

JavaScriptで学ぶ プログラミング入門 丸1日コース 問題集

- コメント（「// ...」や「/* ... */」の部分）は入力する必要はありません。説明です。

1 JavaScriptプログラムの書き方と関数の呼び出し



1.1 ダイアログボックスを使って「こんにちは」と表示せよ

- この問題は既に解答例がファイルfunction.htmlに入っているのをブラウザで表示するだけでよい。
- studentフォルダの中にある、ファイルfunction.htmlをブラウザで読み込む（ブラウザのウィンドウを開いておいて、そこにドラッグ&ドロップ）
- Windowsでは、**圧縮ファイルのままだと表示されない**ので、ダウンロードしたファイルを右クリックして**【すべて展開】**をしておくこと！

1.2 上の問題の「こんにちは」の代わりに別の言葉を指定して表示してみよ

- エクスプローラあるいはファイナダからfunction.htmlのアイコンをエディタのアイコンにドラッグしfunction.htmlをエディタで開くとソースプログラムが見える（「ソース」「ソースコード」「コード」「スクリプト」などとも呼ばれる）。次のようになっている。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>初めてのJavaScriptプログラム</title>
</head>
<body>
  <script type="text/javascript">
    "use strict"; // これがあると間違いが簡単に見つかる。説明は後ほどします
    alert("こんにちは"); // これがダイアログボックスを表示する「関数」
  </script>
</body>
</html>
```

1. 「こんにちは」の部分^{ひきすう}を他の文字列（たとえば「おはよう」）に変え、ファイルを保存
 2. ブラウザで再読み込み（再読み込みはをクリックするのが楽。WindowsではF5、Macでは⌘+Rのショートカットも便利）
- alertの引数（パラメータ）の文字列は'...'あるいは`...`でもよい。「`」はバッククオートあるいはバックティックなどと呼ばれる（入力するにはShfit+@。英語配列のキーボードでは左上にある）。`...`は、タグを出力するときなど文字列中に「"」や「'」を入れたいときや、変数の値などを文字列中に入れたいときにとても便利。
 - うまいいかなかったら、エディタで確認して修正し、保存してからブラウザで再読み込みをする（再読み込みはをクリックするのが楽。WindowsではF5、Macでは⌘+Rのショートカットも便利）

1.3 document.writeを使ってブラウザウィンドウのドキュメント領域内に「こんにちは」と表示せよ

```
document.write("こんにちは"); // ●解答例●
```

1. 「alert」を「document.write」に変える
2. ファイルを保存する
3. ブラウザから再読み込み

1.4 document.writeでタグを出力して、picturesフォルダにある画像を表示せよ（picturesフォルダに画像があることを確認）

```
document.write(``); // ●解答例●
```

- 最初と最後の「`」はバッククオート（「Shfit+@」で入力。英語配列では左上の方のキー）
- 「document.write('');」あるいは「document.write("");」と書いても結果は同じだが、出力したい文字列中に「"」や「'」がある場合は、`...`を使うのが便利。`...`を使うと中に変数を使ったりもできる。
- HTMLやCSSの指定ではバッククオートは**使えない**ので注意。たとえば「src=」の後は"..."あるいは'...'でなければならない。`...`は使えない。

2 分岐

2.1 大吉か大凶が同じ割合で出るおみくじのプログラムを作れ

- この問題は既に解答例がファイルif.htmlに入っているのをブラウザで表示するだけでよい。
- ブラウザで表示したら、if.htmlのアイコンをエディタのアイコンに重ねて開いて、内容を確認する。

2.2 大吉、小吉、末吉、凶が同じ割合で出るおみくじのプログラムを作れ (if.htmlを変更。見えそうな部分をコピーして変更する)

3 繰り返し (ループ)

3.1 picturesフォルダにある画像をpicture000.jpgから順番に8枚連続して表示するループを書け【写経】

```
"use strict";
for (let i=0; i<8; i++) {
  let ファイル名 = `pictures/picture00${i}.jpg`; // 下のコメント参照
  let 画像タグ = `
```

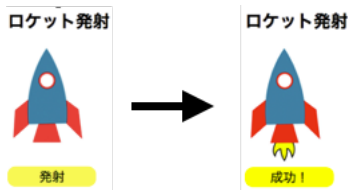
3.2 上のプログラムに、たとえば次のような変更を加えてみよ

- CSSを加えて画像の大きさを変更せよ (「forループ – スタイルの一斉変更」のスライド参照)
- 表示する枚数を変更せよ (4枚、10枚、16枚、...。16枚は少し難しい——ヒント: if文を使う)

4 イベントの処理

4.1 ロケットの画像の下に「発射」ボタンを置き、ボタンを押したらロケットの画像を変えて、発射したように見せよ (rocket.htmlの●●を書き換える)

- 画像はpicturesフォルダの下のrocket1.pngとrocket2.png
- 画像の領域とボタンの領域を設け、2つの領域に名前 (id) を付けて、innerHTMLやsrcを変更



```
...
<style>
  #pict { width: 4cm; } /* ロケットの画像の大きさを指定 */
  #button {
    font-size: 18pt; background-color: yellow; width: 4.4cm;
    text-align: center; border-radius: 15px; /* 角を丸くする */
    cursor: pointer; /* カーソルを手の形に変える */
  }
</style>
</head>
<body>
  <h1>ロケット発射</h1>
  
  <div id="button">
    発射
  </div>
<script type="text/javascript">
  "use strict";
  document.getElementById("button").addEventListener("click", function() {
    document.getElementById("●●").src = "pictures/rocket2.png"; // 画像を変える
    document.getElementById("●●").innerHTML = "成功!";
  })
</script>
...
```

document.getElementById利用時の留意点

document.getElementByIdを使う時は、既にそのIDを含む領域がブラウザ内部で構築されていないなければならない。たとえば上の例では、JSのコードが<div id="pict">...</div>や<div id="button">...</div>の下に置かれていなければならない。