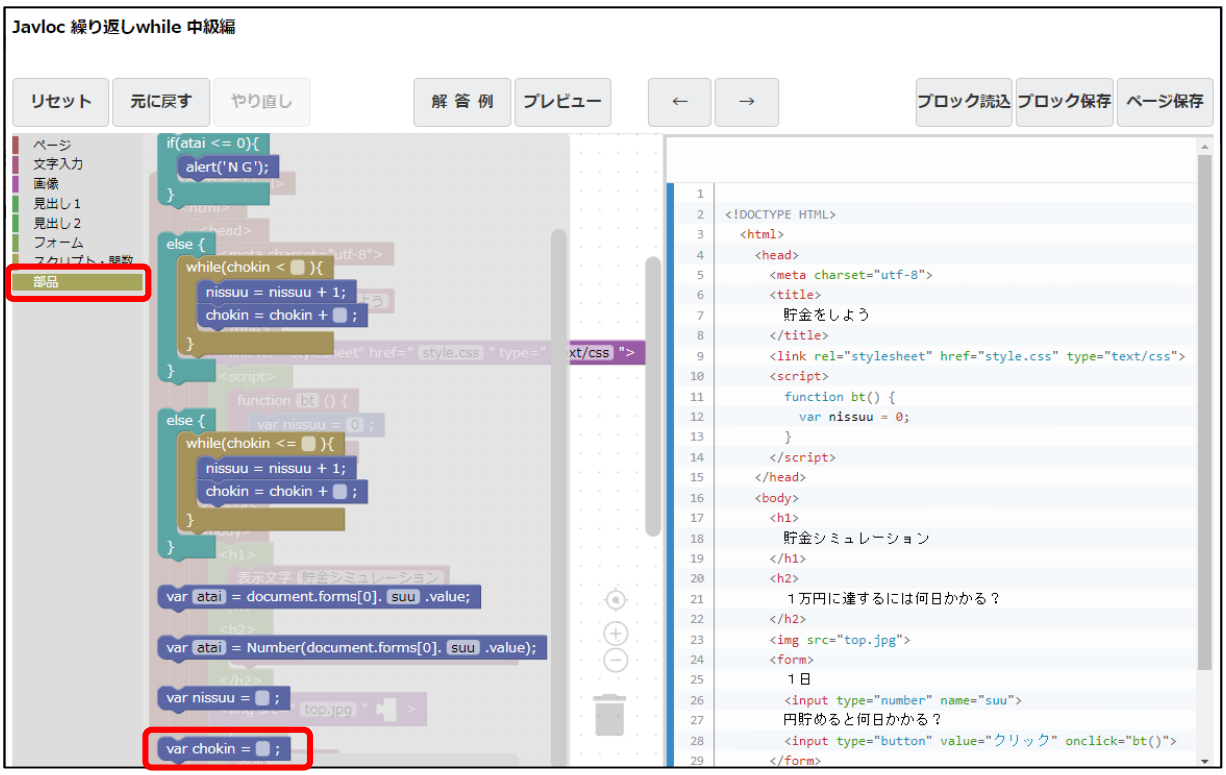
変数(var)：値を入れる箱のようなもの
宣言：変数を用意すること
代入(＝)：変数に値を入れること



②　変数 chokin を宣言し、0 を代入

リスト内[部品]をクリック、枠内にある var chokin =　　;ブロックを選択し、ドラッグして var nissuu = 0; ブロックの下にドロップ。

ブロック選択のヒント　**問題文に「変数」とあったら「var」を含むブロックを選びましょう。**

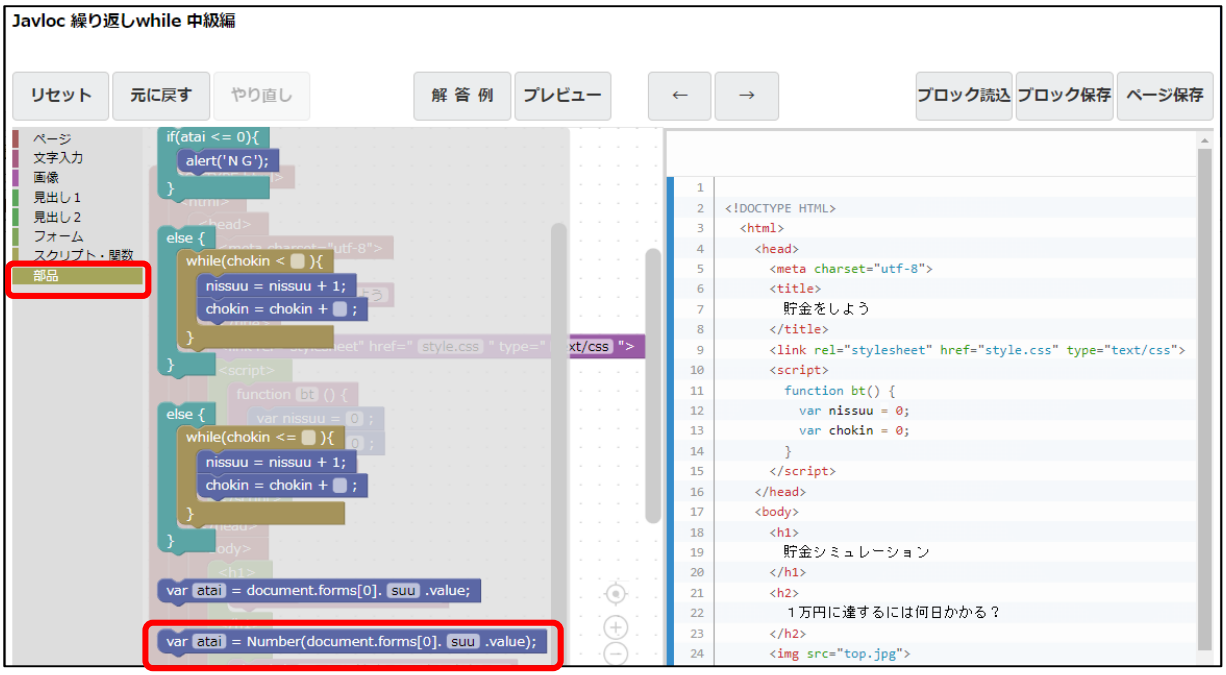


空欄に「0」と入力。※必ず半角で入力しましょう。

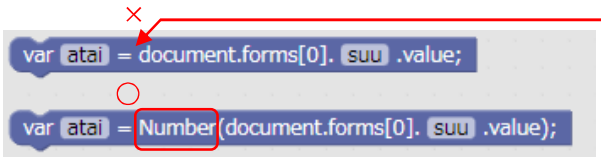
③　変数 atai を宣言し、フォームのナンバー（suu）の値を数値に変換し代入

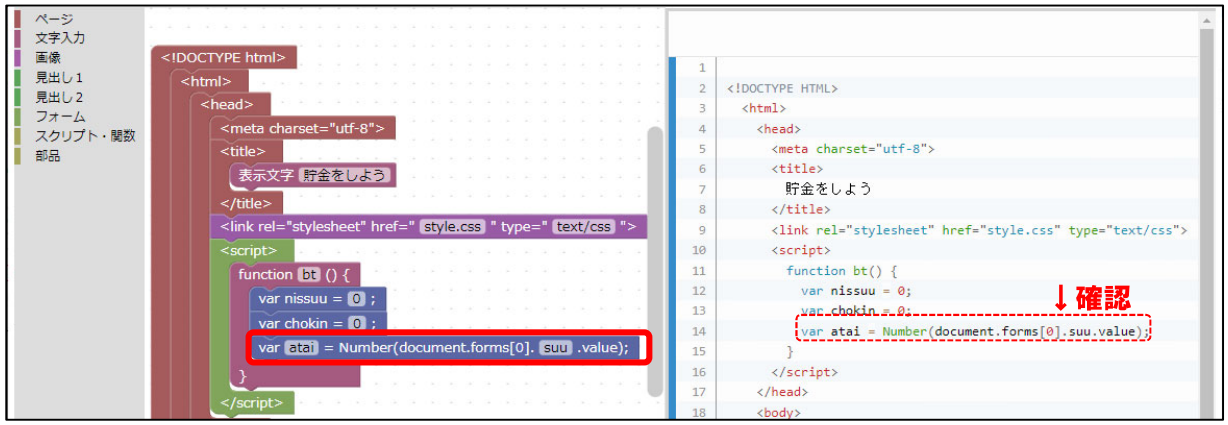
リスト内[部品]をクリック、枠内にある `var atai = Number(～);` ブロックを選択し、ドラッグして `var chokin = 0;` ブロックの下にドロップ。

ブロック選択のヒント　**問題文に「数値に変換」とあったら「Number」を含むブロックを選びましょう。**



[部品]の中には、誤ったブロックが混ざっています。

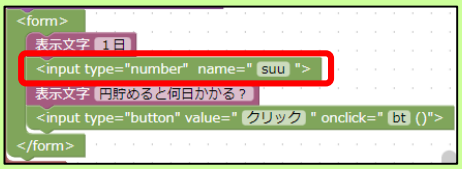




「Number(～)」は、（～）内にある値を**数値に変換**します。

「document.forms[0]」は、今作成している Web ページに最初に登場するフォームのことを指しています。（この問題では、処理条件３－(4)で作成した<form>のことです。）

.suu.value は、右図で示されている **suu という名前の付いた数値入力欄（number）に入力された値**を指します。フォームで入力された値は**文字列**として扱われるため、数値に変換していないと、この値を用いて計算をすることができません。そのため、Number(～)を用いて**数値に変換**する必要があるのです。

