

1. 条件分岐 switch 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 条件分岐switch 初級

<問題>

1. 【条件分岐 switch 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。

<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。
スキンケア選び
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。
自分に合ったスキンケア選び
 - (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。
あなたに合ったスキンケアシリーズが分かる
 - (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（セレクト・ボタン）を挿入しなさい。
 - (5) 【段落】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。
 - (6) 【画像】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 icon。

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】 3種類の商品があり、選択された項目に合致する商品の文章・画像をページ下部に表示させる。

【機能】

- ・フォームのオプションで選択された項目に合致する文章を3-(5)、画像を3-(6)の位置に表示する。
- ・いずれも選択されていない場合には、文章・画像は表示しない。

- (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

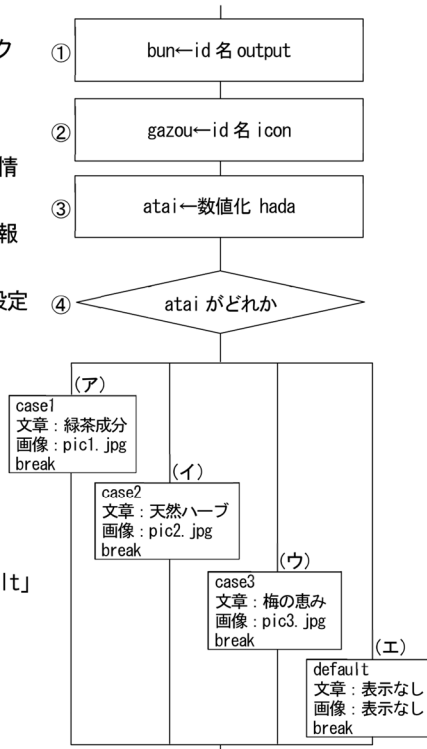
- ① 選択された項目の文章を表示させる場所 (id名 output) の情報を取得し、「bun」に設定する。
- ② 選択された項目の画像を表示させる場所 (id名 icon) の情報を取得し、「gazou」に設定する。
- ③ フォームのセレクト (hada) の値を数値にして「atai」と設定する。
- ④ 「atai」の値によって場合分けを行う。
 - (ア) 「atai」の値が1のときはcase1の処理を実行し終了
処理 文章：緑茶成分 画像：pic1.jpg
 - (イ) 「atai」の値が2のときはcase2の処理を実行し終了
処理 文章：天然ハーブ 画像：pic2.jpg
 - (ウ) 「atai」の値が3のときはcase3の処理を実行し終了
処理 文章：梅の恵み 画像：pic3.jpg
 - (エ) 「atai」の値がどれにも当てはまらないときは「default」を選択し終了
処理 文章・画像いずれも表示しない

5. プレビュー画面で確認しなさい。

<作成ページ例>



<フローチャート>

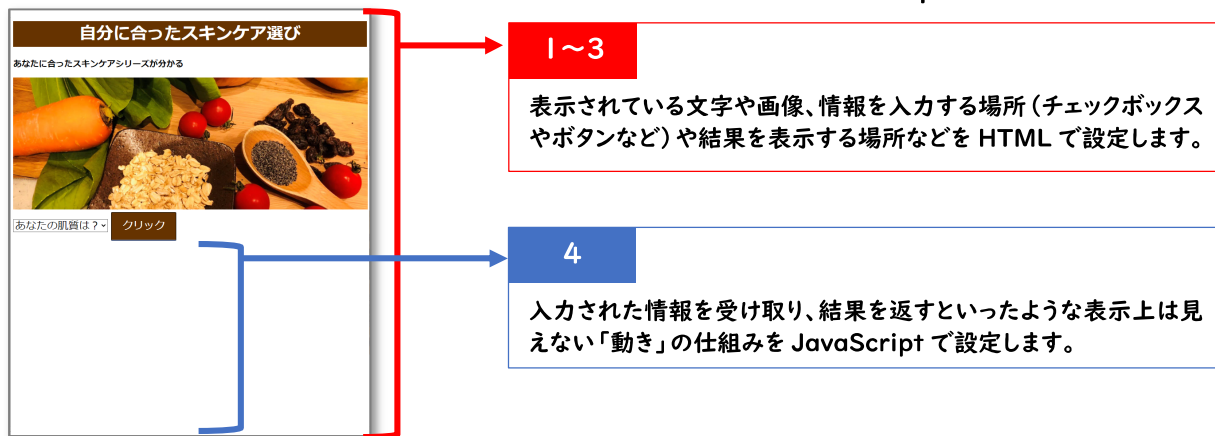


教材サイトからアプリを開く

Javloc 教材サイトから[初級プロジェクト]の[条件分岐「switch」初級]をクリック。

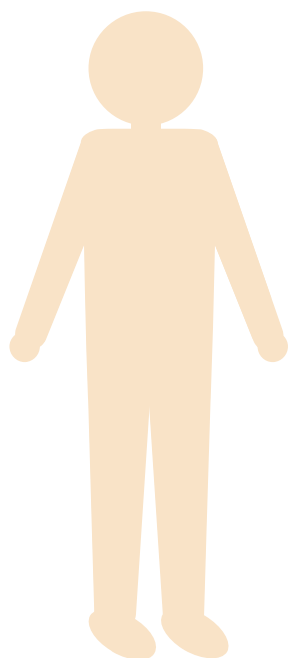


<処理条件> 1～3. では、ページの HTML 部分を作成し、4. では JavaScript を作成します。



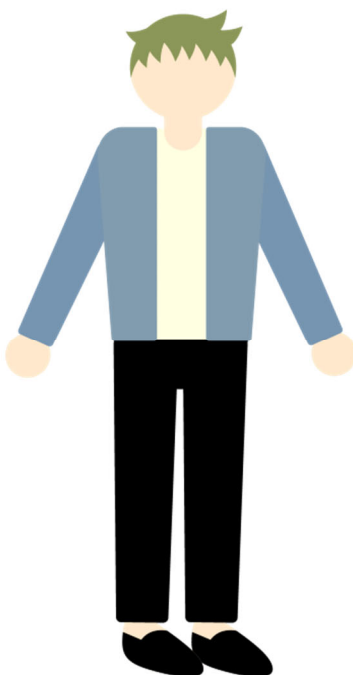
HTML と CSS と JavaScript の役割 イメージ図

HTML のみ



構造

HTML + CSS



デザイン

HTML + CSS
+ JavaScript



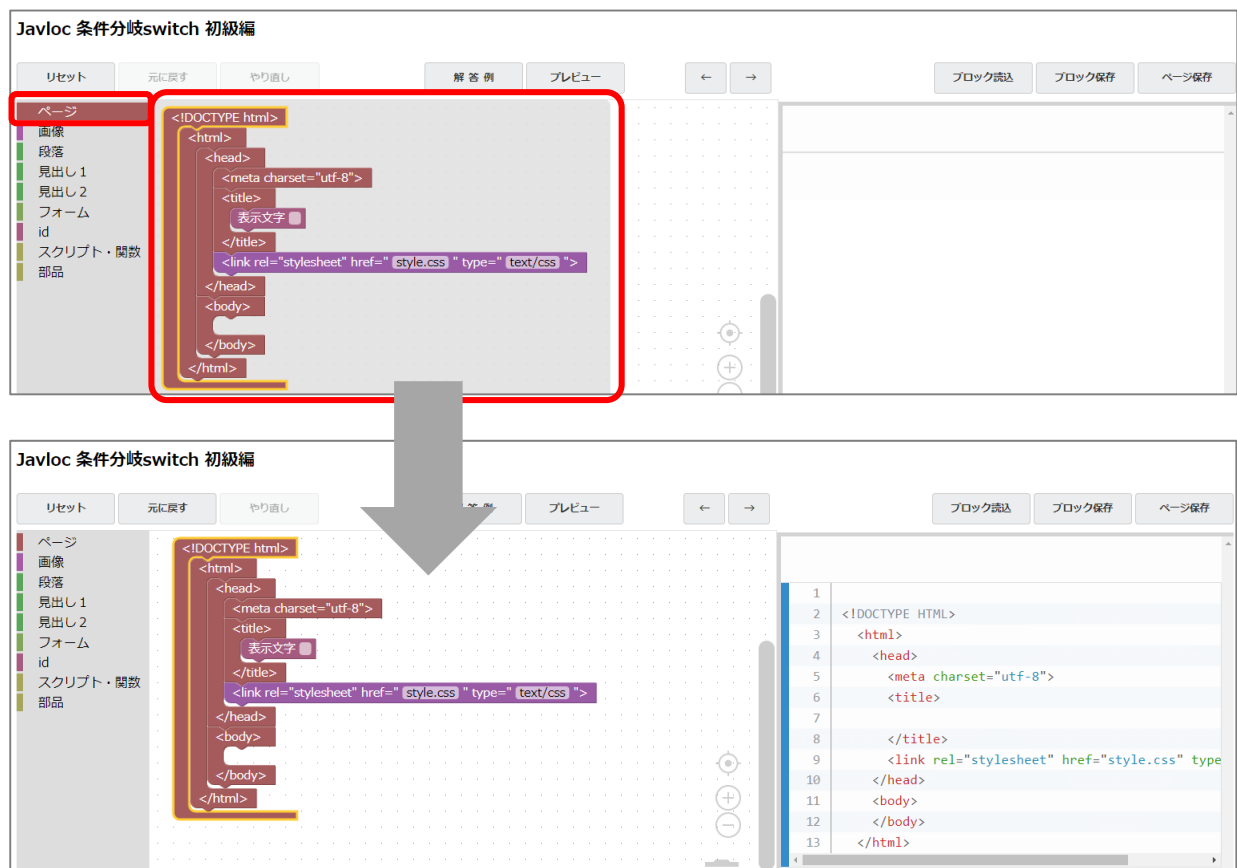
動き

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。

スキンケア選び

<title>ブロックの表示文字ブロックの空欄に「スキンケア選び」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。



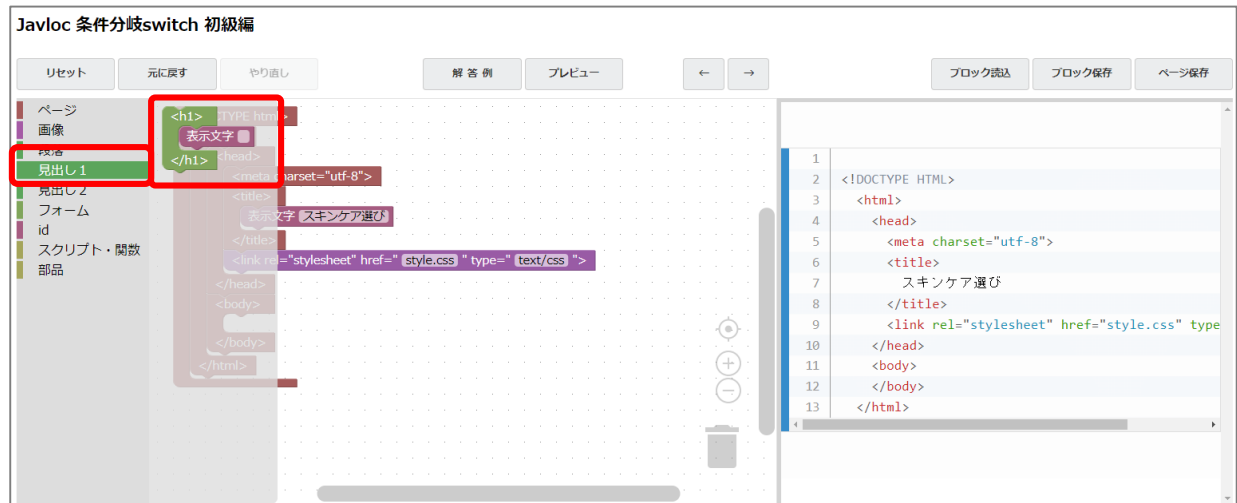
3. <body>ブロックの中に下記の(1)から順にブロックを挿入し、処理をしない。

(1) 【見出し 1】ブロックを挿入し、下記の文字を入力しない。

自分に合ったスキンケア選び

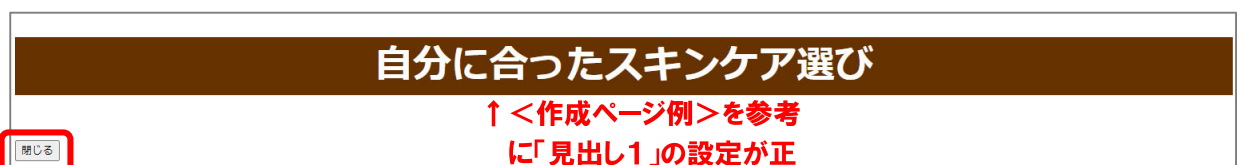
リスト内[見出し 1]をクリック、枠内にある<h1>ブロックをドラッグし、<body>ブロックの中にドロップ。

表示文字ブロックの空欄に「自分に合ったスキンケア選び」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>を参考
に「見出し 1」の設定が正
しく行われているか確認

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。

あなたに合ったスキんケアシリーズが分かる

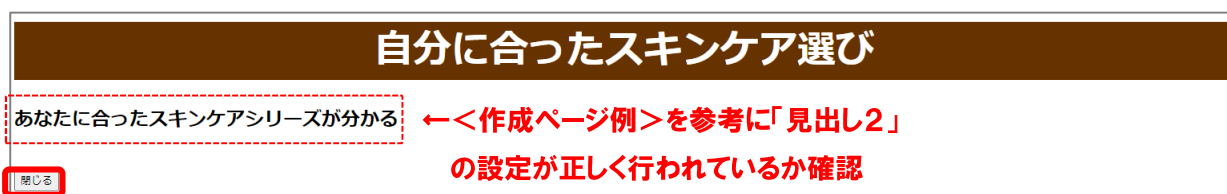
リスト内[見出し2]をクリック、枠内にある<h2>ブロックをドラッグし、<h1>ブロックの下にドロップ。

表示文字ブロックの空欄に「あなたに合ったスキんケアシリーズが分かる」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。
“”の空欄の中に「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。



[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「画像」の設定が正しく
行われているか確認

<作成ページ例>



(4) 【フォーム】のブロックセット（セレクト・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。

The screenshot shows the Javloc editor interface. On the left, a sidebar lists page elements: ページ (Page), 画像 (Image), 段落 (Paragraph), 見出し1 (Heading 1), 見出し2 (Heading 2), フォーム (Form), id, スクリプト・関数 (Script/Function), and 部品 (Component). The 'フォーム' (Form) block is highlighted with a red box. In the main workspace, a red box highlights the HTML code for the form block, which includes a select element with options for skin types and a button labeled 'クリック' (Click).

