

JavaScript

JavaScriptは主にWebページ内に記述され、Webブラウザにて実行されるプログラム言語です。Webページを動的にしたり、機能を追加したり出来ます。

JavaScriptにはjQueryを代表とするライブラリ(部品集)やフレームワーク(ひな形)が多く存在します。

今日のゴールは

- JavaScriptとは何か？
- jQueryって何なのか？
- どんなことが出来るのか？

など説明できるようになることです。

JavaScriptって何？

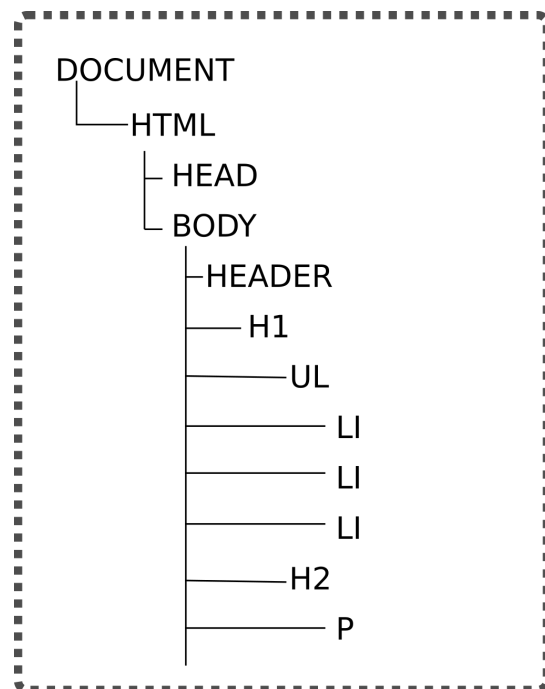
JavaScriptはWebブラウザの上で働くプログラミング言語です。

計算したり、ネットワークを使ったりすることも出来ますがWebページの要素(DOM)を操作することが出来ます。DOMを操作することで書かれている文字列を置き換えたり、画像を変更したり、要素の色を変えたりすることが出来ます。

DOMとは

Document Object Model の頭文字をとったものです。

HTML文書やXML文書を各種プログラムから利用するための仕組みです。



DOMは階層構造になっていて親子の関係になっている。
(親ノード、子ノード)

jQueryとは？

ウェブブラウザで動くJavaScriptコードをより容易に記述できるようにするために設計されたJavaScriptライブラリ(部品集)

JavaとJavaScript

誤解している人も多いのですが、JAVAとJavaScriptは名前こと似ていますが別のプログラミング言語です。JavaScriptを略してJavaと呼ぶと誤解を産みます。正しくJavaScriptと言うか、略すならJSとしましょう。

「ジャバスク」と略す人もいますが、伝わらない場合も多いので要注意です。

JavaScriptを体験しよう

授業用資料javascript.zipを

https://github.com/sozoshakobe/information_architecture

からダウンロードしてください。

file1.htmlとjquery_file1.html、
file2.htmlとjquery_file2.html

は同じ内容をJavaScriptとjQueryで書き分けています。

JavaScriptで出来ることはjQueryを使っても出来ます。
jQueryを使う方が効率的に書けるので行数が少なくなります。

JavaScriptを使うことで

画像を変更したり、色を変更したり動きをつけることが出来ます。
入力値を確認したり、機能的な動作を加えることも出来ます。

どちらもJavaScriptが仕事をしていますが、WebページのDOMを操作することで実現出来ているということを実感しましょう。

jQueryを書いてみよう

新規ファイルを作成

新規HTMLファイルを作成します。
タイトルは「js.html」としましょう。

基本のHTMLを記述し、bodyタグ内に
と記述します。

jQueryライブラリを読み込む

「jQuery CDN」を検索して、jQueryのCDNを探します。
見つけたら、CDNのURLをコピーして、</body>の上に貼り付けます。
script:src + tab でひな形が展開します。

```
<script
src="https://code.jquery.com/jquery-3.4.1.slim.min.js"></script>
</body>
```

