1．繰り返しfor　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for 初級

＜問　　題＞

1．【繰り返し for 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。

クイズ

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。

画像を並べてみよう

(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。

柱が５本になるにはこの画像を何枚並べればよいでしょう。

(3)　【画像】ブロックを挿入し、ファイル名をpic1.jpgにしなさい。

(4)　【フォーム】のブロックセット（テキスト・ボタン）を挿入しなさい。

(5)　【コンテナ】ブロックを挿入し、中に【id】ブロックを挿入しなさい。id名はoutput。

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]

指定された回数分、画像を表示させる。

[機能]

・フォームのテキストに入力された値の数だけ画像を表示する。

・表示先は３－(5)の位置とする。

・画像表示の処理は、表示先をクリアした状態から始める。

(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　画像を表示させる場所（id名output）の情報を取得し、「gazou」に設定する

②　フォームのテキスト（suu）の値を数値にして「atai」に設定する

③　「gazou」の中身を空欄にする

④　フォームに入力した回数以下になるまで、画像pic1.jpgの表示を繰り返す

＜フローチャート＞

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「for」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。

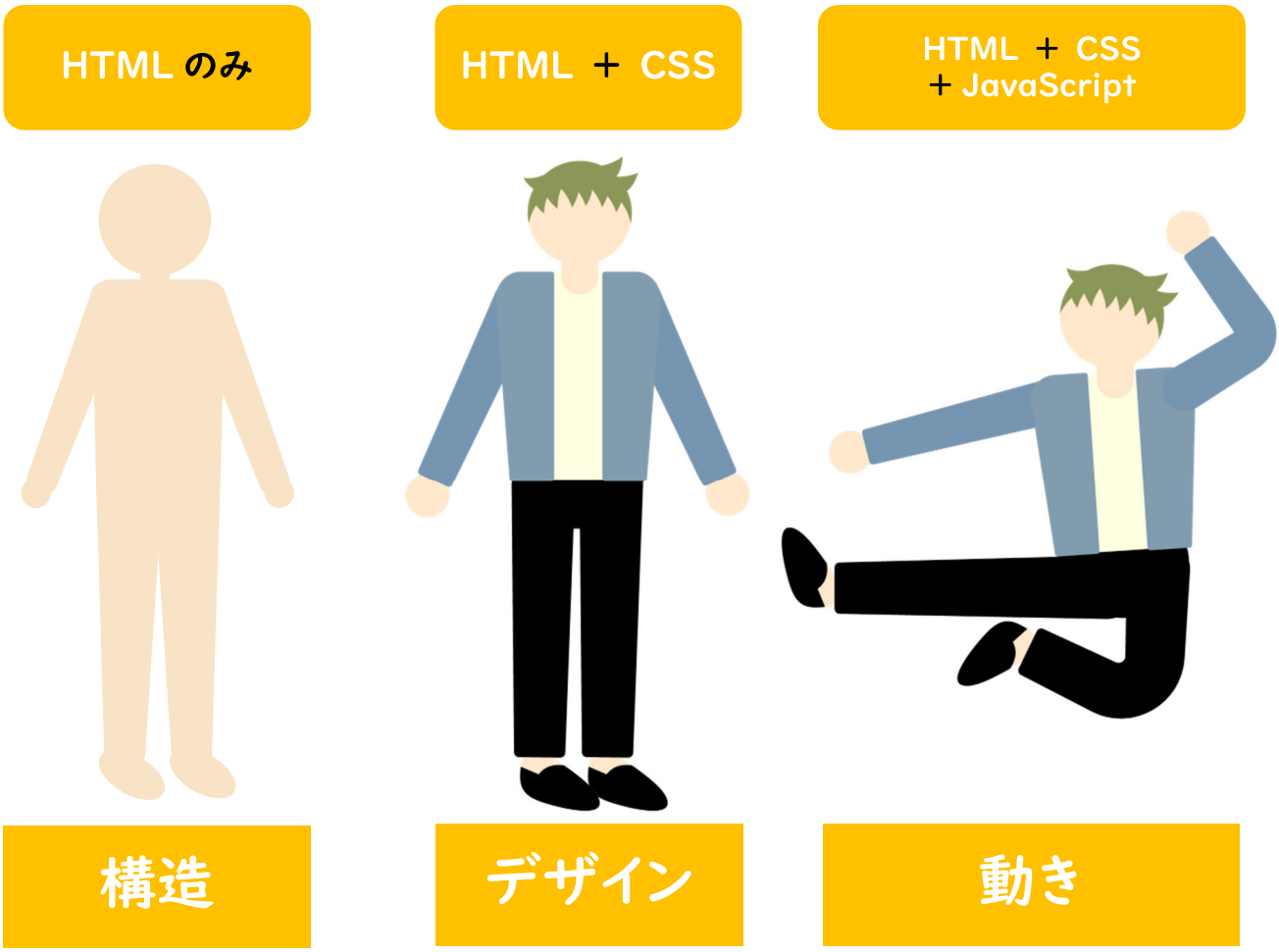
1～3

表示されている文字や画像、情報を入力する場所（チェックボックスやボタンなど）や結果を表示する場所などを 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

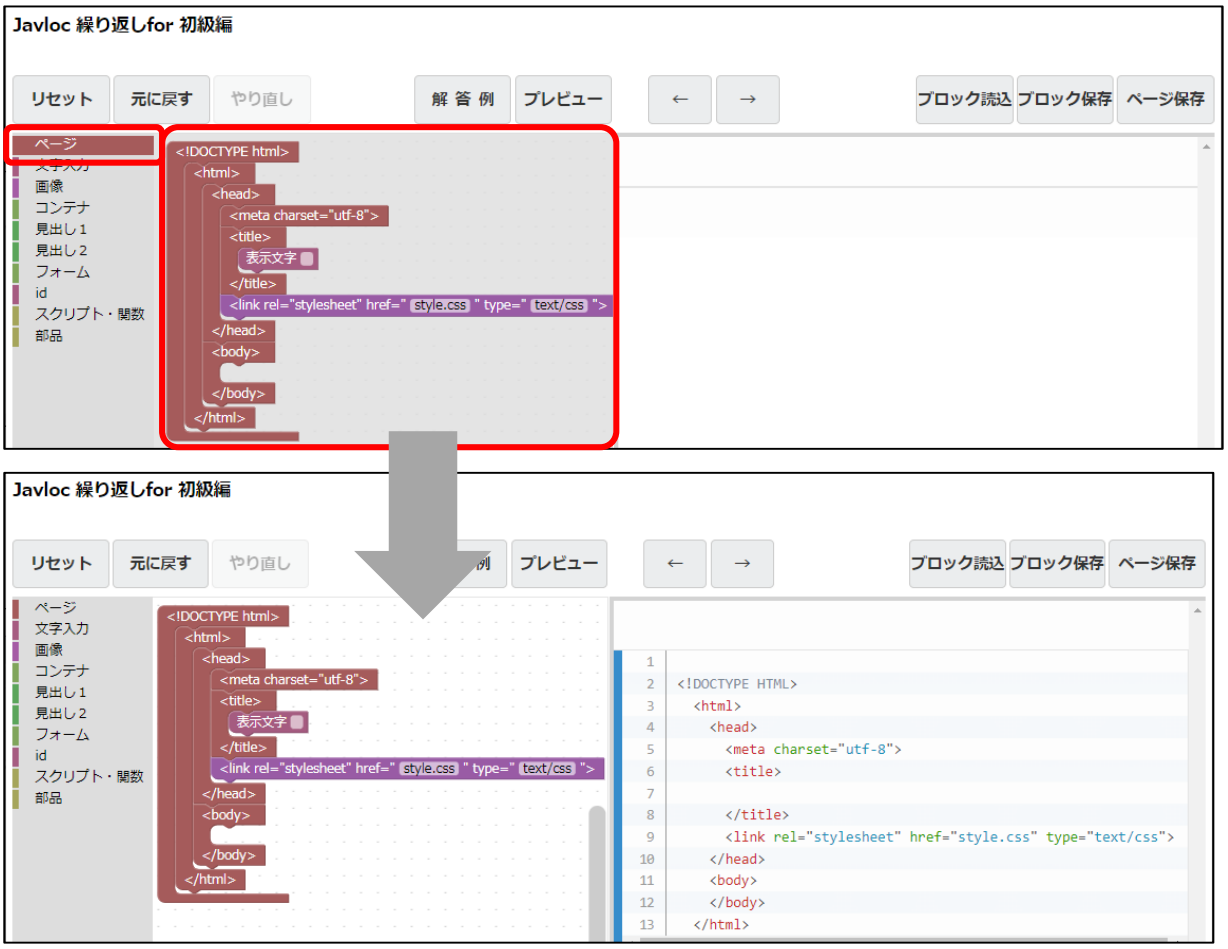


ページを構成（HTML 部分の作成）

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。



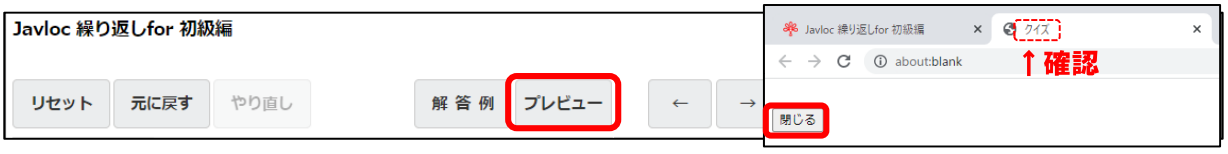
2．<title>ブロックの中にページタイトルを入力しなさい。ページタイトルは下記のとおりとする。

クイズ

<title>ブロックの表示文字ブロックの空欄に「クイズ」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。



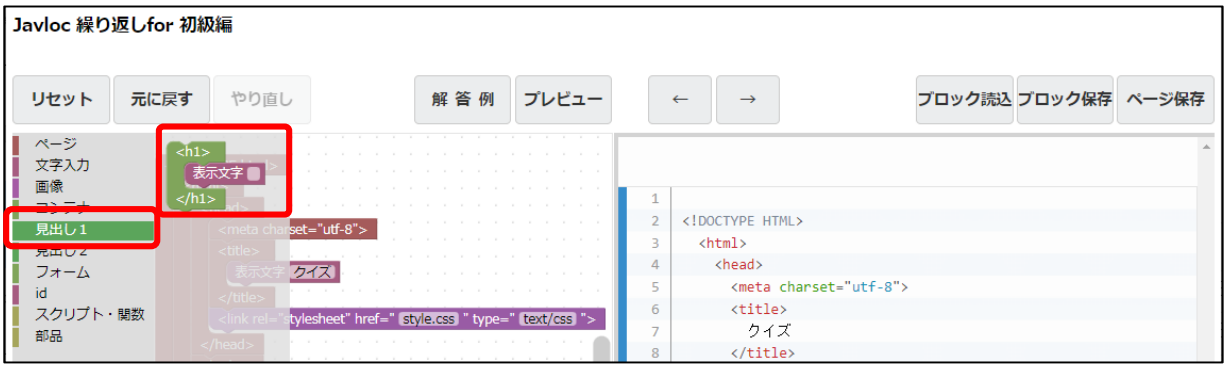
3．<body>ブロックの中に下記の(1)から順にブロックを挿入し、処理をしなさい。

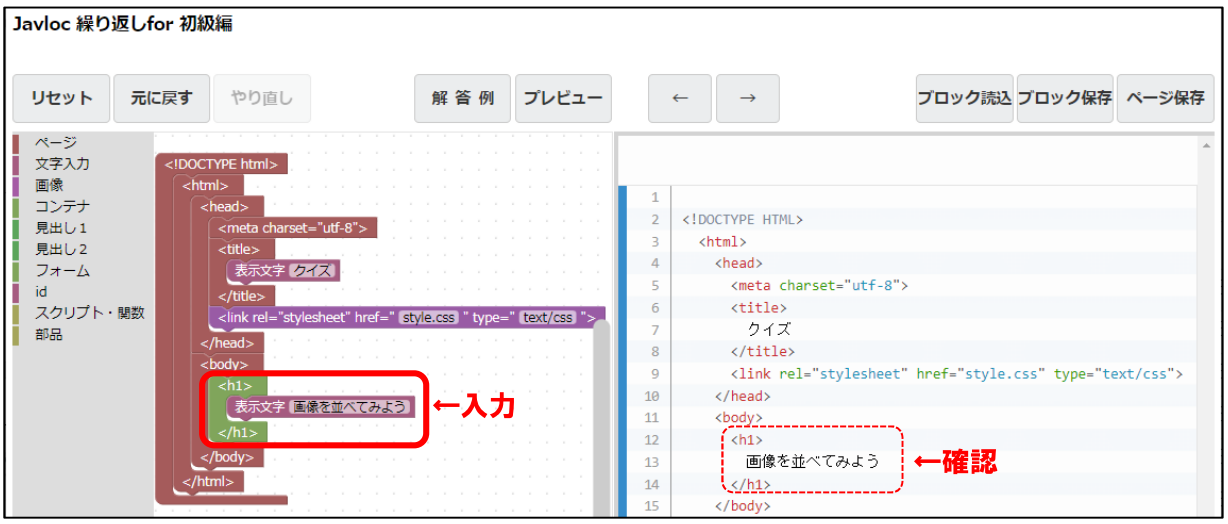
(1) 【見出し１】ブロックを挿入し、下記の文字を入力しなさい。

画像を並べてみよう

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「画像を並べてみよう」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。

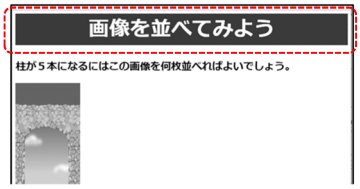




[プレビュー]ボタンをクリックし、正しく設定されているか確認。

[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。

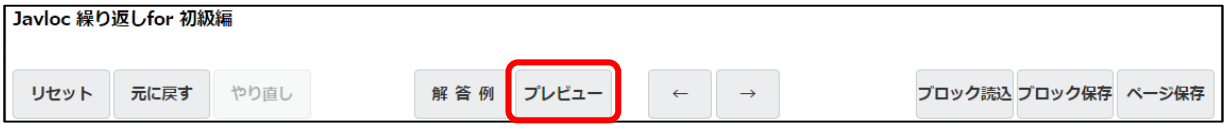
柱が５本になるにはこの画像を何枚並べればよいでしょう。

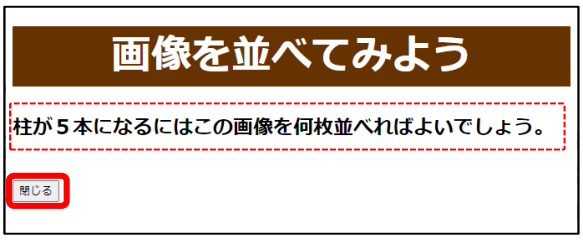
リスト内[見出し２]をクリック、枠内にある<h2>ブロックをドラッグし<h1>ブロックの下にドロップ。

表示文字ブロックの空欄に「柱が５本になるにはこの画像を何枚並べればよいでしょう。」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。

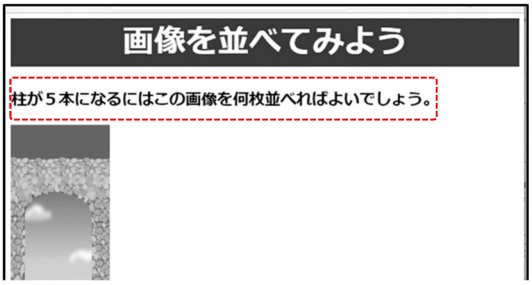
[閉じる]ボタンをクリックし、ブロック操作画面へ戻る。





←＜作成ページ例＞を参考に「見出し２」の設定が正しく行われているか確認

＜作成ページ例＞



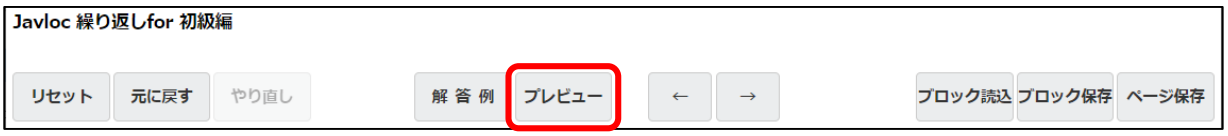
(3) 【画像】ブロックを挿入し、ファイル名を pic1.jpg にしなさい。

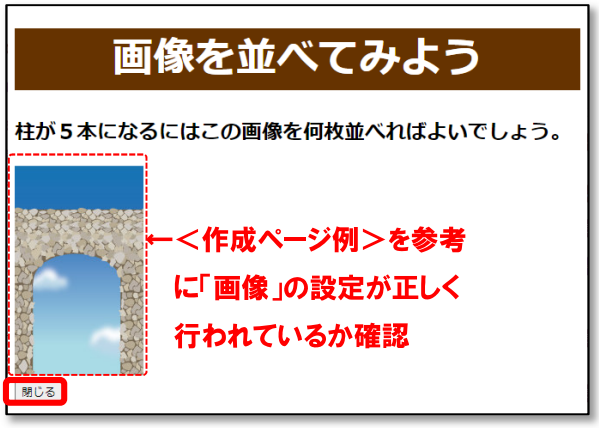
リスト内[画像]をクリック、枠内にある<img>ブロックをドラッグし<h2>ブロックの下にドロップ。 “”の空欄に「pic1.jpg」と入力し、ソース表示エリアで画像が設定されていることを確認。