This screenshot shows the HTML source code for the form block. A red dashed box highlights the code, and a red arrow points to it with the text '確認' (Check). The code defines a select menu with the name 'hada' and four options: 'あなたの肌質は?' (What is your skin type?), '普通肌' (Normal skin), '敏感肌' (Sensitive skin), and '乾燥肌' (Dry skin). Below the select menu is a button with the value 'クリック' (Click) and an onclick event.

[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。

The screenshot shows the Javloc editor interface with the 'プレビュー' (Preview) button highlighted by a red box. Other buttons like 'リセット' (Reset), '元に戻す' (Undo), 'やり直し' (Redo), '解答例' (Answer Example), and 'ブロック読み込み' (Load Block) are also visible.

まだ JavaScript が設定されていないため、肌質を選択し「クリック」ボタンをクリックしても、下の A の箇所には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」ボタンをクリックすると計算が行われ、A の箇所に結果が表示されるようになります。

This diagram illustrates the workflow of the form. On the left, a preview of the form shows a select menu with 'あなたの肌質は?' (What is your skin type?) and three options: '普通肌' (Normal skin), '敏感肌' (Sensitive skin), and '乾燥肌' (Dry skin). Below the select menu is a red box labeled 'A'. A 'クリック' (Click) button is shown next to it. Red arrows connect the UI elements to the HTML code on the right. The code shows the select menu with options and the button with the value 'クリック' and an onclick event. Red text labels the code elements: 'セレクト hada' (Select hada), '既定値として表示する文字' (Text displayed as default value), '一つ目の選択肢として表示する文字' (Text displayed as the first selection option), '二つ目の選択肢として表示する文字' (Text displayed as the second selection option), '三つ目の選択肢として表示する文字' (Text displayed as the third selection option), and 'ボタン' (Button).

(5) 【段落】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[段落]をクリック、枠内にある<p>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段落が設定されていることを確認。

The screenshot shows the Javloc editor interface. On the left, the 'Paragraph' block is selected in the sidebar. In the workspace, a red box highlights the paragraph block being dragged. The source code on the right shows the HTML structure with the new paragraph tag added.

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<p>ブロックのパズル型空欄にドロップ。id 内“ ”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。**

The screenshot shows the Javloc editor interface. On the left, the 'id' block is selected in the sidebar. In the workspace, a red box highlights the id block being dragged. The source code on the right shows the HTML structure with the new id tag added and the name 'output' entered.

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、「段落」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の表示が変化します。

(6) 【画像】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 icon。

リスト内[画像]をクリック、枠内にあるブロックをドラッグし<p>ブロックの下にドロップ。

Javloc 条件分岐switch 初級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

表示文字 普通肌
</option>
<option value="2">
表示文字 敏感肌
</option>
<option value="3">
表示文字 乾燥肌
</option>
</select>
<input type="button" value="クリック" onclick="bt()">
</form>
<p id="output">
</p>

</body>
</html>

```
18   
19 <form>  
20 <select name="hada">  
21 <option value="0" selected>  
22 あなたの肌質は?  
23 </option>  
24 <option value="1">  
25 普通肌  
26 </option>  
27 <option value="2">  
28 敏感肌  
29 </option>  
30 <option value="3">  
31 乾燥肌  
32 </option>  
33 </select>  
34 <input type="button" value="クリック" onclick="bt()">  
35 </form>  
36 <p id="output">  
37 <img src="">  
38 </p>  
39 </body>
```

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。id 内“ ”の空欄に「icon」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。

Javloc 条件分岐switch 初級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

表示文字 普通肌
</option>
<option value="2">
表示文字 敏感肌
</option>
<option value="3">
表示文字 乾燥肌
</option>
</select>
<input type="button" value="クリック" onclick="bt()">
</form>
<p id="output">
</p>

</body>
</html>

```
19 <form>  
20 <select name="hada">  
21 <option value="0" selected>  
22 あなたの肌質は?  
23 </option>  
24 <option value="1">  
25 普通肌  
26 </option>  
27 <option value="2">  
28 敏感肌  
29 </option>  
30 <option value="3">  
31 乾燥肌  
32 </option>  
33 </select>  
34 <input type="button" value="クリック" onclick="bt()">  
35 </form>  
36 <p id="output">  
37 <p>  
38 <img src="" id="icon">  
39 </p>
```

ここでは「画像」タグに icon という名前を付けたということになります。

(6)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、処理をしない。

【前提】 3種類の商品があり、選択された項目に合致する商品の文章・画像をページ下部に表示させる。

【機能】

- ・フォームのオプションで選択された項目に合致する文章を3-(5)、画像を3-(6)の位置に表示する。
- ・いずれも選択されていない場合には、文章・画像は表示しない。



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。

Javloc 条件分岐switch 初級編

リセット 元に戻す やり直し **解答例** プレビュー ← → ブロック表示 ブロック保存 ページ保存

このフォームの部分を操作してみましょう。
プルダウン（セレクト）から項目を選択し、「クリック」ボタンを押してみます。
右図の4パターンを試してみましょう。

自分に合ったスキンケア

あなたに合ったスキンケアシリーズが分かる

あなたの肌質は？ クリック

段落 id="output"

画像 id="icon"

プルダウン（セレクト）で選択された商品の宣伝の文章とイメージ画像を「クリック」ボタンが押された際にフォームの下に表示する。

既定値（デフォルト値）「あなたの肌質は？」が選択され「クリック」ボタンが押された際は、何も表示しない。

普通肌

自分に合ったスキンケア選び

あなたに合ったスキンケアシリーズが分かる

普通肌 クリック

乾燥肌

梅の恵み

敏感肌

自分に合ったスキンケア選び

あなたに合ったスキンケアシリーズが分かる

敏感肌 クリック

乾燥肌

梅の恵み

乾燥肌

自分に合ったスキンケア選び

あなたに合ったスキンケアシリーズが分かる

乾燥肌 クリック

梅の恵み

既定値: あなたの肌質は？

自分に合ったスキンケア選び

あなたに合ったスキンケアシリーズが分かる

あなたの肌質は？ クリック

(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。

The image consists of two screenshots of the Javloc editor interface, titled 'Javloc 条件分岐switch 初級編'.

The top screenshot shows the left sidebar with the 'スクリプト・関数' (Script & Function) block highlighted. A red box highlights the script block set in the editor. The right pane shows the source code with the following structure:

```
<html>
<head>
  <meta charset="utf-8">
  <title>
    スキンケア選び
  </title>
  <link rel="stylesheet" href="style.css" type="text/css">
</head>
<body>
  <h1>
    自分に合ったスキンケア選び
  </h1>
```

The bottom screenshot shows the same editor after the script block has been inserted. A red dashed box highlights the script block in the source code, with a red arrow pointing to it and the text '確認' (Check). The source code now includes the script block:

```
<html>
<head>
  <meta charset="utf-8">
  <title>
    スキンケア選び
  </title>
  <link rel="stylesheet" href="style.css" type="text/css">
  <script>
    function bt() {
    }
  </script>
</head>
<body>
  <h1>
```

<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、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 選択された項目の文章を表示させる場所 (id 名 output) の情報を取得し、「bun」に設定する。

リスト内【部品】をクリック、枠内にある `var bun = 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。

The image shows two screenshots of the Javloc editor interface. The top screenshot shows the '部品' (Parts) list on the left with 'var bun = document.getElementById('output');' highlighted. The bottom screenshot shows the same code block being dropped into the 'function bt()' block in the main editor area. A red arrow points to the code in the bottom screenshot with the text '←確認' (Check).

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 bun = 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、「bun」という箱 (変数) に入れておく。

ということになります。

- ② 選択された項目の画像を表示させる場所 (id 名 icon) 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('icon');` 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。

Javloc 条件分岐switch 初級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ
画像
段落
見出し1
見出し2
フォーム
id
スクリプト・関数
部品

```
var bun = document.getElementById('output');  
var gazou = document.getElementById('icon');  
var atai = Number(document.forms[0].hada.value);  
switch (atai) {  
  case 1:  
    bun.innerHTML = ' ';  
    gazou.src = ' ';  
    break;  
}
```

スクリプト・関数
部品

```
<title>  
  表示文字 スキンケア選び  
</title>  
<link rel="stylesheet" href="style.css" type="text/css">  
<script>  
  function bt() {  
    var bun = document.getElementById('output');  
    var gazou = document.getElementById('icon');  
  }  
</script>  
</head>  
<body>
```

これも①と同じようなことをしています。

ブロックの中身は

```
var gazou = document.getElementById('icon');
```

なので、これは何をしているかというと、

「icon」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。

③ フォームのセレクト (hada) の値を数値にして「atai」と設定する。

リスト内[部品]をクリック、枠内にある `var atai = Number(document.forms[0].hada.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。

The image contains two screenshots of the Javloc editor interface, titled "Javloc 条件分岐switch 初級編".

The top screenshot shows the "部品" (Parts) block selected in the left sidebar. In the main workspace, a JavaScript block `var atai = Number(document.forms[0].hada.value);` is highlighted with a red rectangle. The right pane shows the HTML output, including a script block with a function `bt()` that contains `var bun = document.getElementById('output');` and `var gazou = document.getElementById('icon');`.

The bottom screenshot shows the same workspace after the JavaScript block has been added to the switch statement. The block is now nested within a `case 1:` block. A red dashed box highlights the line `var atai = Number(document.forms[0].hada.value);` with a red arrow pointing to it and the text "確認" (Check) written in red.

「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].hada.value);
```

なので、これは何をしているかというと、

フォームの「hada」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 「atai」の値によって場合分けを行う。
- (ア) 「atai」の値が1のときはcase1の処理を実行し終了
処理 文章：緑茶成分 画像：pic1.jpg
 - (イ) 「atai」の値が2のときはcase2の処理を実行し終了
処理 文章：天然ハーブ 画像：pic2.jpg
 - (ウ) 「atai」の値が3のときはcase3の処理を実行し終了
処理 文章：梅の恵み 画像：pic3.jpg
 - (エ) 「atai」の値がどれにも当てはまらないときは「default」を選択し終了
処理 文章・画像いずれも表示しない

リスト内[部品]をクリック、枠内にある `switch (atai) {~}` 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。

Javloc 条件分岐switch 初級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

部品

```

var bun = document.getElementById('output');
var gazou = document.getElementById('icon');
var atai = Number(document.forms[0].hada.value);

switch (atai) {
  case 1:
    bun.innerHTML = '緑茶成分';
    gazou.src = 'pic1.jpg';
    break;
  case 2:
    bun.innerHTML = '天然ハーブ';
    gazou.src = 'pic2.jpg';
    break;
  case 3:
    bun.innerHTML = '梅の恵み';
    gazou.src = 'pic3.jpg';
    break;
  default:
    // 何も表示しない
}

```

```

<script>
  var bun = document.getElementById('output');
  var gazou = document.getElementById('icon');
  var atai = Number(document.forms[0].hada.value);
</script>
</head>
<body>
  <h1>
    自分に合ったスキンケア選び
  </h1>
  <h2>
    あなたに合ったスキンケアシリーズが分かる
  </h2>
  
  <form>
    <select name="hada">
      <option value="0" selected>
        あなたの肌質は？
      </option>
      <option value="1">
        普通肌
      </option>
      <option value="2">
        敏感肌
      </option>
      <option value="3">
        乾燥肌
      </option>
    </select>
  </form>

```

Javloc 条件分岐switch 初級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

部品

