

# JavaScript基礎実習

## 01. JavaScript概要

株式会社ジードライブ

# 今回学ぶこと

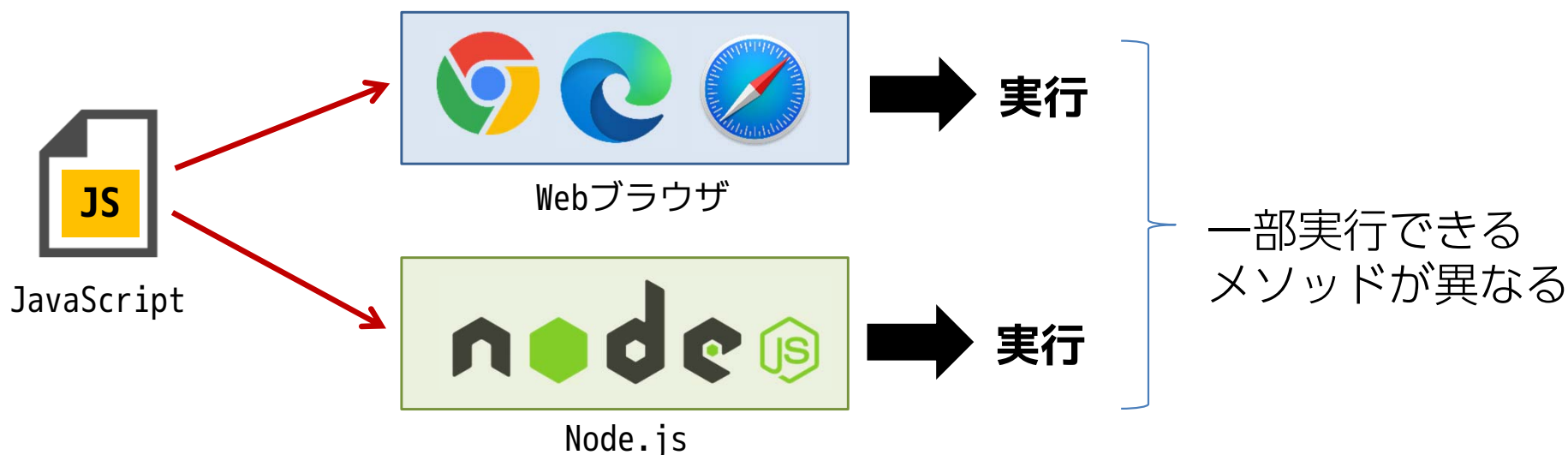
---

- 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

**Run JavaScript Everywhere**

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.

[Download Node.js \(LTS\)](#)

Downloads Node.js **v22.11.0**<sup>1</sup> with long-term support. Node.js can also be installed via package managers.

Want new features sooner? Get **Node.js v23.1.0**<sup>1</sup> instead.

**Create an HTTP Server** Write Tests Read and Hash a File

```
1 // server.mjs
2 import { createServer } from 'node:http'
3
4 const server = createServer((req, res) => {
5   res.writeHead(200, { 'Content-Type': 'text/plain' });
6   res.end('Hello World!\n');
7 });
8
9 // starts a simple http server locally on port 3000
10 server.listen(3000, '127.0.0.1', () => {
11   console.log('Listening on 127.0.0.1:3000');
12 });
13
14 // run with `node server.mjs`
```

JavaScript [Copy to clipboard](#)

Learn more what Node.js is able to offer with our [Learning materials](#).

# 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)

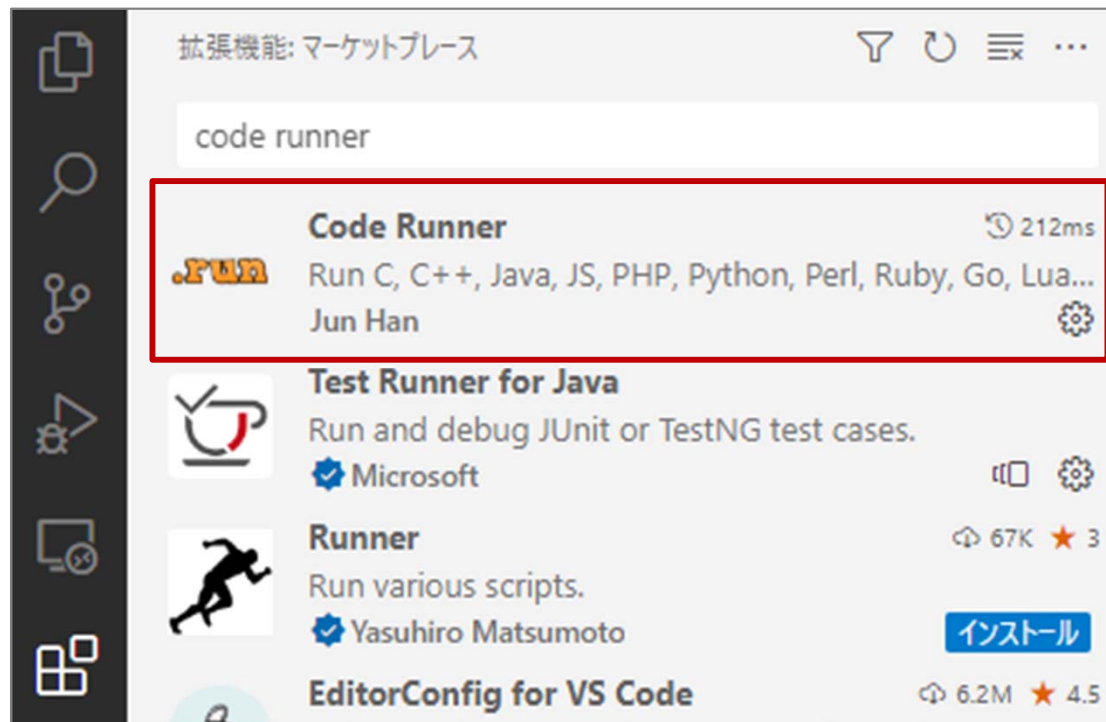
```
C:¥Users¥zdXXXX> node -v  
v20.11.1
```