※必ず半角で入力しましょう。

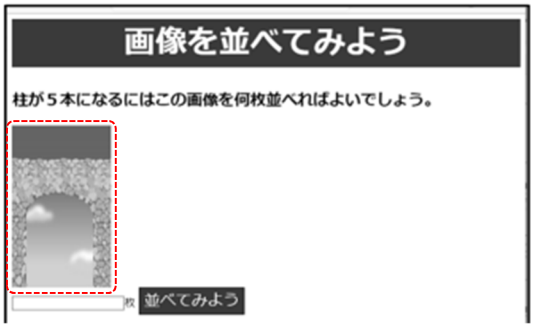
[プレビュー]ボタンをクリックし、正しく設定されているか確認。

[閉じる]ボタンをクリックし、ブロック操作画面へ戻る。



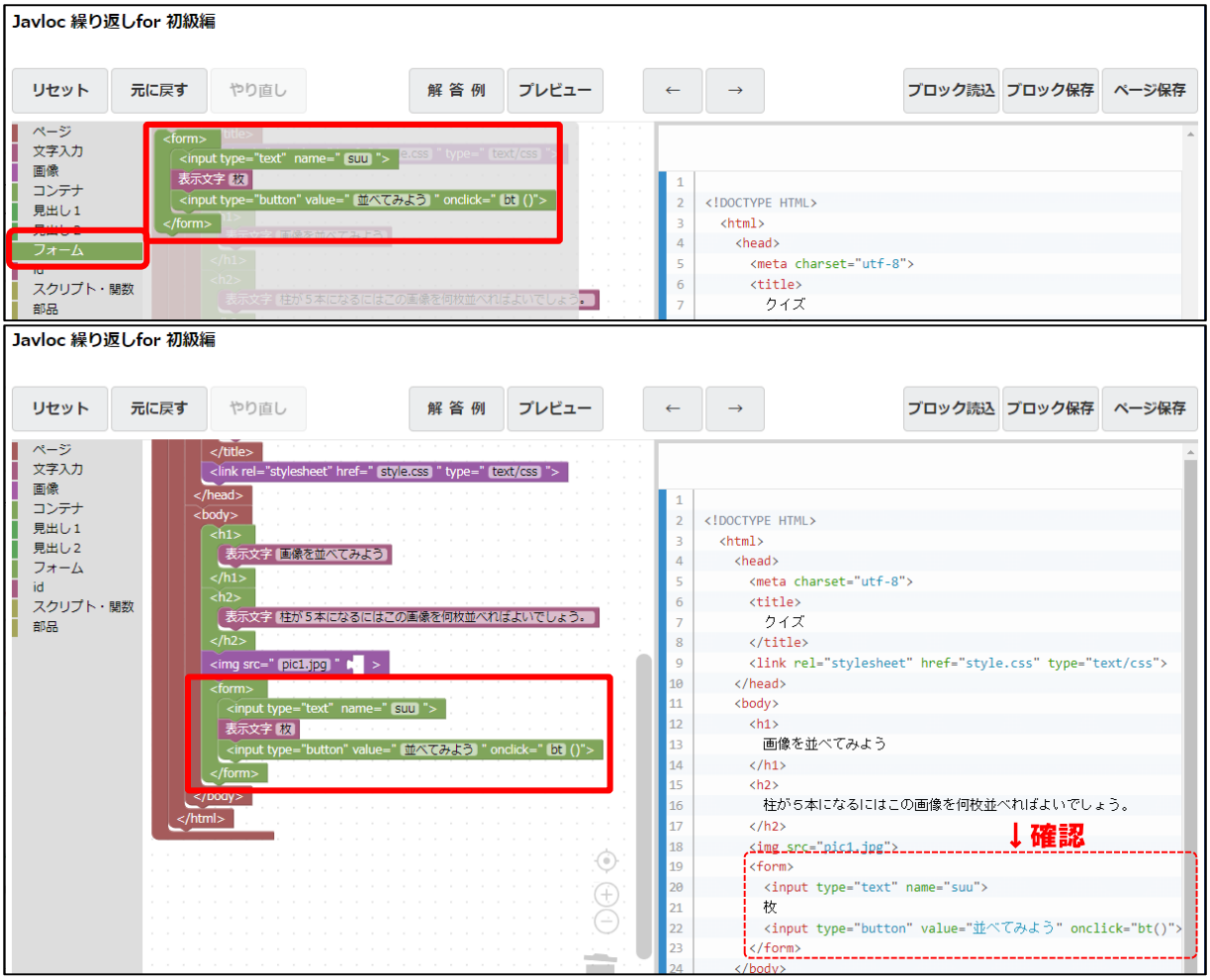


＜作成ページ例＞



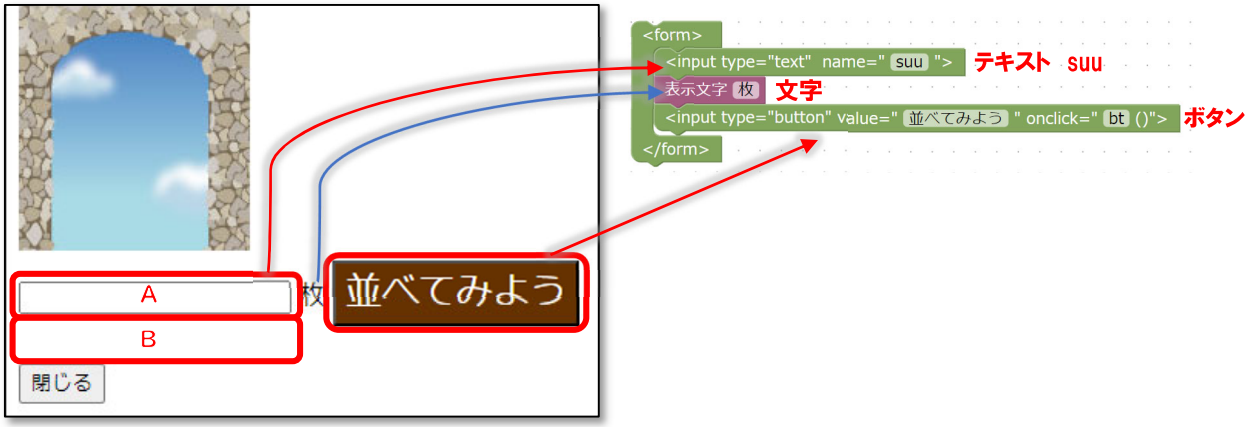
(4) 【フォーム】のブロックセット（テキスト・ボタン）を挿入しなさい。

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。



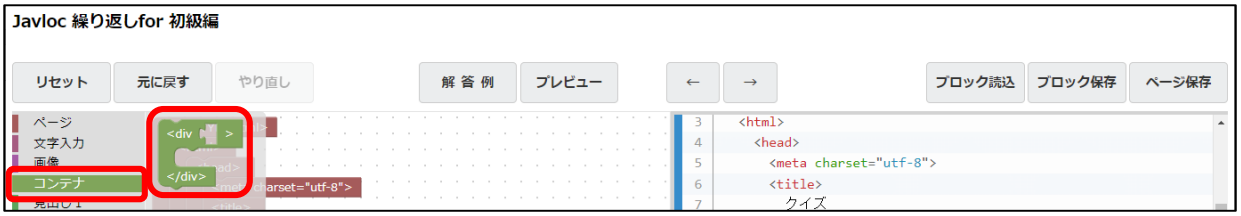
[プレビュー]ボタンをクリックし、＜作成ページ例＞を参考に「フォーム」の設定が正しく行われているか確認。[閉じる]ボタンをクリックし、ブロック操作画面へ戻る。

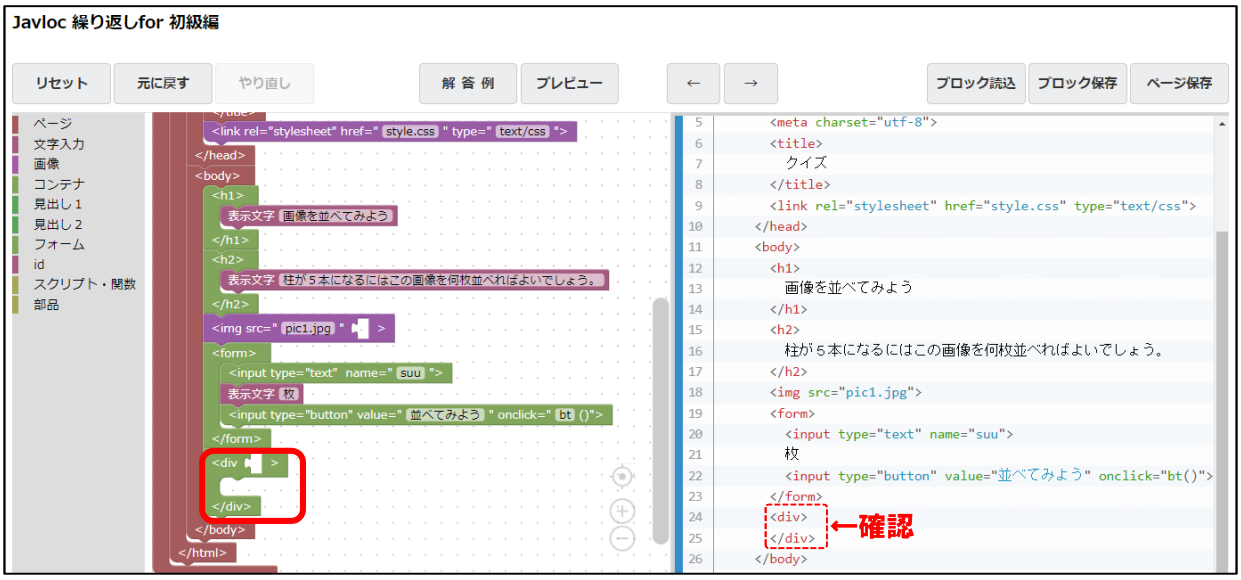
まだ JavaScript が設定されていないため、A のテキスト欄に数値を入力し「並べてみよう」ボタンをクリックしても、B の箇所には何も表示されません。この後設定する JavaScript と(5)によって、「並べてみよう」ボタンをクリックすると処理が実行され、B の箇所に結果が表示されるようになります。



(5) 【コンテナ】ブロックを挿入し、中に【id】ブロックを挿入しなさい。id 名は output。

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

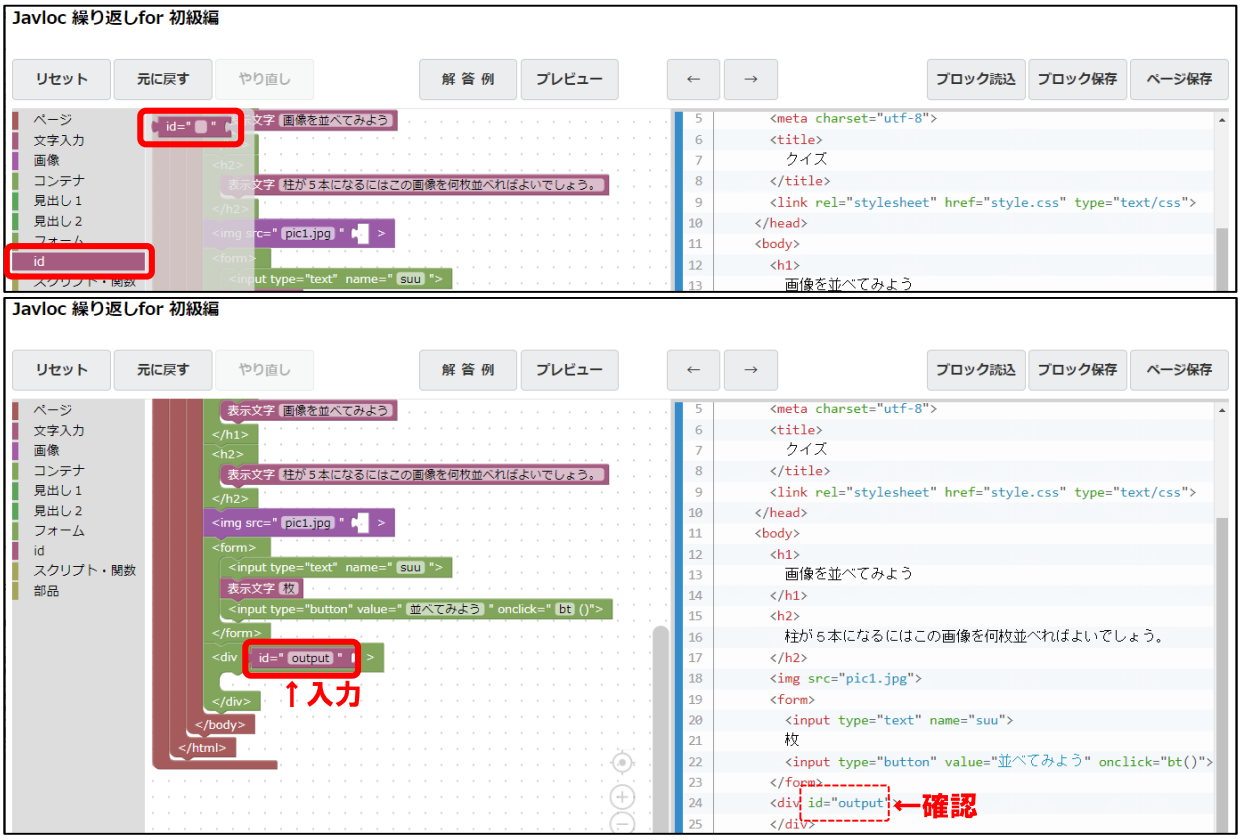




リスト内[id]をクリック、枠内にある<id>ブロックをドラッグし ブロックのパズル型空欄にドロップ。

“”の空欄に「output」と入力し、ソース表示エリアで id 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。



** ～ **は、「コンテナ」と呼ばれ、単体では特に意味を持たないタグですが、この で囲った部分を**一つのかたまりとしてグループ化する**ことができるタグです。

id とは、HTML のタグに**名前を付ける**ことができるというものです。

ここでは、「コンテナ」タグに「**output**」という名前を付けたということになります。

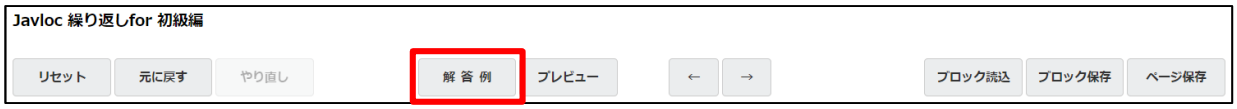
(5)の処理だけでは、プレビュー画面上では何も変化しません。JavaScript での処理によってここの表示が変化します。

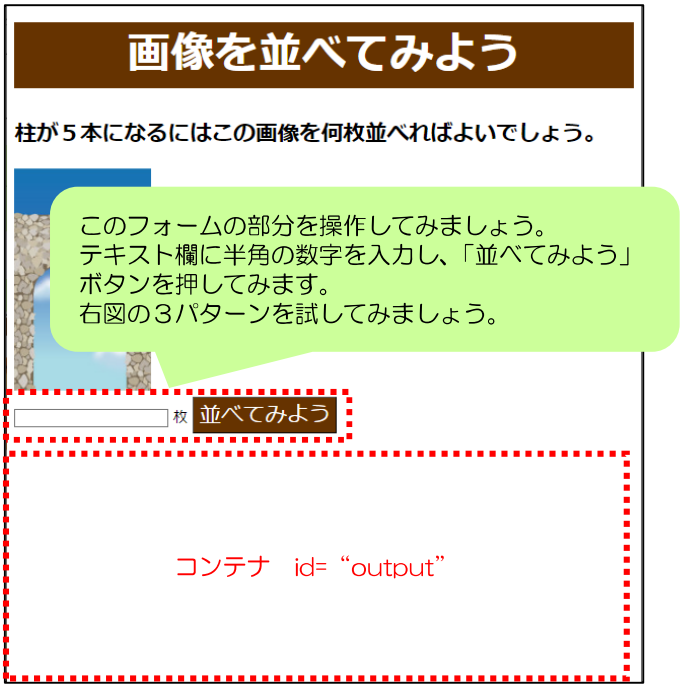
4．〈head〉ブロックの中の〈link～〉ブロックの下に、下記の指示通りブロックを挿入し、処理をしなさい。

［前提］
指定された回数分、画像を表示させる。
［機能］
・フォームのテキストに入力された値の数だけ画像を表示する。
・表示先は３－(5)の位置とする。
・画像表示の処理は、表示先をクリアした状態から始める。

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。

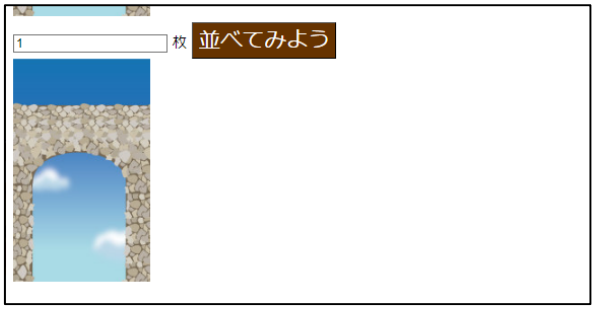
まず、[前提]と[機能]を読み、「解答例」ボタンをクリックし、解答例ページの動きを確認します。