```

var bun = document.getElementById('output');
var gazou = document.getElementById('icon');
var atai = Number(document.forms[0].hada.value);

switch (atai) {
  case 1:
    bun.innerHTML = '緑茶成分';
    gazou.src = 'pic1.jpg';
    break;
  case 2:
    bun.innerHTML = '天然ハーブ';
    gazou.src = 'pic2.jpg';
    break;
  case 3:
    bun.innerHTML = '梅の恵み';
    gazou.src = 'pic3.jpg';
    break;
  default:
    // 何も表示しない
}

```

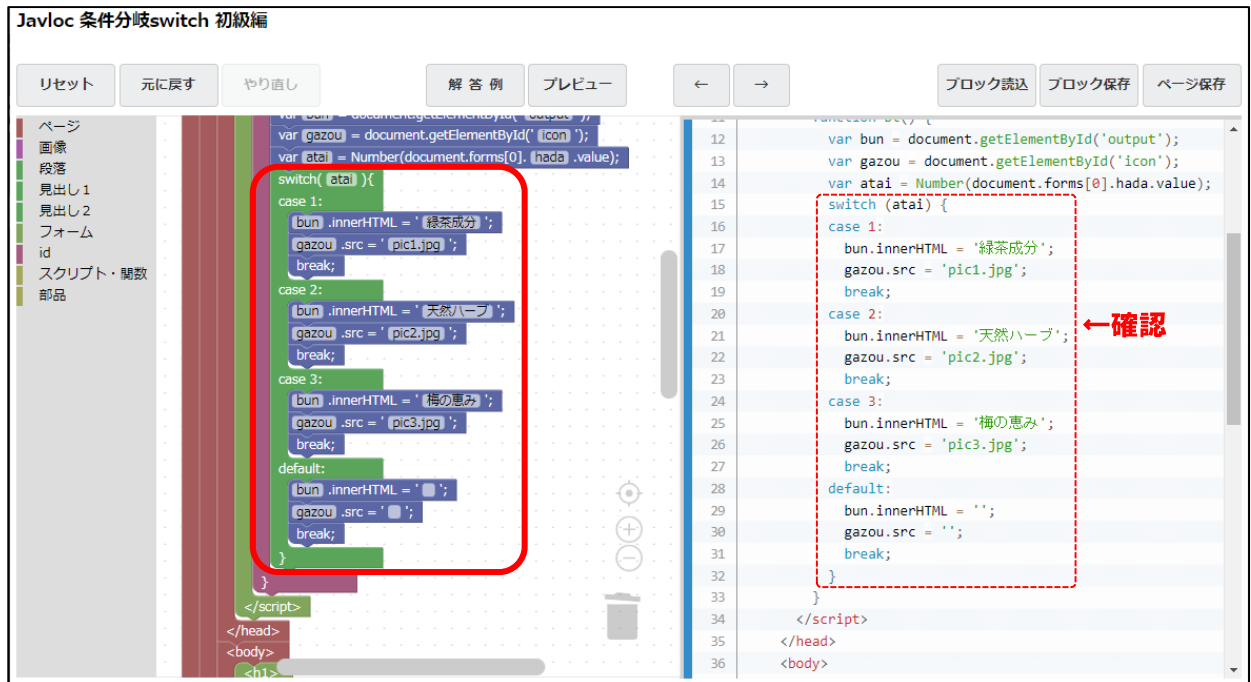
```

<script>
  var bun = document.getElementById('output');
  var gazou = document.getElementById('icon');
  var atai = Number(document.forms[0].hada.value);
  switch (atai) {
    case 1:
      bun.innerHTML = '緑茶成分';
      gazou.src = 'pic1.jpg';
      break;
    case 2:
      bun.innerHTML = '天然ハーブ';
      gazou.src = 'pic2.jpg';
      break;
    case 3:
      bun.innerHTML = '梅の恵み';
      gazou.src = 'pic3.jpg';
      break;
    default:
      // 何も表示しない
  }
</script>
</head>
<body>

```


case 1 の上の“空欄”には「緑茶成分」、下の“空欄”には「pic1.jpg」と入力。
case 2 の上の“空欄”には「天然ハーブ」、下の“空欄”には「pic2.jpg」と入力。
case 3 の上の“空欄”には「梅の恵み」、下の“空欄”には「pic3.jpg」と入力。
default の“空欄”には、何も入力しない。

※「pic1.jpg」、「pic2.jpg」、「pic3.jpg」は必ず半角で入力しましょう。



switch というのは、いろいろな選択肢があるときに、その場合に合った選択肢を選ぶための仕組みです。

例えば、「色」が何かによって処理を変えたい場合、下記のようになります。

switch(色){

case '赤': ←色が赤の場合

ここに赤の場合の処理を書く

break: ←処理が終わったら、抜け出す。青、黄色、default は確認しない。

case '青': ←色が青の場合

ここに青の場合の処理を書く

break: ←処理が終わったら、抜け出す。黄、default は確認しない。

case '黄': ←色が黄の場合

ここに黄色の場合の処理を書く

break: ←処理が終わったら、抜け出す。default は確認しない。

default: ←色が赤・青・黄色以外の場合

ここに赤・青・黄色以外の場合の処理を書く

break: ←処理が終わったら、抜け出す。

}

※break は、実行中の処理から抜け出す処理です。

※default は、どの条件にも当てはまらなかった場合という意味です。

また、今回の処理では、atai が 1 だった場合、「緑茶成分」という文章と「pic1.jpg」という画像を表示するために、「.innerHTML」と「.src」という新しいプロパティが登場します。

```
switch( atai ){  
  case 1:  
    bun.innerHTML = '緑茶成分';  
    gazou.src = 'pic1.jpg';  
    break;  
  case 2:
```

.innerHTML は、HTML 要素の中身を変更するという動きをします。

例えば、atai が 1 の場合に処理するブロックの中身は、

```
bun.innerHTML = '緑茶成分';
```

となっており、これは変数「bun」の HTML 要素の中身を「緑茶成分」に変更するという意味になります。

変数「bun」には id「output」が入っていますので、id「output」の場所に「緑茶成分」と表示されるようになります。

.src は、画像などを読み込むための住所のようなもので、その住所に保存されているファイルを読み込むという動きをします。この問題では住所を省略してファイル名だけ入力すれば表示されるようになっています。

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. 条件分岐 switch 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 条件分岐switch 中級

<問題>

1. 【条件分岐 switch 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。

<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。
スキンケア選び
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。
自分に合ったスキンケア選び
 - (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。
あなたに合ったスキンケアシリーズが分かる
 - (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 【フォーム】からセレクトのブロックを挿入しなさい。name は hada。その中に下記のブロックを挿入し、処理をしなさい。
 - (ア) 【フォーム】からオプションのブロック (value= "0" : 初期設定値) を選択・挿入し、下記の文字を入力。
あなたの肌質は？
 - (イ) 【フォーム】からオプションのブロック (value= "1") を選択・挿入し、下記の文字を入力。
普通肌
 - (ウ) 【フォーム】からオプションのブロック (value= "2") を選択・挿入し、下記の文字を入力。
敏感肌
 - (エ) 【フォーム】からオプションのブロック (value= "3") を選択・挿入し、下記の文字を入力。
乾燥肌
 - ② 【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。
 - (5) 【段落】ブロックを挿入し、その中に【id】ブロックを挿入しなさい。id 名は output。
 - (6) 【画像】ブロックを挿入し、その中に【id】ブロックを挿入しなさい。id 名は icon。
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] 3種類の商品があり、選択された項目に合致する商品の文章・画像をページ下部に表示させる。
[機能]
 - ・フォームのオプションで選択された項目に合致する文章を 3-(5)、画像を 3-(6) の位置に表示する。
 - ・いずれも選択されていない場合には、文章・画像は表示しない。

- (1) 【スクリプト・関数】ブロックを挿入し、3-(4)-②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

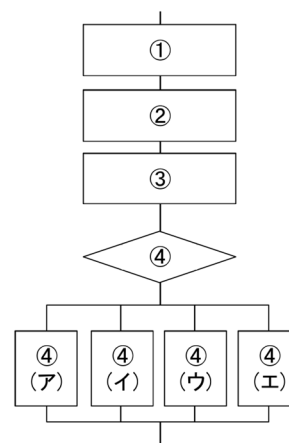
- ① 変数 bun を宣言し、id(output) を取得し代入
- ② 変数 gazou を宣言し、id(icon) を取得し代入
- ③ 変数 atai を宣言し、フォームのセレクト (hada) の値を数値に変換し代入
- ④ 「atai」の値によって条件分岐
 - (ア) 「atai」が1の場合、下記の処理を実行し終了
処理 HTML 挿入部分 (bun) : 緑茶成分
画像 (gazou) : pic1.jpg
 - (イ) 「atai」が2の場合、下記の処理を実行し終了
処理 HTML 挿入部分 (bun) : 天然ハーブ
画像 (gazou) : pic2.jpg
 - (ウ) 「atai」が3の場合、下記の処理を実行し終了
処理 HTML 挿入部分 (bun) : 梅の恵み
画像 (gazou) : pic3.jpg
 - (エ) 「atai」の値がどれにも当てはまらないときは「default」を選択し終了
処理 HTML 挿入部分 (bun) ・画像 (gazou) いずれも表示しない

5. プレビュー画面で確認しなさい。

<作成ページ例>



<フローチャート>



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[条件分岐「switch」初級]をクリック。

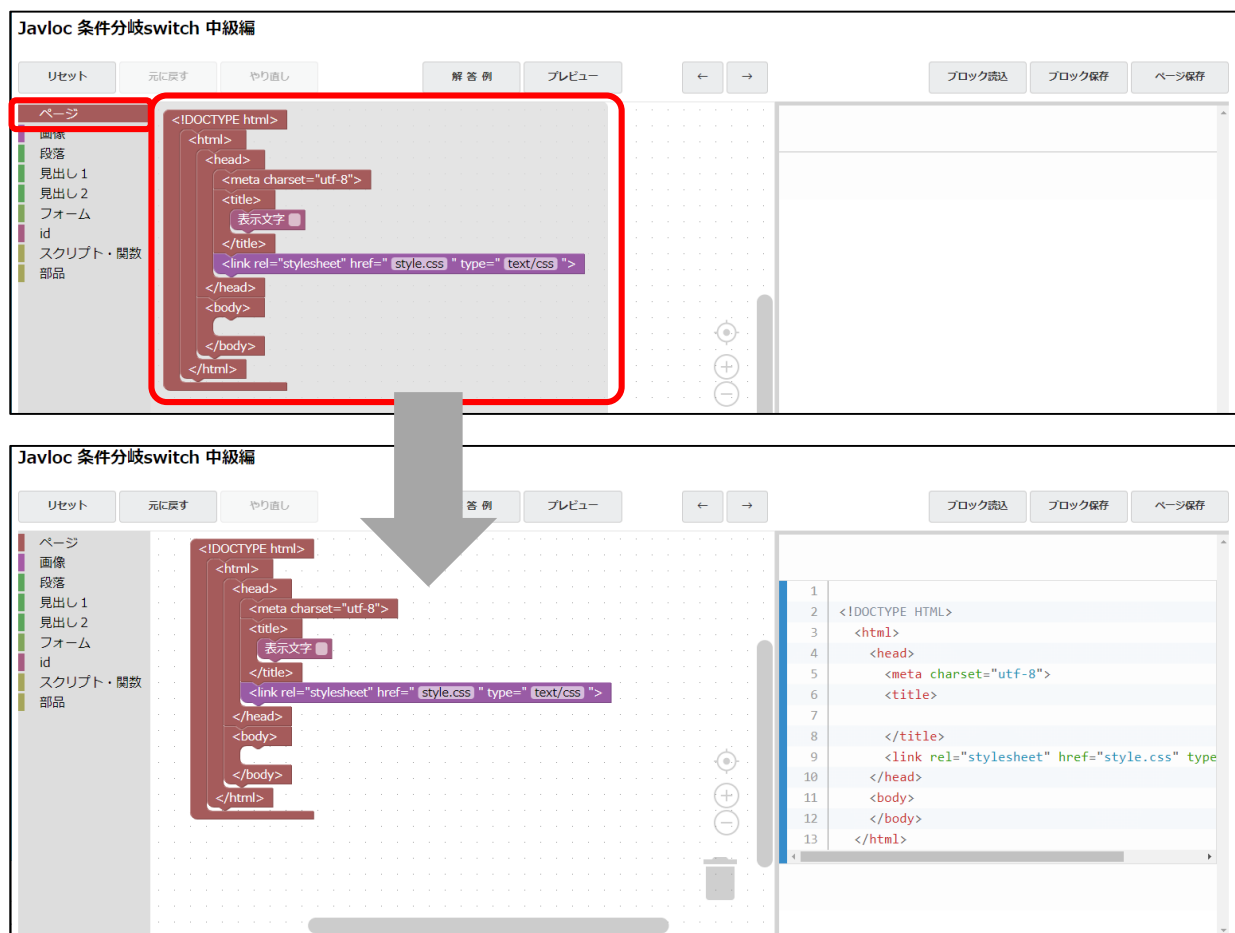


ページを構成（HTML 部分の作成）

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。

スキンケア選び

<title>ブロックの表示文字ブロックの空欄に「スキンケア選び」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。



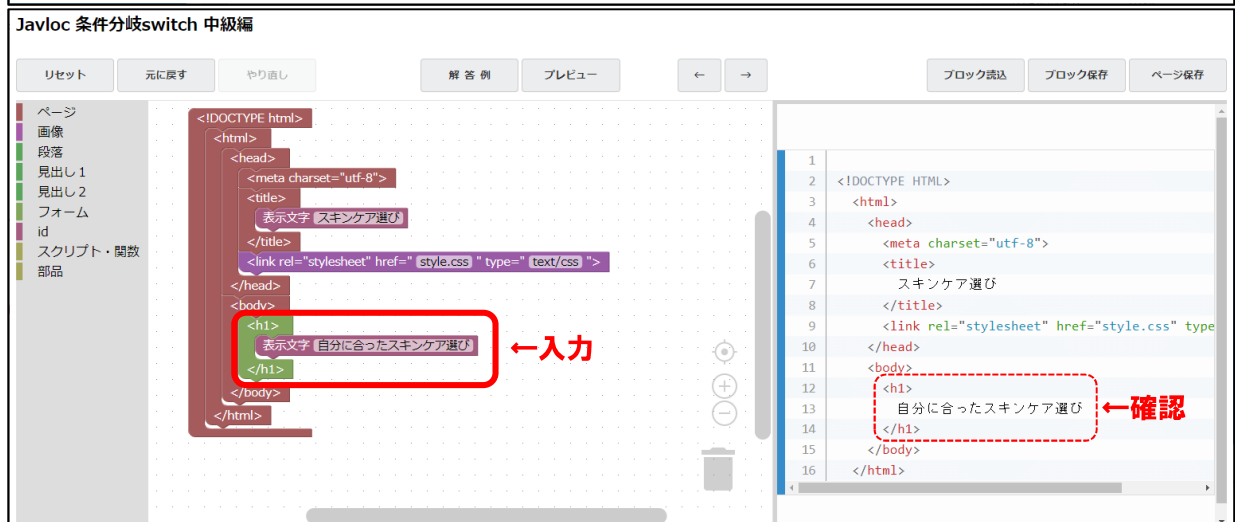
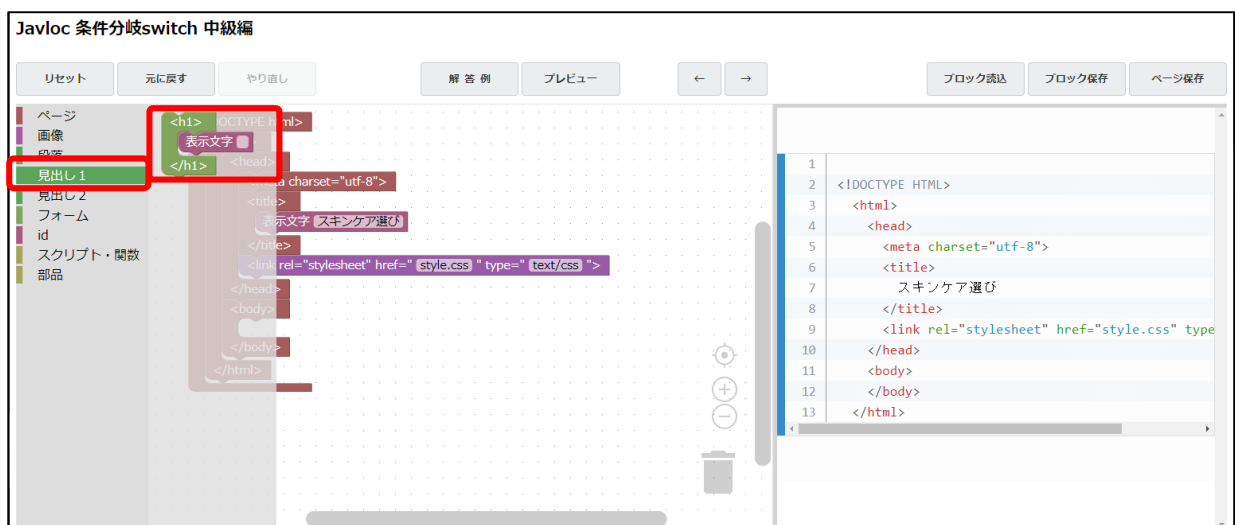
3. <body>ブロックの中に下記の(1)から順にブロックを挿入し、処理をしなさい。

(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。

自分に合ったスキンケア選び

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「自分に合ったスキンケア選び」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。

あなたに合ったスキんケアシリーズが分かる

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「あなたに合ったスキんケアシリーズが分かる」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。

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

<h2> 表示文字

</h2>

1 <!DOCTYPE HTML>
2 <html>
3

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>
表示文字 スキンケア選び
</title>
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body>
<h1>
表示文字 自分に合ったスキンケア選び
</h1>
<h2>
表示文字 あなたに合ったスキんケアシリーズが分かる ←入力
</h2>
</body>
</html>

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>
6 スキンケア選び
7 </title>
8 <link rel="stylesheet" href="style.css" type="text/css">
9 </head>
10 <body>
11 <h1>
12 自分に合ったスキンケア選び ←確認
13 </h1>
14 <h2>
15 あなたに合ったスキんケアシリーズが分かる
16 </h2>
17 </body>
18

(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。

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

画像

1

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>
表示文字 スキンケア選び
</title>
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body>
<h1>
表示文字 自分に合ったスキンケア選び
</h1>
<h2>
表示文字 あなたに合ったスキんケアシリーズが分かる
</h2>
 ↑入力
</body>
</html>

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>
6 スキンケア選び
7 </title>
8 <link rel="stylesheet" href="style.css" type="text/css">
9 </head>
10 <body>
11 <h1>
12 自分に合ったスキンケア選び
13 </h1>
14 <h2>
15 あなたに合ったスキんケアシリーズが分かる
16 </h2>
17 ←確認
18

(4) 【フォーム】のブロックセット（セレクト・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>ブロックをドラッグし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。

The image consists of two screenshots of the Javloc editor interface, titled "Javloc 条件分岐switch 中級編".

The top screenshot shows the left sidebar with a list of blocks: ページ, 画像, 段落, 見出し1, 見出し2, フォーム, スクリプト・関数, 部品. The "フォーム" block is highlighted with a red box. In the main workspace, a red box highlights a green block containing the code: `<form>` and `</form>`. Below it, a blue block contains the code: `<select name=" " >` and `</select>`. The right sidebar shows the source code area with a snippet of HTML: `<!DOCTYPE HTML>`, `<html>`, `<head>`, `<meta charset="utf-8">`, and `<title>`.

The bottom screenshot shows the same editor after the form block has been inserted. The left sidebar remains the same. In the main workspace, the form block is now placed below the image block. A red box highlights the `<form>` and `</form>` code in the workspace. The right sidebar shows the updated source code, where the `<form>` and `</form>` tags have been added to the HTML output. A red arrow points to the `<form>` tag in the source code with the text "←確認" (Check).

<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」と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】からセレクトのブロックを挿入しなさい。name は hada。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<select name=" " ~>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「hada」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。

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

