

PBL第4回 プログラミング練習

山本, 辻, 粕谷研究室

TA: 中井, 山田, 井土, 國廣, 辻, 富久

本年度のスケジュール

第1回 4/24 : ガイダンス & Monaca アカウント作成

第2回 5/08 : HTML, Java, CSS について

第3回 5/22 : プログラミング練習, **中間報告会**

第4回 6/05 : プログラミング練習

第5回 6/19 : サンプルプログラム作成

第6回 7/03 : アプリの計画, 発表スライド作成

第7回 7/10 : 見学, **デモ大会**

第8回 7/24 : PBL 成果発表会

目次

- ◆ switch文
- ◆ オブジェクト指向について
- ◆ htmlとの連携

switch文

switch文

if文のように分岐を行うことができる構文です.

値によって分岐するのはif文と同じだが
if文と違い条件が多くなっても見やすいです.

switch文

switchは次のように記述します.

caseで分岐したあと最後に必ずbreakをつけてください.

```
switch (x) {  
  ... case (15): console.log("fizzbuzz"); break;  
  ... case (3): console.log("fizz"); break;  
  ... case (5): console.log("buzz"); break;  
  ... default: console.log(x); break;  
}
```

switch文

こんな感じでたくさんの分岐が行えるのでif文が長くなりそうなときはswitch文を使おう.

```
switch (second % 10){  
    case 1: console.log("1secind"); break;  
    case 2: console.log("2secind"); break;  
    case 3: console.log("3secind"); break;  
    case 4: console.log("4secind"); break;  
    case 5: console.log("5secind"); break;  
    case 6: console.log("6secind"); break;  
    case 7: console.log("7secind"); break;  
    case 8: console.log("8secind"); break;  
    case 9: console.log("9secind"); break;  
    default: console.log(second); break;  
}
```

オブジェクト指向とは

オブジェクト指向とは

オブジェクト指向とは...

オブジェクト指向 (オブジェクトしこう、[英](#): *object-oriented*) は、[ソフトウェア工学](#)理論の一つであり、ソフトウェア設計とプログラム記述の際に用いられる考え方である。元々は特定の[プログラミング・パラダイム](#)を説明する為に考案された言葉だった。*object-oriented programming*のように修飾語として用いられるのが常であり、それ単体で扱われる事はない漠然とした設計構想でもある

Wikipediaより

オブジェクト指向とは

前回最後にみせたものです

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

こんな感じですよ使える。

```
<script>  
  変数  
  var traffic_light = {  オブジェクト  
    blue : "go",  
    yellow : "slow down",  
    red : "stop"  
  }  
</script>
```

プロパティ名 値

プロパティ

オブジェクト指向とは

また前々回のJavaScriptの説明のでも使用されています。

Dateという組み込みのオブジェクトを利用することでさまざまなことができます。

```
//ひらいたときの時刻を表示
var today = new Date();
var year = today.getFullYear();
var month = today.getMonth() + 1;
var date = today.getDate();
var hour = today.getHours();
var minute = today.getMinutes();
var second = today.getSeconds();
```

オブジェクト指向とは

自分で作成すること場合はこのような形になります。

クラスというオブジェクトの塊を作成しています。
それからオブジェクトを利用するために
newしてオブジェクトを作っています。

constructorはnewしたときに呼びだされます

```
class Dog {
  constructor(name){
    this.name = name;
  }
  your_Name(){
    console.log(this.name);
  }
  bark(){
    console.log("bowbow");
  }
}

var Nora_Inu = new Dog("inu");
console.log(Nora_Inu.name);
Nora_Inu.bark();
Nora_Inu.your_Name();
```

オブジェクト指向とは

オブジェクト指向のいいところ

「カプセル化」

こういった変数や関数を

ひとまとまりで使用することができる.

中にある変数を直接参照できなくする.

```
class Dog {  
    constructor(name){  
        this.name = name;  
    }  
    your_Name(){  
        console.log(this.name);  
    }  
    bark() {  
        console.log("bowbow");  
    }  
}  
  
var Nora_Inu = new Dog("inu");  
console.log(Nora_Inu.name);  
Nora_Inu.bark();  
Nora_Inu.your_Name();
```

オブジェクト指向とは

オブジェクト指向のすごいところ

「継承」

既存のクラスを継承すると関数などをそのまま利用できる。

constructorは継承元と同じものを利用できるので記述しなくてもよい。

クラスを継承したいけど新しいコンストラクタを利用したいときは次のようになる。

super()を利用すればよい。

```
class Pet_dog extends Dog{
  .. constructor(name, Nickname){
  ..   .. super(name);
  ..   .. this.Nickname = Nickname;
  .. }
  .. Nick_Name(){
  ..   .. console.log(this.Nickname);
  .. }
}

var Wanchan = new Pet_dog("Kabosu", "Doge");
Wanchan.your_Name();
Wanchan.Nick_Name();
```

オブジェクト指向とは

オブジェクト指向の面白いところ
「ポリモーフィズム」

元々のクラスの関数を
同じ名前で別の動作に変えることが
できる.

```
class Puppy extends Dog {  
    ··· your_Name(){  
        ··· console.log("Puppy_" + this.name);  
    }  
}  
  
var koinu = new Puppy("koinu");  
koinu.your_Name();
```

オブジェクト指向とは

JavaScriptには前回学んだfor文やif文だけではなく
様々な機能があります。

どんどん使っていきましょう。

htmlとの連携

htmlとの連携

htmlのidを使って色々遊んでみよう.

htmlはタグ付けによって表現されている.

そのタグを対象にJavaScriptを書いて遊ぶ.

htmlとの連携

とても簡単な例

呼び出されるとpushとアラートをだします。

htmlに記述

```
<input type="button" value="押す" onClick="b_push()" >
```

```
function b_push(){  
    alert("push");  
}
```

JavaScriptに記述

ここで対応している。

htmlとの連携

とても簡単な例2
ボタンを押すと箇条書きが増える

getElementByIdという関数でタグを持っ
てきている.

ボタンは先ほど同様

```
function b_inc(){  
    ol = document.getElementById("ol");  
    if(ol.childElementCount < 20){  
        li = document.createElement("li");  
        li.innerHTML = "追加の文";  
        ol.appendChild(li);  
    }  
}
```

```
<input type="button" value="増える" onClick="b_inc()" >
```

```
<ol id="ol">  
    <li>もともとある文</li>  
</ol>
```

htmlとの連携

とても簡単な例3
増やした箇条書きを減らす関数

ボタンは先ほど同様

```
function b_dec(){  
    ol = document.getElementById("ol");  
    if(ol.lastElementChild != null){  
        ol.lastElementChild.remove();  
    }  
}
```

```
<input type="button" value="減る" onClick="b_dec()" >
```

```
<ol id="ol">  
    <li>もともとある文</li>  
</ol>
```

htmlとの連携

実際に動かしてみよう！

配布されたファイルを自由に触ってください。

課題(提出不要)

配布されたfizzbuzz関数をswitch文に書き換えてみよう.

Inputタグのtype=textなどを用いて足し算やかけ算ができる電卓を作ってみよう.

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 : nakai@yamamoto.ist.aichi-pu.ac.jp

前回同様テンプレートを利用して書いてください。