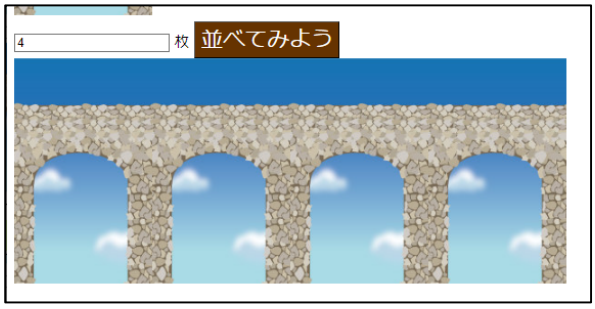


ボタンを押した際に、テキスト欄に入力した値の数だけ画像がフォームの下に表示される。
テキスト欄に入力がない状態でボタンを押した際には、何も表示しない。

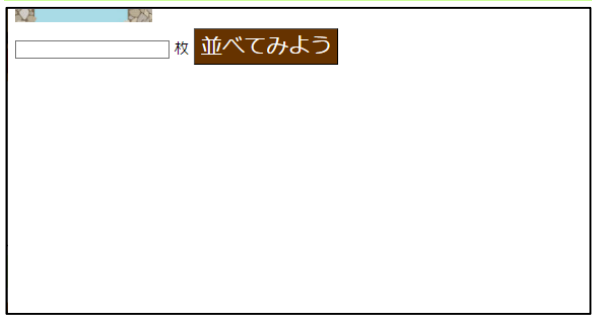
1 と入力



4 と入力

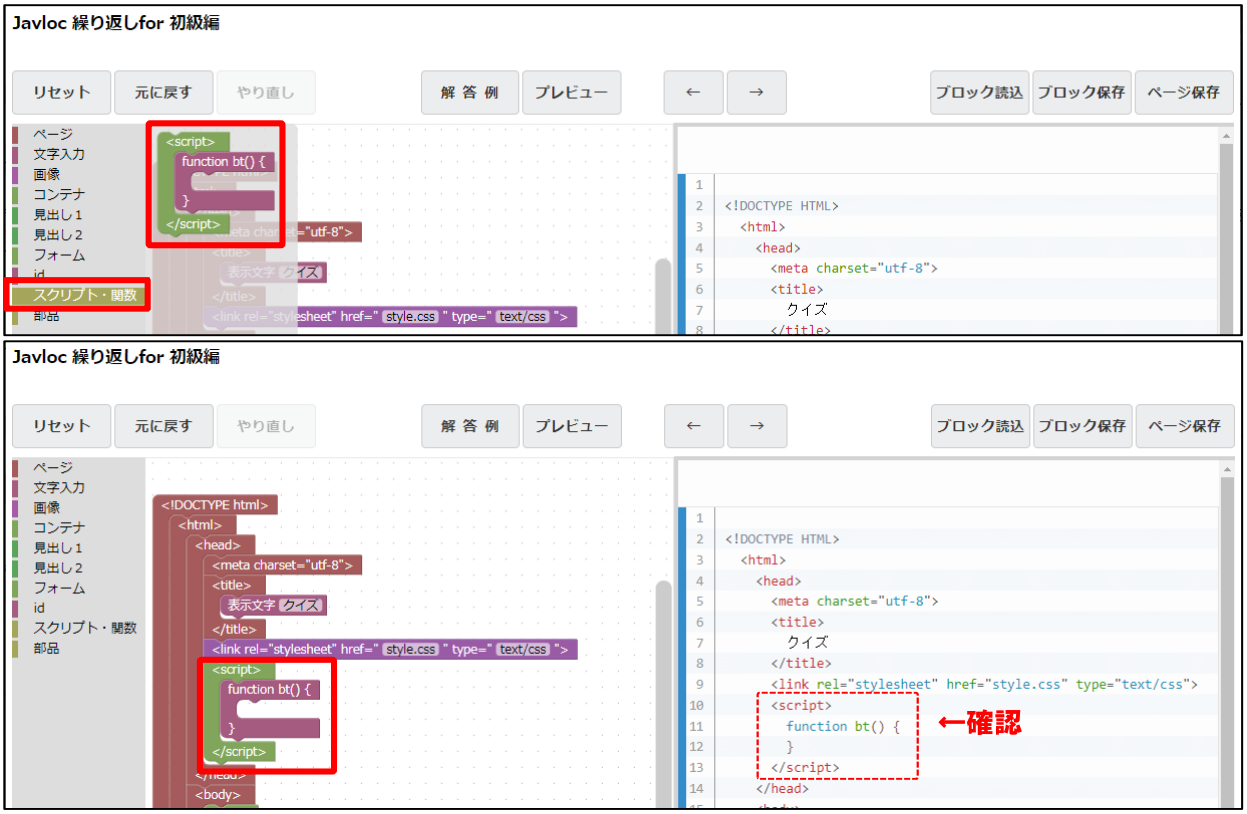


入力しない



(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。

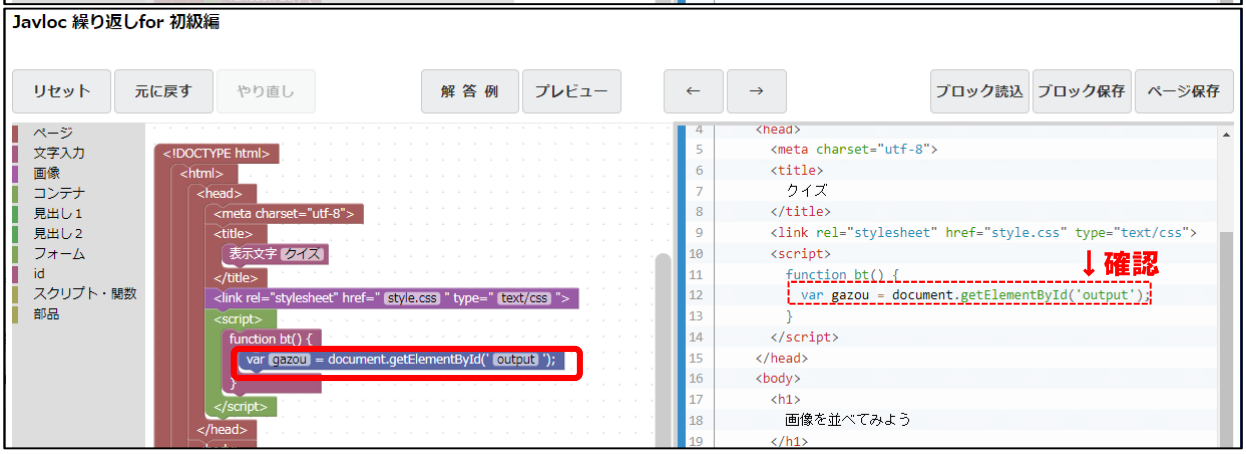
function □（ ）とは、ユーザーが作成する関数を定義する JavaScript です。□部分に関数名を記述します。これは、さまざまな処理を一つにまとめて、名前を付けることができるものです。

(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 画像を表示させる場所（id 名 output）の情報を取得し、「gazou」に設定する

リスト内[部品]をクリック、枠内にある var gazou = document.getElementById('output'); 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。



プログラムはすべて「**半角**」で入力します！そして、大文字と小文字も区別されます。

「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。

「document.getElementById('output')」は、HTML 内で特定の名前（id）が付いた要素を探し出しますよといった意味です。

これにより、その名前（id）の付いた場所を JavaScript で操作することができるようになります。

ブロックの中身は

var gazou = document.getElementById('output');

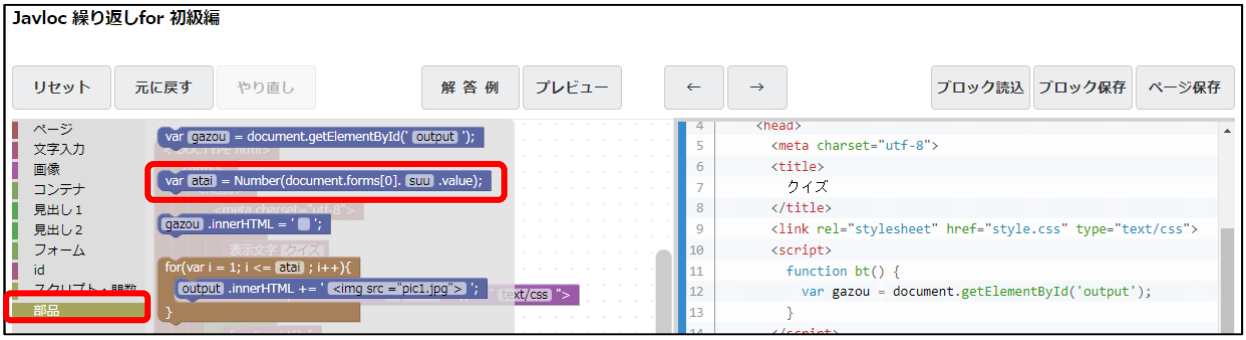
なので、これは何をしているかというと、

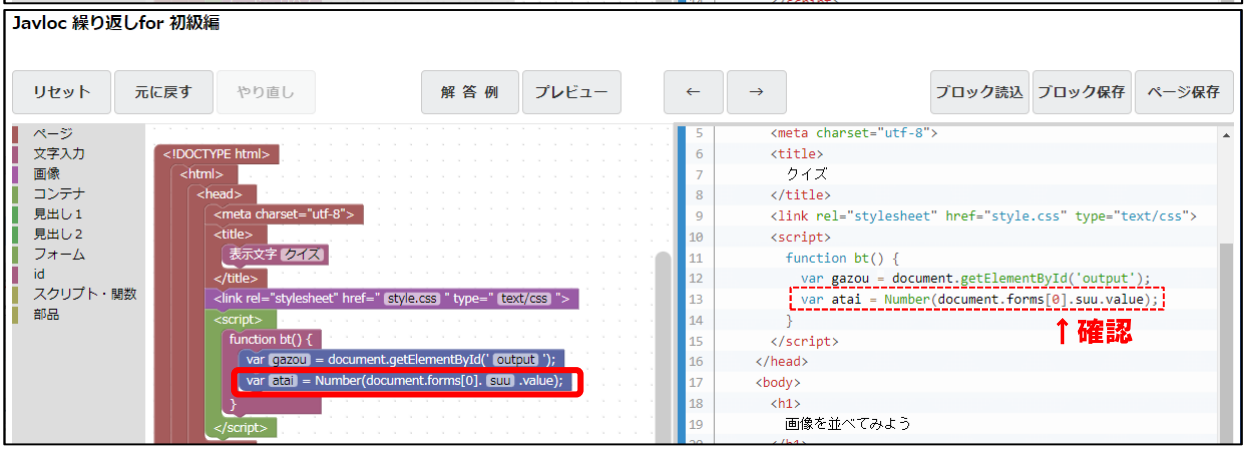
「output」という名前（id）の場所を探してきて、「gazou」という箱（変数）に入れておく。

ということになります。

② フォームのテキスト（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①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);`

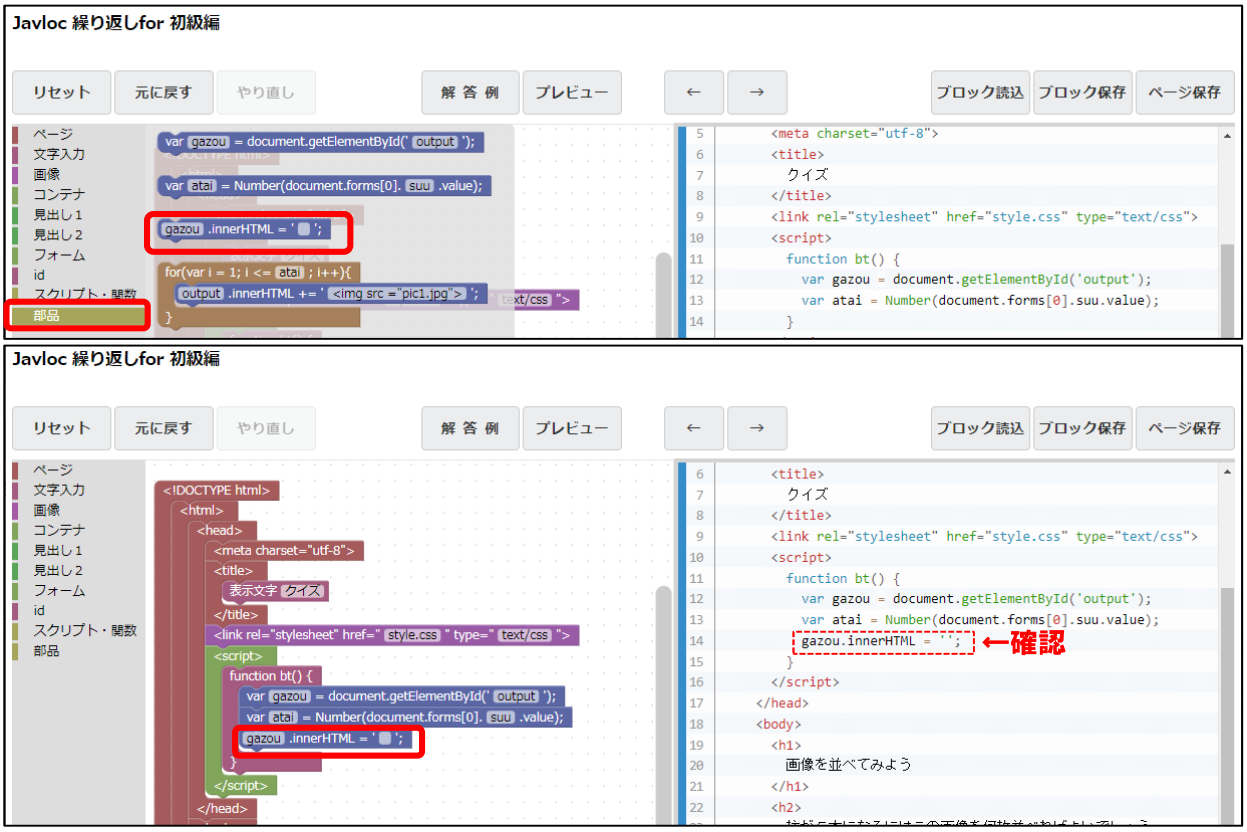
なので、これは何をしているかというと、

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。

③　「gazou」の中身を空欄にする

リスト内[部品]をクリック、枠内にある gazou.innerHTML = ''; 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。



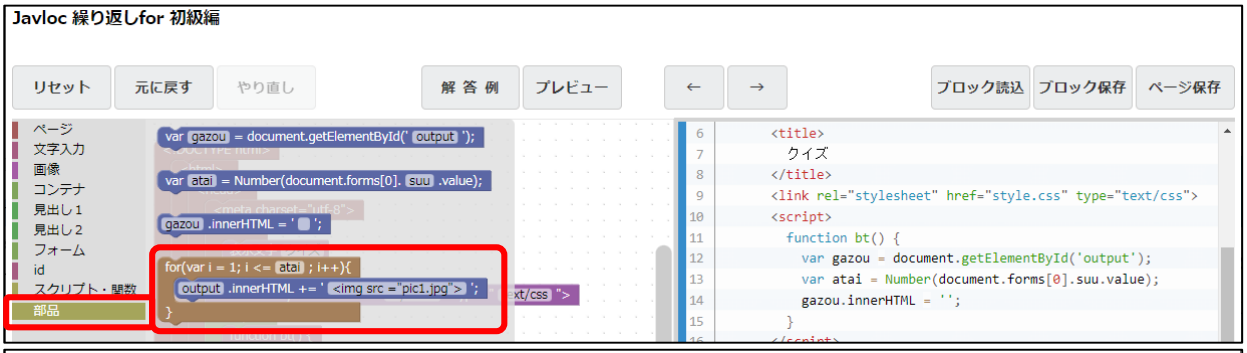
.innerHTML は、**HTML 要素の中身を変更**するという動きをします。
今回の処理は、

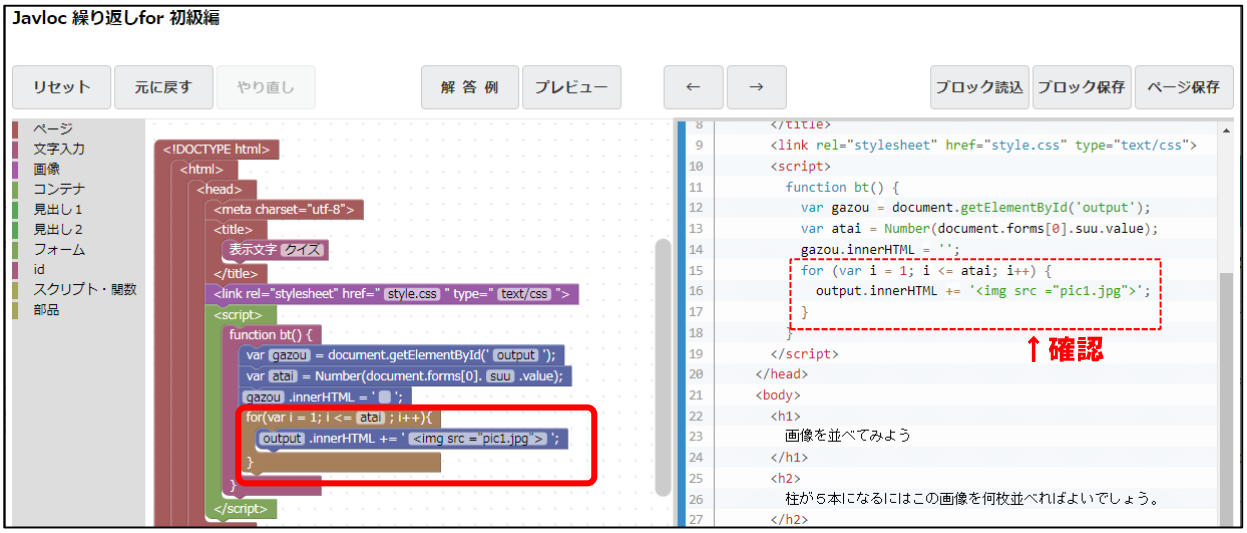
gazou.innerHTML = ‘　’;

となっており、これは変数「gazou」の HTML 要素の中身を「空欄‘　’」に変更するという意味になります。
変数「gazou」には id「output」が入っていますので、id「output」の場所を空欄に変更（何も表示させない）しています。

④ フォームに入力した回数以下になるまで、画像 pic1. jpg の表示を繰り返す

リスト内[部品]をクリック、枠内にある for (var i = 1; i <= atai; i++) {～} ブロックセットを選択し、ドラッグして③で設定したブロックの下にドロップ。





for は、**同じ処理を何度も繰り返す**ための命令文です。
例えば、「1 から 10 までの数字を順番に表示する」という処理をする場合、

```
for (var i = 1; i <= 10 ; i + +) {
   表示する処理
}
```

変数 i に 1 を代入し、**i が 10 以下の間**、**i を 1 ずつ増やしながら**「表示する処理」を繰り返す。
ただ、上記の具体的な流れはそこまで意識する必要はなく、「10」の部分に繰り返したい回数を入れればいいと覚えておくといいでしょう。