JavaScriptを記述する

jQuery CDNの下にSCRIPTタグを作ります。
script + tab でひな形が展開します。

```
<script
src="https://code.jquery.com/jquery-3.4.1.slim.min.js"></script>
<script>
</script>
</body>
```

jQueryの基本構文を書く

```
<script>
    $(document).ready(function() {
        //この中に処理を書きます。
    });
</script>
```

JavaScriptでは

「//」で始まる行

「/*」から「*/」の間はコメントとして扱われます。

処理を書いてみる

下記のとおり書いてみてください。

```
$(document).ready(function() {  
    $("#jquery").attr('src','images/jquery.png');  
});
```

ブラウザで確認してみてください。

HTML内のIMGタグにはSRC属性が設定されていなかったはずですが、画像が表示されています。jQuery(JavaScript)がSRC属性にファイルの場所を追加した事で画像タグが完成し指定した画像が表示されました。

JavaScriptを外部ファイルに書くには

<script>タグ内以外にも外部ファイルに記述することも出来ます。

拡張子は「.js」とします。

「script.js」という名称がよく使われます。

外部ファイルの指定は<script>タグに「src属性」を加えます。

```
<script src="script.js"></script>
```

JavaScriptを少しでも詳しく

計算

JavaScriptでは下記の3つの道具を使って計算を行います。

- 値
- 算術演算子
- 変数

値とは？

プログラムの中で扱う文字や数字を「値」といいます。

値にはいくつか種類がありますが、代表的な3つを理解しましょう。

数値

| 名称 | 例 |
|--------|------|
| 整数 | 100 |
| マイナスの値 | -8 |
| 少数 | 0.99 |

数字はそのまま記述します。クォテーション(“とか”)をつけてしまうと数値として機能しません。“100”と書いてしまうと文字として「100」という意味になります。

文字列

文字からなるデータのこと。「あ」や「キ」など一文字のデータから、「おはよう」などの単語、1000文字の文章も全て文字列として扱います。

文字列データはクォテーションで囲み、文字列であることをJavaScriptに教えます。

| 備考 | 例 |
|------------------|---------------------------------|
| 文字列はクォテーションで囲みます | ‘文字列’ |
| エスケープ | ‘クォテーション内にクォテーションを書く場合は\'と書きます’ |
| 文字列としての数字 | ’10000’ |
| 電話番号など0で始まる数字 | ‘09012345678’ |

真偽値

正しい(真、true)、正しくない(偽、false)かを表す値

算術演算子

| 計算 | 演算子 | 記述例 | 計算結果 |
|-----|-----|--------|------|
| 足し算 | + | 4 + 5 | 9 |
| 引き算 | — | 6 - 3 | 3 |
| 掛け算 | * | 4 * 4 | 16 |
| 割り算 | / | 12 / 3 | 4 |
| 余り算 | % | 24 % 5 | 4 |

変数

変数は値を入れておく入れ物です。数学で出てきた「x」や「y」と同じです。
変数には固有の名前をつけることが出来ます。

変数宣言

変数は使用する前に宣言をします。以下の例はschoolという名前の変数を宣言して、「創造社デザイン専門学校」という値(文字列)を変数に**代入**したという意味になります。

```
var school = “創造社デザイン専門学校”;
```

varはvariableの略で変数の宣言を意味します。

プログラムで登場する “=”(イコール) は左辺と右辺が同じという意味ではなく、左辺に右辺の値を設定することをいいます。(これを**代入**といいます。)

| 変数宣言 | 意味 | 範囲 |
|-------|-------------|-------------|
| var | 再宣言、再代入可 | { }の外でも使える |
| const | 再代入不可、再宣言不可 | { }の中だけで使える |
| let | 再代入可、再宣言不可 | { }の中だけで使える |