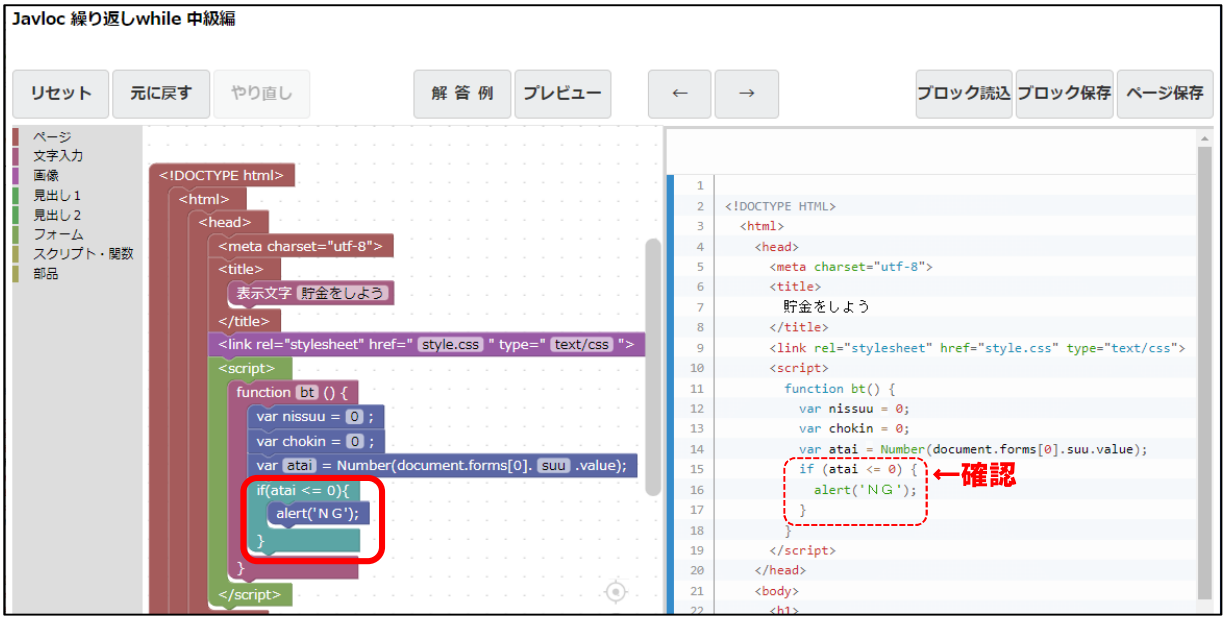


このように、型の処理をしていないと、正しく動作しません。

④　変数 atai が 0 以下の場合、「ＮＧ」と表示

リスト内[部品]をクリック、枠内にある `if (atai <= 0) {〜}` ブロックを選択し、ドラッグして `var atai = Number(〜) ;` ブロックの下にドロップ。

ブロック選択のヒント　問題文に「場合」とあったら条件式。「if」を含むブロックを選びましょう。



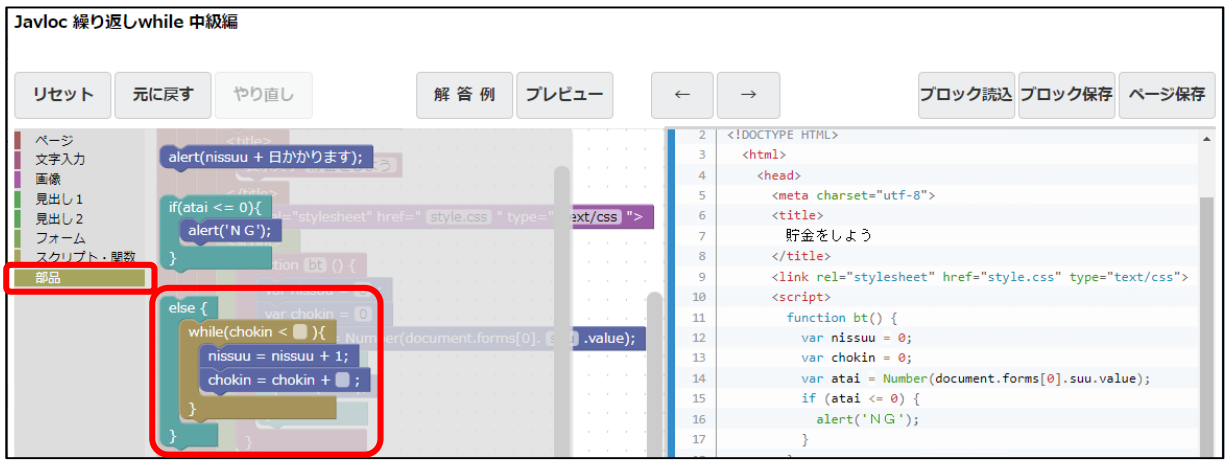
⑤　それ以外の場合

（ア）chokin が 10000 未満の間、下記の処理を繰り返す

nissuu に 1 を加算

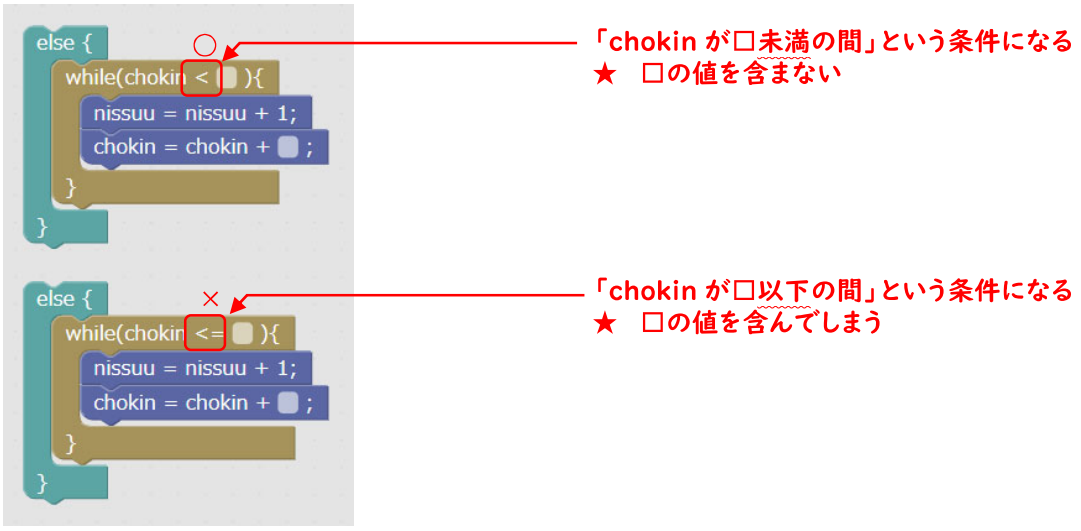
chokin に atai を加算し、chokin に代入

リスト内[部品]をクリック、枠内にある下図のブロックセットを選択。



[部品]の中には、誤ったブロックが混ざっています。

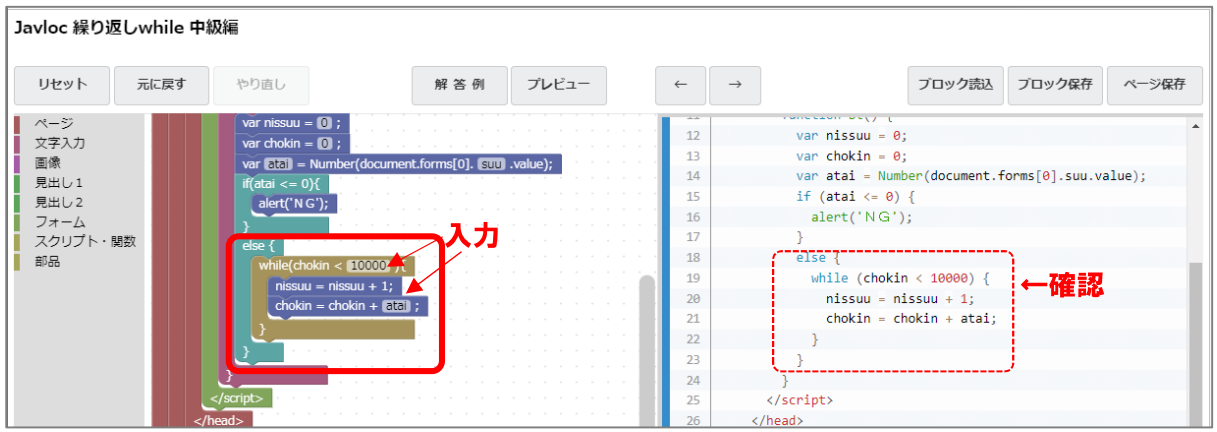
問題文では、「chokin が 10000 未満の間、下記の処理を繰り返す」とあるので



ドラッグして if (atai <= 0) {～}ブロックの下にドロップ。

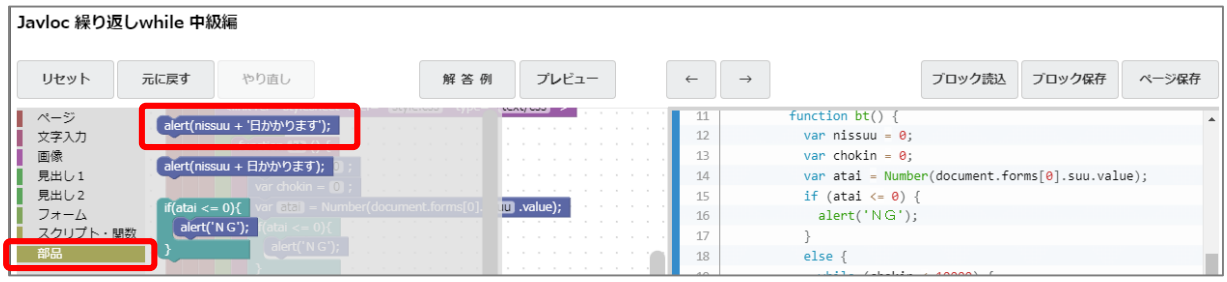
while (chokin < □);の空欄には、「10000」と入力。chokin = chokin + □;の空欄には、「atai」と入力。

※必ず半角で入力しましょう。



（イ）「〇日かかります」と表示　※〇は nissuu の値

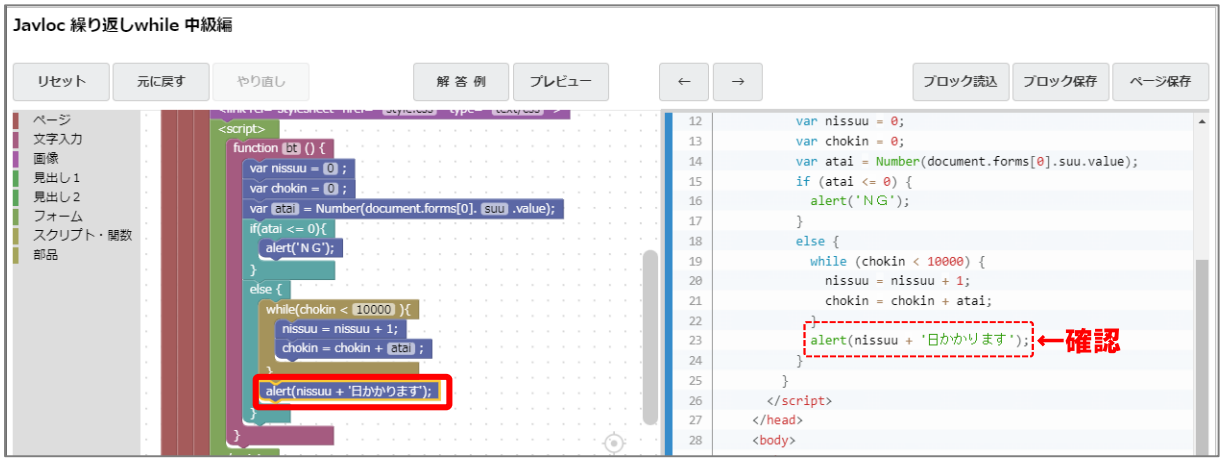
リスト内[部品]をクリック、枠内にある alert(nissuu + '日かかります'); ブロックを選択し、ドラッグ。