それを踏まえて、ブロックの中身を見てみましょう。

```
for (var i = 1; i <= atai ; i + +) {
   gazou.innerHTML += '<img src ="pic1.jpg">';
}
```

変数 atai 以下になるまで

gazou.innerHTML += '<img src ="pic1.jpg">';
の処理を繰り返し行うという動きです。

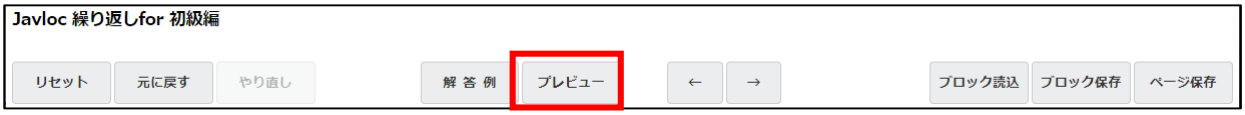
←変数 gazou の中身に<img src ="pic1.jpg">を追加する

<img src ="pic1.jpg">は、pic1.jpg という画像を表示するタグなので
pic1.jpg という画像を atai の回数分繰り返し追加し、表示をするという処理になります。

＜作成ページの動きを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 for　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

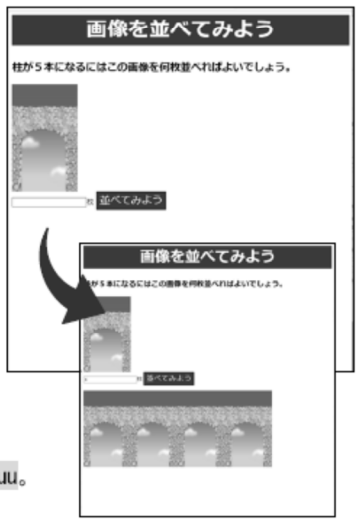
プログラミング問題　繰り返しfor 中級

＜問　　題＞

1．【繰り返し for 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。

［クイズ］

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。

［画像を並べてみよう］

(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。

［柱が５本になるにはこの画像を何枚並べればよいでしょう。］

(3)　【画像】ブロックを挿入し、ファイル名を pic1. 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　【フォーム】からテキストのブロックを選択・挿入しなさい。name は suu。

②　【文字入力】ブロックを挿入し、下記の文字を入力しなさい。

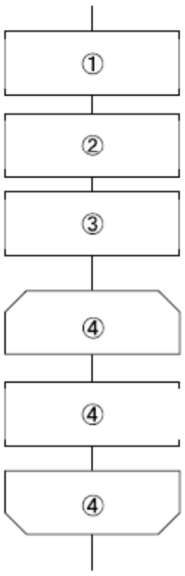
［枚］

③　【フォーム】からボタンのブロックを選択・挿入しなさい。value は並べてみよう、onclick は関数名 bt。

(5)　【コンテナ】ブロックを挿入し、中に【id】ブロックを挿入しなさい。id 名は output。

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。

＜フローチャート＞



［前提］

指定された回数分、画像を表示させる。

［機能］

・フォームのテキストに入力された値の数だけ画像を表示する。

・表示先は３－(5)の位置とする。

・画像表示の処理は、表示先をクリアした状態から始める。

(1)　【スクリプト・関数】ブロックを挿入し、３－(4)－③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 gazou を宣言し、id（output）を取得し代入

②　変数 atai を宣言し、フォームのテキスト（suu）の値を数値に変換し代入

③　変数 gazou の場所にある HTML を空欄に置き換える

④　変数 atai の値の回数分、下記の処理を繰り返す

　　変数 gazou の場所にある HTML に<img src ="pic1. jpg">を追加

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

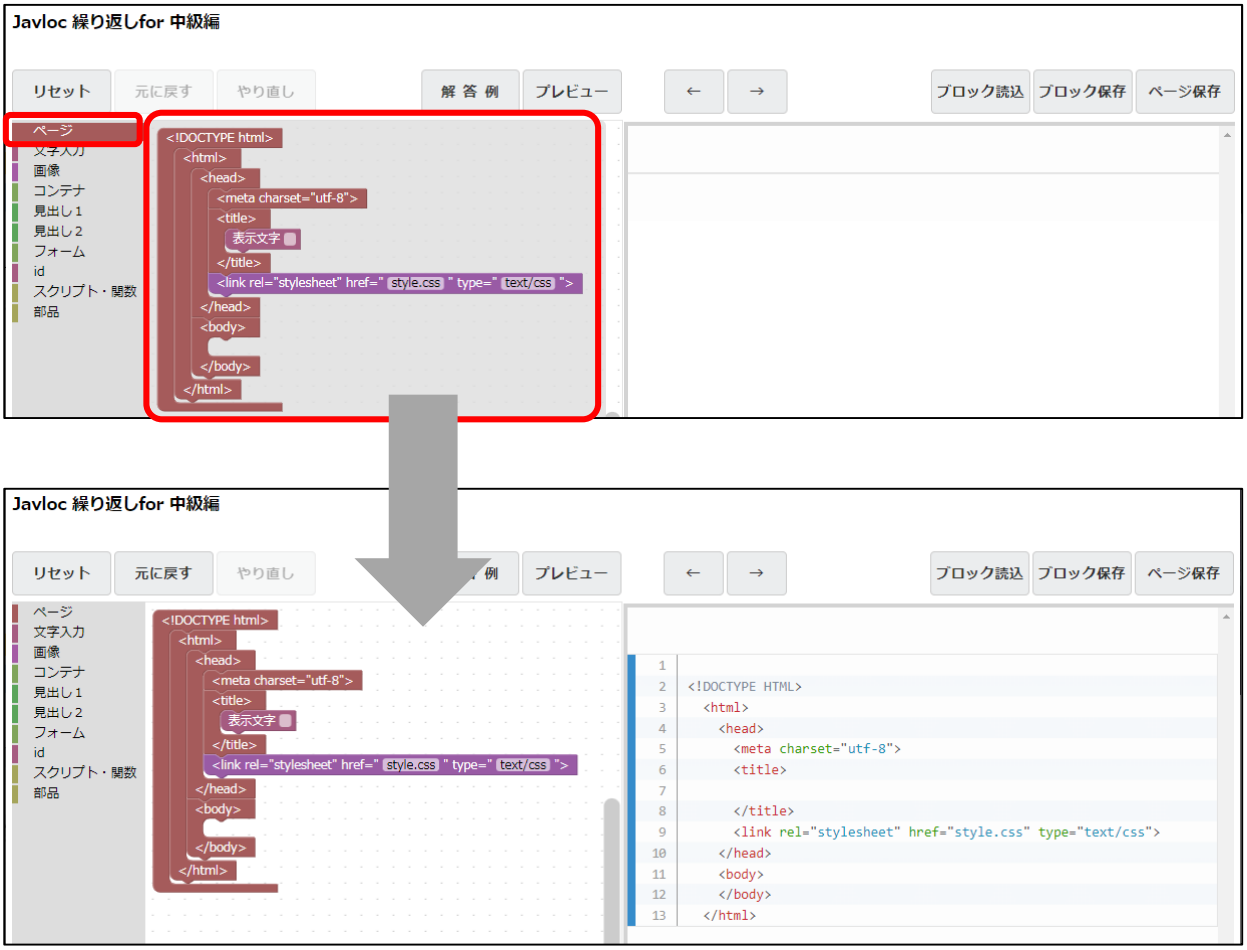
Javloc 教材サイトから[中級プロジェクト]の[繰り返し「for」中級]をクリック。

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。

クイズ

<title>ブロックの表示文字ブロックの空欄に「クイズ」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。

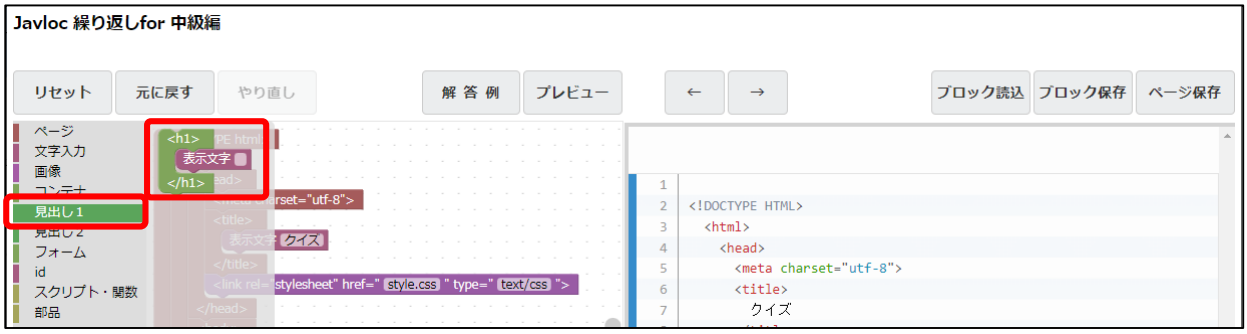
3．<body>ブロックの中に下記の(1)から順にブロックを挿入し、処理をしなさい。

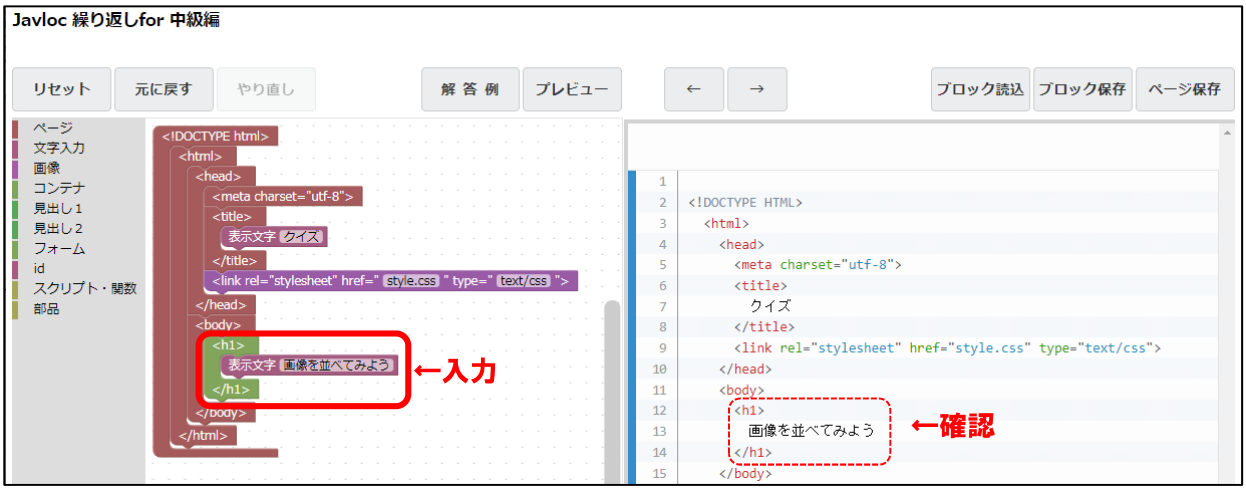
(1) 【見出し１】ブロックを挿入し、下記の文字を入力しなさい。

画像を並べてみよう

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「画像を並べてみよう」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。

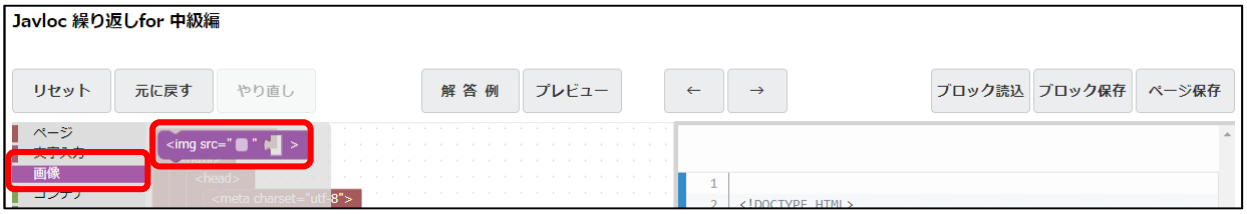
柱が５本になるにはこの画像を何枚並べればよいでしょう。

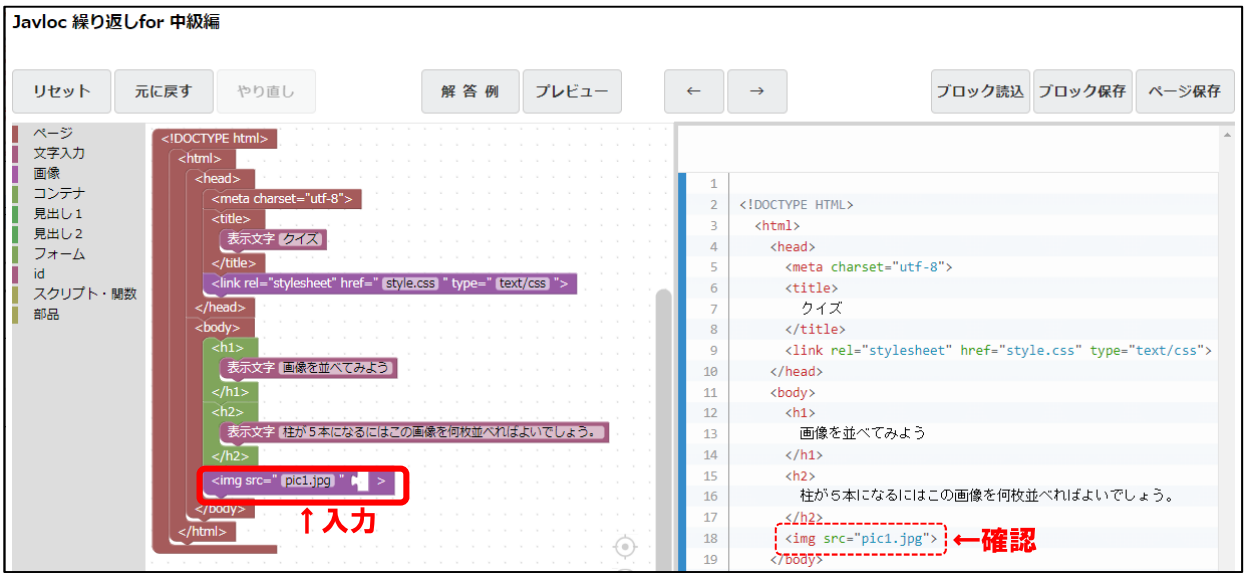
リスト内[見出し２]をクリック、枠内にある<h2>ブロックをドラッグし、<h1>ブロックの下にドロップ。

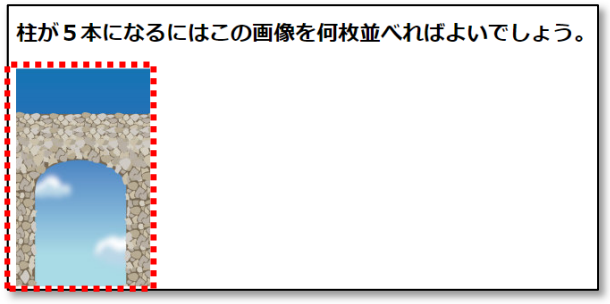
表示文字ブロックの空欄に「柱が５本になるにはこの画像を何枚並べればよいでしょう。」と入力し、ソース表示エリア・プレビューで見出し２が設定されていることを確認。

(3) 【画像】ブロックを挿入し、ファイル名を pic1. 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「pic1.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。

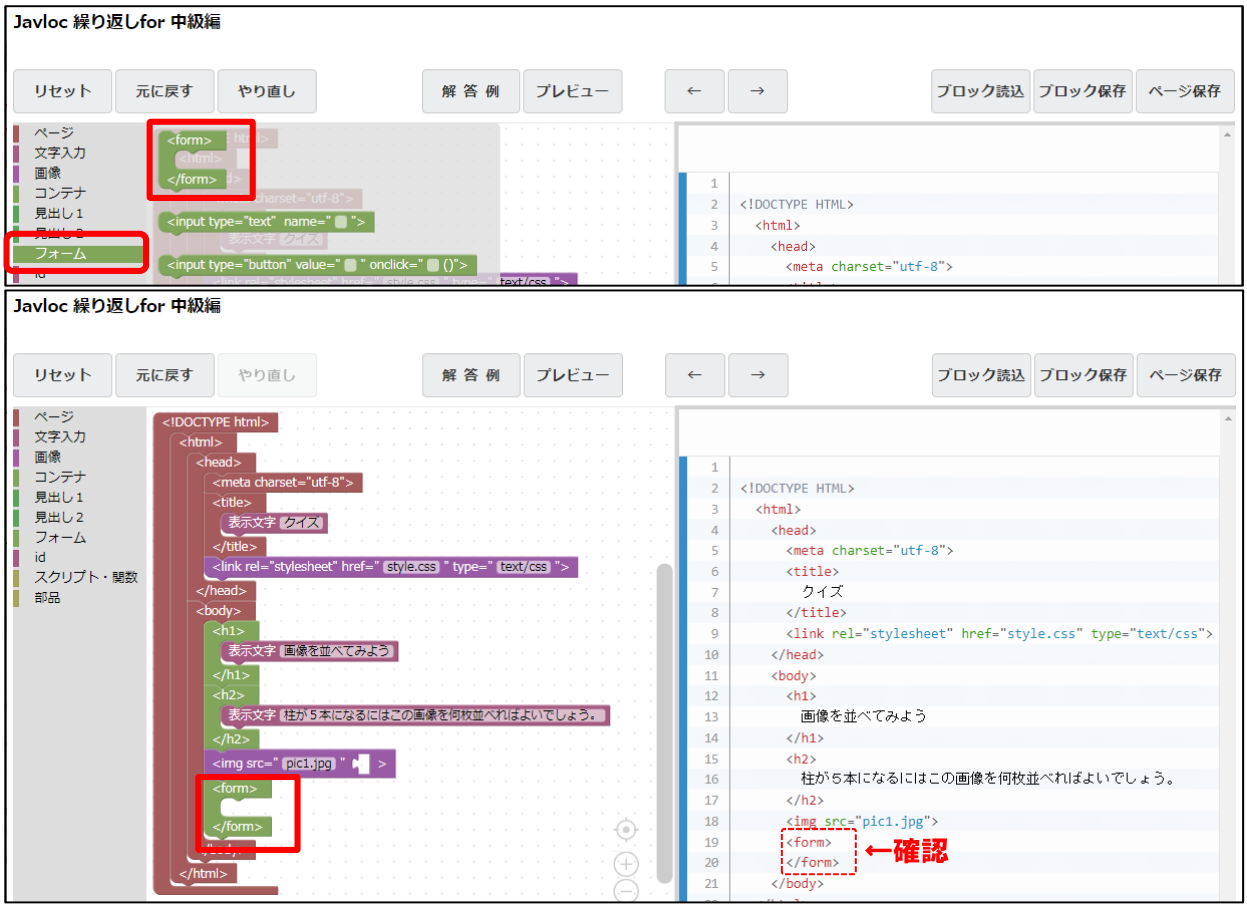






(4) 【フォーム】から<form>ブロックを選択・挿入し、その中に下記のブロックを挿入し、処理をしなさい。

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。



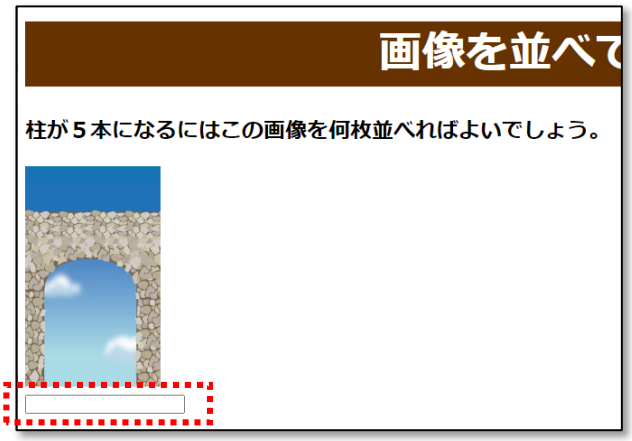
<form>タグは、**入力欄や送信ボタンを作成する際に使用する要素**です。ここで入力された情報はプログラムを使用して Web サーバーへ送信することができるので、アンケートや会員登録などのページで利用されています。

