2023 年 9 月 11 日（月）

『おもしろ電子工作実験室』

（前回の復習）

◇JavaScript のプログラムから HTML の要素を操作するためには**セレクター**を使用する。

セレクターとは

HTML：要素の id 属性 または class 属性 で指定した属性値

タグ名もセレクターとして使用する

```
<button class="btn btn-primary"  id="led-on">LED ON</button>

<div id="print"></div>
```

※id 属性は HTML 内でユニークな（唯一無二な）属性値を設定する

※class 属性は HTML 内で複数設定可能。グループとして使用する。

class 属性内に複数の属性値を設定することも可。

※id 属性や class 属性の属性値には命名規則がある。

（前回のテキストを参照してください）

JavaScript：HTML で設定したセレクターを参照するために使用する

（css も同じように使用する）

```
// id 属性で設定したセレクターを参照する
$("#led-on").click(function() {
}

// class 属性で設定したセレクターを参照する
$(".btn").css("color", "red"); // すべての .btn の色を赤色に変更

// <button>タグを参照する
$("button").text("ボタン");  // すべての<button>の text をボタンに変更
```

※id 属性で設定したセレクターは“#”を先頭に付加する

※class 属性で設定したセレクターは“.”を先頭に付加する

※“#”や“.”が先頭にないセレクターは タグ名 として認識する

◇JavaScript のコードは、<body>タグの一番したに<script>タグを使用して記述する

◇obniz の初期設定を必ず行う。

const obniz = new Obniz(“OBNIZ_ID_HERE”);

［注意］“OBNIZ_ID_HERE”の箇所には所有している obniz ID に変更する

◇obniz がオンラインになったときの処理を記述

obniz.onconnect = async function() { … }

◇LED の初期設定と定義

const led = obniz.wired(“LED”, { … });

◇obniz ボードの画面の初期表示

◇obniz ボードのスイッチの状態が変わったときの処理

obniz.switch.onchange = function(state) { … }

◇処理の情報や状態を画面に表示する

$("#print").text(表示する内容);

◇［Print on obniz］ボタンがクリックされたときの処理

$("#showtime").on("click", function() { … }

◇[LED ON]ボタンをクリックしたときの処理

$("#led-on").click(function() { … }

◇[LED OFF]ボタンをクリックしたときの処理

$("#led-off").click(function() { … }

◇HTML や JavaScript のプログラムは上から下へ解釈・処理が行われる。

※HTML でセレクターを設定した後で、そのセレクターを使用した JavaScript のプログラムを記述しないと動作しない。

●定義した変数が使用できる範囲について（スコープ）

変数とは、プログラム内で一時的にデータを保存する領域のこと。

（別の機会に説明します）

```
const obniz = new Obniz("OBNIZ_ID_HERE");

const led = obniz.wired("LED", { … } );

for (let i = 0; I < 10; i++) { … }
```

変数の定義方法は３つ。

・const　１回のみ値を設定することができる変数。定数。スコープが厳密。
・let　通常の変数の定義。新しい定義方法。スコープが厳密。
・var　通常の変数の定義。古い定義方法。スコープが緩い

const と let は 定義した{ … } 内でのみ使用できる。

```
let ans = 0;
for (let i = 0; i < 10; i++) {
    ans = ans + i;
}
console.log(ans);
console.log(i);  // 範囲外なのでエラーになる
```

i はこの範囲のみ有効

●演習問題（３）

ボタンスイッチモジュールを追加し、ボタンを押したときに LED を ON する

・「obniz　ボタン」で検索
・「Keyestudio_Button」のページを開く
　（https://docs.obniz.com/ja/sdk/parts/Keyestudio_Button/README.md）
・ボタンモジュールのピンアサイン（signal, vcc, gnd）を確認
　（s、 v,　g とプリントされている）

・ボタンモジュールを obniz ボードのどのポートに接続するか検討
（例えば、signal → 9、vcc → 10、 gnd → 11）
・ボタンモジュールの初期設定　（どの位置に記述する？）

```
const button = obniz.wired("Keyestudio_Button", {signal:9, vcc:10, gnd:11});
```

・ボタンが押された時、離された時の処理を記述（どの位置に記述する？）

```
button.onchange = function(pressed){

    // ここにボタンの状態によって、LED を ON または OFF にするコードを記述

};
```

※ボタンの状態は
押すと　➡　pressed が false
離すと　➡　pressed が true

・ボタンの状態を判定するには　if ステートメントを使用する
記述方法は以下の通り

```
if (pressed == true) {
    // pressed が true の場合の処理
}
else {
    // pressed が true でない場合の処理
}
```