

スクリプト言語

実習課題

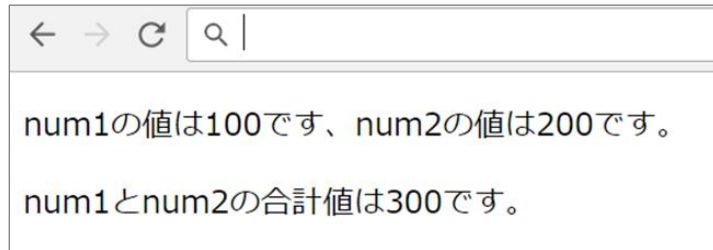
- 1 -

～ JavaScript ～

JavaScript 課題 1 - 変数宣言 (jsKadai01.html)

変数 `num1`、`num2` を宣言します。`num1` に 100、`num2` に 200 を代入し、実行例のように `num1` と `num2` の値を Web ブラウザに表示しましょう。文章は `<p>` タグ（パラグラフタグ）を用いて表示します。

さらに変数 `sum` を宣言し、`num1` と `num2` の合計を代入します。そして `<p>` タグで `sum` の値を表示しましょう。



JavaScript 課題 2 - 変数を使った計算 (jsKadai02.html)

以下を計算して表示しましょう。1 年は 365 日、1 日 24 時間とします。

- 1 年の時間
- 1 年の分
- 1 年の秒

計算する際、変数 `x` に 365 を代入して計算してください。

まず以下のように表示します。

1年を365日とした場合に

1年は8760時間です。

1年は525600分です。

1年は31536000秒です。

上記の表示ができれば、リストタグ `` を用いて以下のように表示してください。

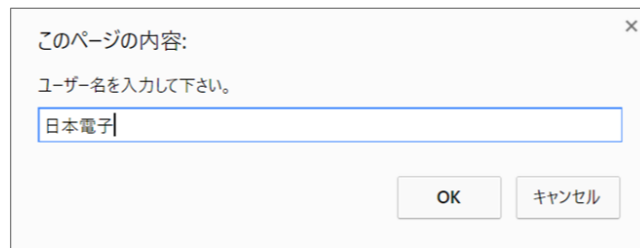
1年を365日とした場合に

- 1年は8760時間です。
- 1年は525600分です。
- 1年は31536000秒です。

`` タグの使い方が不安な方は、インターネット等で調べてみましょう。

JavaScript 課題3 – 値の入力 (jsKadai03.html)

`window.prompt` メソッドを用いてユーザー名を入力し、「〇〇さんがログインしました」(〇〇には入力したユーザー名)と表示しましょう。

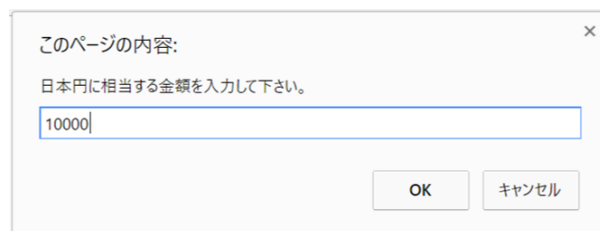


日本電子さんがログインしました

JavaScript 課題4 - 値の入力と計算 (jsKadai04.html)

`window.prompt` メソッドを用いて日本円を入力し、その金額をドルに変換して表示しましょう。

なお、1ドルは110円とし、小数点以下を切り捨てる `Math.floor()` メソッドを用いて整数値で表示してください。



10000円は約90ドルです

JavaScript 課題5 - 配列 (jsKadai05.html)

配列を使用して、以下のように関東の県一覧を表示しましょう。

配列の格納方法はリテラル形式、コンストラクタ形式のどちらを使用しても構いません。

```
kanto[0]:東京  
kanto[1]:埼玉  
kanto[2]:千葉  
kanto[3]:神奈川  
kanto[4]:茨城  
kanto[5]:栃木  
kanto[6]:群馬
```

JavaScript 課題6 - 配列と for 文 (jsKadai06.html)

