

PBL 第3回 プログラミング練習(JAVASCRIPT)

山本・辻・粕谷研究室

TA：中井・山田・井土・國廣・辻・富久

1

JAVASCRIPTとは (※一部説明を省いています)

- プログラミング言語
- Webブラウザなどで使われる
- オブジェクト指向

```
// スラッシュ2つで一行コメント  
console.log("Hello World!!"); // 文の最後にセミコロン  
/* こんな風にも書いてもコメント */
```

JAVASCRIPTとは

この後の説明する内容の詳しい情報は以下のリンクをクリック！

<https://developer.mozilla.org/ja/docs/Web/JavaScript>

一般的なJAVASCRIPTの使い方

- **html**に直接書く（**script**タグで囲んだところに書く）
- ぽげぽげ**.js** というファイルで保存して**html**からロードする
- ブラウザのコンソールなどで試す ← 今回はこのやり方でやります

（**Node.js**などの例外があります。）

JAVASCRIPTの実行方法

- 前回作った”index.html”を**firefox**で開く
- **F12**でウェブコンソールを開く
- **Shift+F4**でスクラッチパッドを開く

- スクラッチパッド**JavaScript**を書く
- 「実行」もしくは**Ctrl+R**で**JavaScript**を実行
- ウェブコンソールに結果が表示される



やってみよう

コンソールに”Hello World!!”と出力せよ.

変数(VAR, CONST, 代入)

	数学	JavaScript
変数の宣言	x を変数とする. x の値は 10 とする.	<code>var x;</code> <code>x = 10;</code>
	y を値が 20 の変数とする.	<code>var y = 20;</code>
定数の宣言	k を定数 5 とする.	<code>const x = 5;</code>
変数に代入する	x に 30 を代入する.	<code>x = 30;</code>
	x に y + 1 を代入する.	<code>x = y + 1;</code>

型の種類

型	例
文字列	"Apple", 'Banana'
数値	1, 3.14, -196, NaN, +Infinity
真偽値 (ブール値)	true, false
null	null
undefined	undefined
オブジェクト	Array, Date など

プリミティブ型
(基本型)

主な演算子

	記号
加減乗除余	+, -, *, /, %
代入	=
関係	>, >=, <, <=
等号	==, !=
論理否定	!
増分減分	i++, i--

やってみよう

変数 x と変数 y を作り，それぞれ10,20を代入.

その後変数 z に変数 x と変数 y の値を加算した値を代入.

変数 z の値をコンソールに出力させよ.

やってみよう

変数 x と変数 y を作り，それぞれ”Apple”と”Banana”を代入．

変数 x と変数 y が等しいかをコンソールに出力させよ．

変数 y に”Apple”を再代入し，

再び変数 x と変数 y が等しいかをコンソールに出力させよ．

配列とは

- 複数の要素を順序付けて格納できるデータ構造
- **Haskell**のリストのようなもの

```
var arr1 = [4, 2, 3, 4];  
console.log(arr1[0]); // => 4
```

```
var arr2 = new Array(5); // 長さ5の配列を作成  
console.log(arr2.length); // => 5
```

配列の操作

```
console.log(arr1[0]); // => 4;
```

```
arr1[0] = 0;
```

```
console.log(arr1[0]); // => 0
```

やってみよう

配列[1, 5, 3, 4]を変数`xs`に代入し

変数`xs`をコンソールに出力させよ.

配列の1番目の要素に2を代入し,

再び変数`xs`をコンソールに出力させよ.

