

# JavaScript基礎実習

## 01. JavaScript概要

株式会社ジーードライブ

# 今回学ぶこと

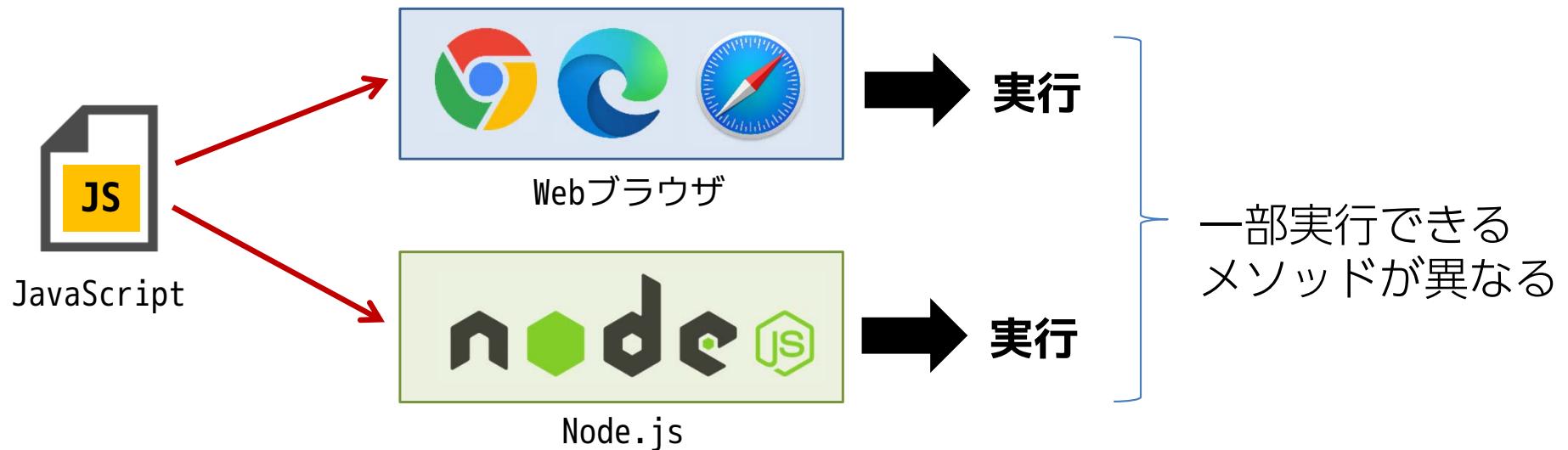
---

- JavaScriptの開発環境
- JavaScriptの実行方法

# JavaScriptの開発環境

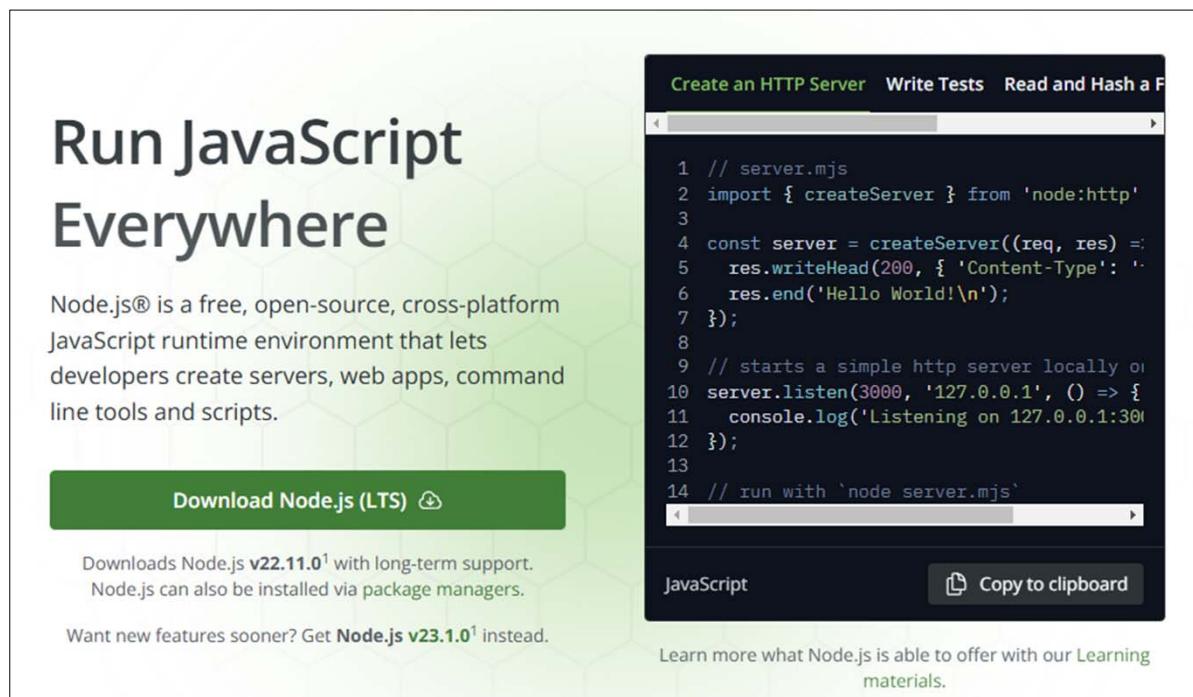
# JavaScriptの開発に必要なもの

- テキストエディタ
  - JavaScriptコードの記述に使用する
  - この講座ではVSCodeを使用
- JavaScript実行環境
  - Webブラウザ（Webクライアント用JavaScript）
