

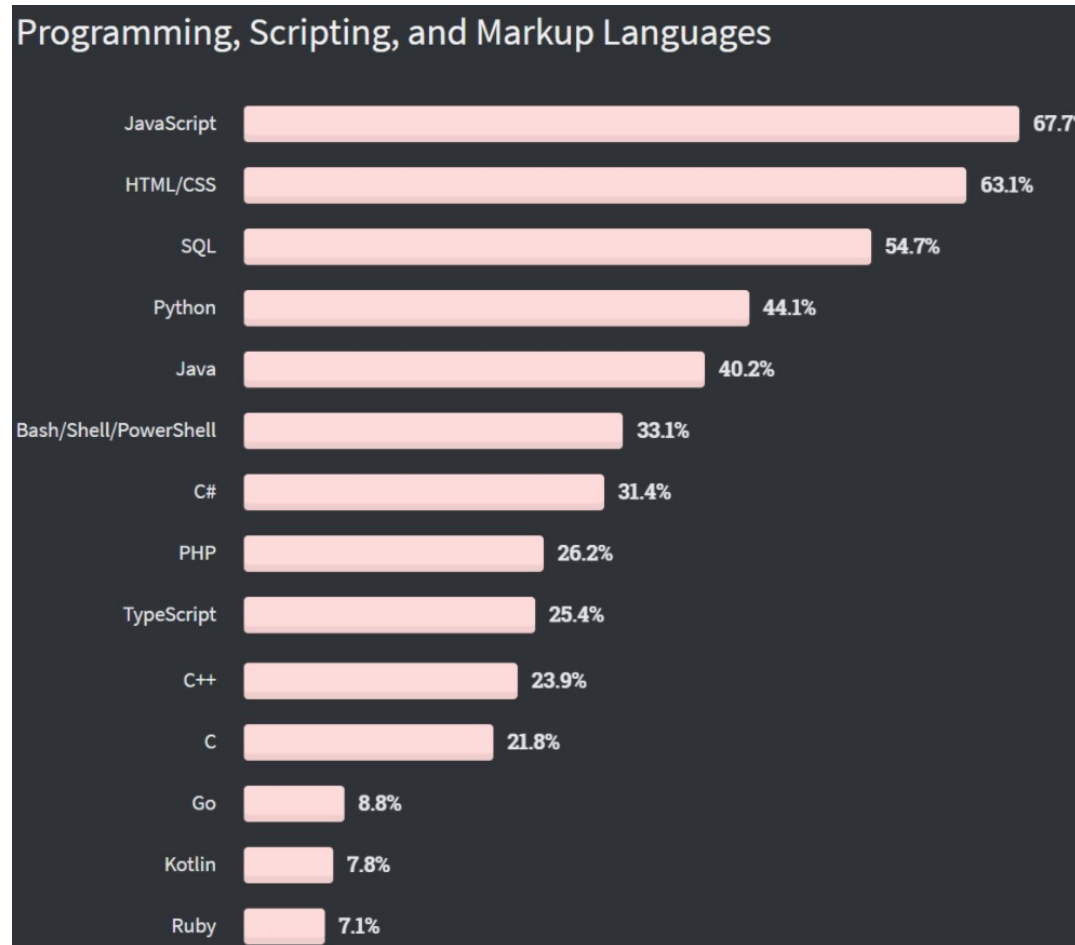
JavaScript

# 11번째 세션

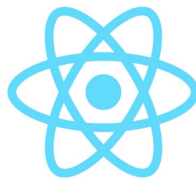
---

NEXT X LIKELION 이한주

# Introduction



# Introduction – JavaScript 생태계



webpack



# | 그래서 JavaScript가 뭔데?

- 파이썬과 같은 동적 타입 언어
- 주로 웹 프론트엔드에 쓰이며, 스크립트 언어에 해당됨
- 여태 인형 옷 입히기를 했으면, 인형을 움직여보자!



# | 동적 타입 언어란? (Dynamic typed language)

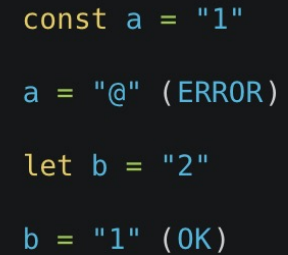
- 런타임에 타입 검사를 시행한다.
- 자바스크립트와 파이썬이 '동적 타입 언어'에 속한다
- 약타입 언어 : 자료형이 맞지 않을 시에 암묵적으로 타입을 변환하는 언어 -> JS
- JS에서  $1 + "1" = "11"$ 이다.
- 강타입 언어 : 자료형이 맞지 않을 시에 에러 발생, 암묵적 변환을 지원하지 않음 -> Python
- Python에서  $1 + "1"$ 은 TypeError
- 결론 : 둘다 근본이 없다.