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
```

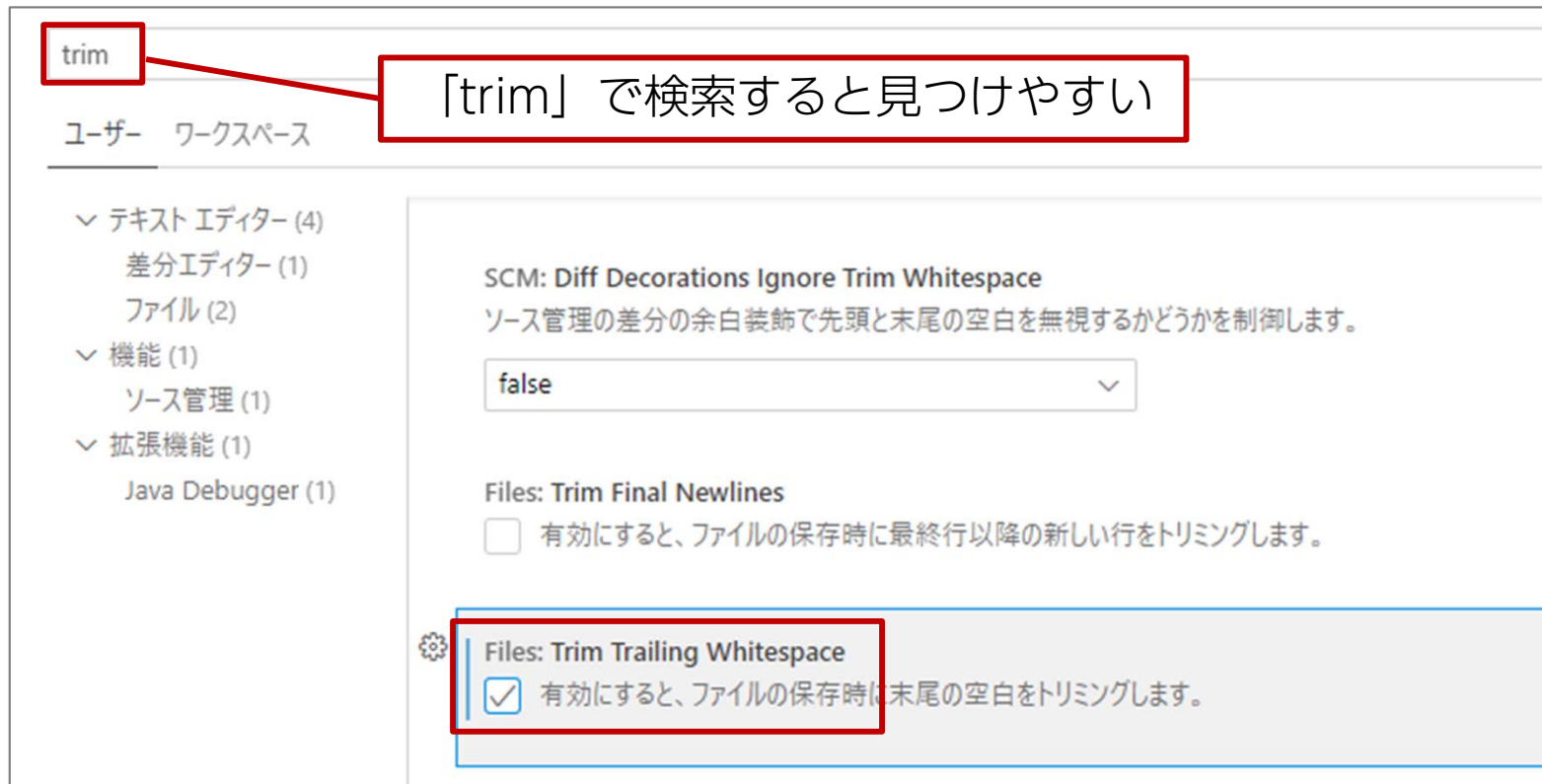
# VSCodeの準備

- 「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上での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
└─ JS 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

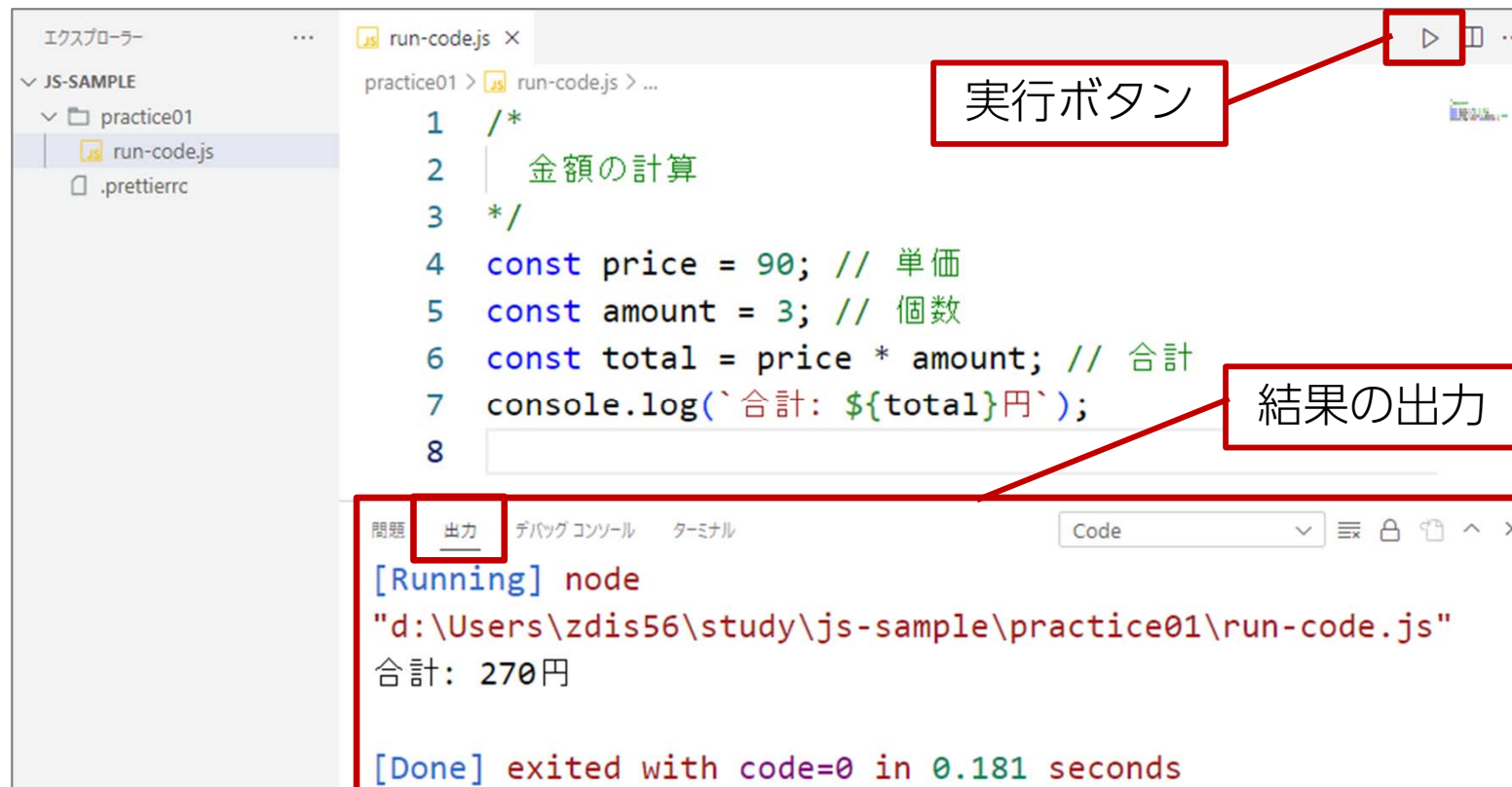
```
zdis56@zd328A MINGW64 /d/Users/zdis56/study/js-sample
$ node index.js
Hello
```

Ctrl+@でターミナルを開く

node 〇〇.js で実行

# 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



# 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

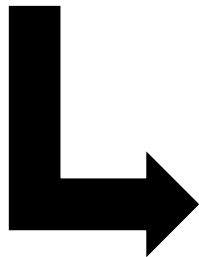
コンソール

合計: 270円

# 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