[部品]の中には、誤ったブロックが混ざっています。

問題文では、『「〇日かかります」と表示　※〇は nissuu の値』とあるので、

（ア）で設定した<else>ブロックの中、<while>ブロックの下にドロップ。



条件分岐のブロックの中身を日本語に読み下して見てみましょう。

```
if (atai <= 0) {
  alert('NG');
}
else {
  while (chokin < 10000) {
      nissuu = nissuu + 1;
      chokin = chokin + atai;
  }
  alert (nissuu + '日かかります');
}
```

```
もし（変数 atai が 0 以下だったら）{
  アラートメッセージで表示（「NG」）
}
それ以外の場合 {
  繰り返す（変数 chokin が 10000 より小さい間）{
      変数 nissuu に 1 を足す ←日数をカウントする箱
      変数 chokin に変数 atai を足す ←貯金箱
  }
    アラートメッセージで表示（「nissuu 日かかります」）
}
```

今回は、「目標額に到達するまで何回貯金を繰り返したか」を知りたいため、while 文を使用しています。

WEB ページ上では、結果しか確認することができません。

そのため次ページでは、具体的に while 文ではどのような処理を繰り返して結果を導き出したのかを見ていきます。

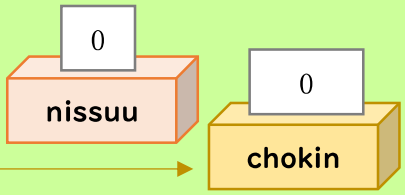
例）

数値入力欄（suu）に「5000」と入力した場合を考えてみましょう

最初の状態

nissuu ＝ 0

chokin ＝ 0

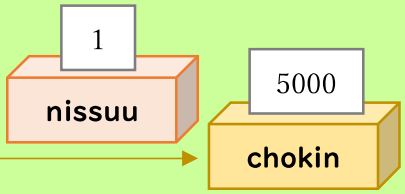


繰り返し処理１回目

①chokin が「0」で、10000 より小さい値のため

nissuu（0）に 1 を足す

chokin（0）に 5000 を足す

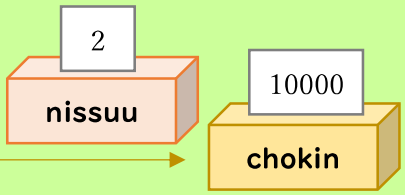


繰り返し処理２回目

②chokin が「5000」で、10000 より小さい値のため

nissuu（1）に 1 を足す

chokin（5000）に 5000 を足す



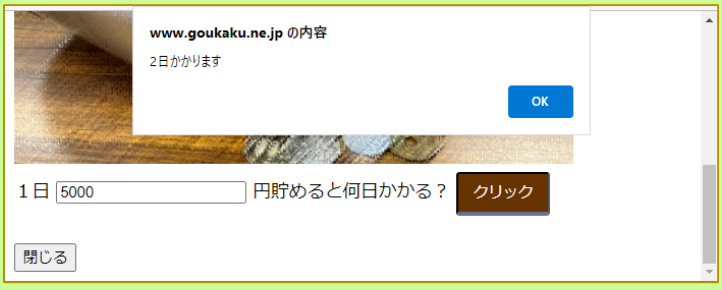
繰り返し処理３回目

③chokin が「10000」で、10000 より小さい値でないため

繰り返し処理を終了し、次の処理（アラートメッセージ表示）へ移行する

nissuu には「2」が入っているため

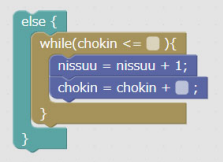
アラートメッセージには「２日かかります」と表示されます



Try

次の設定も終え、正しく動作することを確認したら、

試しに、誤っている方のブロック（chokin<=10000 となる条件）に置き換えてみましょう。



すると、繰り返し処理３回目が行われ、結果は「３日かかります」となってしまいます。

求めたいのは１万円貯めるために必要な日数をカウントすることなので、chokin<10000 と設定する必要があることが体感できるでしょう。

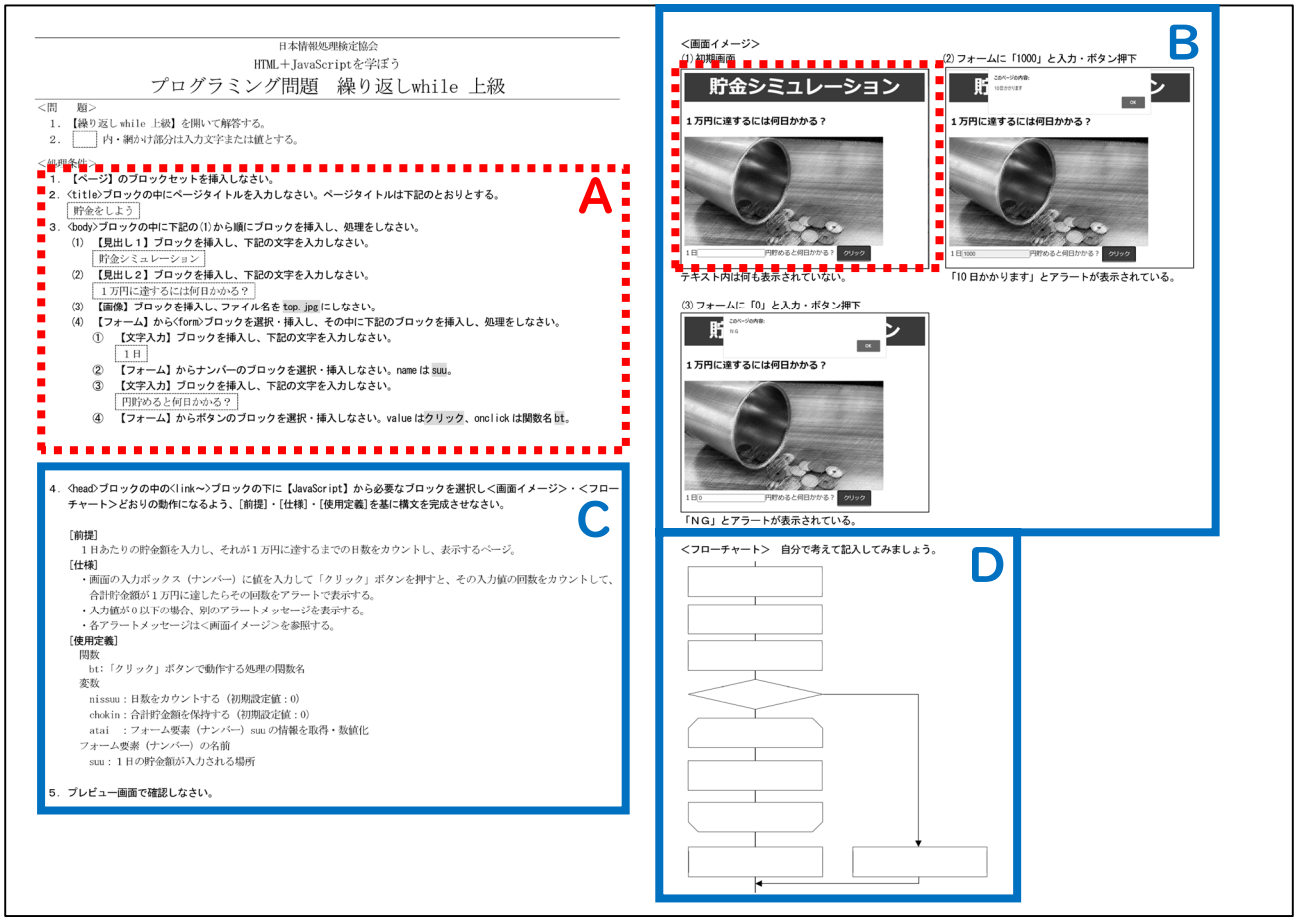
５．プレビュー画面で確認しなさい。

[プレビュー]ボタンをクリックし、動作を確認。

４．で解答例を基に確認した内容と同様の動作になっているか見てみましょう。

3．繰り返しwhile　上級　解答手順

＜問題文＞



A=HTML 部分の指示
C=JavaScript 部分の指示

まず、**AとB-(1)の画面イメージを参照**し、ページ全体の構成を**処理条件のとおりに組み立て**ていきます。
次に、**Cの[前提]・[仕様]・[使用定義]**、そして**Bの画面イメージで動作の確認**をします。
実際にブロックを組み立てる前に、**Dのフローチャートを用いて処理手順の整理**をしてみましょう。
その後、**フローチャートどおりにブロックを組み立て**ていきます。

教材サイトからアプリを開く

Javloc 教材サイトから[上級プロジェクト]の[繰り返し「while」上級]をクリック。

ページを構成（HTML 部分の作成）

１～３は、中級編解答方法をご参照ください。

プログラムを組み立てる（JavaScript 部分の作成）

４．<head>ブロックの中の<link～>ブロックの下に【JavaScript】から必要なブロックを選択し＜画面イメージ＞・＜フローチャート＞どおりの動作になるよう、［前提］・［仕様］・［使用定義］を基に構文を完成させなさい。

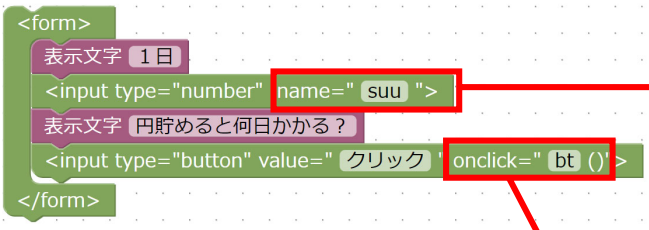
［前提］

１日あたりの貯金額を入力し、それが１万円に達するまでの日数をカウントし、表示するページ。

インプット部分の確認

3.の(4)入力フォーム

＜ブロック＞



＜作成ページ例＞

仕様部分を読み解く

［仕様］

・画面の入力ボックス（ナンバー）に値を入力して「クリック」ボタンを押すと、その入力値の回数をカウントして、合計貯金額が１万円に達したらその回数をアラートで表示する。

→インプット部分の設定を確認。

→合計貯金額を計算して保管しておくための箱（変数）を準備。

→日数をカウントするための箱（変数）を準備。

→合計貯金額が１万円に達するまで、入力された値の加算・回数のカウントを繰り返す。

→カウントした回数はアラートメッセージに表示。

・入力値が０以下の場合、別のアラートメッセージを表示する。

→入力された値が０以下の場合、別のアラートメッセージを表示。

・各アラートメッセージは＜画面イメージ＞を参照する。

→入力値０以下の場合：＜画面イメージ＞（３）を参照。アラートメッセージは「NG」だと確認。

→それ以外の場合：＜画面イメージ＞（２）を参照。アラートメッセージは「○日かかります」だと確認。

これに[使用定義]の内容を照らし合わせてフローチャートを作成

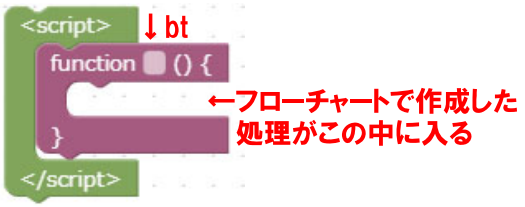
[使用定義]

