

# リスキリング

JavaScript 第3回（全3回） / 全7回

## git

- バージョン管理システム
- 「分散型」

## GitHub

- git を「集中管理」するための Web サービス

# GitHub を使ってみよう！

1. GitHub にアカウントを作成する。
2. 新しいリポジトリを作成する。
3. VSCode で clone する。
4. 何かファイルを作って、commit する。
5. push する。
6. GitHub 上にファイルが来ていることを確認する。

## VSCode で clone

1. 「Ctrl + Shift + P」 でコマンドパレットを開く。
2. 「gitcl」と入力し、「Git: Clone」 コマンドを選択。「Clone From GitHub」 を選択する。
3. GitHub にサインインするように求められたら、サインインプロセスを実行する。
4. リポジトリの URL を入力する。
5. プロジェクトの複製先となるローカル ディレクトリを選択 (または作成) する。

## 試してみよう！

前回の宿題の模範解答を、以下の GitHub に置いてあります。

<https://github.com/tamx/reskilling-sample.git>

試しに clone して実行してみましょう。

# オブジェクト

オブジェクトとは、

(乱暴に言うと) 「データ型」とその「計算方法」の塊です。

データを表す構造とその計算ロジックを「クラス」という塊で表します。

例：分数を表すクラス

```
class Rational {  
    var bunshi;  
    var bunbo;  
}
```

# メソッド

オブジェクトの中で定義されている関数は「メソッド」と呼びます。

```
class Rational {  
    time(rhs) {  
        this.bunshi *= rhs.bunshi;  
        this.bunbo *= rhs.bunbo;  
    }  
}
```

# インスタンス

オブジェクトのデータが入っている本体を「インスタンス」と呼びます。

インスタンスの作成は「コンストラクタ」という特別なメソッドで行います。

コンストラクタを用いて、インスタンスを作成するために、**「new 演算子」**を使います。

```
class Rational {  
    constructor(bunshi, bunbo) {  
        this.bunshi = bunshi;  
        this.bunbo = bunbo;  
    }  
}  
  
var bunsu = new Rational(1, 3);
```



## 静的メソッド

インスタンスと関係のないクラス内のメソッドを「静的メソッド」と呼びます。

```
class Rational {  
    static time(lhs, rhs) {  
        let bunshi = lhs.bunshi * rhs.bunshi;  
        let bonbo = lhs.bunbo * rhs.bunbo;  
        let instance = new Rational(bunshi, bonbo);  
        return instance;  
    }  
}  
  
var bunsu1 = new Rational(1, 3).time(new Rational(3, 2));  
var bunsu2 = Rational.time(new Rational(1, 3), new Rational(3, 2));
```

# インスタンス同士の比較

```
class Rational {  
    equals(rhs) {  
        if (this.bunshi !== rhs.bunshi) {  
            return false;  
        }  
        if (this.bunbo !== rhs.bunbo) {  
            return false;  
        }  
        return true;  
    }  
}  
  
var bunsu1 = new Rational(1, 3).time(new Rational(3, 2));  
var bunsu2 = Rational.time(new Rational(1, 3), new Rational(3, 2));  
console.log(bunsu1.equals(bunsu2));
```

## 演習

- 上の分数クラスを利用して、分数同士の割り算メソッドを作成してください。

## 郵便番号から住所変換

今まで学んだことを活かして、入力した郵便番号から住所を保管する Web ページを作成しましょう。

```
<input type="text" id="zip"></input>
```

```
var element = document.getElementById("zip");  
console.log(element.value);
```

# 演習

香川県の郵便番号の CSV が以下の URL から取得できます。

<http://etp.xsrv.jp/reskilling/>

```
const url = "http://etp.xsrv.jp/reskilling/2/37KAGAWA_ZIP_UTF-8.csv";

fetch(url)
  .then((response) => {
    return response.text();
  })
  .then((data) => {
    console.log(data);
    // CSV を読み込み
    let csvArray = [];
    let lines = data.split(/\r\n|\n/);
    for (let i = 0; i < lines.length; ++i) {
      let line = lines[i].replace(/"/g, "");
      let cells = line.split(",");
      csvArray.push(cells);
    }
    console.log(csvArray);
  });
```

## 演習

```
function search(zip) {  
  for (let index in csvArray) {  
    let data = csvArray[index];  
    if (data[2] === zip) {  
      let address = data[7] + data[8];  
      return address;  
    }  
  }  
  return "";  
}  
  
let address = search("7600019");  
console.log(address);
```

# 休憩

体を動かして、ストレッチしましょう。