```
<form>
  <h1>
  </h1>
  </form>
  表示文字 自分に合ったスキンケア選び
  <select name=" " >
  </select>
  表示文字 あなたに合ったスキンケアシリーズが分かる
  
  <option value="0" selected>
  表示文字
```

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>
6 スキンケア選び
7 </title>
8 <link rel="stylesheet" href="style.css" type="text/css">
9 </head>
```

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

```
<body>
  <h1>
  表示文字 自分に合ったスキンケア選び
  </h1>
  <h2>
  表示文字 あなたに合ったスキンケアシリーズが分かる
  </h2>
  
  <form>
    <select name=" hada " > ←入力
    </select>
  </form>
</body>
</html>
```

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>
6 スキンケア選び
7 </title>
8 <link rel="stylesheet" href="style.css" type="text/css">
9 </head>
10 <body>
11 <h1>
12 自分に合ったスキンケア選び
13 </h1>
14 <h2>
15 あなたに合ったスキンケアシリーズが分かる
16 </h2>
17 
18 <form>
19 <select name="hada"> ←確認
20 </select>
21 </form>
22 </body>
```

自分に合ったスキンケア選び

あなたに合ったスキンケアシリーズが分かる



▼

閉じる

(ア)【フォーム】からオプションのブロック (value= "0" : 初期設定値) を選択・挿入し、下記の文字を入力。

あなたの肌質は？

リスト内[フォーム]をクリック、枠内にある `<option value="0" selected ~>` ブロックをドラッグし、`<select>` ブロックの中にドロップ。“ ”の空欄に「あなたの肌質は？」と入力し、ソース表示エリア・プレビューでチェックボックスが設定されていることを確認。

JavaLoc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム スクリプト・関数 部品

```
<form>
  <h1>
  </h1>
  </form> 表示文字 自分に合ったスキンケア選び
  </h1>
  <select name=" " >
    表示文字 あなたに合ったスキンケアシリーズが分かる
  </select>
  <option value="0" selected>
    表示文字
  </option>
```

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>
6     スキンケア選び
7   </title>
8   <link rel="stylesheet" href="style.css" type="text/css">
9 </head>
10 <body>
11 <h1>
12   自分に合ったスキンケア選び
13 </h1>
```

JavaLoc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム スクリプト・関数 部品

```
<body>
  <h1>
    表示文字 自分に合ったスキンケア選び
  </h1>
  <h2>
    表示文字 あなたに合ったスキンケアシリーズが分かる
  </h2>
  
  <form>
    <select name=" hada " >
      <option value="0" selected>
        表示文字 あなたの肌質は？ ←入力
      </option>
    </select>
  </form>
</body>
</html>
```

```
3 <html>
4 <head>
5   <meta charset="utf-8">
6   <title>
7     スキンケア選び
8   </title>
9   <link rel="stylesheet" href="style.css" type="text/css">
10 </head>
11 <body>
12   <h1>
13     自分に合ったスキンケア選び
14   </h1>
15   <h2>
16     あなたに合ったスキンケアシリーズが分かる
17   </h2>
18   
19   <form>
20     <select name="hada">
21       <option value="0" selected>
22         あなたの肌質は？ ←確認
23       </option>
24     </select>
25   </form>
26 </body>
27 </html>
```

自分に合ったスキンケア選び

あなたに合ったスキンケアシリーズが分かる



あなたの肌質は？
あなたの肌質は？

(イ)【フォーム】からオプションのブロック (value= "1") を選択・挿入し、下記の文字を入力。

普通肌

リスト内[フォーム]をクリック、枠内にある<option value="1" ~>ブロックをドラッグし、<option value="0" selected>ブロックの下にドロップ。“”の空欄に「普通肌」と入力し、ソース表示エリア・プレビューでチェックボックスが設定されていることを確認。

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 フォーム スクリプト・関数 部品

表示文字 自分に合ったスキンケア選び

<option value="0" selected> 表示文字 あなたの肌質は？

</option>

<option value="1"> 表示文字 普通肌 </option>

<option value="2"> 表示文字 あなたの肌質は？ </option>

</form>

<option value="3"> 表示文字 </option>

<input type="button" value=" " onclick=" " >

```
<html>
<head>
<meta charset="utf-8">
<title>
スキンケア選び
</title>
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body>
<h1>
自分に合ったスキンケア選び
</h1>
<h2>
あなたに合ったスキンケアシリーズが分かる
</h2>

<form>
<select name="hada">
<option value="0" selected>
あなたの肌質は？
</option>
</select>
</form>
</body>
</html>
```

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 フォーム スクリプト・関数 部品

表示文字 自分に合ったスキンケア選び

表示文字 あなたに合ったスキンケアシリーズが分かる

<form>

<select name="hada">

<option value="0" selected> 表示文字 あなたの肌質は？

</option>

<option value="1"> 表示文字 普通肌 </option>

</select>

</form>

</body>

</html>

```
<meta charset="utf-8">
<title>
スキンケア選び
</title>
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body>
<h1>
自分に合ったスキンケア選び
</h1>
<h2>
あなたに合ったスキンケアシリーズが分かる
</h2>

<form>
<select name="hada">
<option value="0" selected>
あなたの肌質は？
</option>
<option value="1">
普通肌
</option>
</select>
</form>
</body>
</html>
```

自分に合ったスキンケア選び

あなたに合ったスキンケアシリーズが分かる



あなたの肌質は？

あなたの肌質は？

普通肌

(ウ)【フォーム】からオプションのブロック (value=“2”) を選択・挿入し、下記の文字を入力。

敏感肌

リスト内[フォーム]をクリック、枠内にある<option value="2" ~>ブロックをドラッグし、<option value="1">ブロックの下にドロップ。“”の空欄に「敏感肌」と入力し、ソース表示エリア・プレビューでチェックボックスが設定されていることを確認。

JavaLoc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 フォーム スクリプト・関数 部品

```
<select name="hada">
  <option value="0" selected>
    表示文字 自分に合ったスキンケア選び
  </option>
  <option value="1">
    表示文字 普通肌
  </option>
  <option value="2">
    表示文字 敏感肌
  </option>
</select>
```

```
<meta charset="utf-8">
<title>
  スキンケア選び
</title>
<link rel="stylesheet" href="style.css" type="text/css">
</link>
</head>
<body>
  <h1>
    自分に合ったスキンケア選び
  </h1>
  <h2>
    あなたに合ったスキンケアシリーズが分かる
  </h2>
  
  <form>
    <select name="hada">
      <option value="0" selected>
        あなたの肌質は？
      </option>
      <option value="1">
        普通肌
      </option>
      <option value="2">
        敏感肌
      </option>
    </select>
  </form>
</body>
```

JavaLoc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

```
</h1>
<h2>
  表示文字 あなたに合ったスキンケアシリーズが分かる
</h2>

<form>
  <select name="hada">
    <option value="0" selected>
      表示文字 あなたの肌質は？
    </option>
    <option value="1">
      表示文字 普通肌
    </option>
    <option value="2">
      表示文字 敏感肌
    </option>
  </select>
</form>
```

```
</title>
<link rel="stylesheet" href="style.css" type="text/css">
</link>
</head>
<body>
  <h1>
    自分に合ったスキンケア選び
  </h1>
  <h2>
    あなたに合ったスキンケアシリーズが分かる
  </h2>
  
  <form>
    <select name="hada">
      <option value="0" selected>
        あなたの肌質は？
      </option>
      <option value="1">
        普通肌
      </option>
      <option value="2">
        敏感肌
      </option>
    </select>
  </form>
</body>
```

自分に合ったスキンケア選び

あなたに合ったスキンケアシリーズが分かる



あなたの肌質は？▼
あなたの肌質は？
普通肌
敏感肌

(エ)【フォーム】からオプションのブロック (value= "3") を選択・挿入し、下記の文字を入力。

乾燥肌

リスト内[フォーム]をクリック、枠内にある<option value="3" ~>ブロックをドラッグし、<option value="2">ブロックの下にドロップ。""の空欄に「乾燥肌」と入力し、ソース表示エリア・プレビューでチェックボックスが設定されていることを確認。

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム スクリプト・関数 部品

```
<select name="hada">
  <option value="0" selected>
    表示文字 あなたに合ったスキンケアシリーズが分かる
  </option>
  <option value="1">
    表示文字 普通肌
  </option>
  <option value="2">
    表示文字 敏感肌
  </option>
  <option value="3">
    表示文字 乾燥肌
  </option>
</select>
```

```
</title>
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body>
  <h1>
    自分に合ったスキンケア選び
  </h1>
  <h2>
    あなたに合ったスキンケアシリーズが分かる
  </h2>
  
  <form>
    <select name="hada">
      <option value="0" selected>
        あなたの肌質は？
      </option>
      <option value="1">
        普通肌
      </option>
      <option value="2">
        敏感肌
      </option>
      <option value="3">
        乾燥肌
      </option>
    </select>
  </form>
</body>
```

