

## 『おもしろ電子工作実験室』

### ●obniz について再度確認

- ・ obniz はインターネットに接続して使用する IoT 機器
  - ・ obniz は 2.4GHz 帯の Wi-Fi に接続する
  - ・ obniz は web サイトを構築する技術 (HTML、CSS、JavaScript) を利用する
  - ・ HTML は web ページ全体の情報の記述や部品等の配置を記述するマークアップ言語
  - ・ CSS は HTML 内に配置した部品に色などの装飾を行うスタイルシート言語
  - ・ JavaScript は HTML 内に配置した部品等に動きを与えるためのプログラミング言語
  - ・ obniz の動作処理は JavaScript 内に記述する
- 
- ・ HTML はタグと言われる文字列で記述を行う
  - ・ タグは開始タグと終了タグ内にタグの内容 (タグのコンテンツ) を記述する  
例: `<button>LED を点灯</button>`
  - ・ タグは、終了タグありのタグと、終了タグなしのタグがある
  - ・ HTML は `<html>` タグ内に記述する
  - ・ `<head>` タグ内に web ページの情報や先に読み込む必要がある CSS や JavaScript を記述する
  - ・ `<body>` タグ内に web ページに表示するコンテンツ (表示する内容) を記述する
- 
- ・ `<title>` タグは、サイトのタイトルで、web ブラウザーのヘッダー部分に表示される
  - ・ 必要に応じて CSS を `<head>` タグ内に記述する
  - ・ CSS は `<style>` タグ内に記述し、`<head>` タグの最下位に記述する
- 
- ・ 必要に応じて JavaScript を `<body>` の最下位に記述する
- 
- ・ `<div>` タグは、グループを設定する
  - ・ `<h1>` タグは、見出しを表示する。1~6 まで設定できる
  - ・ `<button>` タグは、ボタンを表示する。ボタンが押されたことを JavaScript に通知する。
  - ・ `<img>` タグは、画像を表示する
  - ・ `<a>` タグは、リンクを設定する
  - ・ `<p>` タグは、段落 (文章のかたまり) を設定する
  - ・ `<br/>` タグは改行を行う。改行を行うところに記述する。
  - ・ `<hr/>` はラインを引く。

- ・半角の空白文字や改行を何文字入力しても 1 文字しか表示されない
- ・半角スペースや < >などを特殊文字と呼び、記述方法がある。  
(記述方法はネットで検索、講座サポートサイト：<https://yoron.tech/kouza/> の参考サイトにも記述してある)

- ・コメントの活用
- ・コメントはプログラミング上のメモや処理内容を記述する動作に影響を与えない文章  
HTML：`<!-- ここにコメントを記述 -->`  
CSS：`/* ここにコメントを記述 */`

- ・タグには属性を記述する場合がある

※JavaScript は id 属性と class 属性要素の目印として使用するので記述方法を覚えておいてください。

※他サイトなどから HTML 等のソースコードをコピー&ペーストを行う場合は、コピー先を十分確認してからペースト（貼り付け）行ってください

※HTML と CSS は誤った記述を行ってもエラーにならない。  
誤った記述を行うと **誤った解釈** or **無視** されます。

※HTML はパソコン内のテキストエディター（メモ帳など）で記述し、確認することが可能。

※動作を確認する場合は、パソコン内で簡単な HTML を作成して動作確認すると良い  
その場合、拡張子を、「.html」に変更すること

※HTML のテンプレートファイルは

「講座サポートサイト：<https://yoron.tech/kouza/>」に掲載してあります

## ●演習問題（1）

下記を表示する HTML を作成してください。

- ・ 開発環境は、obniz の開発環境 または パソコン内のテキストエディター を使用する
- ・ obniz の開発環境を使用する場合、  
現在まで作成したソースコードは削除 または ダウンロードを行ってください。
- ・ 講座サポートページに HTML のテンプレートファイルがあるので使用してください。  
[使用したソースコード] 見出し内の [2] HTML テンプレート
- ・ 講座サポートページに解答例を掲載してあります。

[さらに問題]