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 【フォーム】からテキストのブロックを選択・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="text" ～>ブロックをドラッグし<form>ブロックの中にドロップ。“”の空欄に「suu」と入力し、ソース表示エリア・プレビューでフォーム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。

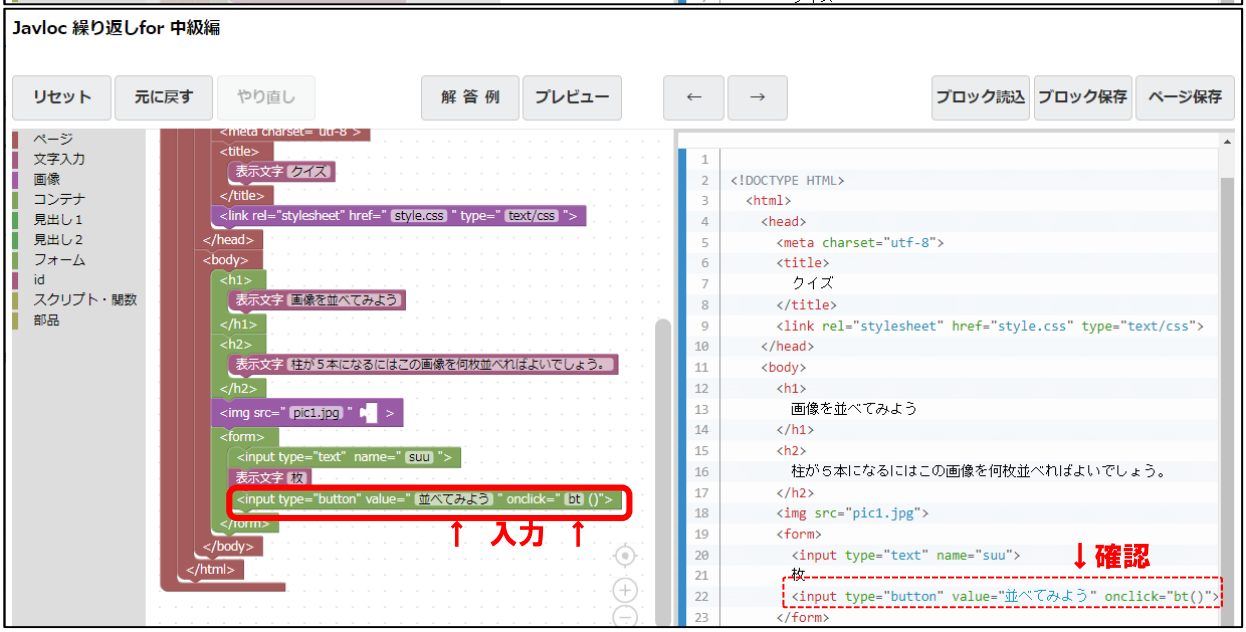


② 【文字入力】ブロックを挿入し、下記の文字を入力しなさい。
枚

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="text" ～>ブロックの下にドロップ。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。

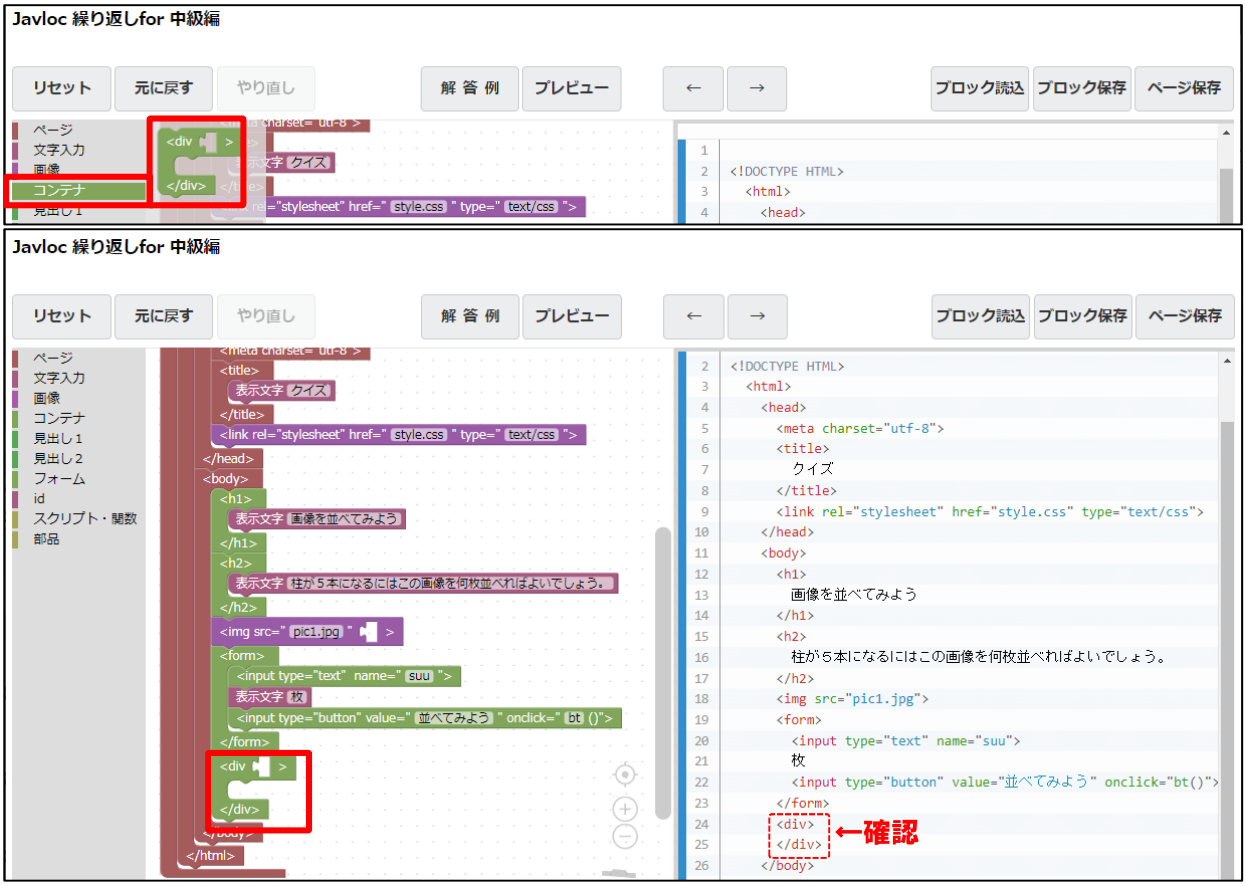
③　【フォーム】からボタンのブロックを選択・挿入しなさい。value は並べてみよう、onclick は関数名 bt。

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「並べてみよう」、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。



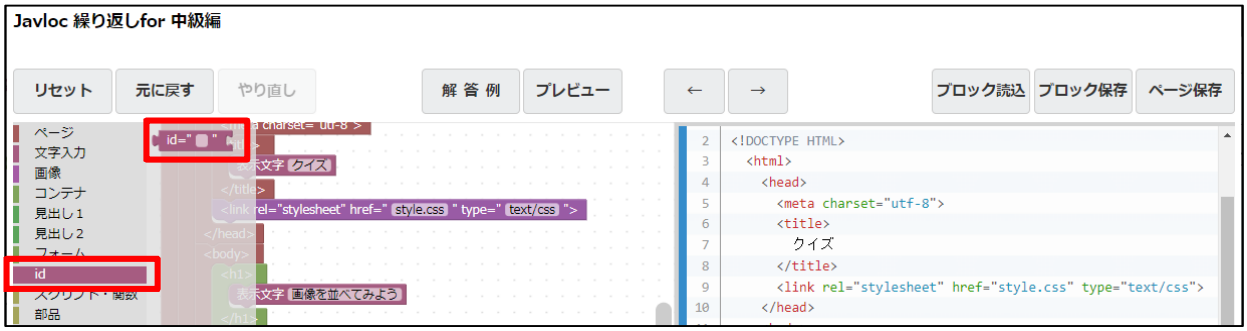
(5) 【コンテナ】ブロックを挿入し、中に【id】ブロックを挿入しなさい。id 名は output。

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



リスト内[id]をクリック、枠内にある<id>ブロックをドラッグし ブロックの空欄にドロップ。“”の空欄に「output」と入力し、ソース表示エリアで id 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．<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］
　指定された回数分、画像を表示させる。
［機能］
・フォームのテキストに入力された値の数だけ画像を表示する。
・表示先は３－(5)の位置とする。
・画像表示の処理は、表示先をクリアした状態から始める。

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。

1 と入力

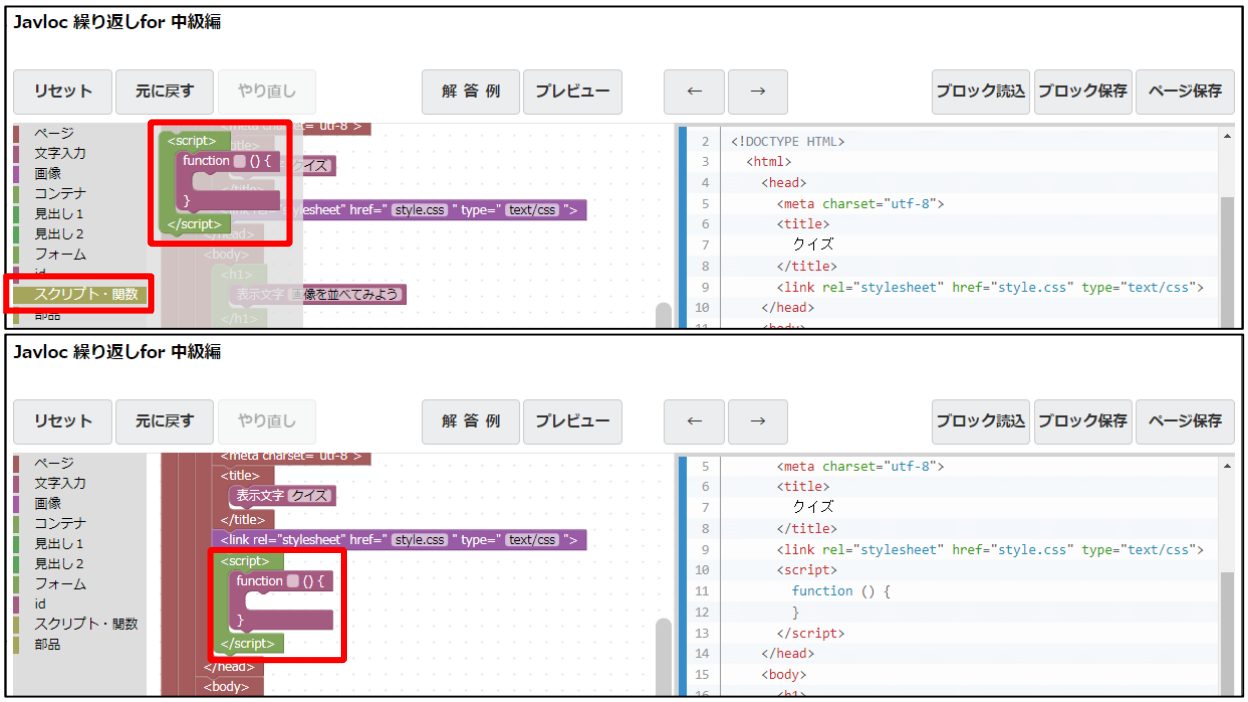
4 と入力

入力しない

ボタンを押した際に、テキスト欄に入力した値の数だけ画像がフォームの下に表示される。
テキスト欄に入力がない状態でボタンを押した際には、何も表示しない。

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

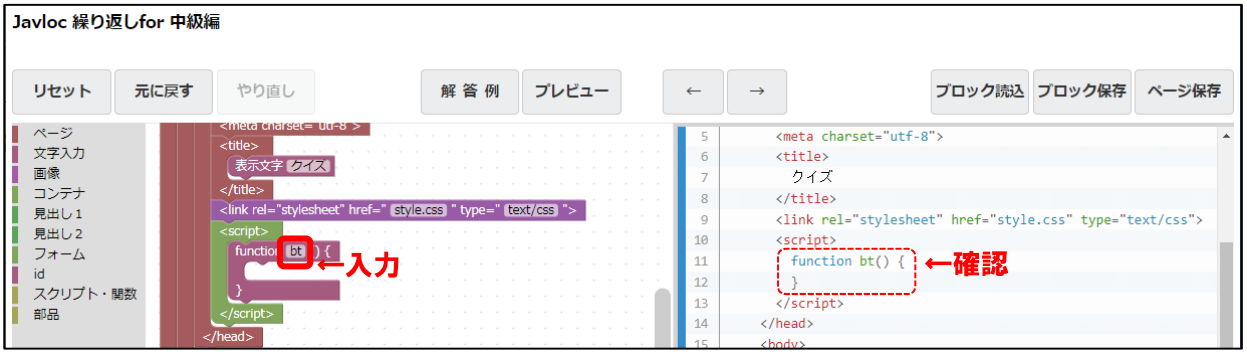
リスト内[スクリプト・関数]をクリック、枠内にある<script>ブロックセットをドラッグし、<link>ブロックの下にドロップ。



3－(4)－③で挿入・入力したボタンのブロック <input type="button" value=" 並べてみよう " 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。

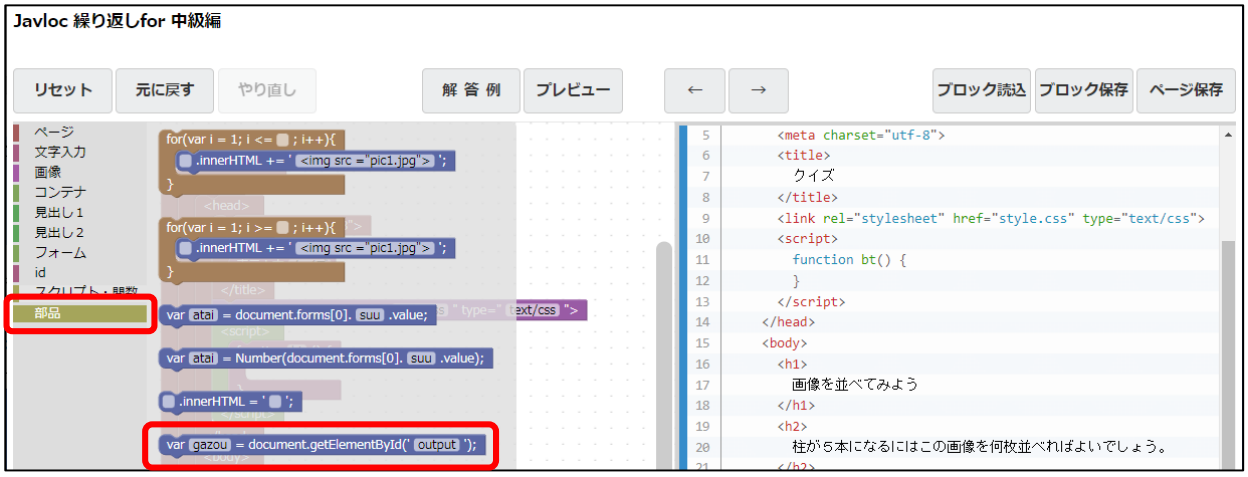
(2)　【部品】内から正しいブロックを選択し、(1)の関数ブロックの中に下記処理手順どおりに組み立てなさい。