関数

bt:「クリック」ボタンで動作する処理の関数名

→処理はすべて関数「bt」の中で処理されるため、

フローチャートに関数は含みません。



＜フローチャート＞　自分で考えて記入してみましょう。

変数

nissuu：日数をカウントする（初期設定値：0）

var nissuu = 0;

変数

chokin：合計貯金額を保持する（初期設定値：0）

var chokin = 0;

変数

atai　：フォーム要素（ナンバー）suu の情報を取得・数値化

フォーム要素（ナンバー）の名前

suu：１日の貯金額が入力される場所

var atai = Number(document.forms[0].suu.value);

もし、入力値が0以下の場合、
‘NG’をアラートメッセージで表示する。

if (atai <= 0) ; {

それ以外の場合、

else {

合計貯金額が１万円に達するまで、入力された値の加算・回数のカウントを繰り返す。

while (chokin < 10000) {

nissuu = nissuu + 1;

chokin = chokin + atai;

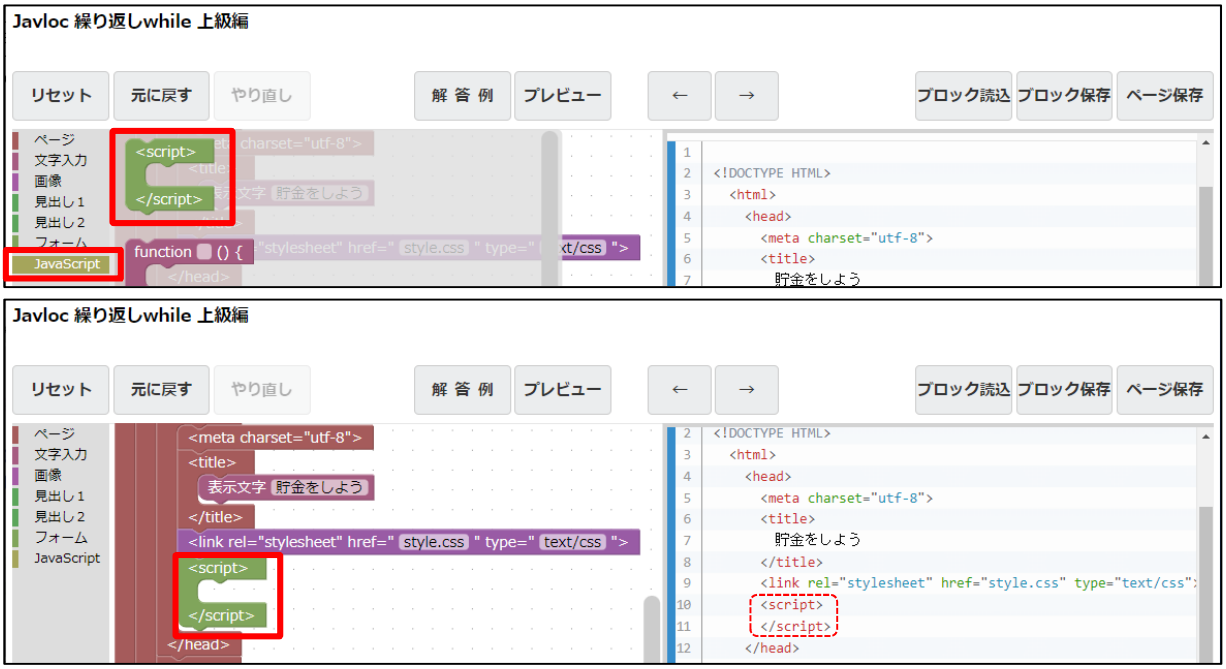
}

カウントした回数はアラートメッセージに表示。

alert(nissuu + '日かかります') ; }

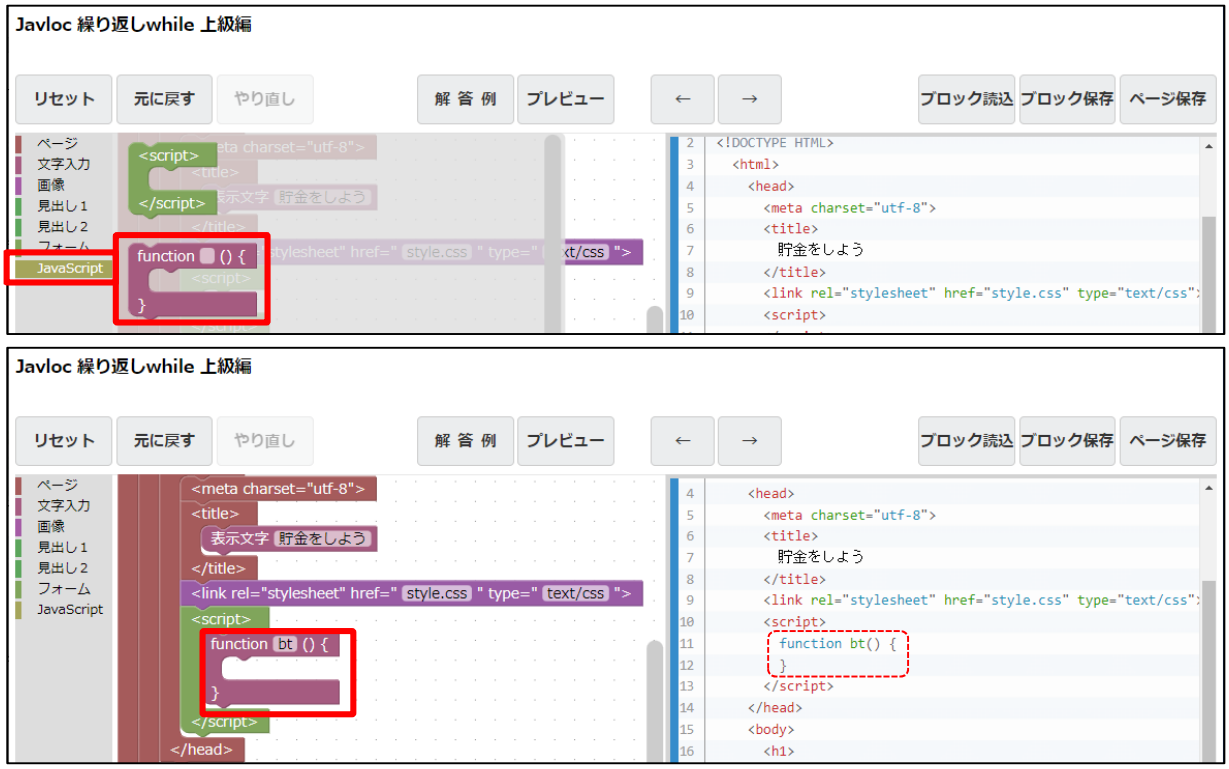
alert('ＮＧ') ; }

リスト内[JavaScript]をクリック、枠内にある<script>ブロックを選択し、ドラッグして＜link>ブロックの下にドロップ。ソース表示エリアで設定を確認。