課題5と同じ動作を for 文を用いて表示しましょう。

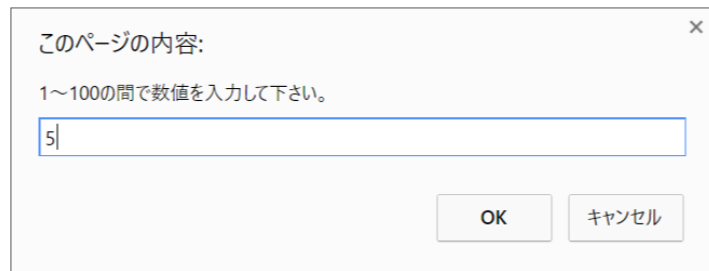
その際、配列の長さを取得し、最後に「kanto[]の長さは〇〇です」と表示してください。

```
kanto[0]:東京  
kanto[1]:埼玉  
kanto[2]:千葉  
kanto[3]:神奈川  
kanto[4]:茨城  
kanto[5]:栃木  
kanto[6]:群馬  
kanto[]の長さは7です
```

JavaScript 課題 7 - for 文と if 文 (jsKadai07.html)

ウィンドウから値を入力し、以下の動作を行いましょう。

入力値	動作
1 から 100 の値	画面上に 1 から入力された値まで表示する。 (例：5 が入力された場合は、1、2、3、4、5 を表示)
100 より大きい値	「100 より大きい値が入力されました」と表示
上記以外 (0、マイナス値、文字列など...)	「0、または不正な値が入力されました」と表示



正しい入力の場合

1
2
3
4
5

不正な入力の場合

100より大きい値が入力されました

不正な値が入力されました

なお、画面表示にあたってはすべて HTML タグを適正に含むこと。

JavaScript 課題 8 - 値の変換 (jsKadai08.html)

ウィンドウから値を入力し、以下の動作を行きましょう。

入力値	動作
1 から 255 までの値	入力された値を 2 進数に変換して表示する。 (例：192 が入力された場合は、2 進数「11000000」を表示)
255 より大きい値	「255 より大きい値が入力されました」と表示する
上記以外	「ゼロ、または不正な値が入力されました」と表示する

このページの内容

10進数→2進数変換：0から255までの値を入力して下さい。

OK
キャンセル



10 進数 192 を 2 進数変換した結果は「11000000」です

画面表示にあたってはすべて HTML タグを適正に含むこと。

➤ ヒント

2 進数の変換には `parseInt()` メソッドを使用するとよい。

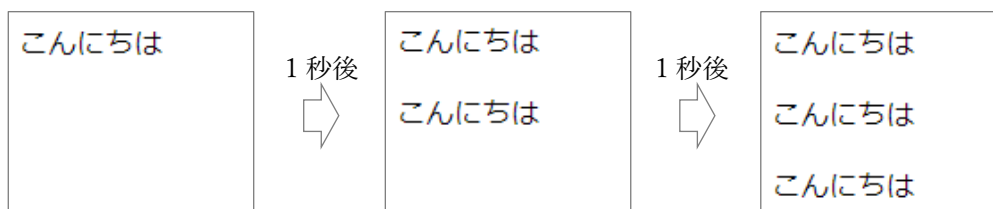
```
parseInt(変換したい値, 基数).toString(2)
```

例：10 進数の 192 を 2 進数に変換する

```
parseInt(192, 10).toString(2);    ⇒ 11000000
```

JavaScript 課題 9 - setInterval による繰り返し (jsKadai09.html)

1 秒ごとに「こんにちは」を表示しましょう。「こんにちは」がだんだん増えていきます。



➤ ヒント

`setInterval` を使用すると、関数を一定間隔ごとに実行することができます。

```
setInterval(実行したい関数名, 実行間隔ミリ秒)
```

関数の中で、`document.write` で「こんにちは」を表示すると良いです。

JavaScript 課題 10-1 - ボタン操作 (jsKadai10-1.html)

jsSample16.html を改造し、「犬に変更」と「猫に戻す」の2つのボタンを作成しましょう。
「猫に戻す」ボタンが押されると「我が輩は猫である」を表示します。

JavaScriptにおけるDOMの扱い方

DOMによって、「id1」のidがついたpタグの中身を変更します。

我が輩は猫である。

JavaScript 課題 10-2 - ボタン操作 (jsKadai10-2.html)

実習課題 10-1 をさらに改造し、「犬に変更」と「猫に戻す」の2つのボタンの他、「赤色に変更」、「青色に変更」、「黒色に変更」の合計5つのボタンを作成しましょう。