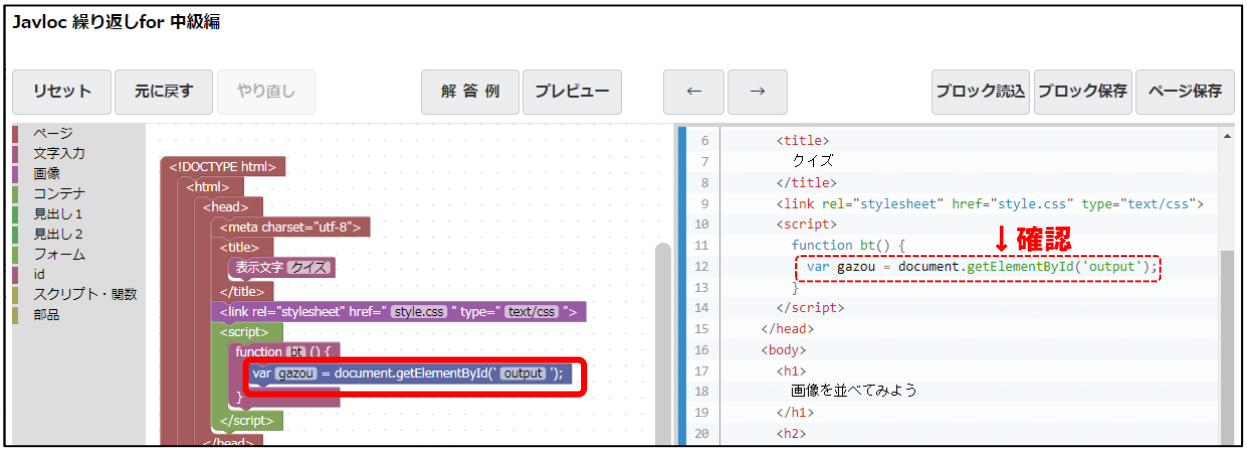
■処理手順

①　変数 gazou を宣言し、id（output）を取得し代入

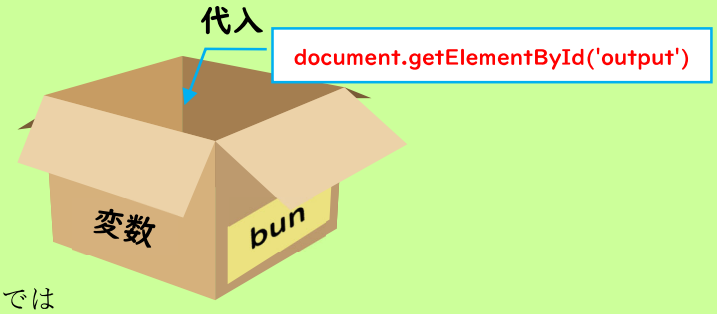
リスト内[部品]をクリック、枠内にある var gazou =　　; ブロックを選択し、ドラッグして function bt (){～}ブロックの中にドロップ。

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。





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



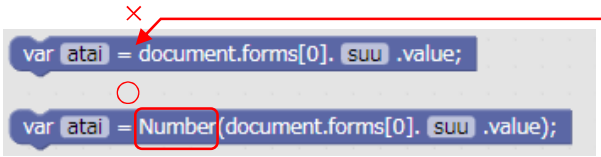
document.getElementById('output')では
HTML にある output という id 要素を取ってくる処理を行います。
それを変数 gazou に代入しています。

②　変数 atai を宣言し、フォームのテキスト（suu）の値を数値に変換し代入

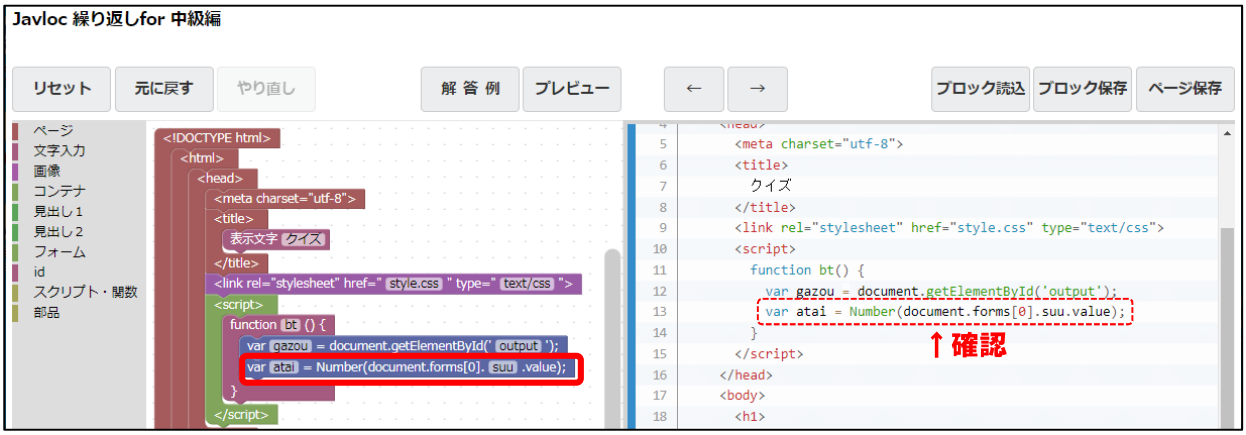
リスト内[部品]をクリック、枠内にある var atai = Number　; ブロックを選択し、ドラッグして var gazou =　　;ブロックの下にドロップ。

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()がない
処理条件に「数値に変換し」とあるため、
数値変換の処理が必要です。



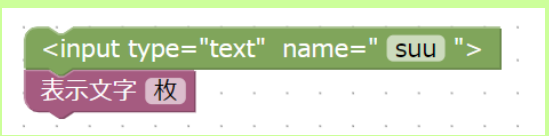
「Number(～)」は、（～）内にある値を**数値に変換**します。

「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。

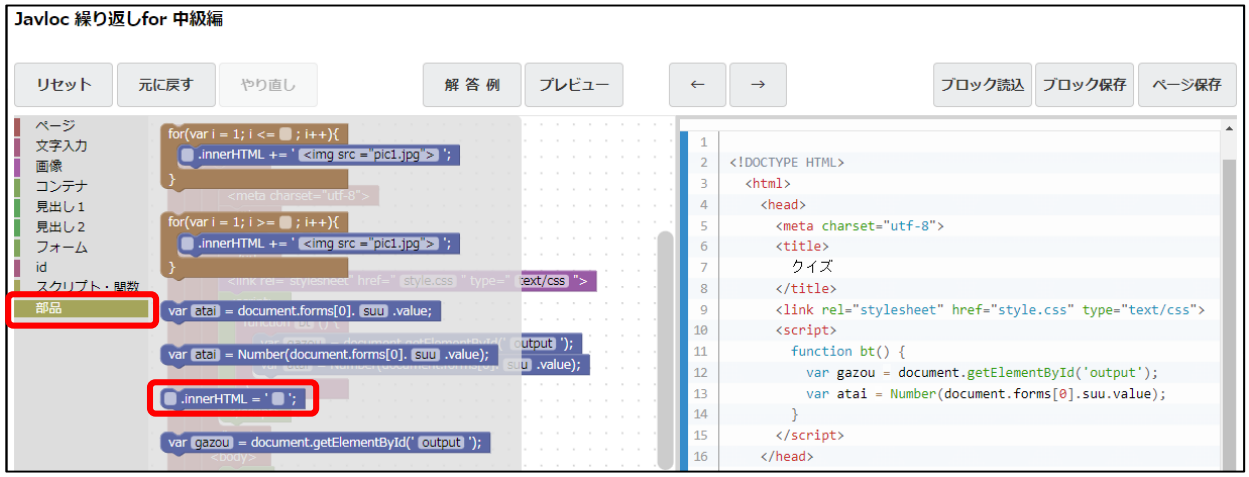
フォームで入力された値は**文字列**として扱われます。このあと設定する④の for の処理で数値として使用するため、Number(～)を用いて**数値に変換**する必要があるのです。

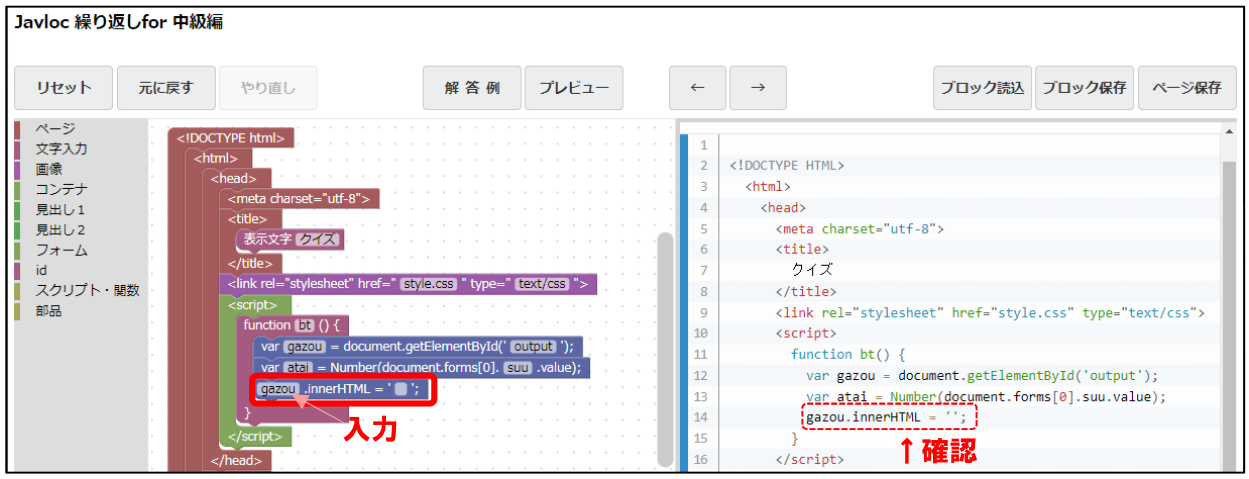
このように、型の処理をしていないと、正しく動作しません。



③　変数 gazou の場所にある HTML を空欄に置き換える

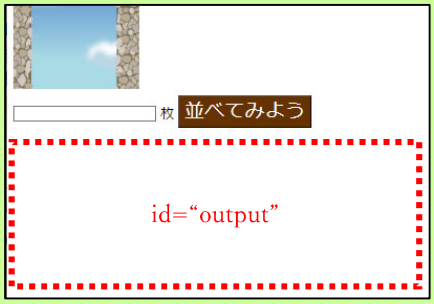
リスト内[部品]をクリック、枠内にある□.innerHTML = ‘ ’; ブロックを選択し、ドラッグして var atai = Number　;ブロックの下にドロップ。



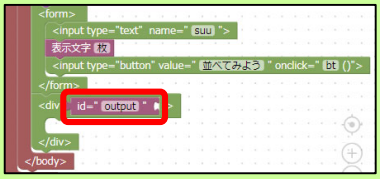


.innerHTML は、**HTML の中身を変更**する処理を行います。
変数 gazou には、id（output）が代入されています。（処理手順①にて設定）
gazou.innerHTML = ‘　’; は、id = “output” にある HTML の中身を空欄に置き換える（何も表示しない）という処理を行っています。

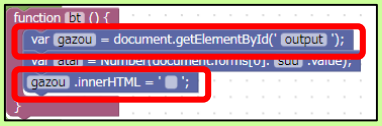
＜作成ページ例＞div 部分



<HTML>



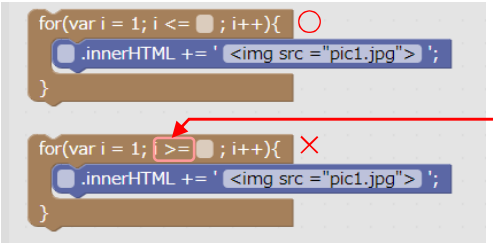
<JavaScript>



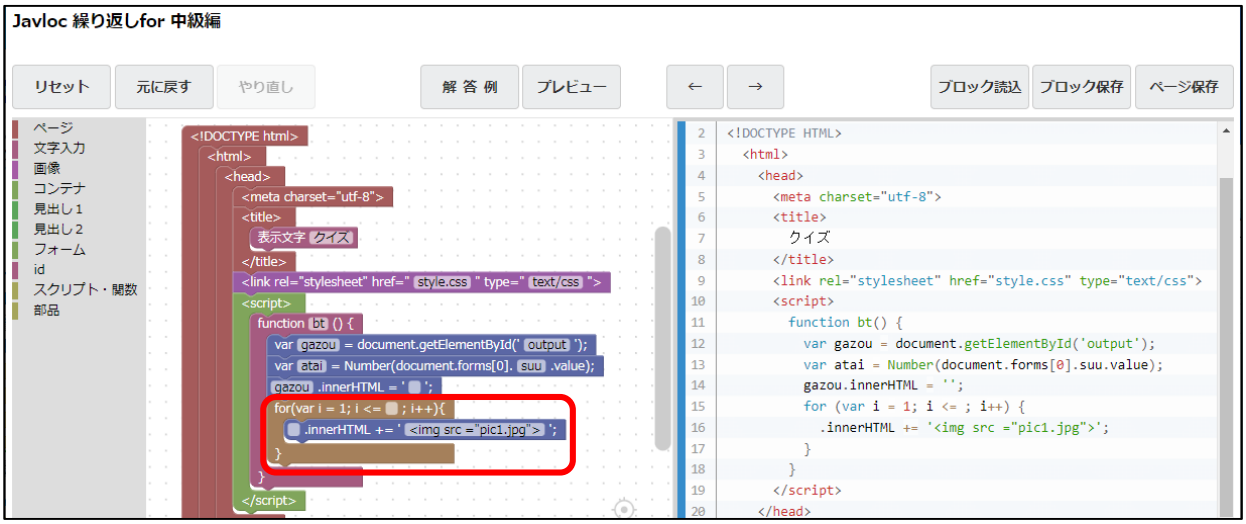
④　変数 atai の値の回数分、下記の処理を繰り返す
　　変数 gazou の場所にある HTML に<img src ="pic1.jpg">を追加

リスト内[部品]をクリック、枠内にある for(var i =1; <=　;i++){～}のブロックセットを選択し、ドラッグして gazou.innerHTML = ‘ ’ ;ブロックの下にドロップ。

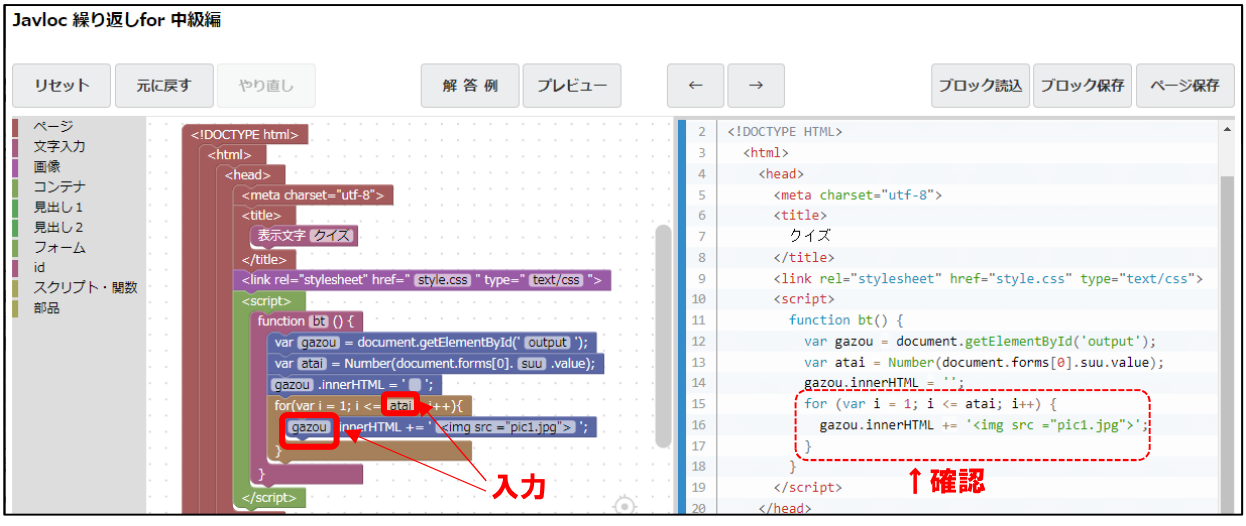
[部品]の中には、誤ったブロックが混ざっています。



不等号が逆
atai の回数分→atai を以下となっている必要あり



for の空欄に「atai」と入力。.innerHTML～の前の空欄に「gazou」と入力。
※必ず半角で入力しましょう。



for は、**同じ処理を何度も繰り返す**ための命令文です。
例えば、「1 から 10 までの数字を順番に表示する」という処理をする場合、

```
for (var i = 1; i <= 10 ; i + +) {
   表示する処理
}
```

変数 i に 1 を代入し、**i が 10 以下の間**、**i を 1 ずつ増やしながら**「表示する処理」を繰り返す。
具体的には下記のような流れになります。

1. 「i」に 1 を代入する
2. 「i」が 10 以下の場合、「表示する処理」を実行する
3. 「表示する処理」が終了したら、「i」に 1 を加える
4. 「i」が 10 以下であれば、2 に戻る
5. 「i」が 10 より大きくなった場合、繰り返しを終了する

それを踏まえて、ブロックの中身を見てみましょう。

```
for (var i = 1; i <= atai ; i + +) {
   gazou.innerHTML += '<img src ="pic1.jpg">';
}
```

これは**変数 atai 以下になるまで**

```
gazou.innerHTML += '<img src ="pic1.jpg">';
```

←変数 gazou の中身に<img src ="pic1.jpg">を追加する

の処理を繰り返し行い、その結果を HTML 要素（id=output）に渡すという動きです。

<img src ="pic1.jpg">は、pic1.jpg という画像を表示するタグなので
pic1.jpg という画像を atai の回数分繰り返し追加し、表示をするという処理になります。