- ・ bootstrap の web サイトを参照してボタンの色や形を変えてみる  
(※使用している Bootstrap のバージョンは v4.5 です。検索したホームページは v5.3 が表示されているので、右上のバージョンを v4.5 に切り替えてから参照してください)
- ・ 性別を選択する radio ボタンを追加してみる  
※radio ボタンは<input>タグを使用し、type 属性に”radio”を指定する

### (演習 1) 簡単なチャット

メニュー

メッセージの送受信  
設定  
ログアウト

---

メッセージ送信

---

メッセージ

台風が近づいています。  
台湾方面へ行く予報になっていますね！  
でも明後日から船は欠航になるみたいですよ

### (演習 1) 簡単なチャット

メニュー

メッセージの送受信  
設定  
ログアウト

---

メッセージ送信

女性 男性 答えたくない

---

メッセージ

台風が近づいています。  
台湾方面へ行く予報になっていますね！  
でも明後日から船は欠航になるみたいですよ

これから JavaScript に関する解説を行います。

obniz の開発環境で作成したソースコードはすべて削除してください。

作成したソースコードが必要な場合は、ダウンロード等を行ってください。

「[1] LED を点灯／消灯する」を使用します。

講座サポートサイトから obniz の開発環境へコピー&ペーストを行い、動作確認を行ってください。(6月19日のテキスト参照)

## ●JavaScript の動作を確認する

※JavaScript は<body>タグの一番下 (</body>の上) に記述する

※JavaScript は<script>タグ内に記述する

```
<script>

  const obniz = new Obniz("OBNIZ_ID_HERE"); ← ①

  // obniz がオンラインになったら
  obniz.onconnect = async function() { ← ②
    // LED を定義
    const led = obniz.wired("LED", { anode: 0, cathode: 1 }); ← ③

    // obniz の画面に初期表示
    obniz.display.clear(); ← ④
    obniz.display.print("Hello World"); ← ⑤

  }

}; ← ⑤
</script>
```

① obniz ボードの使用を設定する。

※"OBNIZ\_ID\_HERE" を所有している obniz ボードの ID に変更する必要がある。

② obniz ボードが Wi-Fi に接続し動作できる状態になったら { ~ } までの処理を行う

③ LED の接続設定を行う

④ obniz ボードの液晶画面をクリアする

⑤ obniz ボードの液晶画面に “Hello World” と表示する

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

const : 定数の宣言

obniz : 変数 (定数) 名

※変数名なので obniz でなくても可。例えば hatake と定義しても可。

ただし、以降 obniz の処理には hatake と記述する必要がある

Obniz() : obniz の処理が定義されているオブジェクト

new : new に続くオブジェクトを新規に作成して使用可能な状態にする

= : 代入演算子、右側の演算 (新規に作成したオブジェクト) を 左の変数に代入する

処理の意味 :

Obniz()のオブジェクトを作成し obniz という変数 (定数) で使用可能にする。

obniz ボードを使用する場合、必ずすべての処理の前に定義を行っておく必要がある。

※変数とは、プログラム内で一時的に使用するデータのこと

例えば、10 回処理を行う場合、1,2,3 . . . 8,9,10 のように何回目かを一時的に記憶するために使用する

※定数とは、1 度だけ設定が可能な変数

※オブジェクトとは、関連のあるデータを 1 つにまとめたもの。

例えば、社員名簿の場合 {社員名、生年月日、部署} を 1 つのデータとして扱う

```
obniz.onconnect = async function() { ... };
```

obniz.onconnect : obniz ボードが Wi-Fi に接続し使用可能になったときの処理を格納する変数

async : 同期処理。ここでは説明しません。

function() : 一連の処理を { ~ } 内にまとめて記述する。JavaScript 言語では「関数」という。

処理の意味 :

obniz ボードが Wi-Fi に接続し使用可能になったときの処理を { ~ } に記述する。

obniz ボードを使用する場合、obniz に関する処理はここに ( { ~ } 内に) 記述する。

※obniz の Wi-Fi が切れた場合の処理は obniz.onclose = async function() { };