  - Node.js（サーバー用JavaScript）



# Node.jsのインストール

- <https://nodejs.org/en/>
- 基本的にはLTS(長期サポート)版をインストールする
  - Node.jsをインストールすると、 **npm** (Node Package Manager)も併せてインストールされる



# Node.jsのインストール

- コマンドプロンプト等でバージョン表示のコマンドを実行することで、インストールの確認ができる

Node.jsのインストール確認(黄字部分を入力し、Enter)

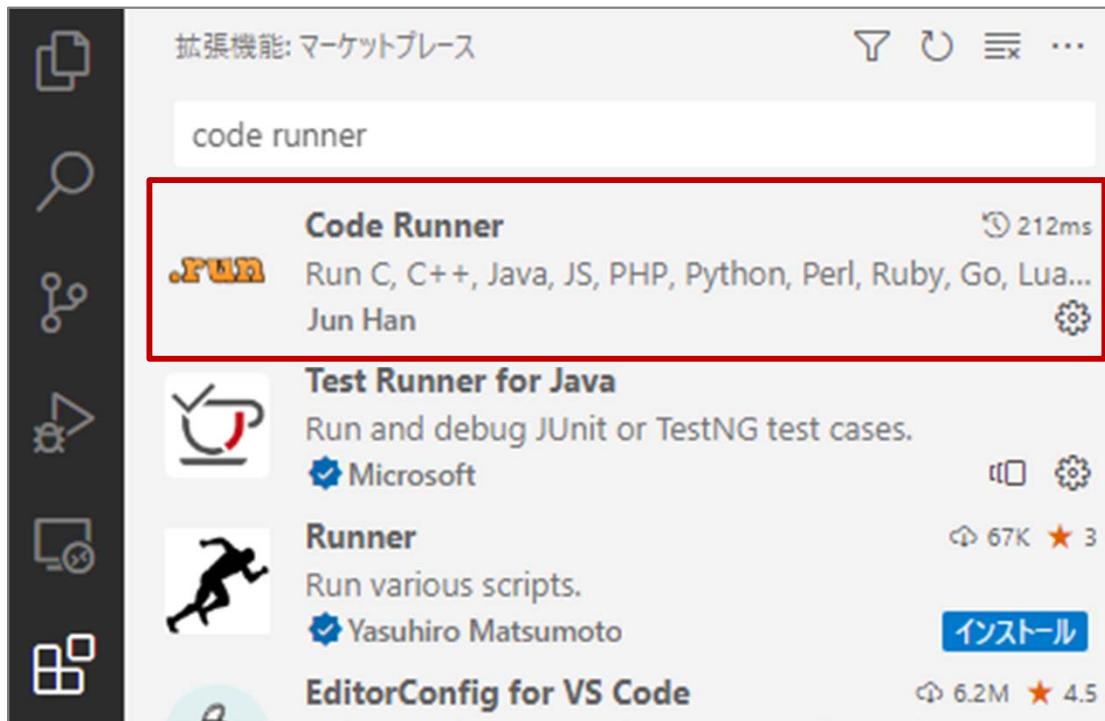
```
C:\Users\zdXXXX> node -v  
v20.11.1
```

