

# IT 基礎 2-2



講師：王紹宇

2023-12-23

# ウォーミングアップ

- 11111 は素数ですか、合成数ですか?

答えは、合成数です。

$$41 * 271 = 11111$$

# プログラミングで計算する

```
function isPrime(x) {  
  if (x <= 1) {  
    return false  
  }  
  for (let i = 2; i <= Math.sqrt(x); i++) {  
    if (x % i === 0) {  
      return false  
    }  
  }  
  return true  
}  
// isPrime(1) => false  
// isPrime(2) => true  
// isPrime(3) => true  
// isPrime(4) => false
```

# 今週のコンテンツ

## 2-2、JAVASCRIPT プログラミング（土曜）

- Javascript プログラミング
- 開発環境とツール
- Docker

## 2-3、フロントエンド開発基礎（日曜）

- HTML 概要
- Javascript 概要
- CSS 概要

# JAVASCRIPT プログラミング

[See code](#)

```
1 // use library to get user input
2 const prompt = require('prompt-sync')({ sigint: true })
3
4 // output num1 is lower or higher than num2
5 // returns if num1 and num2 are equal
6 function compare(num1, num2) {
7   let res = false
8   if (num1 < num2) {
9     console.log(`${num1} is lower`)
10  } else if (num1 > num2) {
11    console.log(`${num1} is higher`)
12  } else {
13    res = true
14  }
15  return res
```

# TEMPLATE STRINGS

```
const age = 20
const name = 'David'
const message = `Hello ${name}, your age is ${age}.`
console.log(message) // => Hello David, your age is 20.
```

# 配列の操作

- Object を新規する
  - `names = []`
  - `names = Array()`
- 要素を追加する
  - `names.push("Bob")`
  - `names.push("Alice", "Bob", "Eve")`
  - `names[4] = "Charles"`
- 要素をアクセスする
  - `firstName = names[0]`
  - `secondName = names[1]`
- `Array.push()`
- `Array.pop()`
- `Array.slice()`
- `Array.splice()`
- `Array.map()`
- `Array.length`



# オブジェクトの操作

- Object を新規する
  - `a = {}`
  - `a = Object()`
- 属性を追加する
  - `a.name = "Bob"`
  - `a["age"] = 20`
- 属性をアクセスする
  - `b = a.name`
  - `b = a["age"]`
- `Object.keys()`
- `Object.values()`
- `Object.entries()`
- `Object.assign()`
- `length: Object.keys(a).length`

# ループ

- do...while
- while
- for
- for...of
- for...in
- break
- continue

# 関数

- Function
- Arrow function =>

```
const plusOne = (x) => x + 1
console.log(plusOne(100)) // => 101
```

```
const minusOneIfOdd = (x) => {
  if (x % 2 !== 0) {
    return x - 1
  }
  return x
}
```

- Recursive function

```
f(x) = 1;           if x == 0
f(x) = f(x-1) * x;  if x > 0
```

```
function factorial(x) {
  if (x === 0) {
    return 1
  }
  return factorial(x - 1) * x
}
```

# インポートとエクスポート

- import
- export

# 開発環境とツール

- Visual Studio Code おすすめのプラグイン
  - フォーマッター: [Prettier - Code formatter](#)
  - ファイル比較: [Partial Diff](#)
- GUI git 管理ツール
  - [Source Tree](#)

豆知識へ

# 宿題(来週金曜まで)

## プログラミング練習(JavaScript)

### 1. 素因数分解

- input: 120
- output: [2, 2, 2, 3, 5]
- input: 11111
- output: [41, 271]

### 2. $n$ の階乗を計算する: $n!$

- input: 0
- output: 1
- input: 5
- output: 120

### 3. 第 $n$ 項までのフィボナッチ数を入力する(第 0 項から始まる)

- input: 0
- output: [0]
- input: 5
- output: [0, 1, 1, 2, 3, 5]

# 質問応答

つづく