# | JavaScript VS Python 자료형 비교

Javascript	Python
Number	int / float
String	str
Boolean	bool
Array	list
Object	dict
Function	def

# JavaScript 변수 선언하기

- 블록 유효범위 변수를 선언
- `const today = 'monday'`
- `let num = 123`
- `const`는 변수 재선언, 재할당 불가능
- `let`은 변수 재선언 불가능, 재할당 가능



```
const a = "1"

a = "@" (ERROR)

let b = "2"

b = "1" (OK)
```

# JavaScript Object

- 파이썬의 dict와 같음

생성할 때

```
const obj = {  
  name : "12",  
  age: 123  
}
```

참조할 때

```
obj.name  
obj["name"]
```

만약에 key 이름을 잘 못 넣거나,  
없는 key를 찾을때는 **undefined**  
반환

할당할 때

```
obj.name = "OneWeek"  
obj["name"] = "OneWeek"
```

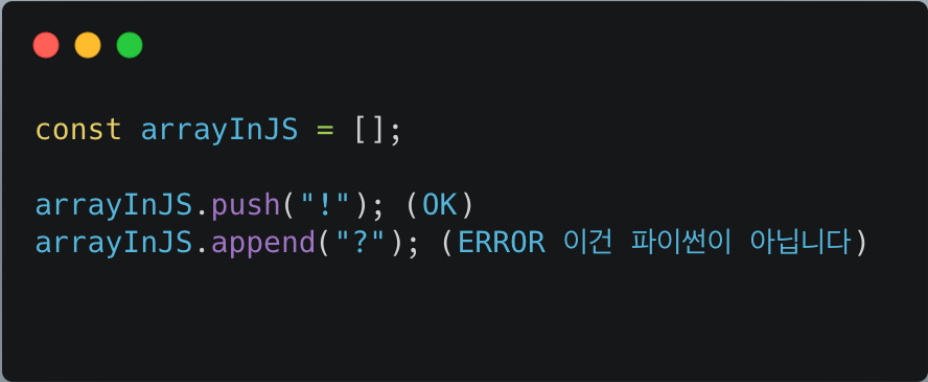
동적으로 추가 및 수정 가능

- 참조형 타입 (Reference Type)
- Array, Function, Date 등등 다수가 Object를 기반으로 만들어져 있음 (몰라도됨)



# JavaScript Array

- 파이썬의 list와 같음
- 하지만 Method 사용 방법이 다름
- 메소드가 궁금하다면 알아서 공부(너무 많습니다..) ([https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/Array))
- 다른 언어의 Array와 동일하지 않음 -> Object를 기반으로 만들어졌기 때문에 JS에서의 Array는 Array-Like한 Object임 (몰라도 됨)



```
const arrayInJS = [];  
  
arrayInJS.push("!"); (OK)  
arrayInJS.append("?"); (ERROR 이걸 파이썬이 아닙니다)
```

# JavaScript Function

- 파이썬의 def와 같음
- 함수를 쓰는 방법은 2가지가 있음

1. Function으로 정의하기
2. Arrow Function으로 정의하기

2가지 방법은 JS엔진 상에 차이점이 있으나, 몰라도 됨 (궁금하면 검색)

Function으로 함수 정의하기

```
function add (a, b) {  
    return a + b  
}
```

Arrow Function으로 함수 정의하기

```
const add = (a, b) => {  
    return a + b  
}
```

# | JavaScript Console.log(“message”)

- 파이썬의 print와 같음
- Message를 찍어보거나, 변수를 넘겨주면 알아서 찍음

# JavaScript의 비교 연산자

- JS는 약타입 언어

```
Console.log(1 == "1")
```

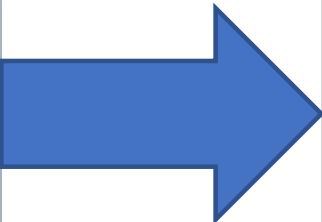
// true => 하나는 숫자형, 하나는 string인데 두개가 같다고 생각함

약타입 언어이기 때문에, 묵시적 형변환을 하기 때문 (궁금하면 검색)

- 그래서, JavaScript에서는 == 대신 ===을 통해 type과 변수를 모두 검사한다.

# JavaScript의 삼항 연산자

- 파이썬의 if else를 간단하게 사용 가능
- (조건문) ? (조건문이 참인 경우 반환) : (조건문이 거짓인 경우 반환)



```
def 홀수?(num):  
    if(num % 2 == 0):  
        return False  
    else:  
        return True
```

```
const 홀수? = (num) => num % 2 === 0 ? false : true
```

# | JavaScript 반복문

- for
- for in
- for of
- while
- do while
- forEach, map, reduce 등과 같은 고차 함수도 존재 (궁금하면 검색)

# | JavaScript 조건문

- if
- switch case
- 삼항 연산자  $( ) ? ( ) : ( )$
- 짧은 조건문  
a || b => a나 b 둘 중 하나 참일 경우 실행  
a && b => a와 b 모두 참이어야 실행

**| JavaScript 소개는 여기까지..**

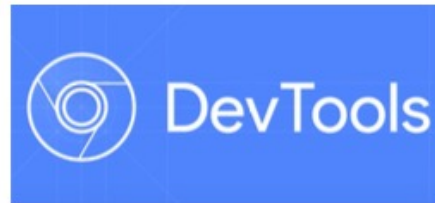


# | DOM

- DOM === Document Object Model
- HTML과 XML 문서의 API
- 브라우저가 HTML과 CSS를 읽으면 DOM이 된다.
- 이 DOM을 조작하는 것이 자바스크립트!