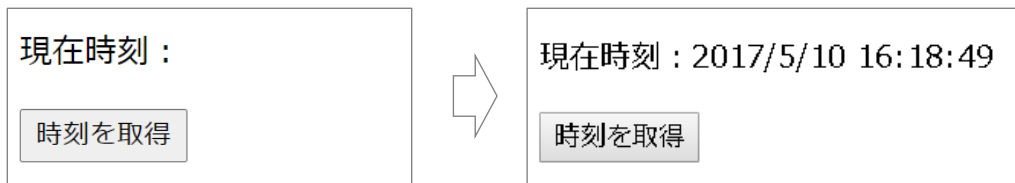
JavaScriptにおけるDOMの扱い方

DOMによって、「id1」のidがついたpタグの中身を変更します。

我が輩は猫である。

JavaScript 課題 11 - 時刻表示 (jsKadai11.html)

初めに「現在時刻：」と「時刻を取得」のボタンだけ表示します。
「時刻を取得」ボタンを押すと、実行例のように現在の時刻を表示しましょう。

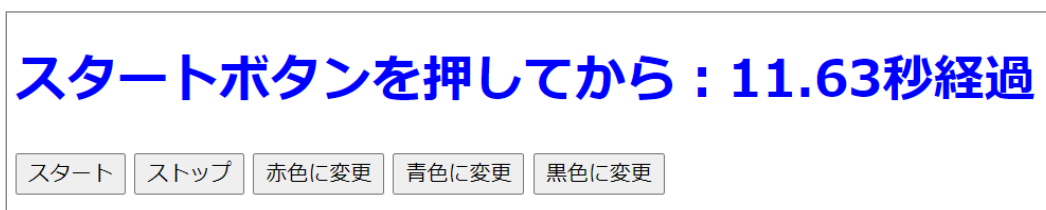


なお、上記のように時刻を表示するには、`Date` オブジェクトの `toLocaleString()` メソッドを使用するとよいです。

JavaScript 課題 12-1 - 色の変更 (jsKadai12-1.html)

`jsSample17-2.html` をさらに改造し、課題 10-2 と同じく赤色、青色、黒色に変化するボタンを追加しましょう。

色を変化させるボタンを押しても、タイマーには影響しないこと。



JavaScript 課題 12-2 – ランダム色 (jsKadai12-2.html) StepUP

jsKadai12-1.html をさらに改造し、押すと色がランダムに変わるボタンを追加しましょう。

スタートボタンを押してから : 11.06秒経過

スタート ストップ 赤色に変更 青色に変更 色をランダムに変更 黒色に変更

色は #000000～#FFFFFF の間で表すことができます。

また、`Math.random()` を実行すると、0.0～1.0 未満のランダムな値（乱数）を取得することができます（0.618...など）。

➤ ヒントその1

以下により、16 進数の 0 から FFFFFFFF までの範囲でランダムな数値（実数）を取得することができる。

```
Math.random() * 0xFFFFFFFF
```

この値には小数点以下も含まれるため、`Math.floor` で小数点以下を切り捨てる。

```
Math.floor(Math.random() * 0xFFFFFFFF)
```

ただし、このままでは 10 進数のため、16 進数に変換する必要がある（前回の 2 進数の課題を応用し、値を 16 進数に変換する）

➤ ヒントその2 :

取得した乱数値は数値のため、HTML のカラー値指定に必要な「#」の文字は付いていない。また、桁が足りない場合は、0 の数が足りない（例：0 だった場合は、「000000」ではなく、単に「0」となる）。そのため、足りない「#」と 0 を補う必要がある。（桁数は `length` プロパティでチェックできる）

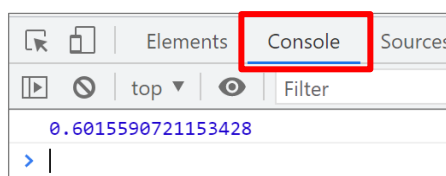
この課題は難易度が高いため、変数に現在どのような値が入っているかを確認しながら進めると良い。変数の値は `console.log()` で確認できる。

```
console.log(変数);
```

例：

```
let i = Math.random();
console.log(i);           // 変数の値をデバッグ出力
```

出力結果は、デベロッパーツール（検証）⇒ Console から確認できる。



JavaScript 課題 12-3 (jsKadai12-3.html) Ultimate StepUp

jsKadai12-2.html をさらに改造し、「色をランダムに変更」ボタンをクリックすると、1 秒ごとに色が変わるようにしましょう。

スタートボタンを押してから : 85.70秒経過

スタート ストップ 赤色に変更 青色に変更 色をランダムに変更 色の変更をストップ 黒色に変更

なお、一度クリックしてしまうと、他の色のボタンを押しても色が変わり続けてしまうため、色を止めるボタンも追加して下さい。

JavaScript 課題 13-1 – 回転ボタン (jsKadai13-1.html)

jsSample18.html を改造し、回転を開始するボタン、回転をストップするボタンを追加しましょう。

CSS3による回転アニメーション

回転 ストップ

JavaScript 課題 13-2 – 回転ボタン (jsKadai13-2.html) StepUp

jsKadai13-1.html をさらに改造し、X 軸、Y 軸、Z 軸にそれぞれ回転を開始するボタンを追加しましょう。もちろん、各軸の回転を止めるストップボタンについても追加します。