npmのインストール確認(黄字部分を入力し、Enter)

```
C:\Users\zdXXXX> npm -v  
10.2.4
```

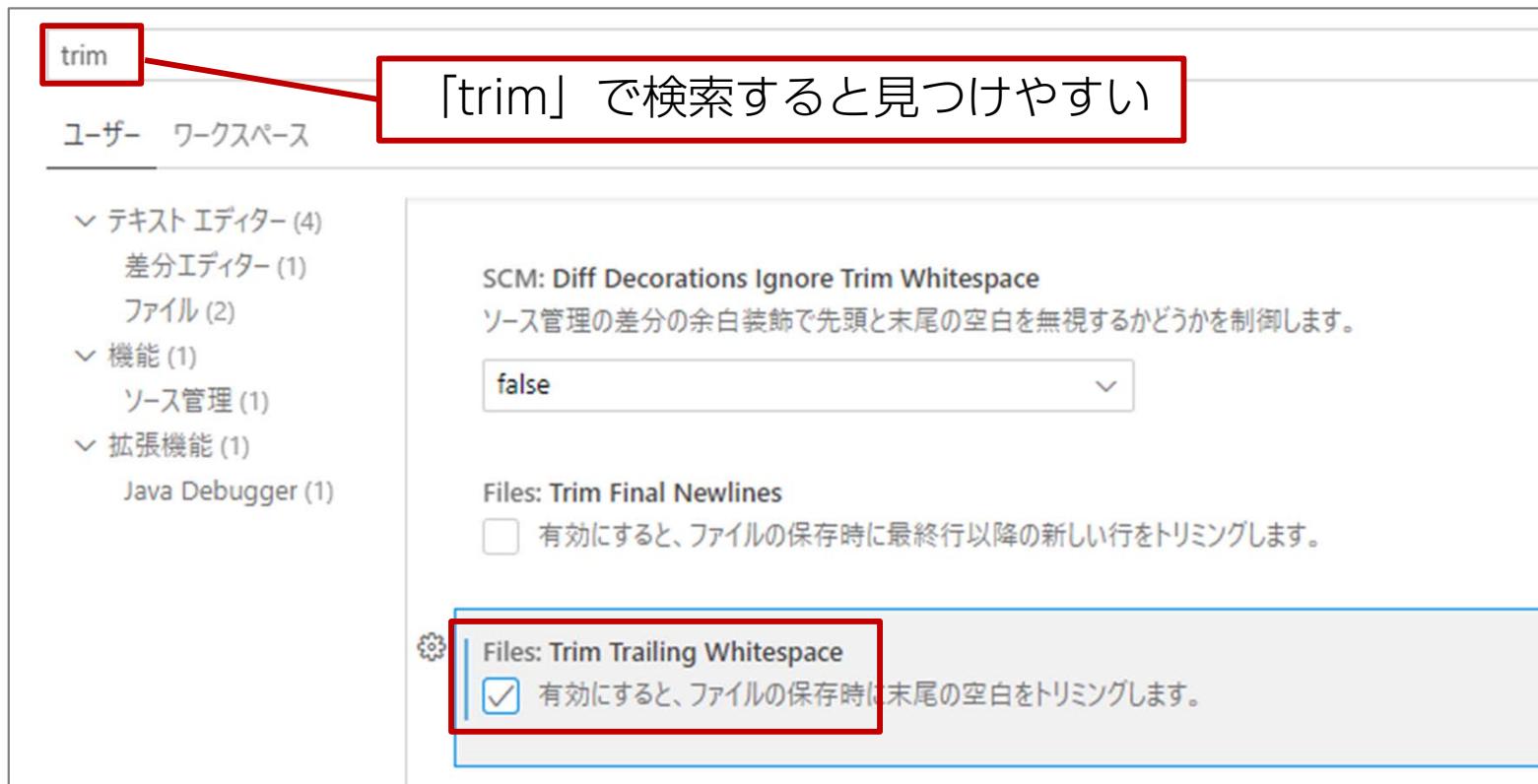
# VSCODEの準備

- 「Code Runner」という拡張機能をインストールする
  - これによりVSCODE上でJavaScriptのプログラムをNode.js環境で手軽に実行できるようになる



# VSCodeのユーザー設定

- ファイル保存時に行末の空白を削除する
  - [ファイル]⇒[ユーザー設定]⇒[設定]を開く
  - 「Files: Trim Trailing Whitespace」にチェックを入れる



# VSCodeのユーザー設定

- デバッグコンソールでの表示を見やすくする
  - [ファイル]⇒[ユーザー設定]⇒[設定]を開く
  - 「Debug > Console: Collapse Identical Lines」のチェックを外す



