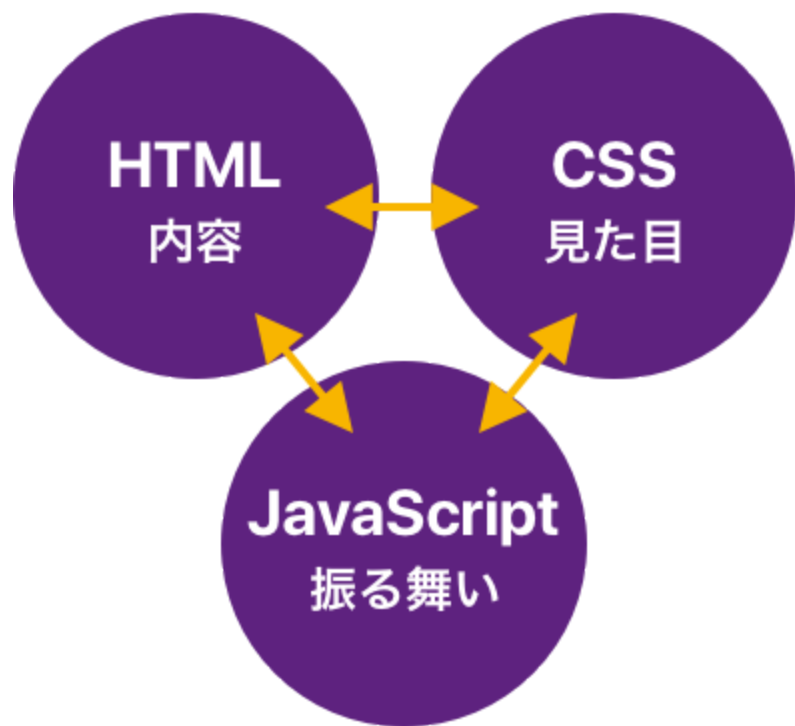


JavaScript

- JavaScriptの役割
- JavaScriptの記述場所
- JavaScriptの文法
- コンソール
- コメント

JavaScriptの役割

3つとも異なる言語でやりとりが必要



JavaScriptの記述場所

1. HTML内に記述
2. 外部jsファイルに記述

1. HTML内に記述

xxx.html

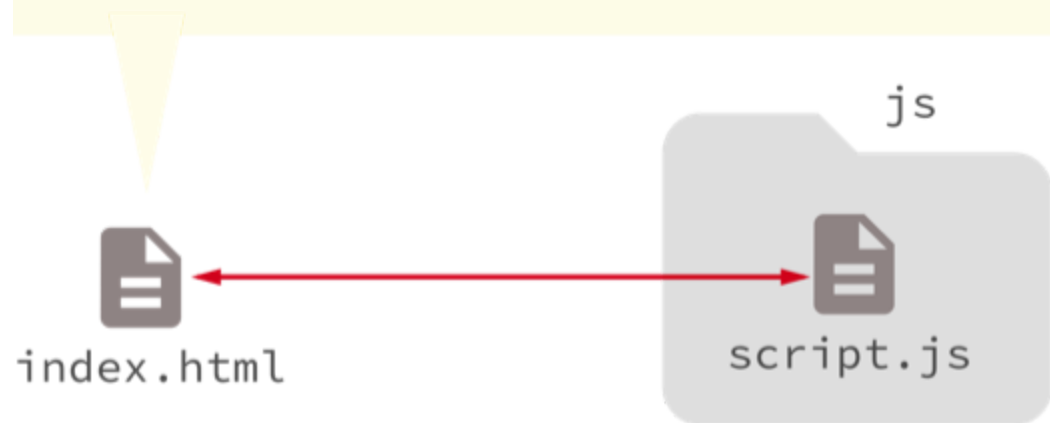
```
    </footer>
    <script>

    // ここにJavaScriptを記述する

    </script>
  </body>
</html>
```

2. 外部jsファイルに記述

```
<script src="js/script.js"></script>
```



JavaScriptの文法

3つの記述パターン

1. 命令式 `()`
2. 代入式 `=`
3. グループ `{}`

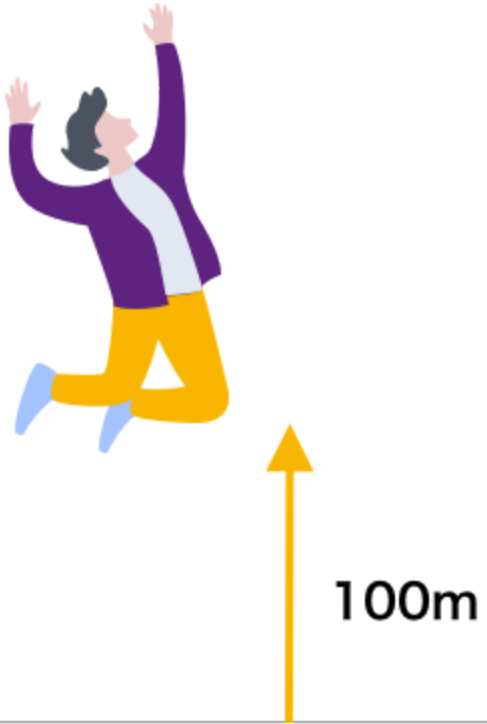
1. 命令文

()