制御構造 - IF文

とある条件によって処理を分岐できる構文.

```
if (条件式A){  
    条件式Aがtrueのとき実行される文  
}  
else if (条件式B) {  
    条件式Aがfalseかつ条件式Bがtrueのとき実行される文  
}  
else {  
    条件式Aも条件式Bもfalseのとき実行される文  
}
```

省略可能

制御構造 - IF文 - 例

```
var x = 3;

if (x == 0) {
    console.log("x は0と等しい");
}
else if (x > 0) {
    console.log("x は0より大きい");
}
else {
    console.log("x は0より小さい");
}
```

やってみよう

自分の学籍番号を変数に代入し、
それが偶数か奇数かをif文で判定し
“偶数”か“奇数”かを出力せよ。

制御構造 - WHILE文

条件が満たされている間,処理を繰り返す構文.

```
while (条件式) {  
    条件式がtrueの間繰り返す  
}
```

制御構造 - WHILE文 - 例

```
var i = 0;

while(i < 10) {
    console.log("i = " + i);

    i = i + 1;
}
```

制御構造 - FOR文

指定した回数を繰り返す処理が書きやすくなる構文.

```
for (初期化式; 条件式; 更新式) {  
    条件式がtrueの間繰り返す  
}
```

制御構造 - FOR文 - 例

```
for (var i = 0; i < 10; i = i + 1) {  
    console.log("i = " + i);  
}
```



やってみよう

for文 もしくはwhile文を使って**10**の階乗の計算結果を出力せよ.

関数文

宣言

```
function 関数名 (仮引数1, 仮引数2, ...) {  
    実行内容  
}
```

呼び出し

```
関数名 (実引数1, 実引数2, ...);
```

関数文 - 例

宣言

```
function f (n) {  
  console.log("Input Number is: " + n);  
}
```

呼び出し

```
f(5); // => Input Number is: 5
```

制御構造 - RETURN文

関数の実行を終了し,呼び出し元に返す値を指定する.

`return` 返り値;

制御構造 - RETURN文 - 例

```
function add (x, y) {  
  return x + y;  
}  
  
var z = add(10, 20);  
  
console.log(z); // => 30
```

やってみよう

引数を**2**つ受け取り、一つ目の引数から二つ目の引数の値を減算し返す関数**sub**を作り、関数**sub**を使って**20-10**をせよ。

オブジェクト

オブジェクトはプロパティの集まりであり、プロパティは値か関数と考えても問題ない。オブジェクトに所属している関数をメソッドと呼ぶ。

```
<script>
```

変数

オブジェクト

```
var traffic_light = {
```

```
  blue : "go",
```

```
  yellow : "slow down",
```

```
  red : "stop"
```

プロパティ名 値

プロパティ

```
</script>
```

出典：
TECHACADEM

オブジェクトの例

さっきのオブジェクト

```
console.log(traffic_light.blue); // => go  
console.log(traffic_light.yellow); // => slow down
```

Mathオブジェクト（標準ビルドインオブジェクト）

```
console.log(Math.PI); // => 3.141592653589793  
console.log(Math.abs(-15)); // => 15
```

やってみよう - FIZZBUZZ

1から30までの数字を出力せよ。

ただし数字が3の倍数のときは数字の代わりに"Fizz"を、

5の倍数のときは数字の代わりに"Buzz"を、

3と5の倍数のときは数字の代わりに"FizzBuzz"と出力せよ。

課題(要提出)

PBL日報の提出

前回同様班のリーダーがまとめて提出してください。

ファイル名：**201x3110xx_PBL_XX.pdf**(xには自身の学籍番号をお願いします)

提出先：yl-apu-pbl19@yamamoto.ist.aichi-pu.ac.jp

課題(提出不要)

JavaScriptの復習をしておく.

Monacaについて予習しておく.

課題(提出不要)

2人の名前(アルファベットのみで構成される文字列)が与えられたとき、その2人の相性度を出力するプログラムを作れ。

相性度は2人の名前の各文字に割り当てられた文字番号の和を11で割った余りである。文字番号はAが1, Bが2, ...Zが26とする。

ヒント: 文字列"ALICE"に対するi番目の文字に対応する数字は「"ALICE".charCodeAt(i) - 64」で求められる。

例えば ALICE と BOB の相性度は $(1 + 12 + 9 + 3 + 5 + 2 + 15 + 2) \% 11 \rightarrow 49 \% 11 \rightarrow 5$ より5となる。なお、名前は全て大文字で与えられる。よって "ALICE" と "BOB" の相性度は5のようになる。