CSS3による回転アニメーション

X軸に回転 X軸回転をストップ Y軸に回転 Y軸回転をストップ Z軸に回転 Z軸回転をストップ

なお、2 軸以上同時に回す場合は rotate3d メソッドを使用するが、今回はそこまで行わなくてよい。(2 軸以上を同時に動かすことはしなくてよい)

JavaScript 課題 14 - カウントダウン (jsKadai14.html)

ビルトインオブジェクトである **Date** オブジェクトを使用し、「2025 年大阪万博まであと〇〇日××時間△△分□□秒」と表示しましょう。

カウントダウンは以前の課題でも使用した **setInterval** メソッドを使用し、一秒ごとに実行します。

大阪万博の開催日付は 2025 年 5 月 3 日 10 時とします。

➤ ヒントその 1 :

2025 年大阪万博の日付のセットは、以下を参考に **Date** オブジェクトの引数に値を渡す。

```
new Date(年、月、日、時、分、秒、ミリ秒)
```

ただし、月だけは 0 から数えられているため、-1 する必要がある (1 月なら 0 を指定する)。今回、分以下は省略しても構わない。

➤ ヒントその 2 :

ストップウォッチ JsSample17.html を参考にして、差分の時間を求める。その際、差分はミリ秒まで出ますが、使用しないので 1000 で割る。その値に対してそれぞれ日、時間、分、秒を求め、最後に **Math.floor** 関数を使用して小数点以下を丸める。

JavaScript 課題 15 - ブラウザオブジェクト (jsKadai15.html)

jsSample19.html を参考にし、以下のようにブラウザを判別して画面に表示しましょう。

ブラウザ	表示
Firefox で閲覧した場合	「Firefox で閲覧しています」
Chrome で閲覧した場合	「Chrome で閲覧しています」
その他のブラウザ	「Firefox、または Chrome を使用してください。」

ブラウザの判別には、ナビゲータオブジェクトである **navigator** の特定のプロパティ値を使用します。

➤ ヒント :

ある文章の中に指定した文字列「〇〇」が入っているかどうかを判別するには、**str.indexOf()** メソッドを使用する。

例：変数 **str** の中に 'Blue' という文字列が入っているか確認する

```
let str = 'Red Green Blue';
str.indexOf('Blue');    ⇒ 10   (見つかった最初の文字位置を返す)
str.indexOf('blue');    ⇒ -1   (見つからない。大文字小文字は区別される)
```

文字列が入っていた場合は「指定された値が最初に現れたインデックス (最初の文字を 0 番目とし文字が現れた位置に対する n 番目)」が、ない場合は「-1」が返る。

なお、**indexOf** メソッドでは大文字、小文字は別の文字として判別される。そのため、大文字、小文字どちらも検索できるようにするには、次のように **toLowerCase()** で文字列を小文字に変換すると良い。

```
str.toLowerCase().indexOf("blue");    // 検索文字列は小文字を指定する
```

JavaScript 課題 16-1 - ブラウザサイズの判別 (jsKadai16-1.html)

jsSample19.html を参考にし、以下のようにブラウザのサイズを判断して画面に表示しましょう。

ブラウザのサイズ	表示
横幅 960px 以下	スマートフォンまたはタブレットで閲覧している可能性があります。
それ以上	PC で閲覧しています。

ブラウザのサイズは、ナビゲータオブジェクトの特定のプロパティ値を使用して取得できる。(ディスプレイのサイズではないので注意)

JavaScript 課題 16-2 (jsKadai16-2.html) Super StepUp

jsKadai16-1.html を改造し、ブラウザの横幅に応じて、画像 jsKadai16img.jpg を画面いっぱいに表示しましょう。



なお、`setInterval` メソッド等を用いてブラウザの横幅を変えた際にリアルタイムに画像サイズが変更されるよう工夫してください。

➤ ヒント:

画像の横幅は `getElementById` メソッドで取得したオブジェクトの `.width` プロパティで、確認・変更を行うことができる。