＜作成ページの動きを確認してみましょう＞

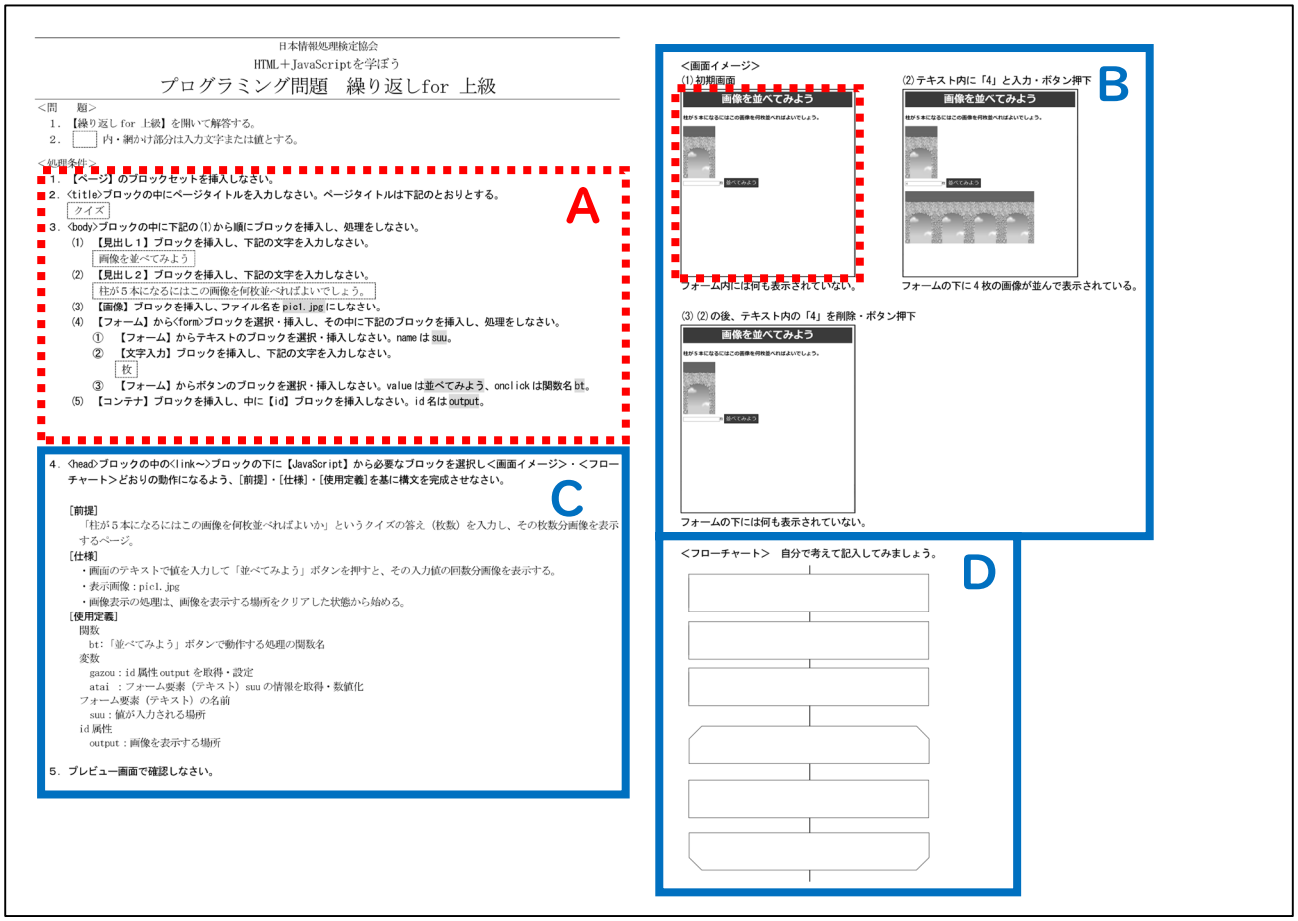
5．プレビュー画面で確認しなさい。

[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。

3．繰り返し for　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「for」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。

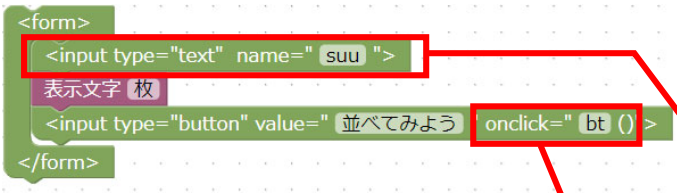
［前提］

「柱が５本になるにはこの画像を何枚並べればよいか」というクイズの答え（枚数）を入力し、その枚数分画像を表示するページ。

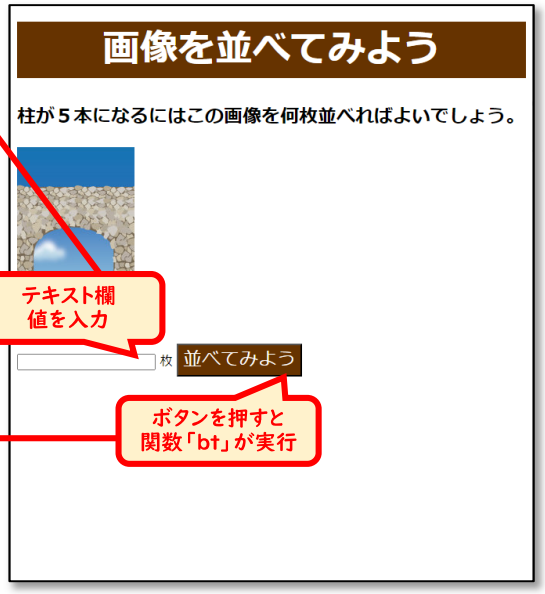
インプット部分の確認

3.の(4)入力フォーム

<ブロック>



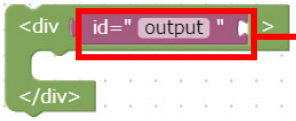
<作成ページ例>



アウトプット部分の確認

3.の(5)コンテナ

<ブロック>



<作成ページ例>

仕様部分を読み解く

［仕様］

・画面のテキストで値を入力して「並べてみよう」ボタンを押すと、その入力値の回数分画像を表示する。

→インプット部分・アウトプット部分の各設定値を確認。

→画像を表示する場所（id＝“output”）を取得し、設定するための箱（変数）を準備。

→テキスト欄に入力された値を入れておくための箱（変数）を準備。

→入力された値の回数分「画像を表示する」という処理を繰り返す。

・表示画像：pic1.jpg

→画像表示タグ<img src ="pic１.jpg">

・画像表示の処理は、画像を表示する場所をクリアした状態から始める。

→画像表示の処理の前に、画像を表示する場所（id ＝“output”）に空欄を設定。

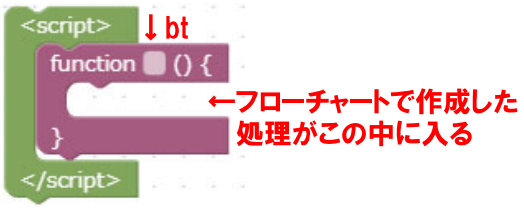
これに[使用定義］の内容を照らし合わせてフローチャートを作成

［使用定義］

関数

bt:「並べてみよう」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

gazou：id 属性 output を取得・設定

フォーム要素（テキスト）の名前

suu：値が入力される場所

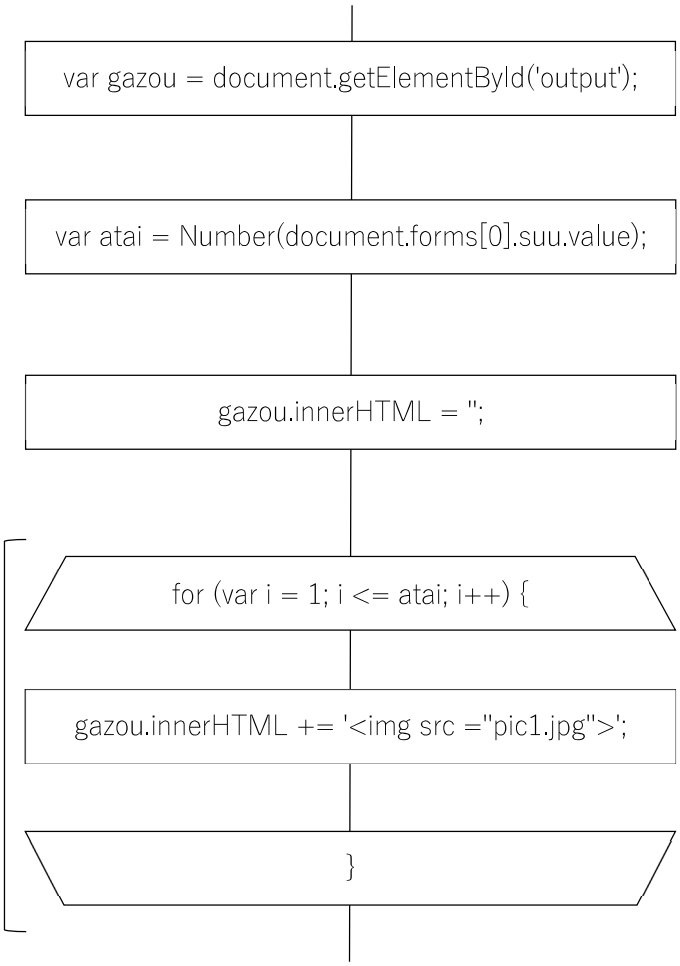
変数

atai　：フォーム要素（テキスト）suu の情報を取得・数値化

画像表示の処理の前に、
画像を表示する場所（id ＝“output”）
に空欄を設定。

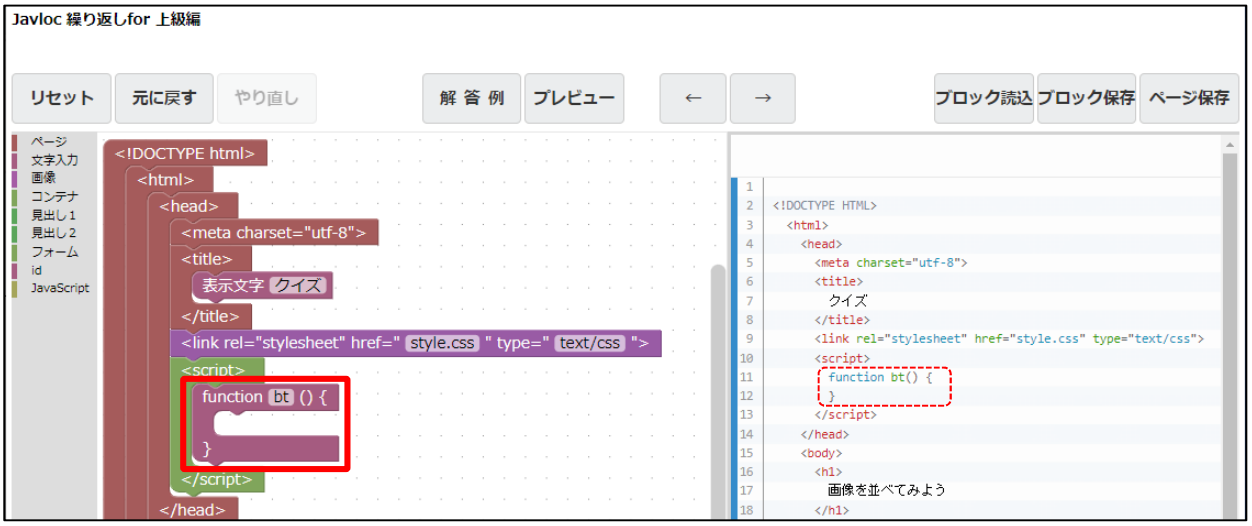
入力された値の回数分
「画像を表示する」という処理を繰り返す。

画像表示タグ<img src ="pic１.jpg">

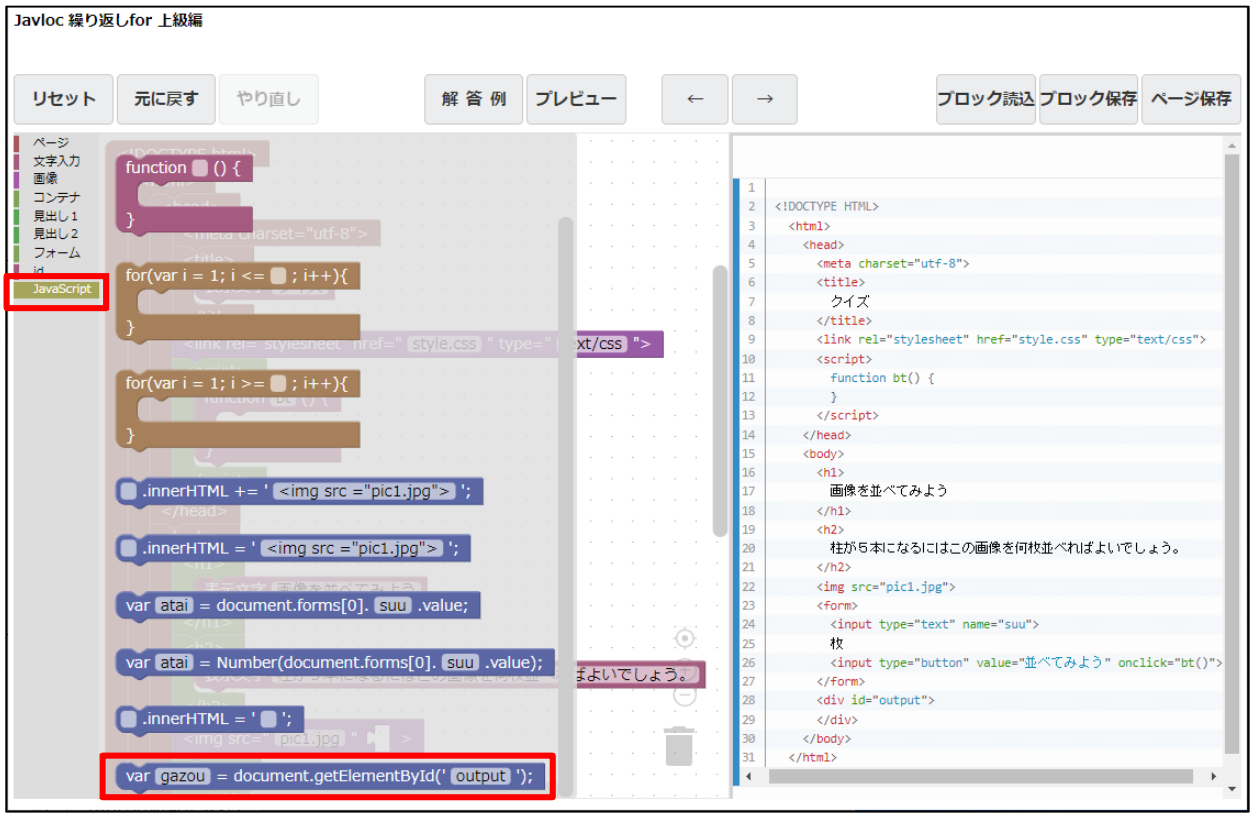


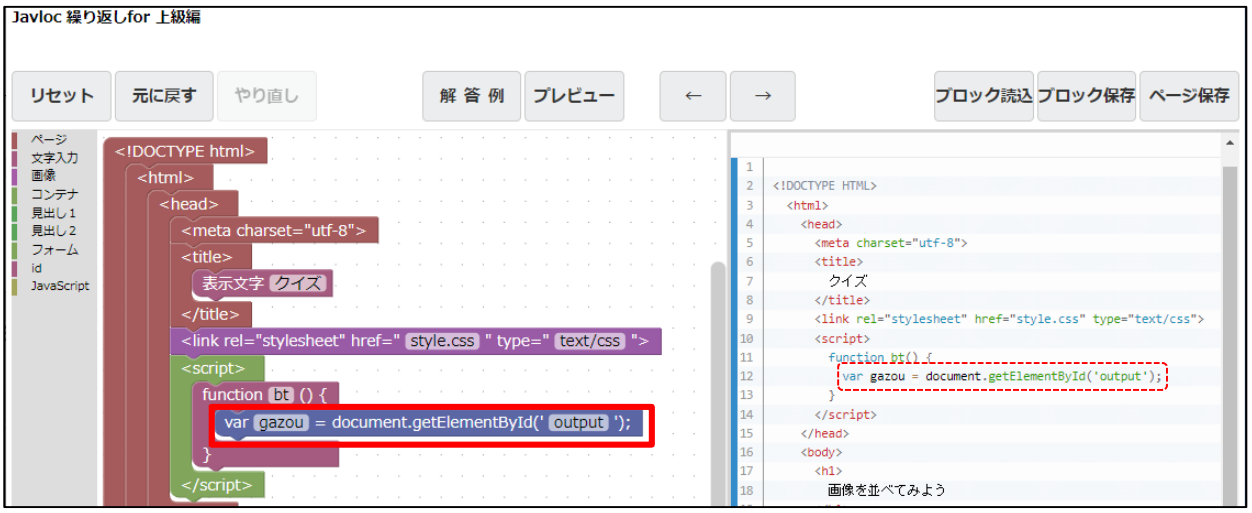
リスト内[JavaScript]をクリック、枠内にある<script>ブロックを選択し、ドラッグして＜link>ブロックの下にドロップ。ソース表示エリアで設定を確認。