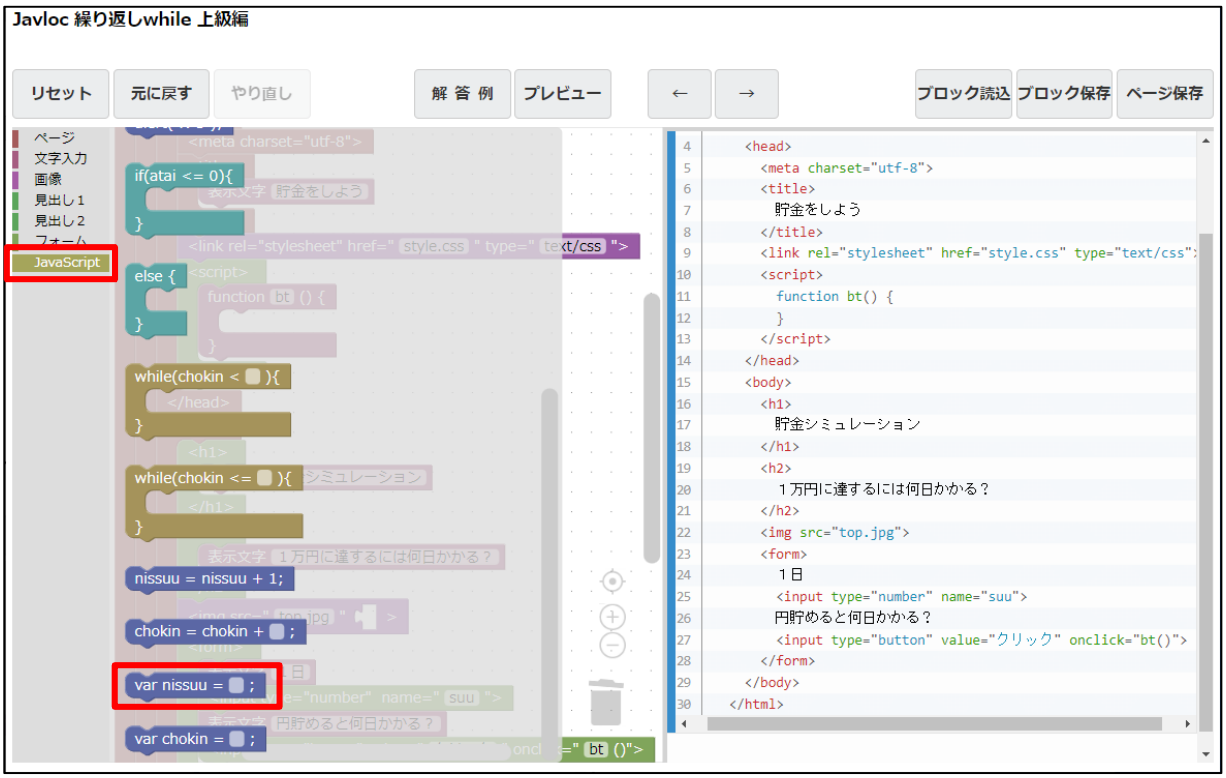


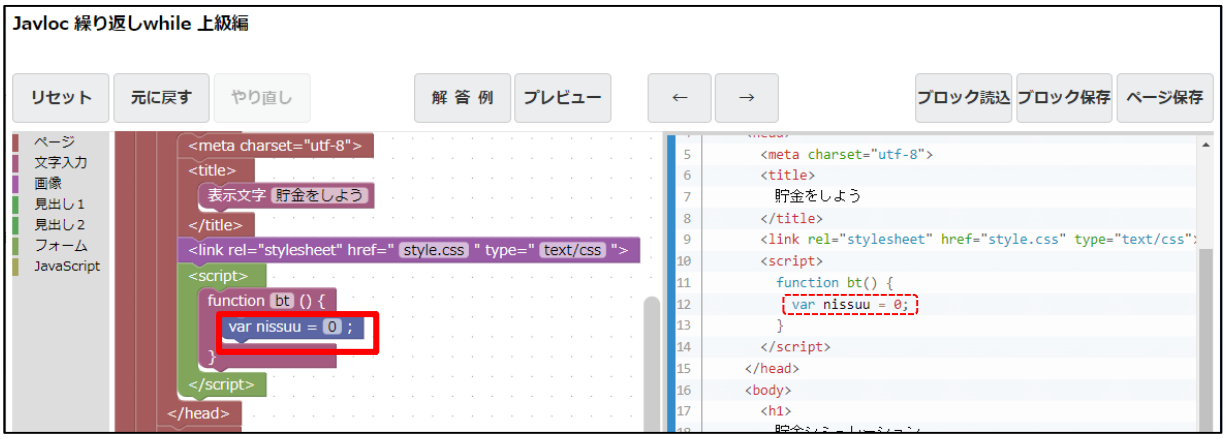
リスト内[JavaScript]をクリック、枠内にある<function>ブロックを選択し、ドラッグして＜script>ブロックの中にドロップ。関数名「bt」を設定。ソース表示エリアで設定を確認。

※必ず半角で入力しましょう。

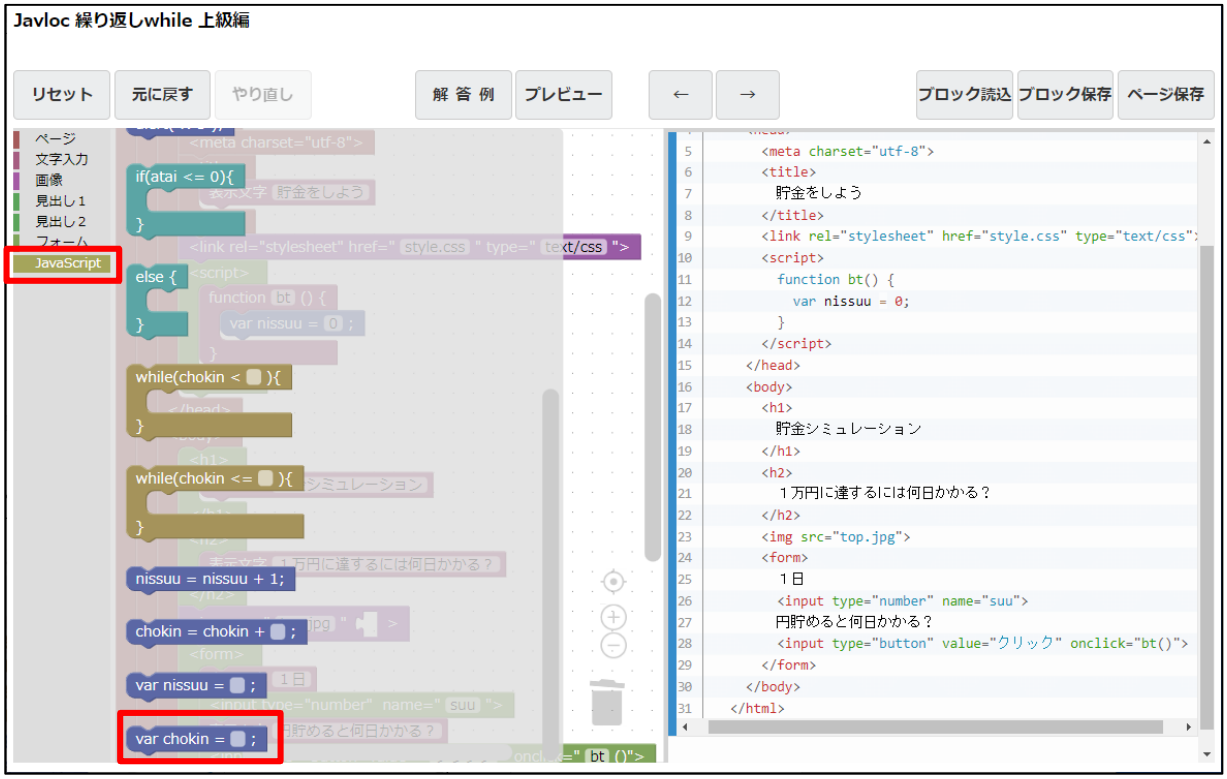


リスト内[JavaScript]をクリック、枠内にある var nissuu = 　; ブロックを選択し、ドラッグして<function>ブロックの中にドロップ。数値「0」を設定。ソース表示エリアで設定を確認。

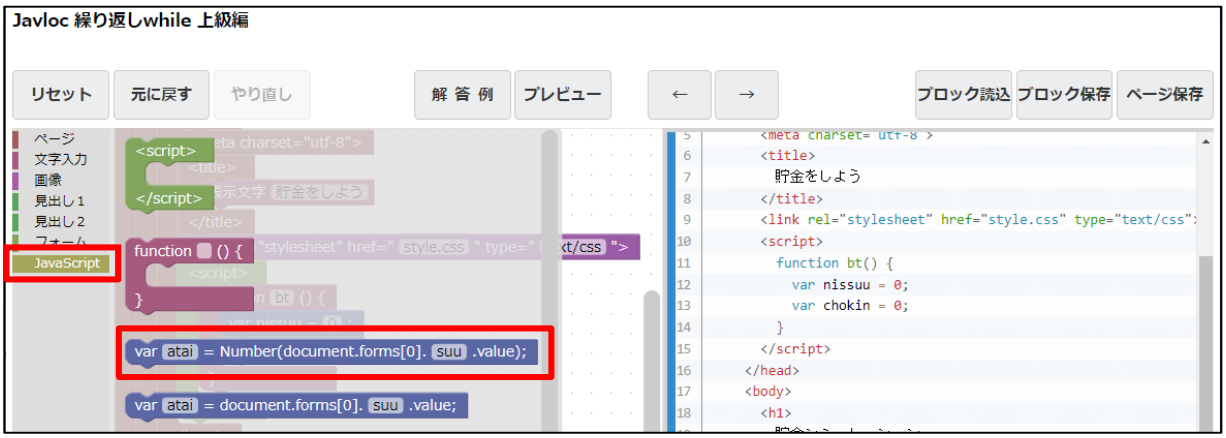


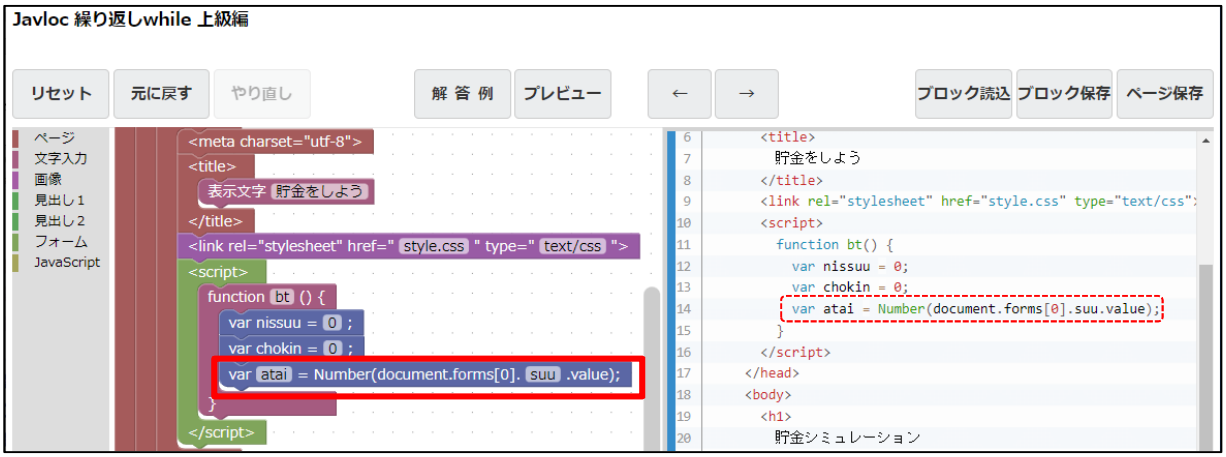


リスト内[JavaScript]をクリック、枠内にある var chokin = ; ブロックを選択し、ドラッグして var nissuu = 0; ブロックの下にドロップ。数値「0」を設定。ソース表示エリアで設定を確認。