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

```
</h2>

<form>
  <select name="hada">
    <option value="0" selected>
      表示文字 あなたの肌質は？
    </option>
    <option value="1">
      表示文字 普通肌
    </option>
    <option value="2">
      表示文字 敏感肌
    </option>
    <option value="3">
      表示文字 乾燥肌
    </option>
  </select>
</form>
</body>
</html>
```

←入力

```
</head>
<body>
  <h1>
    自分に合ったスキンケア選び
  </h1>
  <h2>
    あなたに合ったスキンケアシリーズが分かる
  </h2>
  
  <form>
    <select name="hada">
      <option value="0" selected>
        あなたの肌質は？
      </option>
      <option value="1">
        普通肌
      </option>
      <option value="2">
        敏感肌
      </option>
      <option value="3">
        乾燥肌
      </option>
    </select>
  </form>
</body>
</html>
```

↓確認

あなたに合ったスキンケアシリーズが分かる



あなたの肌質は？
あなたの肌質は？
普通肌
敏感肌
乾燥肌

- ② 【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。

①で設定した<select>ブロックの下にドロップ。value の“”の空欄に「クリック」、onclick の“”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。

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム スクリプト・関数 部品

`<select name="hada">`
`<option value="0" selected>`
表示文字 あなたの肌質は？
`</option>`
`<option value="1">`
表示文字 普通肌
`</option>`
`<option value="2">`
表示文字 敏感肌
`</option>`
`<option value="3">`
表示文字 乾燥肌
`</option>`
`</select>`

`<input type="button" value="" onclick="" ()">`

```
10 </head>
11 <body>
12 <h1>
13   自分に合ったスキンケア選び
14 </h1>
15 <h2>
16   あなたに合ったスキンケアシリーズが分かる
17 </h2>
18 
19 <form>
20   <select name="hada">
21     <option value="0" selected>
22       あなたの肌質は？
23     </option>
24     <option value="1">
25       普通肌
26     </option>
27     <option value="2">
28       敏感肌
29     </option>
30     <option value="3">
31       乾燥肌
32     </option>
33   </select>
34 </form>
35 </body>
36 </html>
```

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

`<option value="0" selected>`
表示文字 あなたの肌質は？
`</option>`
`<option value="1">`
表示文字 普通肌
`</option>`
`<option value="2">`
表示文字 敏感肌
`</option>`
`<option value="3">`
表示文字 乾燥肌
`</option>`
`</select>`

`<input type="button" value="クリック" onclick="bt ()">` ↓ 入力

```
40 
41 <form>
42   <select name="hada">
43     <option value="0" selected>
44       あなたの肌質は？
45     </option>
46     <option value="1">
47       普通肌
48     </option>
49     <option value="2">
50       敏感肌
51     </option>
52     <option value="3">
53       乾燥肌
54     </option>
55   </select>
56   <input type="button" value="クリック" onclick="bt ()">
57 </form>
58 </body>
59 </html>
```

↓ 確認

あなたに合ったスキンケアシリーズが分かる



あなたの肌質は？

クリック

閉じる

(5) 【段落】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[段落]をクリック、枠内にある<p>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段落が設定されていることを確認。

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ
画像
段落
見出し1
見出し2
フォーム
id
スクリプト・関数
部品

<p>
</p>

表示文字 自分に合ったスキンケア選び

表示文字 あなたに合ったスキンケアシリーズが分かる

<form>
<select name="hada">
<option value="0" selected>
表示文字 あなたの肌質は？
</option>
<option value="1">
表示文字 普通肌
</option>
<option value="2">
表示文字 敏感肌
</option>
<option value="3">
表示文字 乾燥肌
</option>
</select>
<input type="button" value="クリック" onclick="ks()" />
</form>
</body>
</html>

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ
画像
段落
見出し1
見出し2
フォーム
id
スクリプト・関数
部品

</option>
<option value="1">
表示文字 普通肌
</option>
<option value="2">
表示文字 敏感肌
</option>
<option value="3">
表示文字 乾燥肌
</option>
</select>
<input type="button" value="クリック" onclick="ks()" />
</form>
<p>
</p>
</body>
</html>

<h2>
あなたに合ったスキンケアシリーズが分かる
</h2>

<form>
<select name="hada">
<option value="0" selected>
あなたの肌質は？
</option>
<option value="1">
普通肌
</option>
<option value="2">
敏感肌
</option>
<option value="3">
乾燥肌
</option>
</select>
<input type="button" value="クリック" onclick="ks()" />
</form>
<p>
</p>
</body>
</html>

←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<p>ブロックのパズル型空欄にドロップ。id内“”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。

JavaLoc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

id=" " </option> <option value="1"> 表示文字 普通肌 </option> <option value="2"> 表示文字 敏感肌

```
15 <h2>
16 あなたに合ったスキンケアシリーズが分かる
17 </h2>
18 
19 <form>
20 <select name="hada">
21 <option value="0" selected>
22 あなたの肌質は？
23 </option>
```

JavaLoc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

</option> <option value="1"> 表示文字 普通肌 </option> <option value="2"> 表示文字 敏感肌 </option> <option value="3"> 表示文字 乾燥肌 </option> </select> <input type="button" value=" クリック " onclick=" ks ()"> </form> <p id=" output " > ↑ 入力 </p> </body>

```
16 あなたに合ったスキンケアシリーズが分かる
17 </h2>
18 
19 <form>
20 <select name="hada">
21 <option value="0" selected>
22 あなたの肌質は？
23 </option>
24 <option value="1">
25 普通肌
26 </option>
27 <option value="2">
28 敏感肌
29 </option>
30 <option value="3">
31 乾燥肌
32 </option>
33 </select>
34 <input type="button" value="クリック" onclick="ks()">
35 </form>
36 <p id="output"> ← 確認
37 </p>
38 </body>
```

(6) 【画像】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 icon。

リスト内[画像]をクリック、枠内にあるブロックをドラッグし<p>ブロックの下にドロップ。

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ブロックリスト: ページ, 画像, 段落, 見出し1, 見出し2, フォーム, id, スクリプト・関数, 部品

ブロック内容:

```
</option>
<option value="2">
  表示文字 敏感肌
</option>
<option value="3">
  表示文字 乾燥肌
</option>
</select>
<input type="button" value="クリック" onclick="ks()">
</form>
<p id="output">
</p>
<img src="">
```

ソース表示:

```
17 </h2>
18 
19 <form>
20 <select name="hada">
21 <option value="0" selected>
22 あなたの肌質は？
23 </option>
24 <option value="1">
25 普通肌
26 </option>
27 <option value="2">
28 敏感肌
29 </option>
30 <option value="3">
31 乾燥肌
32 </option>
33 </select>
34 <input type="button" value="クリック" onclick="ks()">
35 </form>
36 <p id="output">
37
38 <img src=""> ←確認
39 </body>
```

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。id内“”の空欄に「icon」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。

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ブロックリスト: ページ, 画像, 段落, 見出し1, 見出し2, フォーム, id, スクリプト・関数, 部品

ブロック内容:

```
</option>
<option value="2">
  表示文字 敏感肌
</option>
<option value="3">
  表示文字 乾燥肌
</option>
</select>
<input type="button" value="クリック" onclick="ks()">
</form>
<p id="output">
</p>
<img src="" id="icon">
```

ソース表示:

```
17 </h2>
18 
19 <form>
20 <select name="hada">
21 <option value="0" selected>
22 あなたの肌質は？
23 </option>
24 <option value="1">
25 普通肌
26 </option>
27 <option value="2">
28 敏感肌
29 </option>
30 <option value="3">
31 乾燥肌
32 </option>
33 </select>
34 <input type="button" value="クリック" onclick="ks()">
35 </form>
36 <p id="output">
37 </p>
38 <img src="" id="icon"> ←確認
39 </body>
```

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、処理をしない。

〔前提〕

3種類の商品があり、選択された項目に合致する商品の文章・画像をページ下部に表示させる。

〔機能〕

- ・フォームのオプションで選択された項目に合致する文章を3-(5)、画像を3-(6)の位置に表示する。
- ・いずれも選択されていない場合には、文章・画像は表示しない。



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。

Javloc 条件分岐switch 初級編

リセット 元に戻す やり直し **解答例** プレビュー ← → ブロック表示 ブロック保存 ページ保存

このフォームの部分を操作してみましょう。
プルダウン（セレクト）から項目を選択し、「クリック」ボタンを押してみます。
右図の4パターンを試してみましょう。

あなたの肌の質は？

段落 id="output"

画像 id="icon"

プルダウン（セレクト）で選択された商品の宣伝の文章とイメージ画像を「クリック」ボタンが押された際にフォームの下に表示する。

既定値（デフォルト値）「あなたの肌の質は？」が選択され「クリック」ボタンが押された際は、何も表示しない。

普通肌

自分に合ったスキンケア選び

あなたに合ったスキンケアシリーズが分かる

普通肌 クリック

緑茶成分

敏感肌

自分に合ったスキンケア選び

あなたに合ったスキンケアシリーズが分かる

敏感肌 クリック

天然ハーブ

乾燥肌

自分に合ったスキンケア選び

あなたに合ったスキンケアシリーズが分かる

乾燥肌 クリック

梅の恵み

既定値: あなたの肌の質は？

自分に合ったスキンケア選び

あなたに合ったスキンケアシリーズが分かる

あなたの肌の質は？ クリック

(1) 【スクリプト・関数】ブロックを挿入し、3-(4)-②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。

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

```
<script>
function () {
}
</script>
```

表示文字 スキンケア選び

<link rel="stylesheet" href="style.css" type="text/css">

</head>

<body>

<h1>

表示文字 自分に合ったスキンケア選び

</h1>

<h2>

表示文字 あなたに合ったスキンケアシリーズが分かる

</h2>

<form>

<select name="hada">

<option value="0" selected>

あなた肌の質は？

</option>

<option value="1">

普通肌

</form>

</body>

</html>

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>
表示文字 スキンケア選び
</title>
<link rel="stylesheet" href="style.css" type="text/css">
<script>
function () {
}
</script>
</head>
<body>
<h1>
表示文字 自分に合ったスキンケア選び
</h1>
<h2>
表示文字 あなたに合ったスキンケアシリーズが分かる
</h2>
```


<form>

<select name="hada">

<option value="0" selected>

あなた肌の質は？

</option>

<option value="1">

普通肌

</form>

</body>

</html>

3-(4)-②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="bt()"` 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。

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ
画像
段落
見出し 1
見出し 2
フォーム
id
スクリプト・関数
部品

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>
  表示文字
</title>
<link rel="stylesheet" href="style.css" type="text/css">
<script>
  function bt() { ←入力
  }
</script>
</head>
<body>
<h1>
  表示文字 自分に合ったスキンケア選び
</h1>
<h2>
  表示文字 あなたに合ったスキンケアシリーズが分かる
</h2>

</body>
</html>
```

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>
6
7
8 </title>
9 <link rel="stylesheet" href="style.css" type="text/css">
10 <script>
11   function bt() { ←確認
12   }
13 </script>
14 </head>
15 <body>
16 <h1>
17   自分に合ったスキンケア選び
18 </h1>
19 <h2>
20   あなたに合ったスキンケアシリーズが分かる
21 </h2>
```

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。

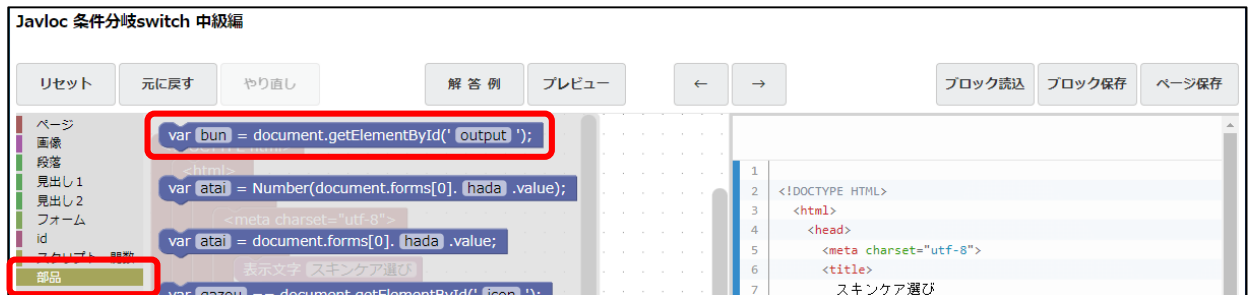
(7) 【部品】内から正しいブロックを選択し、(1)の関数ブロックの中に下記処理手順どおりに組み立てなさい。

■処理手順

① 変数 bun 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 bun =` ；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。

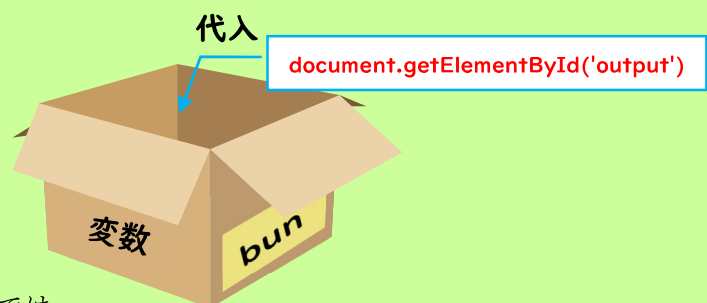
ブロック選択のヒント 問題文に「変数」とあったら「var」を含むブロックを選びましょう。



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 `bun` に代入しています。

③ 変数 gazou を宣言し、id (icon) 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 `var bun =` ; 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。

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