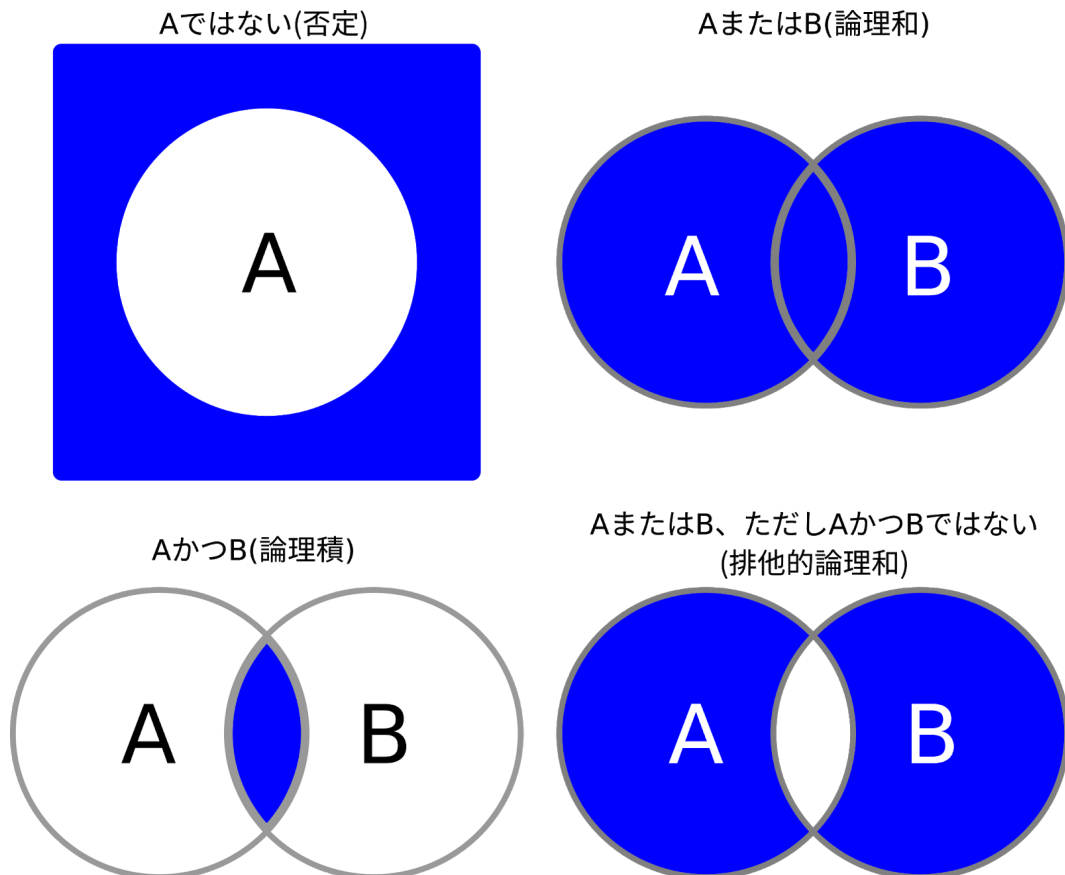
「かつ」と「または」

「学校に来るのは月曜日、火曜日です。」
この文章には二通りの解釈が出来ます。

- 月曜日と火曜日に学校に来る
- 月曜日か火曜日に学校に来る

たった一文字しか違いませんが、内容は大きく異なります。
こういった曖昧さをなくすことを論理といいます。

プログラミングでは論理演算子をつかって表現します。



論理演算子

| 表現 | 名称 | 演算子 |
|-------|-----|-----|
| Aではない | 否定 | ! |
| AまたはB | 論理和 | |
| AかつB | 論理積 | && |

否定(～ではない)

| 入力内容 | 意味 | 結果 |
|---------|-------|-------|
| ! true | 真ではない | false |
| ! false | 偽ではない | true |

論理和(AまたはB)

| 入力内容 | 意味 | 結果 |
|----------------|-----------|-------|
| true true | Aは真またはBは真 | true |
| true false | Aは真またはBは偽 | true |
| false true | Aは偽またはBは真 | true |
| false false | Aは偽またはBは偽 | false |