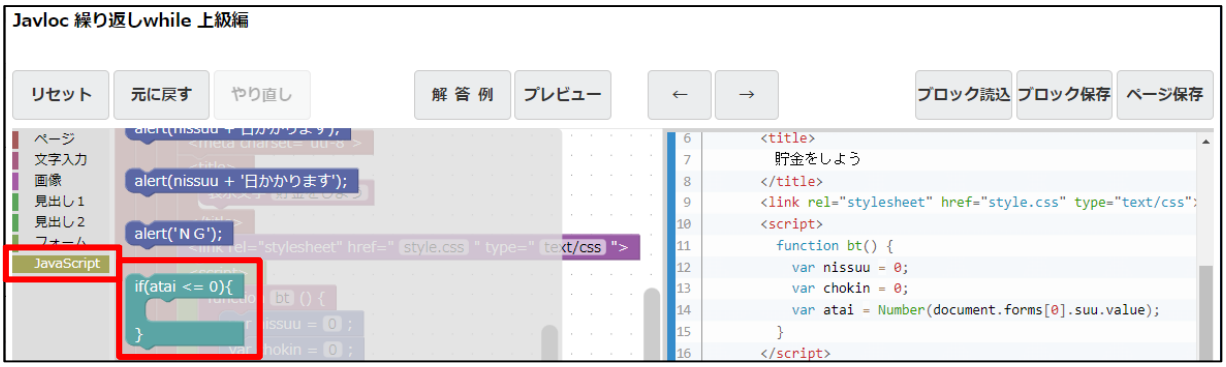


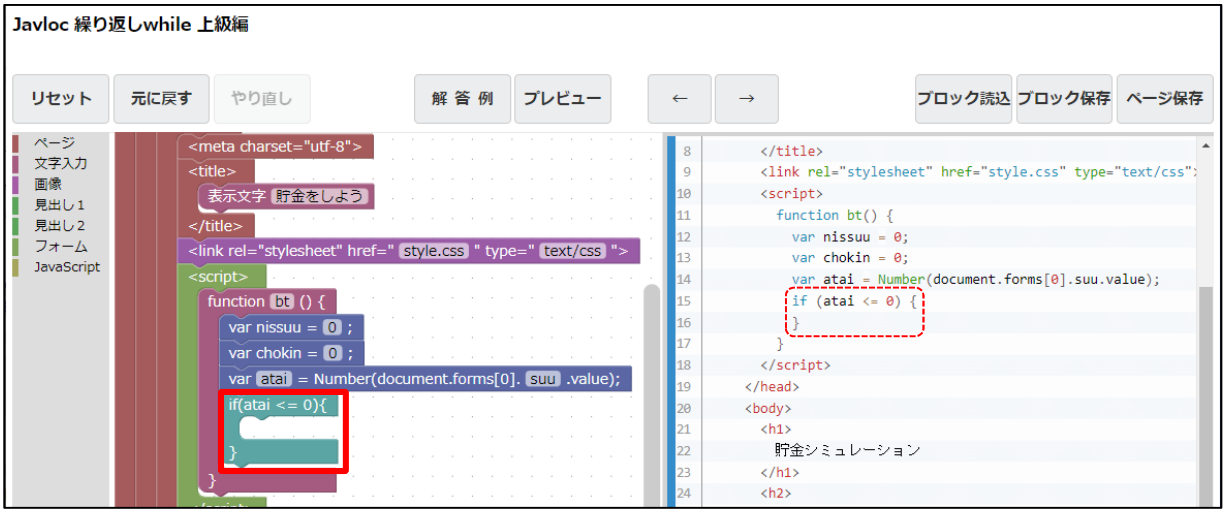
リスト内[JavaScript]をクリック、枠内にある var atai = Number(～) ; ブロックを選択し、ドラッグして var chokin = 0; ブロックの下にドロップ。ソース表示エリアで設定を確認。



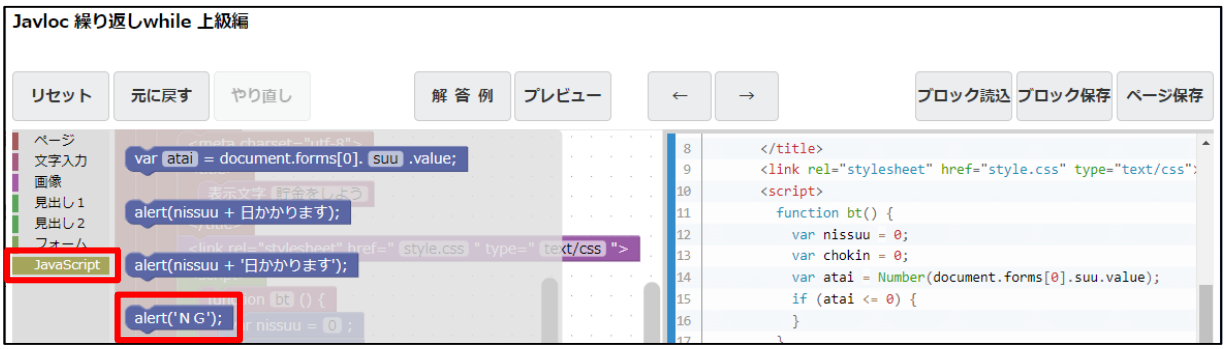


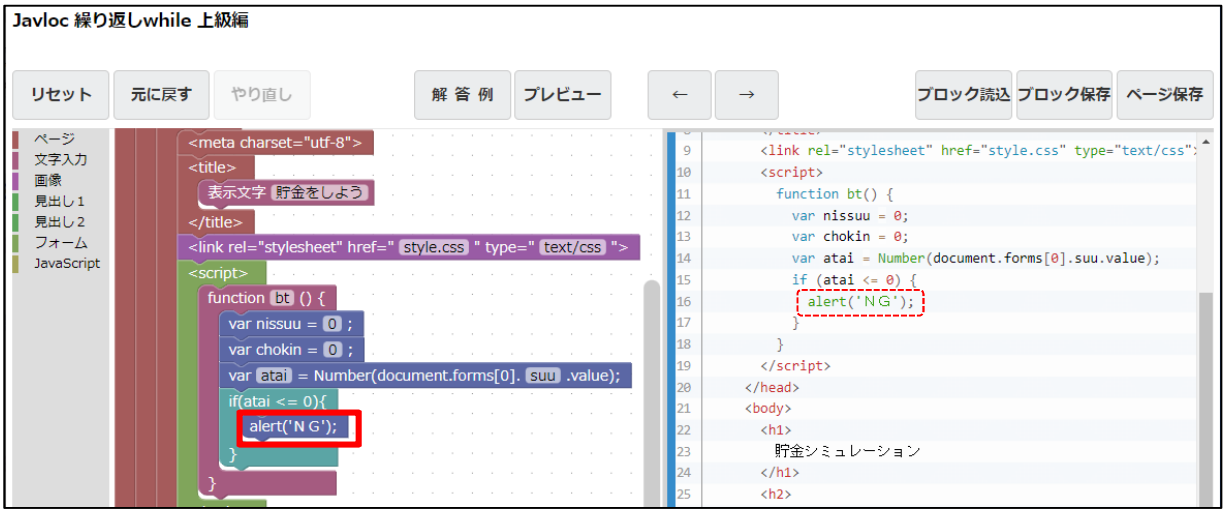
リスト内[JavaScript]をクリック、枠内にある if (atai <= 0) {～} ブロックを選択し、ドラッグして var atai = Number(～) ; ブロックの下にドロップ。ソース表示エリアで設定を確認。



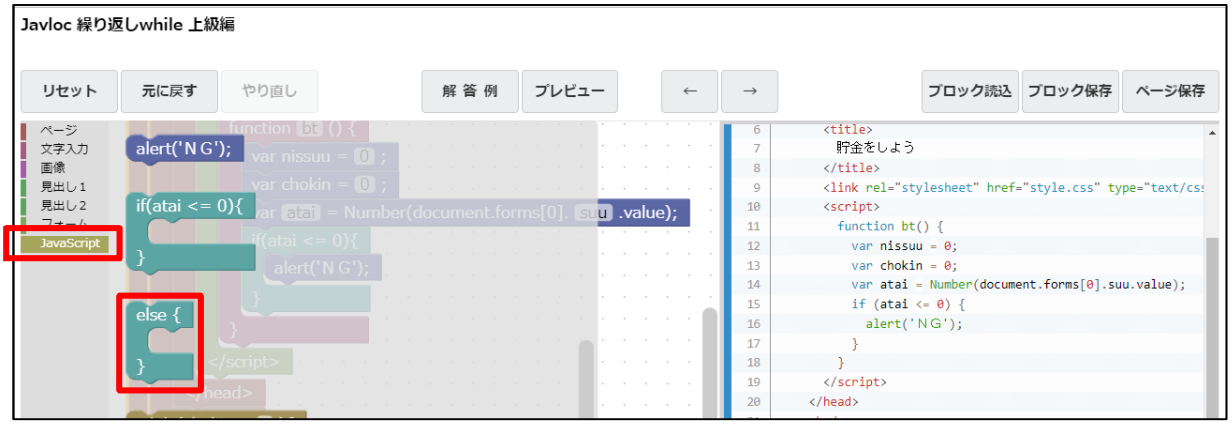


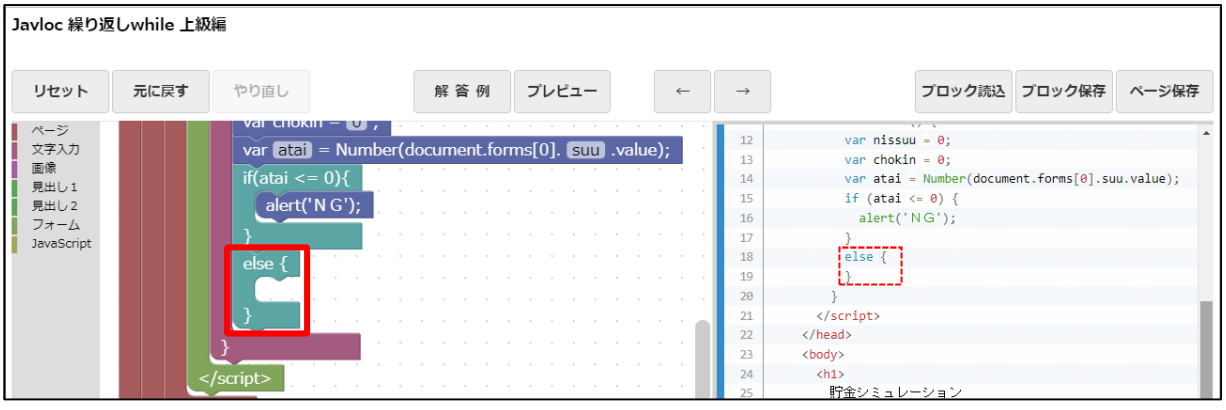
リスト内[JavaScript]をクリック、枠内にある alert('ＮＧ'); ブロックを選択し、ドラッグして if (atai <= 0) {～} ブロックの中にドロップ。ソース表示エリアで設定を確認。