```
var bun = document.getElementById('output');
<html>
var atai = Number(document.forms[0].hada.value);
<meta charset="utf-8">
var atai = document.forms[0].hada.value;
表示文字 スキンケア選び
var gazou = document.getElementById('icon');
var gazou = document.getElementById('icon');
switch ( ) {
  case 1:
    .innerHTML = ' ';
    .src = ' ';
    break;
  case 2:
    .innerHTML = ' ';
    .src = ' ';
    break;
  case 3:
    .innerHTML = ' ';
    .src = ' ';
    break;
}
```

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>
6 スキンケア選び
7 </title>
8 <link rel="stylesheet" href="style.css" type="text/css">
9 <script>
10 function ks() {
11   var bun = document.getElementById('output');
12 }
13 </script>
14 </head>
15 <body>
16 <h1>
17 自分に合ったスキンケア選び
18 </h1>
19 <h2>
20 あなたに合ったスキンケアシリーズが分かる
21 </h2>
22 
23 <form>
24 <select name="hada">
```

[部品]の中には、誤ったブロックが混ざっています。

×

```
var gazou == document.getElementById('icon');
```

○

```
var gazou = document.getElementById('icon');
```

← 代入 = になっていない
比較演算子 == になっている

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>
表示文字 スキンケア選び
</title>
<link rel="stylesheet" href="style.css" type="text/css">
<script>
function ks() {
  var bun = document.getElementById('output');
  var gazou = document.getElementById('icon');
}
</script>
```

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>
6 スキンケア選び
7 </title>
8 <link rel="stylesheet" href="style.css" type="text/css">
9 <script>
10 function ks() {
11   var bun = document.getElementById('output');
12   var gazou = document.getElementById('icon');
13 }
14 </script>
15 </head>
16 <body>
```

↑ 確認

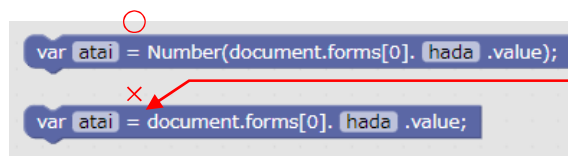
④ 変数 atai を宣言し、フォームのセレクト (hada) の値を数値に変換し代入

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。



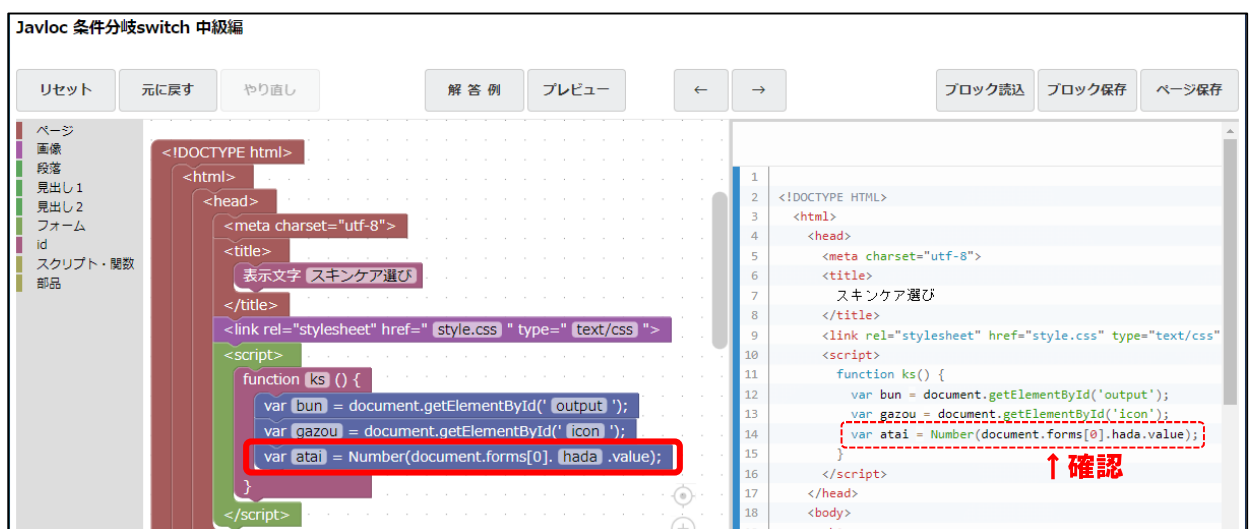
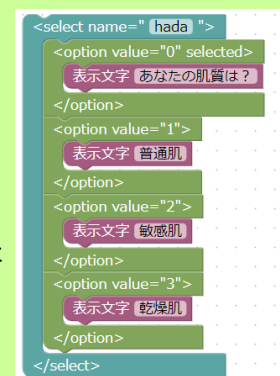
Number()がない
処理条件に「数値に変換し」とあるため、数値変換の処理が必要です。

Number(～)は、(～) 内にある値を数値に変換します。

document.forms[0]は、そのページにある form 要素の中で1番最初に出現するもののことです。

.hada.value は、右図で示されている hada という名前の付いたセレクトボックス内で選択された項目の値を指します。

この値は文字列ですが、このあと設定する④の switch の case は数値となっているため、Number(～)を用いて数値に変換する必要があります。このように、型を合わせる処理をしていないと、正しく動作しません。



④ 「atai」の値によって条件分岐

(ア) 「atai」が1の場合、下記の処理を実行し終了

処理 HTML 挿入部分 (bun): 緑茶成分

画像 (gazou): pic1.jpg

(イ) 「atai」が2の場合、下記の処理を実行し終了

処理 HTML 挿入部分 (bun): 天然ハーブ

画像 (gazou): pic2.jpg

(ウ) 「atai」が3の場合、下記の処理を実行し終了

処理 HTML 挿入部分 (bun): 梅の恵み

画像 (gazou): pic3.jpg

(エ) 「atai」の値がどれにも当てはまらないときは「default」を選択し終了

処理 HTML 挿入部分 (bun)・画像 (gazou) いずれも表示しない

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。

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム ID フォント 部品

```
switch( ) {  
  case 1:  
    .innerHTML = '  
    .src = '  
    break;  
  case 2:  
    .innerHTML = '  
    .src = '  
    break;  
  case 3:  
    .innerHTML = '  
    .src = '  
    break;  
  default:  
    .innerHTML = '  
    .src = '  
    break;  
}
```

スキニングアタチ

```
</title>  
<link rel="stylesheet" href="style.css" type="text/css">  
<script>  
  function ks() {  
    var bun = document.getElementById('output');  
    var gazou = document.getElementById('icon');  
    var atai = Number(document.forms[0].hada.value);  
  }  
</script>  
</head>  
<body>  
<h1>  
  自分に合ったスキニングアタチ  
</h1>  
<h2>  
  あなたに合ったスキニングアタチシリーズが分かる  
</h2>  
  
<form>  
  <select name="hada">  
    <option value="0" selected>  
      あなたの肌質は？  
    </option>  
    <option value="1">  
      普通肌  
    </option>  
    <option value="2">  
      敏感肌  
    </option>  
    <option value="3">
```

[部品]の中には、誤ったブロックが混ざっています。

まず、条件分岐を行う「switch」を含むブロックに絞る、

処理条件に (エ) 「atai」の値がどれにも当てはまらないとき「default」を選択し終了

とあるため、default があるものを選択しましょう。

○ switch() {
 case 1:
 .innerHTML = '
 .src = '
 break;
 case 2:
 .innerHTML = '
 .src = '
 break;
 case 3:
 .innerHTML = '
 .src = '
 break;
 default:
 .innerHTML = '
 .src = '
 break;
}

✗ switch() {
 case 1:
 .innerHTML = '
 .src = '
 break;
 case 2:
 .innerHTML = '
 .src = '
 break;
 case 3:
 .innerHTML = '
 .src = '
 break;
}

ドラッグして③で設定したブロックの下にドロップ。

JavaLoc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

```
var gazou = document.getElementById('icon');
var atai = Number(document.forms[0].hada.value);

switch( ) {
  case 1:
    .innerHTML = ' ';
    .src = ' ';
    break;
  case 2:
    .innerHTML = ' ';
    .src = ' ';
    break;
  case 3:
    .innerHTML = ' ';
    .src = ' ';
    break;
  default:
    .innerHTML = ' ';
    .src = ' ';
    break;
}
</script>
</head>
```

スキンケア選び

```
</title>
<link rel="stylesheet" href="style.css" type="text/css">
<script>
function ks() {
  var bun = document.getElementById('output');
  var gazou = document.getElementById('icon');
  var atai = Number(document.forms[0].hada.value);
  switch () {
    case 1:
      .innerHTML = '';
      .src = '';
      break;
    case 2:
      .innerHTML = '';
      .src = '';
      break;
    case 3:
      .innerHTML = '';
      .src = '';
      break;
    default:
      .innerHTML = '';
      .src = '';
      break;
  }
}
</script>
</head>
<body>
<h1>
```

switch の空欄に、「**atai**」と入力。

case 1 の.innerHTML の前の空欄に、「**bun**」、の空欄に「緑茶成分」、

.src の前の空欄に「**gazou**」、の空欄に「**pic1.jpg**」と入力。

case 2 の.innerHTML の前の空欄に、「**bun**」、の空欄に「天然ハーブ」、

.src の前の空欄に「**gazou**」、の空欄に「**pic2.jpg**」と入力。

case 3 の.innerHTML の前の空欄に、「**bun**」、の空欄に「梅の恵み」、

.src の前の空欄に「**gazou**」、の空欄に「**pic3.jpg**」と入力。

default の.innerHTML の前の空欄に、「**bun**」、の空欄には何も入力しない、

.src の前の空欄に「**gazou**」、の空欄には何も入力しない。

※赤文字の箇所は、必ず半角で入力しましょう。

JavaLoc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id スクリプト・関数 部品

```
var gazou = document.getElementById('icon');
var atai = Number(document.forms[0].hada.value);

switch( atai ){
  case 1:
    bun.innerHTML = '緑茶成分';
    gazou.src = 'pic1.jpg';
    break;
  case 2:
    bun.innerHTML = '天然ハーブ';
    gazou.src = 'pic2.jpg';
    break;
  case 3:
    bun.innerHTML = '梅の恵み';
    gazou.src = 'pic3.jpg';
    break;
  default:
    bun.innerHTML = ' ';
    gazou.src = ' ';
    break;
}
</script>
</head>
```

スキンケア選び

```
</title>
<link rel="stylesheet" href="style.css" type="text/css">
<script>
function ks() {
  var bun = document.getElementById('output');
  var gazou = document.getElementById('icon');
  var atai = Number(document.forms[0].hada.value);
  switch (atai) {
    case 1:
      bun.innerHTML = '緑茶成分';
      gazou.src = 'pic1.jpg';
      break;
    case 2:
      bun.innerHTML = '天然ハーブ';
      gazou.src = 'pic2.jpg';
      break;
    case 3:
      bun.innerHTML = '梅の恵み';
      gazou.src = 'pic3.jpg';
      break;
    default:
      bun.innerHTML = ' ';
      gazou.src = ' ';
      break;
  }
}
</script>
</head>
<body>
<h1>
```

←確認

↑何も入力しないこと

switch (□)は、□の値を読み込んで、それぞれ場合を分けて処理を行います。
 case (△)は、switch の□が△だった場合という意味になります。
 innerHTML は、HTML の中身を変更する処理を行います。
 .src は、画像や動画などのメディア要素のソースを指定する処理を行います。
 break で、実行中の処理から抜け出す処理を行います。
 default は、どの条件にも当てはまらなかった場合という意味になります。

<作成ページ例>form 部分

<HTML>

```
<form>
  <select name="hada">
    <option value="0" selected>
      表示文字 あなたの肌質は？
    </option>
    <option value="1">
      表示文字 普通肌
    </option>
    <option value="2">
      表示文字 敏感肌
    </option>
    <option value="3">
      表示文字 乾燥肌
    </option>
  </select>
  <input type="button" value="クリック" onclick="js()" />
</form>
<p id="output"></p>
</p>
<img src="" id="icon">
```

<JavaScript>

```
<script>
function bt () {
  var bun = document.getElementById('output');
  var picou = document.getElementById('icon');
  var eta = Number(document.forms[0].hada.value);
  switch( eta ){
    case 1:
      bun.innerHTML = '緑茶成分';
      picou.src = 'pic1.jpg';
      break;
    case 2:
      bun.innerHTML = '天然ハーブ';
      picou.src = 'pic2.jpg';
      break;
    case 3:
      bun.innerHTML = '梅の恵み';
      picou.src = 'pic3.jpg';
      break;
    default:
      bun.innerHTML = ' ';
      picou.src = ' ';
      break;
  }
}
```

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。

Javloc 条件分岐switch 中級編

リセット 元に戻す やり直し 解答例 **プレビュー** ← ← → ブロック読み込み ブロック保存 ページ保存

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。

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]

肌質に合わせた化粧品を紹介する。選択された肌質に合わせた商品の商品名・画像を表示するページ。
商品の種類は3種類。

インプット部分の確認

3. の(4)入力フォーム

<ブロック>

```
<form>
  <select name="hada">
    <option value="0" selected>
      表示文字 あなたの肌質は?
    </option>
    <option value="1">
      表示文字 普通肌
    </option>
    <option value="2">
      表示文字 敏感肌
    </option>
    <option value="3">
      表示文字 乾燥肌
    </option>
  </select>
  <input type="button" value=" クリック " onclick=" bt () ">
</form>
```