論理積(AかつB)

| 入力内容 | 意味 | 結果 |
|----------------|----------|-------|
| true && true | Aが真かつBが真 | true |
| true && false | Aは真かつBは偽 | false |
| false && true | Aは偽かつBは真 | false |
| false && false | Aが偽かつBが偽 | false |

比較演算子

| 入力内容 | 意味 | 結果 |
|--------------|---------------|-------|
| 1 < 2 | 右辺は左辺より大きい | true |
| 1 <= 2 | 右辺は左辺以上 | true |
| 1 > 2 | 右辺は左辺より小さい | false |
| 1 >= 2 | 右辺は左辺以下 | false |
| 1 === 2 | 右辺と左辺は値も型も等しい | false |
| 100 == "100" | 右辺と左辺は値が等しい | true |
| 1 != 2 | 右辺と左辺は値が異なる | true |

IF文(条件分岐)

プログラムである条件が真の時はAの処理、偽の時にはBの処理というような場合分けを**条件分岐**といい、**IF文**で表現します。

以下の文は年齢が格納された変数ageにより実行結果を変える例です。

```
const age = 18; // 年齢に整数18を設定
if(age < 20){
    document.write('未成年です');
}else{
    document.write('成人です');
}
```

偽の場合が不要な時はelse以降を省略できます。

```
if(age < 20)
    document.write('未成年です');
```

*処理が1行だけなら、{} も省略可能です。

条件が複数ある場合

```
const age = 13;
if(age <= 12){
    document.write('子供料金です');
}else if(age >= 60){
    document.write('シニア料金です');
}else{
    document.write('大人料金です');
}
```

FOR文(繰り返し、ループ)

プログラムで繰り返し処理を行うことをループ(LOOP)と言います。
LOOPの方法には色々ありますが代表的なFOR文を紹介します。

```
for( let i = 0; i < 100 ; i++ ){
    document.write('ナンバー'+ i);
}
```

for文の()の中は三つのブロックに別れています。

最初の `let i = 0` はカウンター変数です。ループする回数を管理しています。

二つ目の `i < 100` はループする条件です。

カウンター変数が100より小さい時繰り返しが発生します。(100を超えたら終了)

三つ目の `i++` はカウンター変数の調整です。この例では一回繰り返すごとに*i*を1増やしています。これをインクリメントといいます。

上の例を実行するとナンバー0 ~ ナンバー99 まで繰り返します。

コレクション

値や要素の集まりをコレクションといいます。JavaScriptで代表的なコレクションは配列です。

配列(Array)

配列とは複数の値を連続的に並べたもの。

```
const majors =
['グラフィック','イラスト','ウェブ','ディスプレイ','インテリア','プロダクト'];
```

例ではmajorsという変数に

- グラフィック
- イラスト
- ウェブ
- ディスプレイ
- インテリア
- プロダクト

の6個の要素を格納しました。

取り出す場合は `majors[添字]` で表現します。添字は0から始まります。

`majors[0]` は 「グラフィック」が入っています。

| majors | | | | | |
|--------|------|-----|--------|-------|-------|
| [0] | [1] | [2] | [3] | [4] | [5] |
| グラフィック | イラスト | ウェブ | ディスプレイ | インテリア | プロダクト |

配列を順番に取り出す

```
for(let i = 0 ; i < majors.length ; i++){
  document.write(majors[i]);
```

```
}
```

JSON

JavaScript Object Notationの略、軽量なデータ記述言語の1つである。

JavaScriptの表記法を基本としていますが、様々なソフトウェアやプログラミング言語間におけるデータの受け渡しに使えるよう設計されています。

記述例：

```
const myData = { name : '太田' , age : 30 , school : '創造社デザイン専門学校' }
```

データにアクセスする場合は

`myData.name` とすることで '太田' が出力されます。

JSON形式はjQueryのプラグインを使用する場合に必要になります。