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。

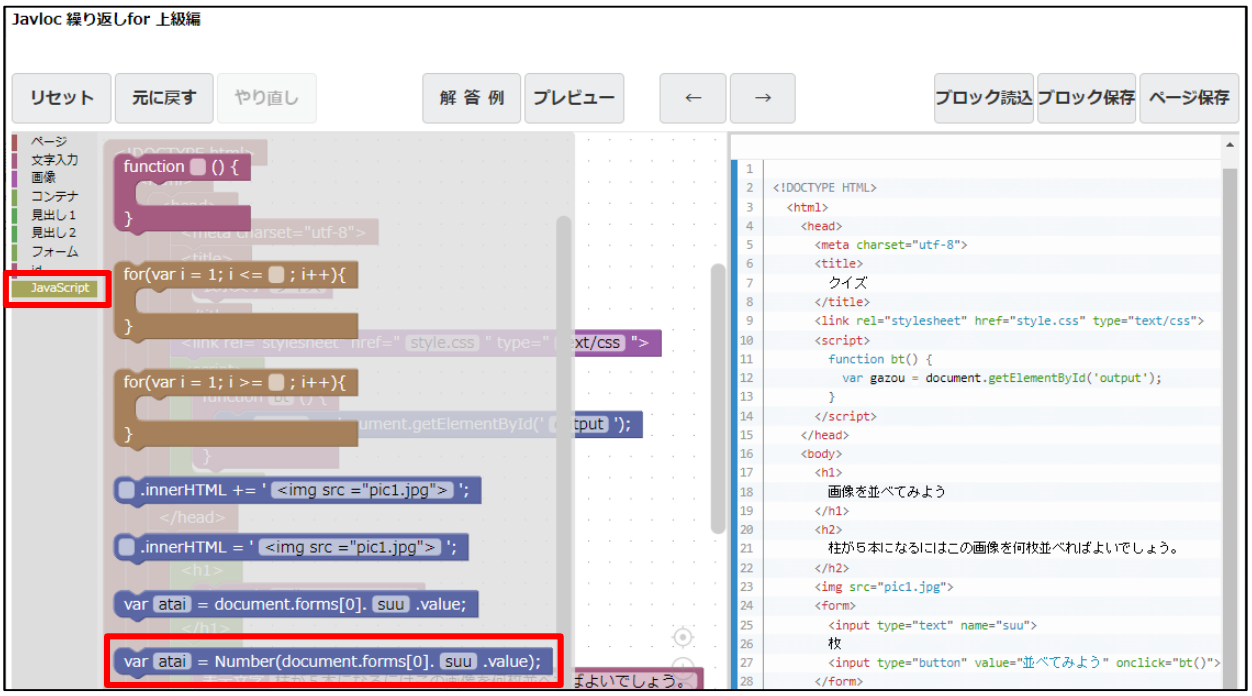


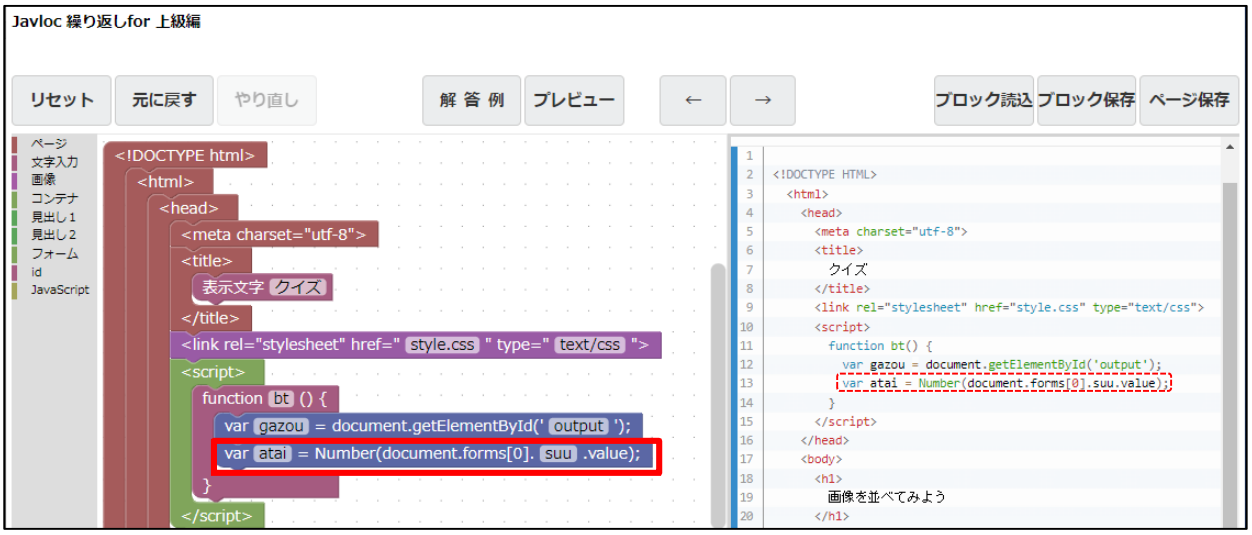
リスト内[JavaScript]をクリック、枠内にある var gazou =　;ブロックを選択し、ドラッグして<function>ブロックの中にドロップ。ソース表示エリアで設定を確認。



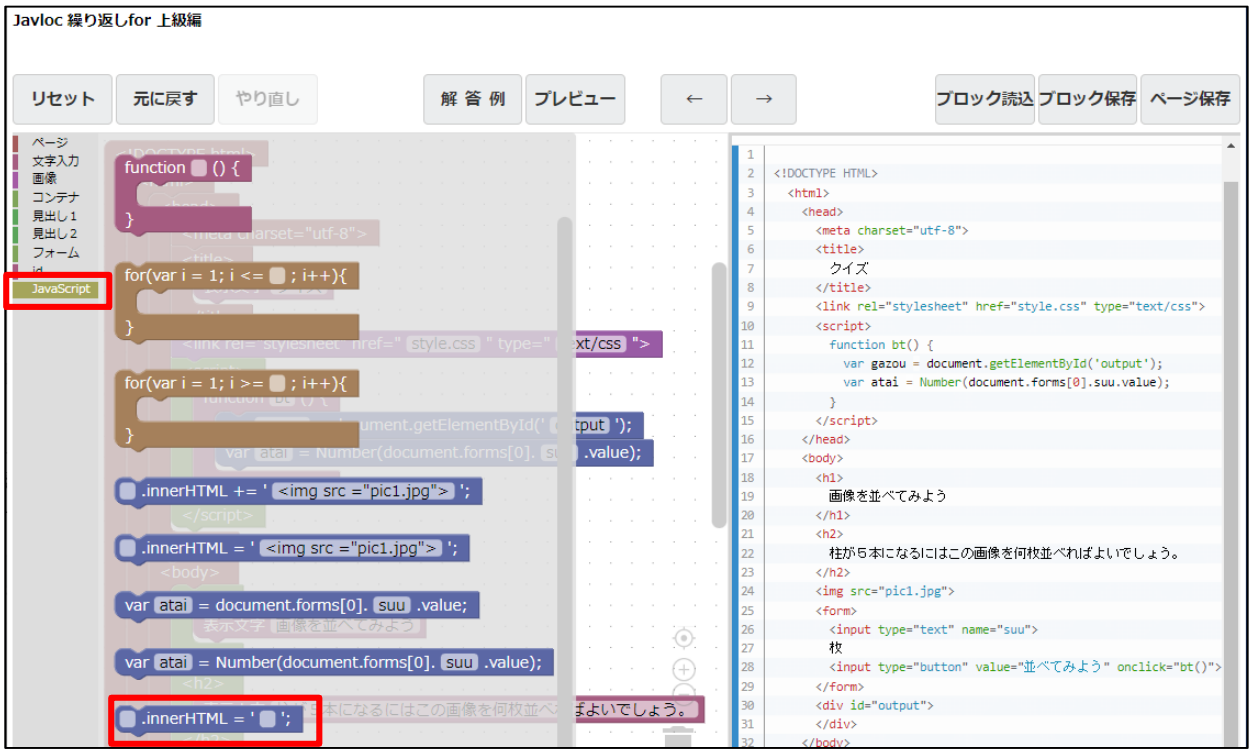


リスト内[JavaScript]をクリック、枠内にある var atai = Number(～)　;ブロックを選択し、ドラッグして var gazou =　;ブロックの下にドロップ。ソース表示エリアで設定を確認。

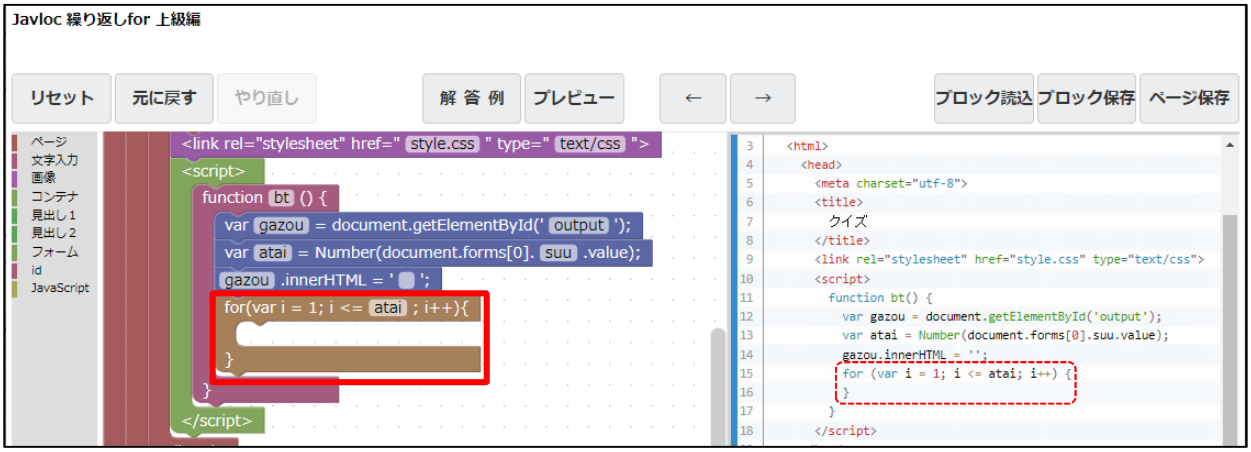




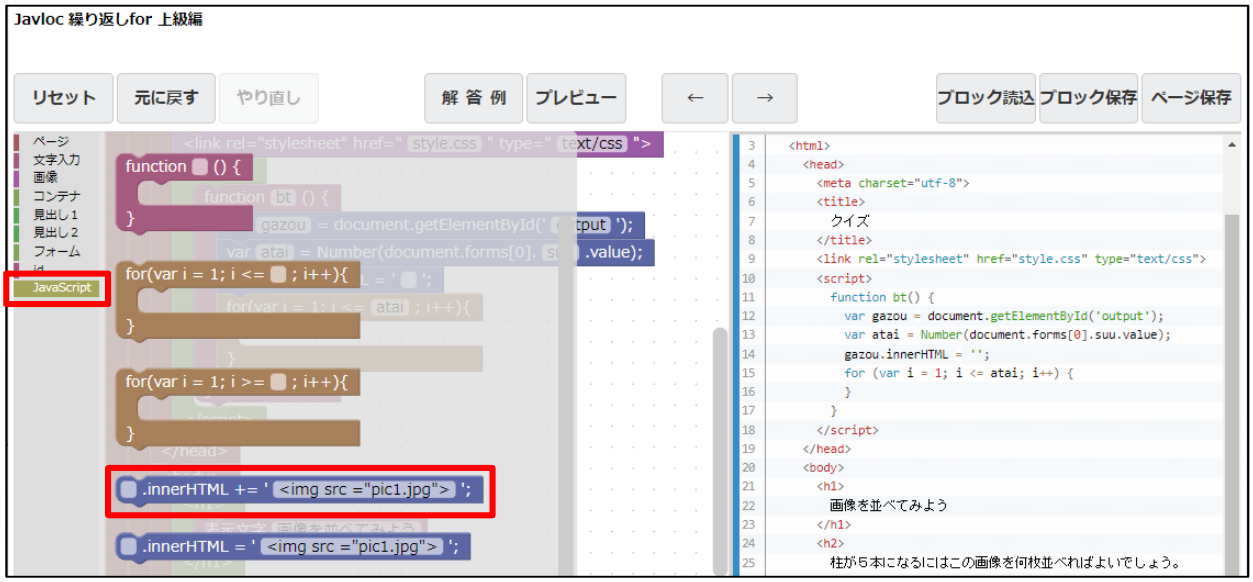
リスト内[JavaScript]をクリック、枠内にある□.innerHTML= '□' ;ブロックを選択し、ドラッグして var atai = Number(～)　;ブロックの下にドロップ。変数「gazou」を設定。ソース表示エリアで設定を確認。

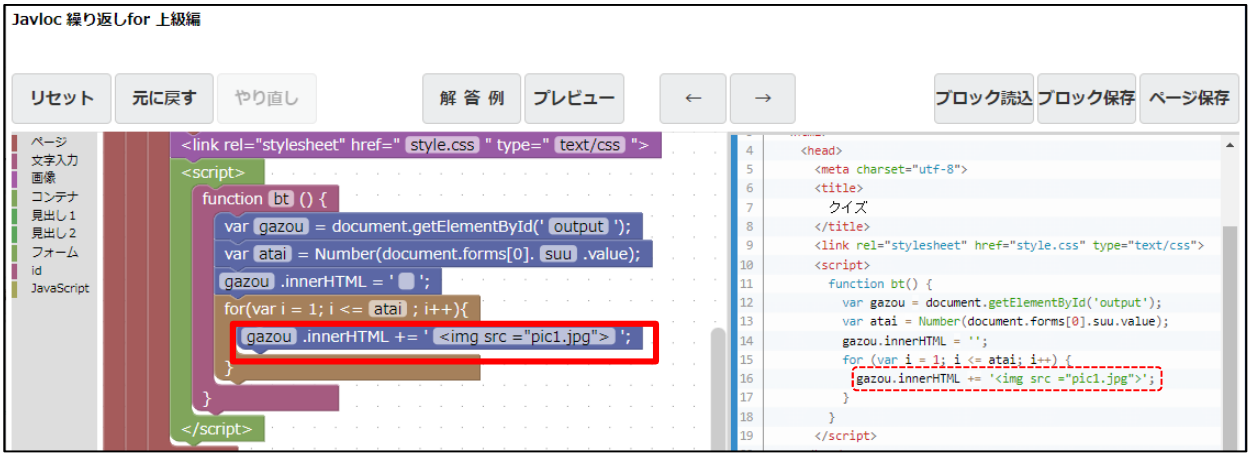


リスト内[JavaScript]をクリック、枠内にある for (var i = 1; i <= □; i++)　ブロックを選択し、ドラッグして gazou.innerHTML=　;ブロックの下にドロップ。変数「atai」を設定。ソース表示エリアで設定を確認。



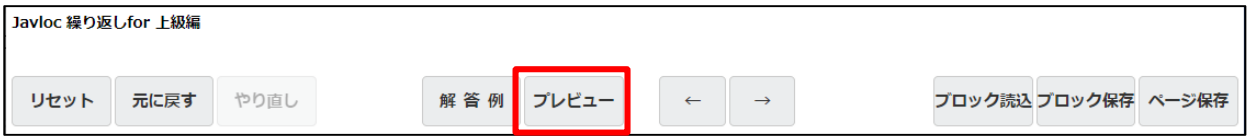
リスト内[JavaScript]をクリック、枠内にある□.innerHTML +=　;ブロックを選択し、ドラッグして for (var i = 1; i <= atai; i++)　の中にドロップ。変数「gazou」を設定。ソース表示エリアで設定を確認。





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。



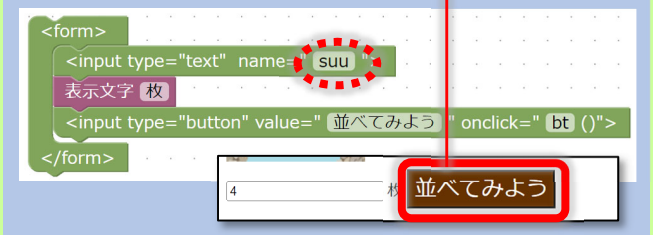
＜画面イメージ＞と同様の動作になっているか見てみましょう。

プログラム全体の流れ

⓪ テキスト欄に値が入力され、並べてみようボタンが押されると、関数が実行される。
① id“output”を取得し、変数 gazou に代入する。
② 変数 atai に⓪のテキスト欄「suu」を数値化した値を代入する。
③ 変数 gazou に、HTML（空欄）を代入する。
④ for 文を使用し、下記の処理を変数 atai の回数分、繰り返す。
　　変数 gazou に画像を表示させるタグを１つ追加する。

【HTML 部分・作成ページ例】

⓪



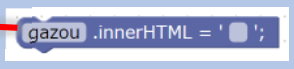
【JavaScript 関数 bt が実行】

例）４と入力した場合

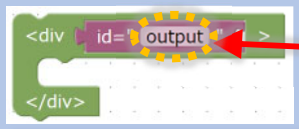
①

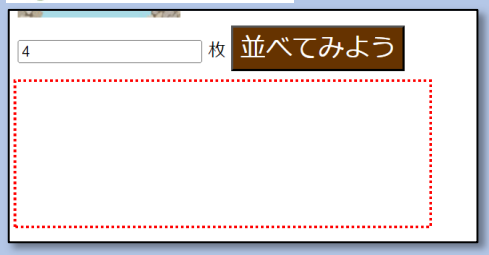
②

③

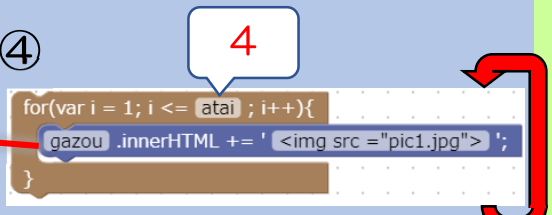


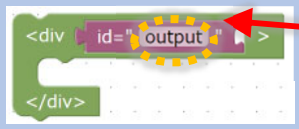
中身を空にする

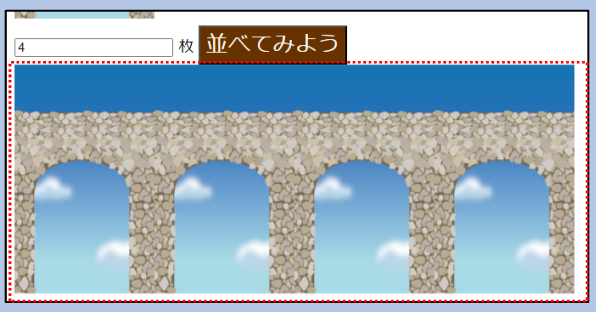


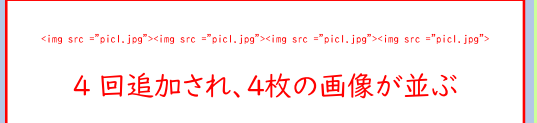


④









＜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。