<作成ページ例>



アウトプット部分の確認

3. の(5)段落、(6)画像

<ブロック>

```
<p id="output">
</p>
<img src="" id="icon">
```

<作成ページ例>



仕様部分を読み解く

【仕様】

- ・画面の選択肢表示ボックス（セレクト）で肌質を選択して「クリック」ボタンを押すと、選択された肌質に合った商品名と画像を表示する。

→インプット部分・アウトプット部分の各設定値を確認。

→商品名を表示する場所 (id=output) を取得し、設定するための箱（変数）を準備。

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

→選択肢表示ボックスで選択された値を入れておくための箱（変数）を準備。

→選択された値によって場合分けをする。case 1～case 3 にそれぞれ表示する商品名・画像を設定。

- ・商品 3 種類の選択肢以外の場合は、商品名・画像は表示しない。

→case 1～case 3 以外の場合、商品名・画像は空欄を設定。

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

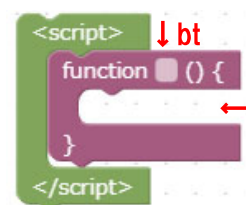
【使用定義】

関数

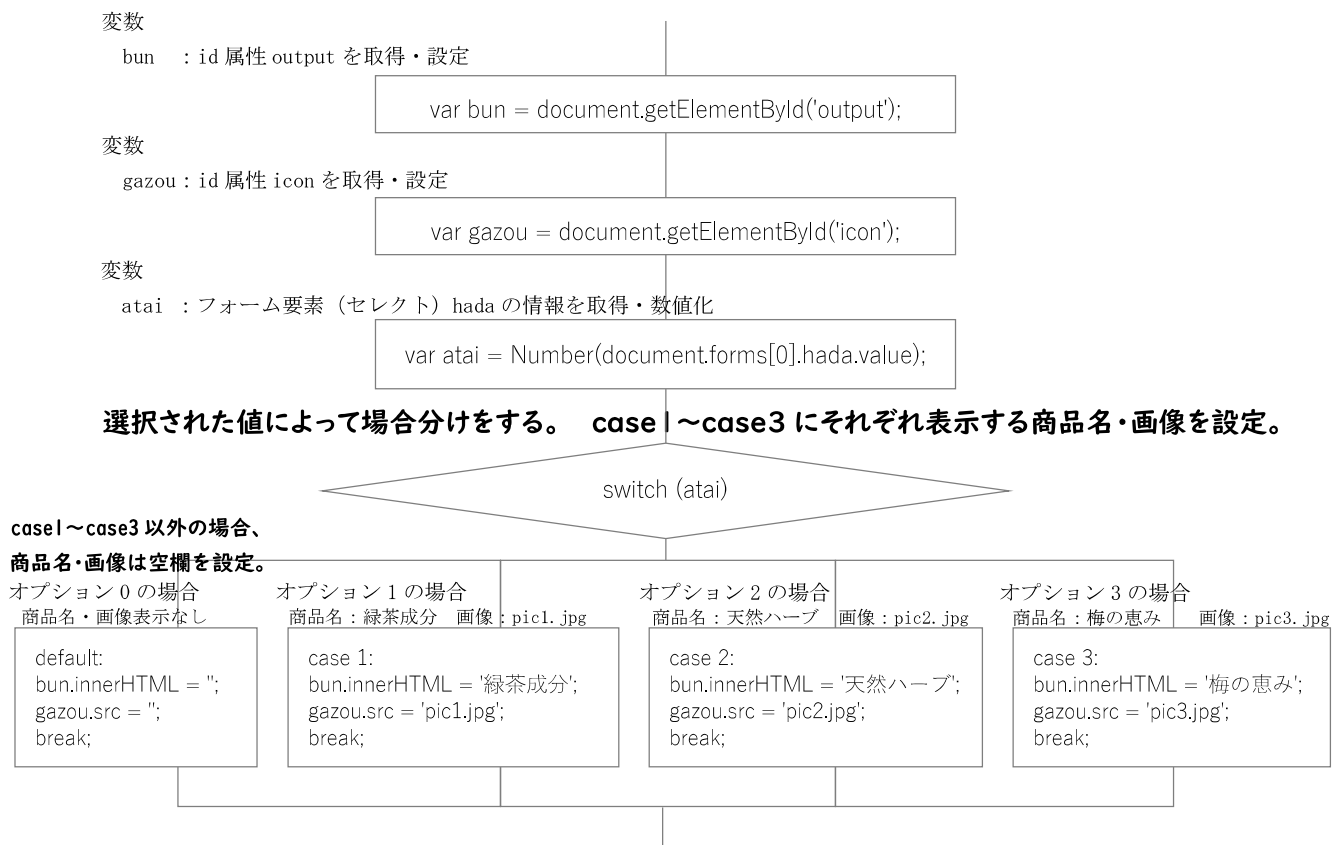
bt: 「クリック」ボタンで動作する処理の関数名

→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。



フローチャートを基にブロックを組み立てる（プログラミング）

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。

The first screenshot shows the Javloc editor interface. On the left, a list of blocks includes 'JavaScript'. In the center workspace, a '<script>' block is highlighted with a red box. On the right, the source code area shows the initial HTML structure with a '<link rel="stylesheet" href="style.css" type="text/css">' block at the bottom.

The second screenshot shows the same interface after the '<script>' block has been moved. It is now located inside the '<link>' block in the source code, as indicated by a red dashed box. The workspace also shows the updated structure with the '<script>' block nested within the '<link>' block.

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。

The first screenshot shows the Javloc editor interface. On the left, a list of blocks includes 'JavaScript'. In the center workspace, a '<function>' block is highlighted with a red box. On the right, the source code area shows the initial HTML structure with a '<script>' block at the bottom.

The second screenshot shows the same interface after the '<function>' block has been moved. It is now located inside the '<script>' block in the source code, as indicated by a red dashed box. The workspace also shows the updated structure with the '<function>' block nested within the '<script>' block. The function name 'bt' is visible in the workspace.

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

Javloc 条件分岐switch 上級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id JavaScript

```
case 2: <html>
</html>
case 3: <meta charset="utf-8">
</meta>
JavaScript
}
表示文字 スキンケア選び
switch( ) {
case 1: <link rel="stylesheet" href="style.css" type="text/css">
</link>
case 2: <script>
function bt() {
}
</script>
case 3:
default: <input>
</input>
}
<h1>
</h1>
var gazou = document.getElementById('icon ');
var gazou == document.getElementById('icon ');
var bun = document.getElementById('output ');
```

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>
6 スキンケア選び
7 </title>
8 <link rel="stylesheet" href="style.css" type="text/css">
9 <script>
10 function bt() {
11 }
12 </script>
13 </head>
14 <body>
15 <h1>
16 自分に合ったスキンケア選び
17 </h1>
18 <h2>
19 あなたに合ったスキンケアシリーズが分かる
20 </h2>
21 
22 <form>
23 <select name="hada">
24 <option value="0" selected>
25 あなたの肌質は？
26 </option>
27 <option value="1">
28 普通肌
29 </option>
30
```

Javloc 条件分岐switch 上級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id JavaScript

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>
表示文字 スキンケア選び
</title>
<link rel="stylesheet" href="style.css" type="text/css">
</link>
<script>
function bt() {
var bun = document.getElementById('output ');
}
</script>
</head>
<body>
```

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>
6 スキンケア選び
7 </title>
8 <link rel="stylesheet" href="style.css" type="text/css">
9 <script>
10 function bt() {
11 var bun = document.getElementById('output ');
12 }
13 </script>
14 </head>
15 <body>
16 <h1>
17 自分に合ったスキンケア選び
18 </h1>
19 <h2>
20 あなたに合ったスキンケアシリーズが分かる
21
```


リスト内[JavaScript]をクリック、枠内にある `var gazou = document.getElementById('icon');` ブロックを選択し、ドラッグして `var bun = document.getElementById('output');` 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。

Javloc 条件分岐switch 上級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム Id JavaScript

```
case 2: <html>
</html>
case 3:
<meta charset="utf-8">
}
</head>
<h1>
表示文字 スキンケア選び
</h1>
switch( ) {
case 1:
<link rel="stylesheet" href="style.css" type="text/css">
</link>
<script>
case 2:
function bt() {
var bun = document.getElementById('output');
case 3:
}
default:
</script>
}
</body>
</html>
var gazou = document.getElementById('icon');
</html>
var gazou = document.getElementById('icon');
表示文字 あなたに合ったスキンケアシリーズがわかる
var bun = document.getElementById('output');
```

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>
6 スキンケア選び
7 </title>
8 <link rel="stylesheet" href="style.css" type="text/css">
9 <script>
10 function bt() {
11 var bun = document.getElementById('output');
12 }
13 </script>
14 </head>
15 <body>
16 <h1>
17 自分に合ったスキンケア選び
18 </h1>
19 <h2>
20 あなたに合ったスキンケアシリーズがわかる
21 </h2>
22 
23 <form>
24 <select name="hada">
25 <option value="0" selected>
26 あなたの肌質は？
27 </option>
28 <option value="1">
29 普通肌
30
```

Javloc 条件分岐switch 上級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム Id JavaScript

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>
表示文字 スキンケア選び
</title>
<link rel="stylesheet" href="style.css" type="text/css">
<script>
function bt() {
var bun = document.getElementById('output');
var gazou = document.getElementById('icon');
}
</script>
</head>
```

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>
6 スキンケア選び
7 </title>
8 <link rel="stylesheet" href="style.css" type="text/css">
9 <script>
10 function bt() {
11 var bun = document.getElementById('output');
12 var gazou = document.getElementById('icon');
13 }
14 </script>
15 </head>
16 <body>
17 <h1>
18 自分に合ったスキンケア選び
19 </h1>
```

リスト内[JavaScript]をクリック、枠内にある `var atai = Number(document.forms[0].hada.value);` ブロックを選択し、ドラッグして `var gazou = document.getElementById('icon');` 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。

Javloc 条件分岐switch 上級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

JavaScript

```

<script>
<html>
<head>
<meta charset="utf-8">
function () {
    // 表示文字 (スキニングア)
}
</script>
<link rel="stylesheet" href="style.css" type="text/css">
<script>
    function bt() {
        var bun = document.getElementById('output');
        var gazou = document.getElementById('icon');
    }
</script>
</head>
<body>
<h1>

```

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

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

var atai = document.forms[0].hada.value;

リセット

元に戻す

やり直し

解答例

プレビュー

←

→

ブロック読込

ブロック保存

ページ保存

ページ

画像

段落

見出し 1

見出し 2

フォーム

id

JavaScript

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>
表示文字 スキンケア選び
</title>
<link rel="stylesheet" href="style.css" type="text/css">
<script>
function bt () {
var bun = document.getElementById('output');
var gazou = document.getElementById('icon');
var atai = Number(document.forms[0].hada.value);
}
</script>

```

```

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>
6 スキンケア選び
7 </title>
8 <link rel="stylesheet" href="style.css" type="text/css">
9 <script>
10 function bt() {
11     var bun = document.getElementById('output');
12     var gazou = document.getElementById('icon');
13     var atai = Number(document.forms[0].hada.value);
14 }
15 </script>
16 </head>
17 <body>
18 <h1>
19 自分に合ったスキンケア選び
20 </h1>

```

リスト内[JavaScript]をクリック、枠内にある下図ブロックを選択し、ドラッグして
var atai = Number(document.forms[0].hada.value);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。

Javloc 条件分岐switch 上級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム JavaScript

```
switch( ) {  
  case 1:  
  case 2:  
  case 3:  
}
```

```
function bt() {  
  var bun = document.getElementById('output');  
  var gazou = document.getElementById('icon');  
  var atai = Number(document.forms[0].hada.value);  
}
```

```
<link rel="stylesheet" href="style.css" type="text/css">  
<script>  
  function bt() {  
    var bun = document.getElementById('output');  
    var gazou = document.getElementById('icon');  
    var atai = Number(document.forms[0].hada.value);  
  }  
</script>  
</head>  
<body>  
  <h1>  
    自分に合ったスキンケア選び  
  </h1>  
  <h2>  
    あなたに合ったスキンケアシリーズが分かる  
  </h2>  
    
  <form>  
    <select name="hada">  
      <option value="0" selected>  
        あなたの肌質は？  
      </option>  
      <option value="1">  
        普通肌  
      </option>  
      <option value="2">  
        敏感肌  
      </option>  
      <option value="3">
```

Javloc 条件分岐switch 上級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム JavaScript

```
function bt() {  
  var bun = document.getElementById('output');  
  var gazou = document.getElementById('icon');  
  var atai = Number(document.forms[0].hada.value);  
  switch (atai) {  
    case 1:  
    case 2:  
    case 3:  
    default:  
  }  
}
```

```
<!DOCTYPE HTML>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>  
      スキンケア選び  
    </title>  
    <link rel="stylesheet" href="style.css" type="text/css">  
    <script>  
      function bt() {  
        var bun = document.getElementById('output');  
        var gazou = document.getElementById('icon');  
        var atai = Number(document.forms[0].hada.value);  
        switch (atai) {  
          case 1:  
          case 2:  
          case 3:  
          default:  
        }  
      }  
    </script>  
  </head>
```

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 case 1: の中にドロップ。変数「bun」、文字列「緑茶成分」を設定。ソース表示エリアで設定を確認。

Javloc 条件分岐switch 上級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム JavaScript

```
break;
function bt () {
  var bun = document.getElementById('output');
  var atai = Number(document.forms[0].hada.value);
  var icon = document.getElementById('icon');
  var atai = Number(document.forms[0].hada.value);
  var atai = document.forms[0].hada.value;
  .innerHTML = ' ';
  .src = ' ';
  case 3:

```