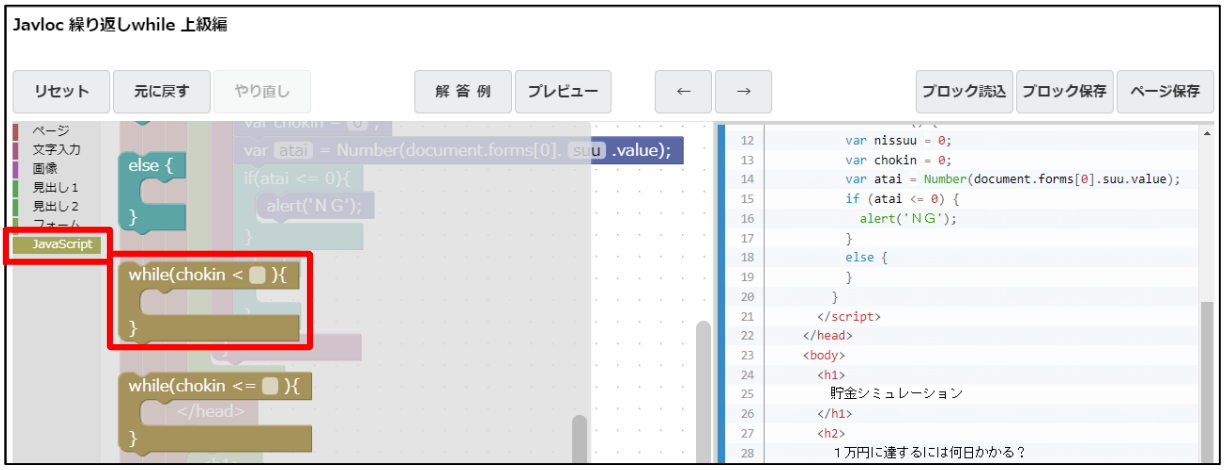


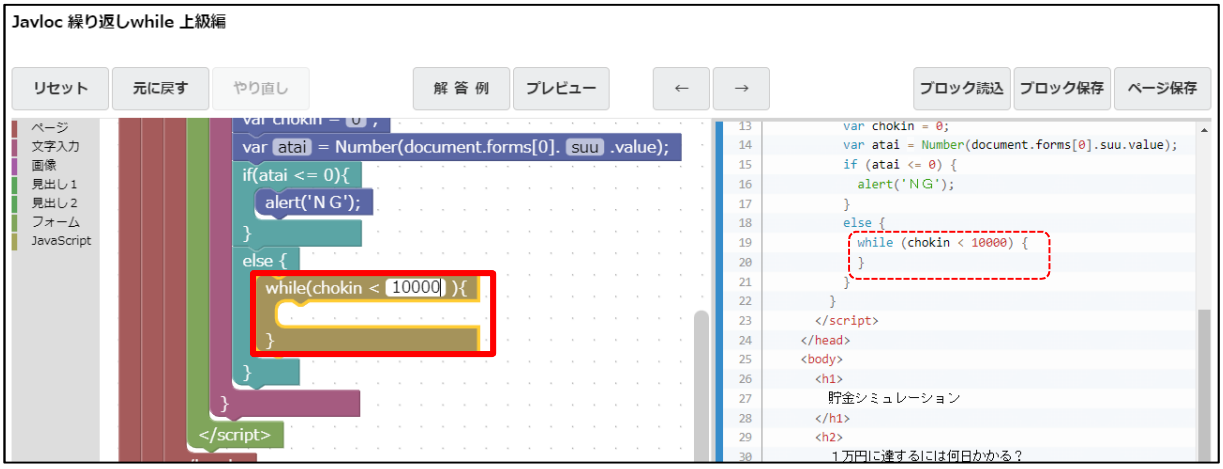
リスト内[JavaScript]をクリック、枠内にある else {～} ブロックを選択し、ドラッグして if (atai <= 0) {～} ブロックの下にドロップ。ソース表示エリアで設定を確認。



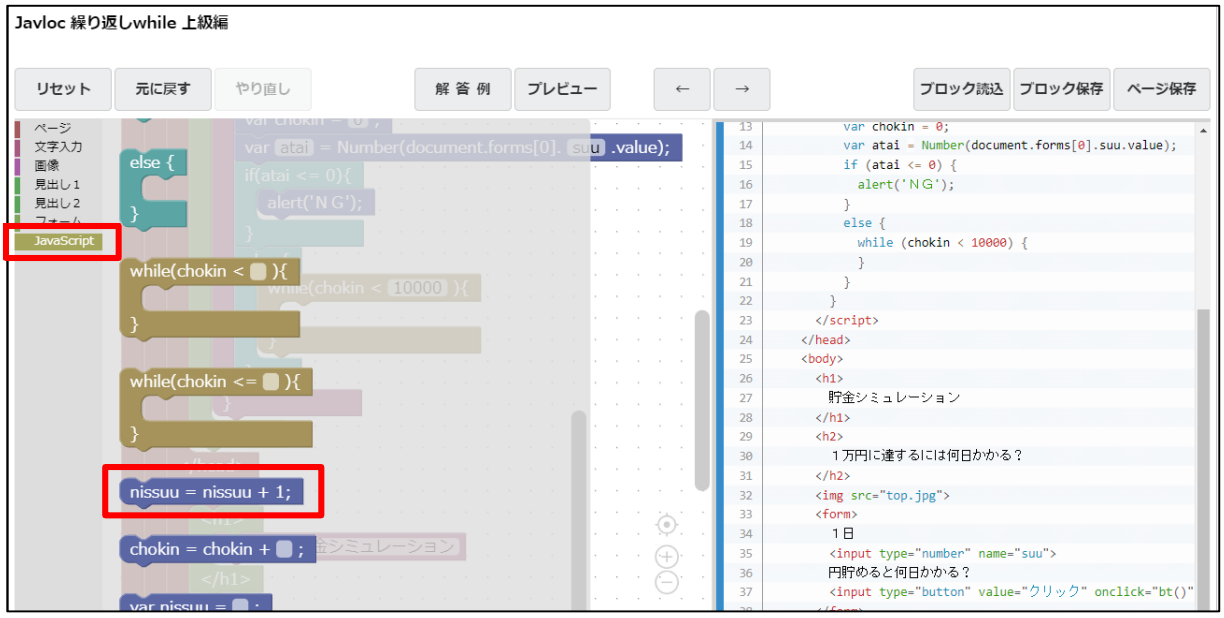


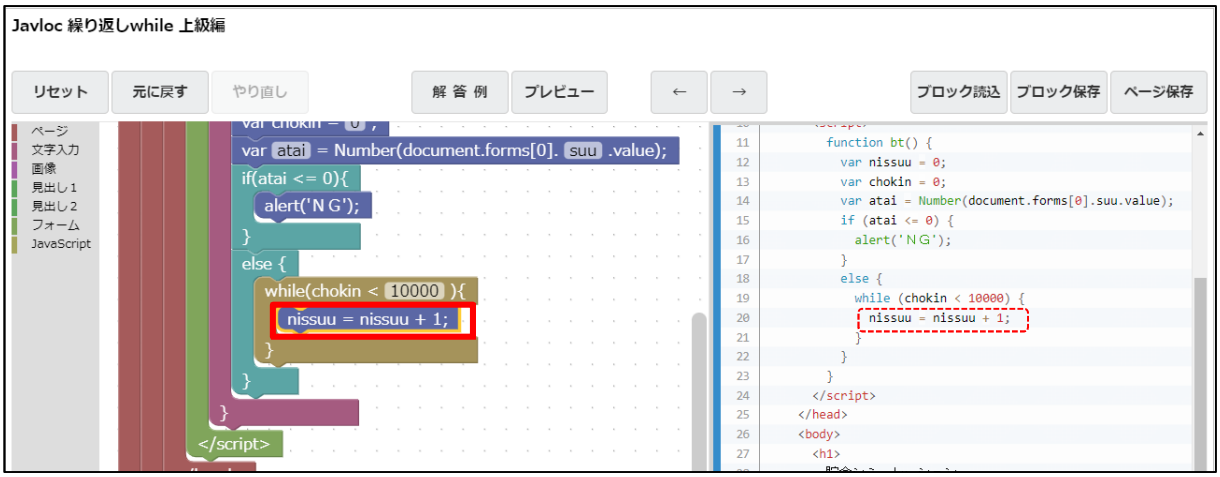
リスト内[JavaScript]をクリック、枠内にある while (chokin < □) {～} ブロックを選択し、ドラッグして else {～} ブロックの中にドロップ。数値「10000」を設定。ソース表示エリアで設定を確認。



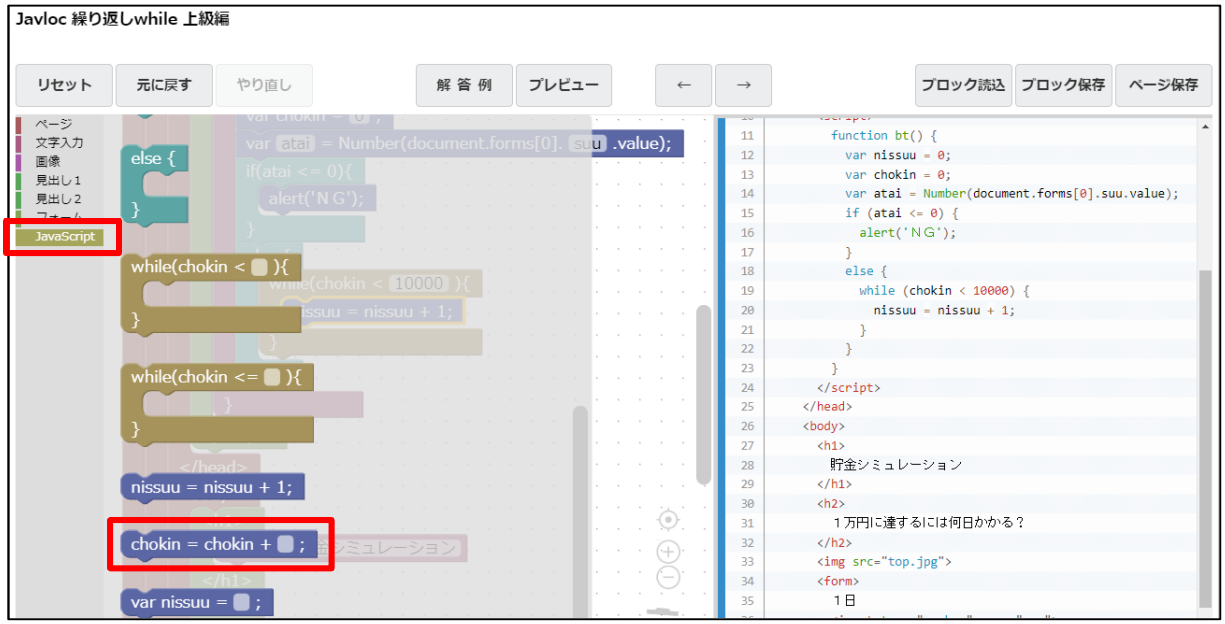


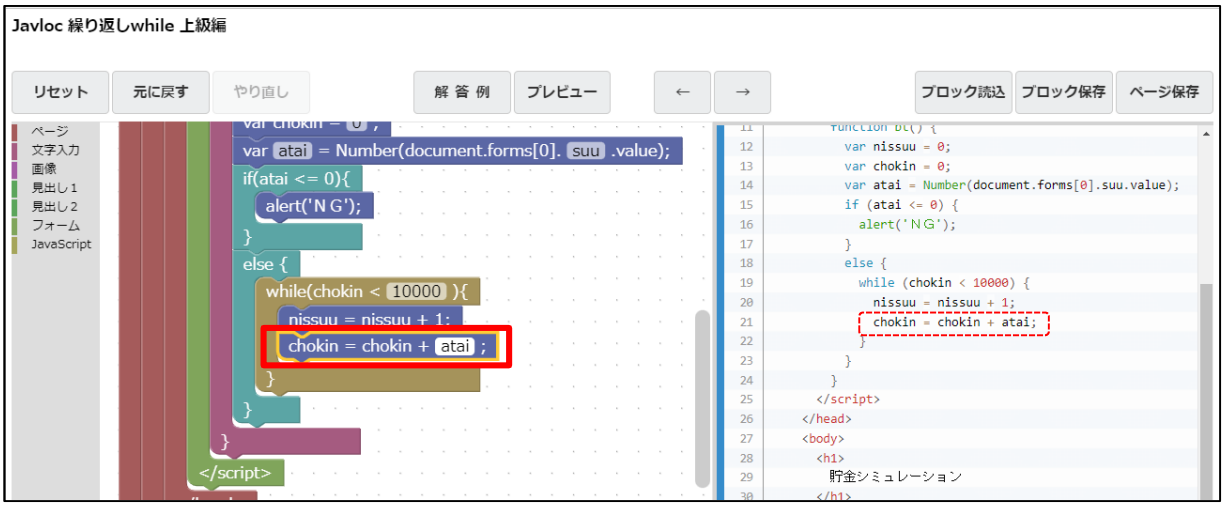
リスト内[JavaScript]をクリック、枠内にある nissuu = nissuu + 1; ブロックを選択し、ドラッグして while (chokin < 10000) {～} ブロックの中にドロップ。ソース表示エリアで設定を確認。