命令を出す

命令の名称(引数);

```
jump(100);
```



jump という命令名があって、() の中に値を入れるとその分だけ実行

引数とは命令に渡す値(パラメーター)

```
alert('alertのテスト');
```

- 記述は半角英数
- セミコロンが文の区切りになる

2. 代入式

=

何らかの値を保存させる

```
要素.属性 = 値;
```

- `=` の前後に半角スペースを入れてもOK

プログラミングの `=` (イコール) は 左辺に右辺の値を保存すること

```
document.querySelector('h1').style.color = '#FF0000';
```

この文を日本語に訳すと、

HTML `document` からタグ `querySelector` を取り出して、CSS `style` の文字色 `color` を `#FF0000` に変更する

ドットシンタックス

オブジェクトと命令を .(ドット) でつなげて記述する

```
document.querySelector('h1').style.color = '#FF0000';
```

3. グループ

```
{ }
```

処理をグループにする場合に使う

```
命令(){  
    処理1  
    処理2  
    処理3  
};
```


{} の中身はインデントする習慣をつけてください

```
function setColor(){  
    document.querySelector('h1').style.color = '#FF0000';  
    document.querySelector('h1').style.backgroundColor = '#DDDD';  
}
```

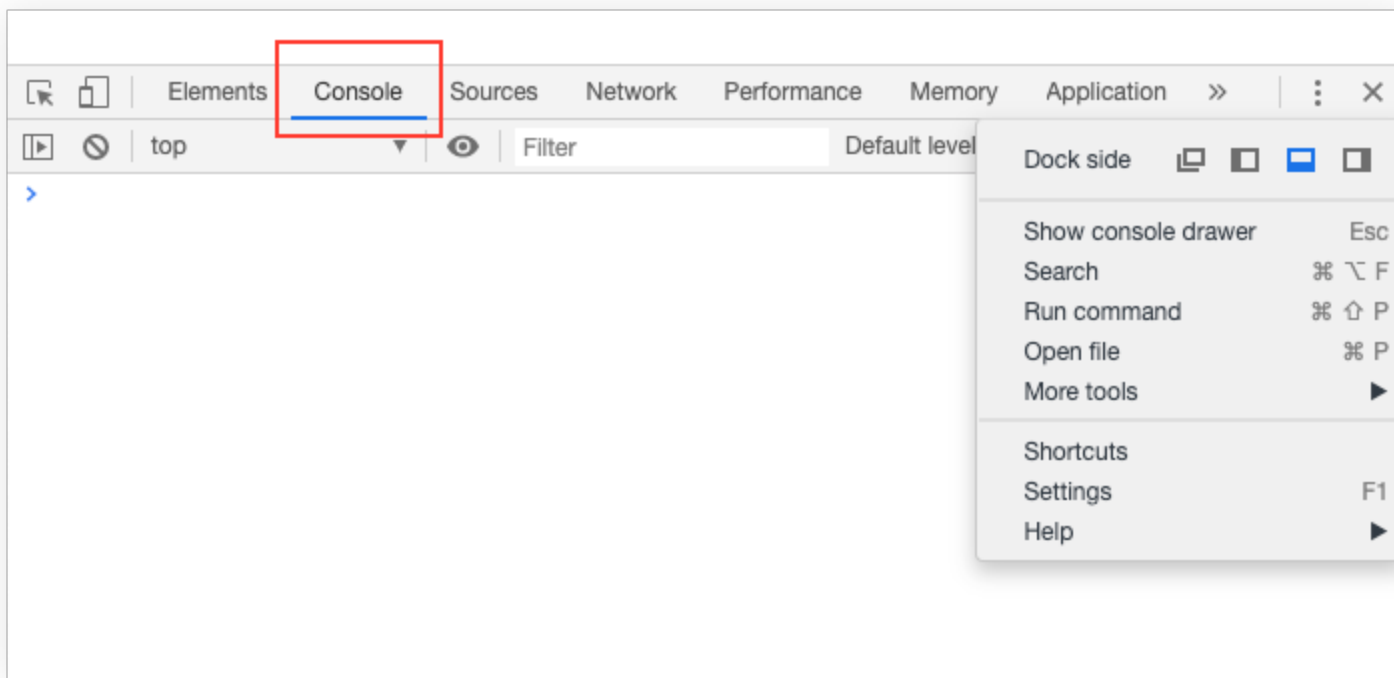
その他

コンソール

JavaScriptで書いたソースの内容を確認する方法

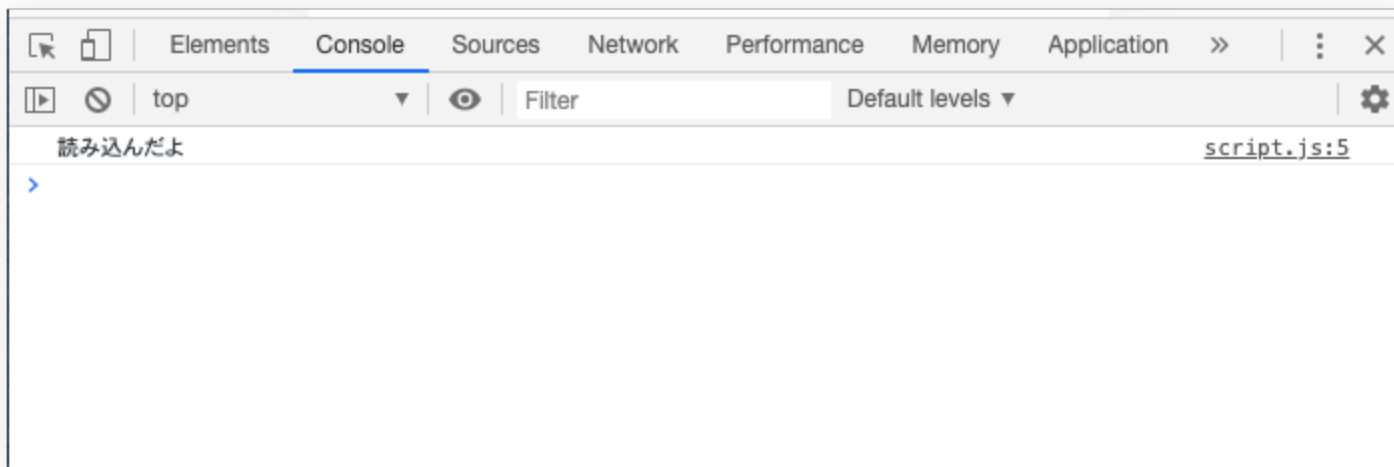
コンソールの出し方

1. ブラウザ上で右クリックして 検証
2. Console タブを選択

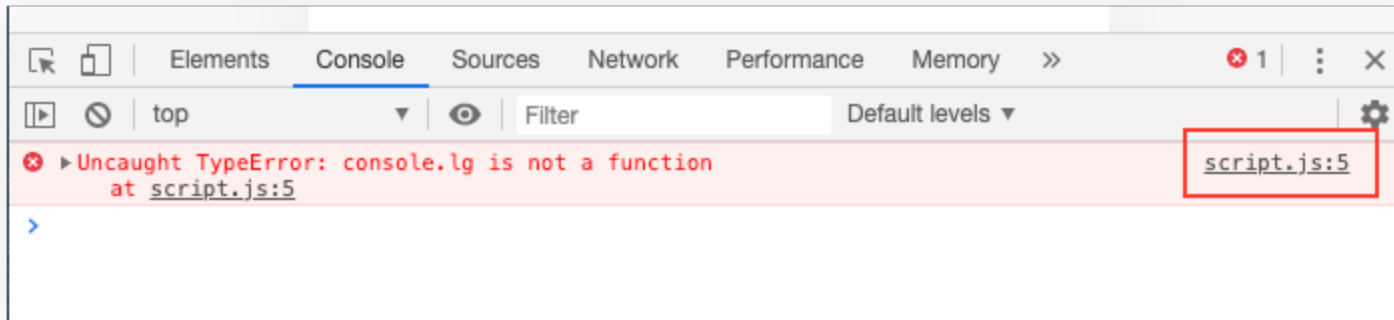


コンソールの実行

```
console.log('読み込んだよ');
```



コンソールでエラー確認



エラーの場合でも確認ができる。
どのファイルの何行目がエラーか教えてくれる。

コメント

コメントの書き方

コメントとはプログラミング内に記述する「メモ書き」のようなもの

```
// 1行のコメント
```

```
/*  
複数行コメント  
複数行コメント  
*/
```