# JavaScriptの実行方法

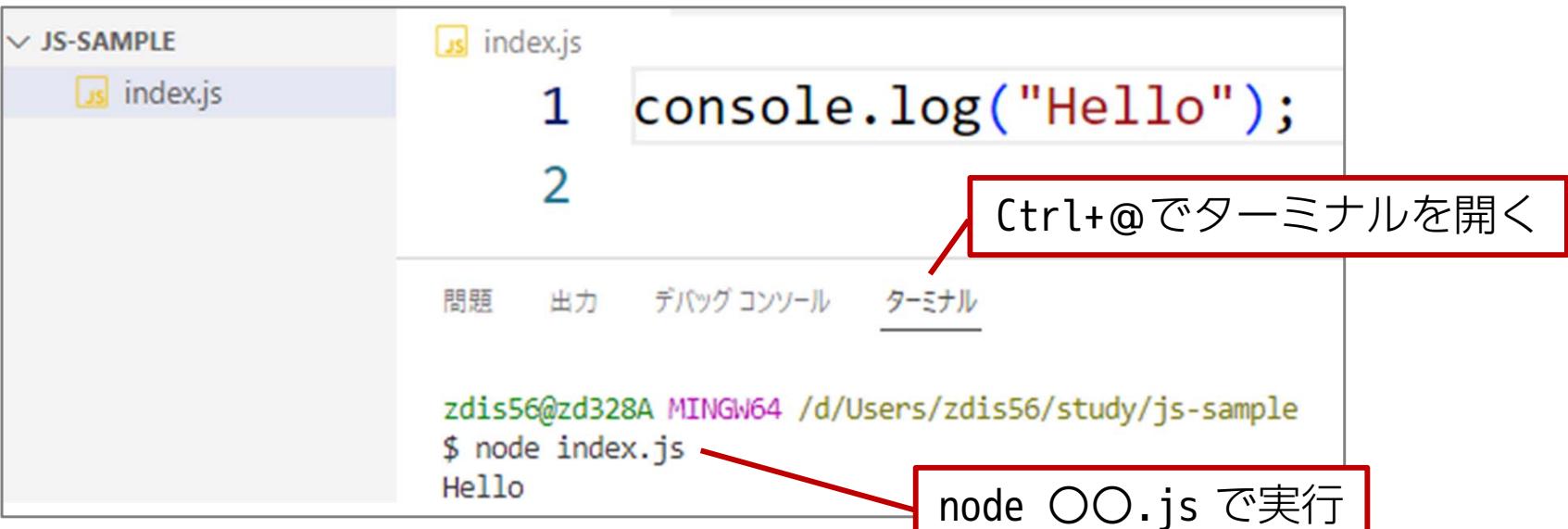
# ブラウザでのJavaScript実行

---

- Webブラウザ上(Webページ内)でJavaScriptを実行させる場合は、JavaScriptの記述方法には以下の3通りの方法がある
  - (1) 独立したファイルに記述してHTML内に読み込む
  - (2) HTML内に記述する
  - (3) HTMLタグの属性として記述する
- JavaScriptが埋めこまれたWebページをWebブラウザで表示することで、JavaScriptプログラムが実行される

# Node.js上でJavaScript実行

- Node.js上でJavaScriptを実行する場合は、独立したファイル(.jsファイル)に記述する
- コマンドラインでNode.jsのnodeコマンドを使ってJavaScriptプログラムを実行する



```
JS-SAMPLE
index.js
1 console.log("Hello");
2

問題 出力 デバッグコンソール ターミナル
zdis56@zd328A MINGW64 /d/Users/zdis56/study/js-sample
$ node index.js
Hello
```

# Code RunnerでのJavaScriptの実行

- 拡張機能「Code Runner」をインストールしている場合、画面右上の実行ボタンを押すことで、Node.jsによるJavaScriptの実行ができる

The screenshot shows the VS Code interface with the 'Code Runner' extension installed. A red box highlights the '実行ボタン' (Run button) in the top right corner of the code editor. Another red box highlights the '結果の出力' (Output) tab in the bottom navigation bar. The code editor displays a file named 'run-code.js' containing the following code:

```
1 /*  
2  | 金額の計算  
3 */  
4 const price = 90; // 単価  
5 const amount = 3; // 個数  
6 const total = price * amount; // 合計  
7 console.log(`合計: ${total}円`);  
8
```

The output panel shows the results of running the code with Node.js:

```
[Running] node  
"d:\Users\zdis56\study\js-sample\practice01\run-code.js"  
合計: 270円  
  
[Done] exited with code=0 in 0.181 seconds
```

# JavaScriptの例

- コメントの書き方など、Javaとの共通点も多い
- ステートメント末尾のセミコロンは基本的に省略可  
⇒自動的にセミコロンが補われる

```
/*
  金額の計算
*/
const price = 90; // 単価
const amount = 3; // 個数
const total = price * amount; // 合計
alert('合計：' + total + '円');
```

省略可

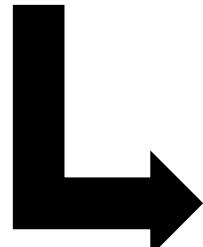
合計：270円

コンソール

# コンソール画面への出力

- `console.log()`を使うと、出力画面にメッセージや計算の結果などの情報を表示させることができる
  - Javaの`System.out.println()`のようなもの
  - 引数はカンマ区切りで列挙可能

```
console.log('こんにちは');
console.log('こんにちは', '元気ですか？');
```



```
問題 出力 デバッグコンソール ターミナル ポート
[Running] node "c:\Users\zdis56\study\javascript-practice\test.js"
こんにちは
こんにちは 元気ですか？

[Done] exited with code=0 in 0.116 seconds
```

# 練習

---

- 練習01-1