```
1
2 <!DOCTYPE HTML>
3 <html>
4 <head>
5   <meta charset="utf-8">
6   <title>
7     スキンケア選び
8   </title>
9   <link rel="stylesheet" href="style.css" type="text/css">
10  <script>
11    function bt() {
12      var bun = document.getElementById('output');
```

Javloc 条件分岐switch 上級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

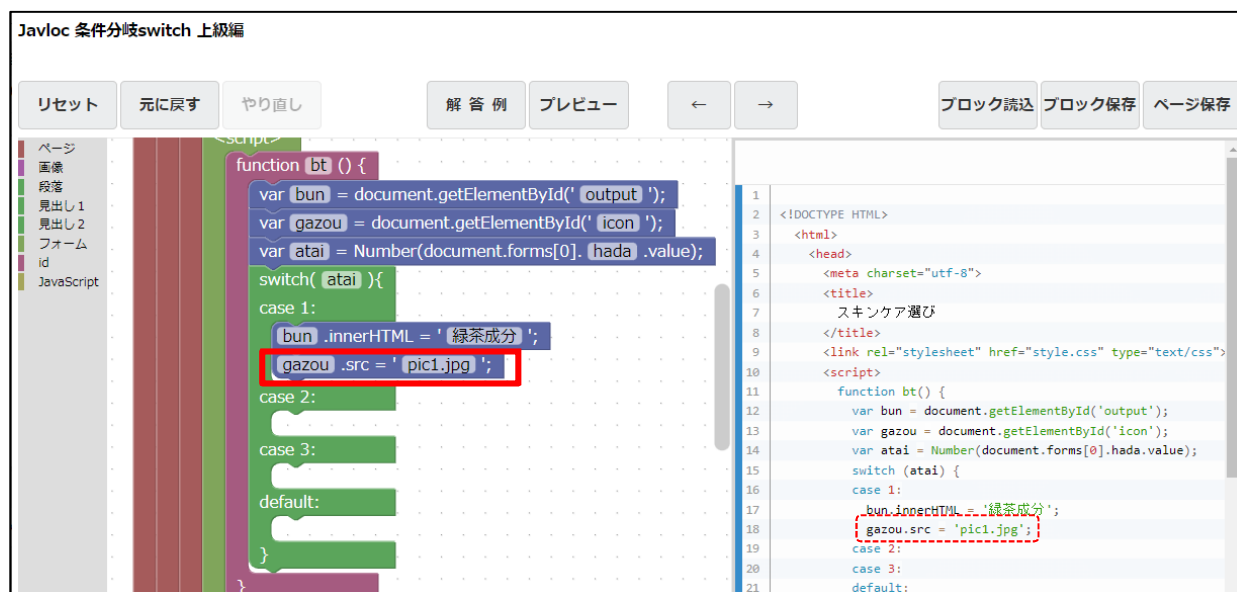
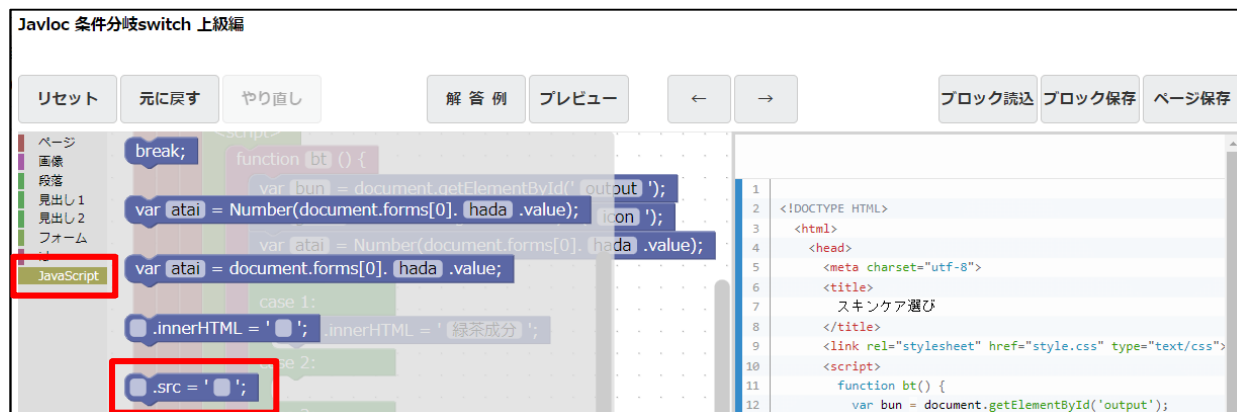
ページ 画像 段落 見出し1 見出し2 フォーム id JavaScript

```
function bt () {
  var bun = document.getElementById('output');
  var gazou = document.getElementById('icon');
  var atai = Number(document.forms[0].hada.value);
  switch (atai) {
    case 1:
      bun.innerHTML = '緑茶成分';
    case 2:
    case 3:
    default:
  }
}

```

```
1
2 <!DOCTYPE HTML>
3 <html>
4 <head>
5   <meta charset="utf-8">
6   <title>
7     スキンケア選び
8   </title>
9   <link rel="stylesheet" href="style.css" type="text/css">
10  <script>
11    function bt() {
12      var bun = document.getElementById('output');
13      var gazou = document.getElementById('icon');
14      var atai = Number(document.forms[0].hada.value);
15      switch (atai) {
16        case 1:
17          bun.innerHTML = '緑茶成分';
18        case 2:
19        case 3:
20        default:
```

リスト内[JavaScript]をクリック、枠内にある□.src = ;ブロックを選択し、ドラッグして□.innerHTML= ;の下にドロップ。変数「gazou」、画像ファイル名「pic1.jpg」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 break;ブロックを選択し、ドラッグして gazou.src = ;の下にドロップ。ソース表示エリアで設定を確認。

Javloc 条件分岐switch 上級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id JavaScript

```
function bt() {  
  var bun = document.getElementById('output');  
  var gazou = document.getElementById('icon');  
  var atai = Number(document.forms[0].hada.value);  
  var atai = Number(document.forms[0].hada.value);  
}
```

```
1 <!DOCTYPE HTML>  
2 <html>  
3 <head>  
4   <meta charset="utf-8">  
5   <title>  
6     スキンケア選び  
7
```

Javloc 条件分岐switch 上級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id JavaScript

```
function bt() {  
  var bun = document.getElementById('output');  
  var gazou = document.getElementById('icon');  
  var atai = Number(document.forms[0].hada.value);  
  switch (atai) {  
    case 1:  
      bun.innerHTML = '緑茶成分';  
      gazou.src = 'pic1.jpg';  
      break;  
    case 2:  
    case 3:  
    default:  
  }  
}
```

```
1 <!DOCTYPE HTML>  
2 <html>  
3 <head>  
4   <meta charset="utf-8">  
5   <title>  
6     スキンケア選び  
7 </title>  
8 </head>  
9 <link rel="stylesheet" href="style.css" type="text/css">  
10 <script>  
11   function bt() {  
12     var bun = document.getElementById('output');  
13     var gazou = document.getElementById('icon');  
14     var atai = Number(document.forms[0].hada.value);  
15     switch (atai) {  
16       case 1:  
17         bun.innerHTML = '緑茶成分';  
18         gazou.src = 'pic1.jpg';  
19         break;  
20       case 2:  
21       case 3:  
22       default:
```


case 2: も case 1: と同様にブロックを選択し、ドロップ。変数「bun」、文字列「天然ハーブ」、変数「gazou」、画像ファイル名「pic2.jpg」を設定。ソース表示エリアで設定を確認。

Javloc 条件分岐switch 上級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id JavaScript

```
function bt () {  
  var bun = document.getElementById('output');  
  var gazou = document.getElementById('icon');  
  var atai = Number(document.forms[0].hada.value);  
  switch(atai){  
    case 1:  
      bun.innerHTML = '緑茶成分';  
      gazou.src = 'pic1.jpg';  
      break;  
    case 2:  
      bun.innerHTML = '天然ハーブ';  
      gazou.src = 'pic2.jpg';  
      break;  
    case 3:  
      // ...  
    default:  
      // ...  
  }  
}
```

```
<!DOCTYPE HTML>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>  
      スキンケア選び  
    </title>  
    <link rel="stylesheet" href="style.css" type="text/css">  
    <script>  
      function bt() {  
        var bun = document.getElementById('output');  
        var gazou = document.getElementById('icon');  
        var atai = Number(document.forms[0].hada.value);  
        switch(atai) {  
          case 1:  
            bun.innerHTML = '緑茶成分';  
            gazou.src = 'pic1.jpg';  
            break;  
          case 2:  
            bun.innerHTML = '天然ハーブ';  
            gazou.src = 'pic2.jpg';  
            break;  
          case 3:  
            // ...  
          default:  
            // ...  
        }  
      }  
    </script>  
  </head>  
</html>
```

case 3: も case 1: 、case 2: と同様にブロックを選択し、ドロップ。変数「bun」、文字列「梅の恵み」、変数「gazou」、画像ファイル名「pic3.jpg」を設定。ソース表示エリアで設定を確認。

Javloc 条件分岐switch 上級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

ページ 画像 段落 見出し1 見出し2 フォーム id JavaScript

```
function bt () {  
  var bun = document.getElementById('output');  
  var gazou = document.getElementById('icon');  
  var atai = Number(document.forms[0].hada.value);  
  switch(atai){  
    case 1:  
      bun.innerHTML = '緑茶成分';  
      gazou.src = 'pic1.jpg';  
      break;  
    case 2:  
      bun.innerHTML = '天然ハーブ';  
      gazou.src = 'pic2.jpg';  
      break;  
    case 3:  
      bun.innerHTML = '梅の恵み';  
      gazou.src = 'pic3.jpg';  
      break;  
    default:  
      // ...  
  }  
}
```

```
<!DOCTYPE HTML>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>  
      スキンケア選び  
    </title>  
    <link rel="stylesheet" href="style.css" type="text/css">  
    <script>  
      function bt() {  
        var bun = document.getElementById('output');  
        var gazou = document.getElementById('icon');  
        var atai = Number(document.forms[0].hada.value);  
        switch(atai) {  
          case 1:  
            bun.innerHTML = '緑茶成分';  
            gazou.src = 'pic1.jpg';  
            break;  
          case 2:  
            bun.innerHTML = '天然ハーブ';  
            gazou.src = 'pic2.jpg';  
            break;  
          case 3:  
            bun.innerHTML = '梅の恵み';  
            gazou.src = 'pic3.jpg';  
            break;  
          default:  
            // ...  
        }  
      }  
    </script>  
  </head>  
</html>
```

default:も case 1: 、case 2: 、case 3: と同様にブロックを選択し、ドロップ。変数「bun」、変数「gazou」を設定。ソース表示エリアで設定を確認。

Javloc 条件分岐switch 上級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

```
switch( atai ){
  case 1:
    bun.innerHTML = '緑茶成分';
    gazou.src = 'pic1.jpg';
    break;
  case 2:
    bun.innerHTML = '天然ハーブ';
    gazou.src = 'pic2.jpg';
    break;
  case 3:
    bun.innerHTML = '梅の恵み';
    gazou.src = 'pic3.jpg';
    break;
  default:
    bun.innerHTML = ' ';
    gazou.src = ' ';
    break;
}
```

```
<link rel="stylesheet" href="style.css" type="text/css">
<script>
function bt() {
  var bun = document.getElementById('output');
  var gazou = document.getElementById('icon');
  var atai = Number(document.forms[0].hada.value);
  switch (atai) {
    case 1:
      bun.innerHTML = '緑茶成分';
      gazou.src = 'pic1.jpg';
      break;
    case 2:
      bun.innerHTML = '天然ハーブ';
      gazou.src = 'pic2.jpg';
      break;
    case 3:
      bun.innerHTML = '梅の恵み';
      gazou.src = 'pic3.jpg';
      break;
    default:
      bun.innerHTML = ' ';
      gazou.src = ' ';
      break;
  }
}
</script>
</head>
<body>
<div>
```

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。

Javloc 条件分岐switch 上級編

リセット 元に戻す やり直し 解答例 プレビュー ← → ブロック読み込み ブロック保存 ページ保存

<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

- ① セレクトボックスが選択され、クリックボタンが押されると、関数が実行される。
 - ② 変数 `atai` に①のセレクトボックス「hada」を数値化した値を代入する。
 - ③ id “output” を取得し、変数 `bun` に代入する
 - ④ id “icon” を取得し、変数 `gazou` に代入する。
- ④ switch 文を使用し場合分けを行い、`bun・gazou` に代入する。
その結果が段落・画像タグに反映される。

【HTML 部分・作成ページ例】

①

```
<form>
  <select name="hada">
    <option value="0" selected>
      表示文字 あなたの肌質は?
    </option>
    <option value="1">
      表示文字 普通肌
    </option>
    <option value="2">
      表示文字 敏感肌
    </option>
    <option value="3">
      表示文字 乾燥肌
    </option>
  </select>
  <input type="button" value="クリック" onclick="bt ()">
</form>
```

```
<p id="output">
</p>
<img src="" id="icon">
```



【JavaScript 関数 bt が実行】

```
① var atai = Number(document.forms[0].hada.value);
② var bun = document.getElementById('output');
③ var gazou = document.getElementById('icon');
```

④

```
switch( atai ){
case 1:
  bun.innerHTML = ' 緑茶成分 ';
  gazou.src = ' pic1.jpg ';
  break;
case 2:
  bun.innerHTML = ' 天然ハーブ ';
  gazou.src = ' pic2.jpg ';
  break;
case 3:
  bun.innerHTML = ' 梅の恵み ';
  gazou.src = ' pic3.jpg ';
  break;
default:
  bun.innerHTML = ' ';
  gazou.src = ' ';
  break;
}
```

＜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。