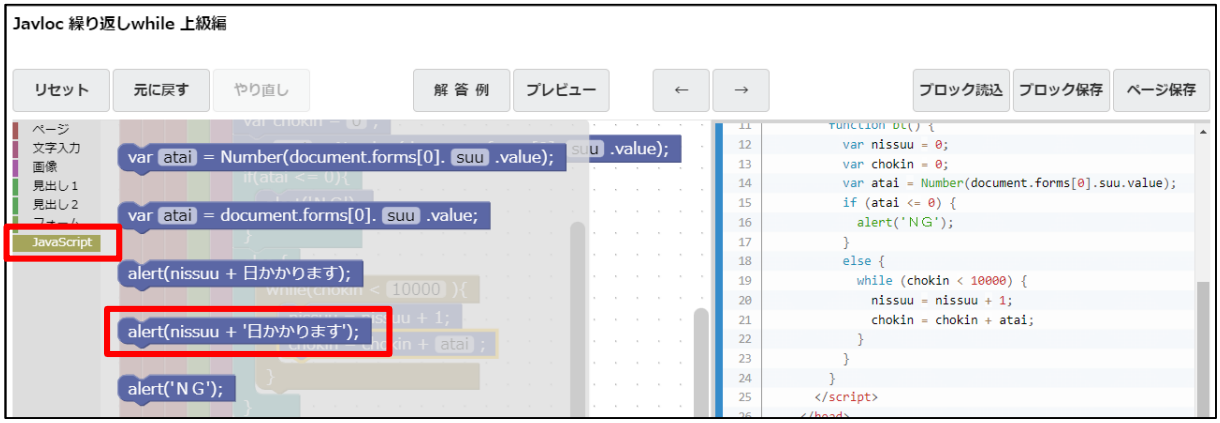


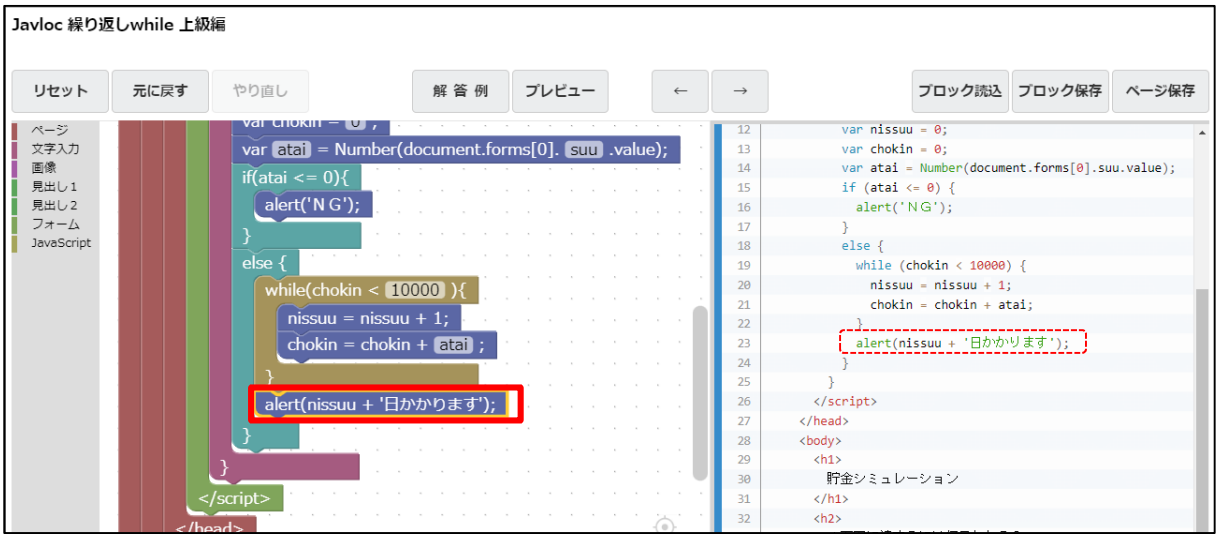
リスト内[JavaScript]をクリック、枠内にある chokin = chokin + □; ブロックを選択し、ドラッグして nissuu = nissuu + 1; ブロックの下にドロップ。変数名「atai」を設定。ソース表示エリアで設定を確認。





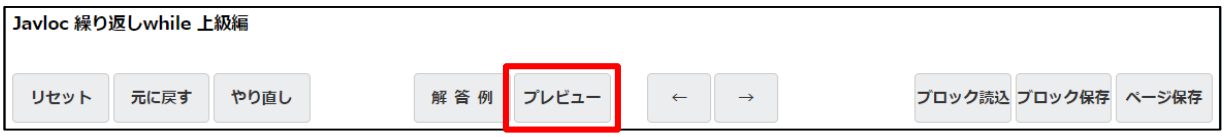
リスト内[JavaScript]をクリック、枠内にある alert(nissuu + '日かかります'); ブロックを選択し、ドラッグして while (chokin < 10000) {～} ブロックの下にドロップ。ソース表示エリアで設定を確認。





5．プレビュー画面で確認しなさい。

[プレビュー]ボタンをクリックし、動作を確認。



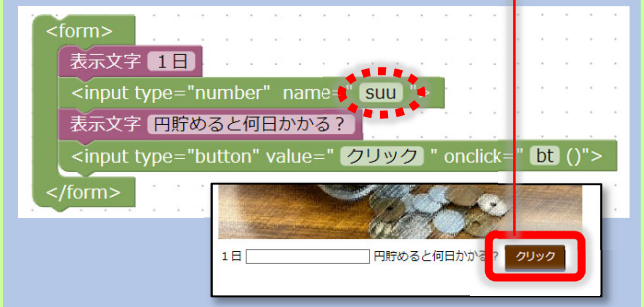
＜画面イメージ＞と同様の動作になっているか見てみましょう。

プログラム全体の流れ

⓪ 数値入力欄から値が入力され、クリックボタンが押されると、関数が実行される。
① 変数 nissuu に 0 を代入する。
② 変数 chokin に 0 を代入する。
③ 変数 atai に⓪の数値入力欄「suu」を数値化した値を代入する。
④ もし変数 atai が 0 以下だった場合、‘ NG ’というアラートメッセージを表示する。
⑤ ④の条件でない場合、(ア)・(イ)の処理を行う。
(ア) while 文を使用し、変数 atai が 10000 未満の間、以下の処理を繰り返す。
変数 nissuu に 1 を足す・変数 chokin に変数 atai を足す
(イ) 変数 nissuu の値と「日かかります」を文字列としてアラートメッセージに表示する。

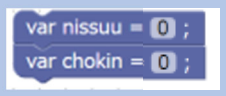
【HTML 部分・作成ページ例】

⓪

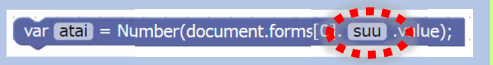


【JavaScript 関数 bt が実行】

①
②



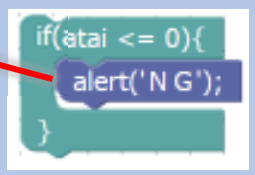
③



0と入力した場合

atai = 0

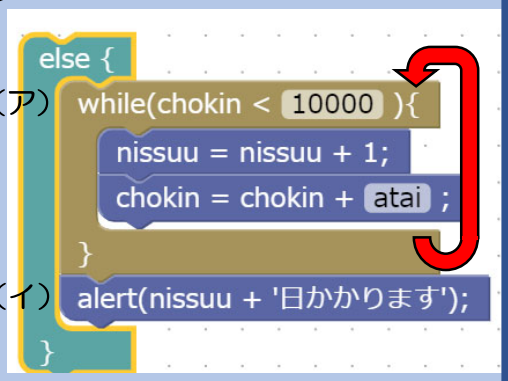
④



1000 と入力した場合

atai = 1000

⑤



＜巻末資料＞作成上の基本ルール

全般

・プログラムは半角英数字・半角スペースで記述する。（文字列には全角使用可）
・大文字と小文字は別物として扱われる。（区別される）
・一つの文（命令）の終わりには「;」（セミコロン）を付ける。 ； は命令文の終わりを意味する。
※ブラウザが推測して解釈してくれるため、省略が可能。ただし、思わぬ動きをする可能性もあるため付けることが推奨される。
・文字列は「'」（シングルクォーテーション）または「"」（ダブルクォーテーション）で囲む。
他に「`」（バッククォート）で囲む方法もある。

主な演算子

種類	意味
=	代入
==	等しい
!=	等しくない
<	未満（～より小さい）
<=	以下
>	超過（～より大きい）
>=	以上

種類	意味
+	数値の足し算（加算）
	文字列の結合
−	引き算（減算）
*	かけ算（乗算）
/	割り算（除算）
++	1 増やす
−−	1 減らす

種類	意味
+=	加算代入
−=	減算代入
*=	乗算代入
/=	除算代入
,	式の区切り
.	～の
!	ではない

変数の命名ルール

・数字のみ、先頭が数字の名前は禁止。（NG 例：3name　OK 例：name3）
・予約語と呼ばれる JavaScript で別の目的で既に使用することが決まっている名前は使用不可。
（NG 例：if、for、break など）
・大文字と小文字が厳密に区別される。（例：Box と box は別物となる）

型について

文字列や数値などのデータの種類のことを「型（Type）」と呼ぶ。

■よく使われる型

型名	種類	例
Number	数値	0、123、-4、1.5
String	文字列	'こんにちは'、"あいうえお"
Boolean	論理値	true、false

■型の変換

＋演算子は数値の場合は足し算（加算）、文字列の場合は結合になる。
文字列とその他の型を＋演算した場合、文字列に変換される。

例	結果	結果の型
1 ＋ 2	3	Number 型（数値）
'1' ＋ '2'	12	String 型（文字列）
1 ＋ '2'	12	String 型（文字列）
'1' ＋ 2	12	String 型（文字列）

他にも、あらかじめ関数を用いて型変換を行う方法もある。
本教材では、文字列を数値型に型変換する際、Number( )；関数を使用。