## 실습 환경

- 웹브라우저에 인터프리터가 내장되어있음
- Pipenv같은 거 필요없음 === 웹브라우저만 있으면 실행 가능

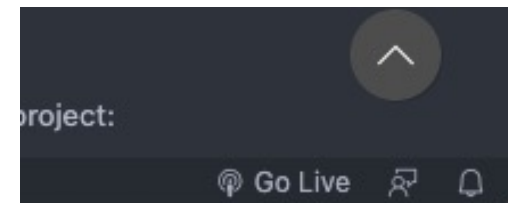
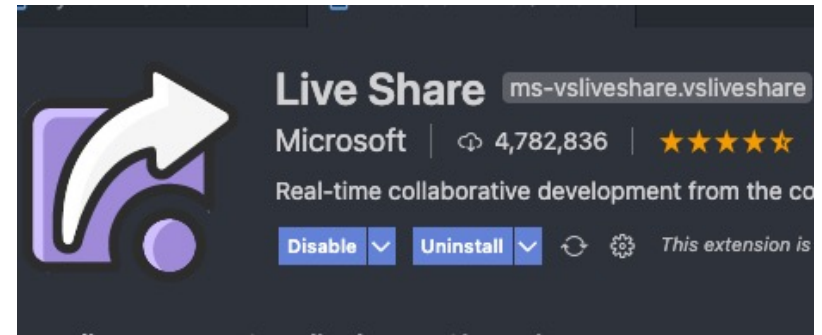


**Chrome Devtools를 사용합니다**

**실행방법: 크롬을 켜 뒤  
(Mac OS) cmd+opt+i  
(windows) F12**

# DOM 조작 실습 셋팅

- 터미널 열고
- mkdir 아무 폴더
- cd 아무 폴더
- touch index.html
- code .
- Vscode에서 "Live server" extension 설치하기



오른쪽 하단에 생겼나 확인

# | 그래서 Script의 위치는 어디에?

- <https://junhobaik.github.io/js-script-position/>
- JavaScript를 사용할 수 있는 <script> </script>의 위치는 어디에?
- <head> 여기에 들어가는 스크립트는 문서를 초기화 하거나 설정하는 가벼운 스크립트 </head>
- <body> Body Tag의 최하단에 위치하는게 좋음 (이유가 궁금하면 검색) </body>

# DOM 조작 실습 – Script 태그 만들기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

  // 반드시 Body의 맨 밑에 Script tag를 답시다
  <script type="text/javascript">
    console.log("시작해 봅시다")
  </script>
</body>
</html>
```

# DOM 조작 실습 – DOM 고르기

```
<body>
  <article>
    <h1 id="title" class="title">자스 어려워요.</h1>
  </article>
  <script type="text/javascript">
    const title1 = document.getElementById('title');

    const title2 = document.querySelector('#title');
    const title3 = document.querySelector('.title');

    console.log(title1 === title2); //true;
    console.log(title1, title2, title3);
  </script>
</body>
```

document.getElementById

- DOM의 Elements를 가지고 id로 가지고 오겠다

document.querySelector (첫번째 원소)

- Css 선택자를 통해 DOM의 Element를 가지고 오겠다.

document.querySelectorAll (배열)

- Css 선택자를 통해 DOM의 Element"s"를 가지고 오겠다.

getElementsByClassName, getElementsByTagName등 다양하지만 다 몰라도된다.

# DOM 조작 실습 – DOM 조작하기

```
<body>
  <article>
    <h1 id="title" class="title">자스 어려워요.</h1>
  </article>
  <script type="text/javascript">
    const title1 = document.getElementById('title');

    title1.innerHTML = '자스 재밌어요';
    title1.style.color = 'red';
  </script>
</body>
```



자스 재밌어요

# DOM 조작 실습 – DOM 조작하기

```
</head>
<body>
  <article>
    <h1 id="title" class="title">자스 어려워요.</h1>
    <div></div>
    <div></div>
    <div></div>
    <div></div>
    <div></div>
    <div></div>
    <div></div>
    <div></div>
  </article>
  <script type="text/javascript">
    const title1 = document.getElementById('title');

    title1.innerHTML = '자스 재밌어요';
    title1.style.color = 'red';

    const divs = document.querySelectorAll('div');
    for (let i = 0; i < divs.length; i++) {
      divs[i].style.width = '100%';
      divs[i].style.height = '100px';
      divs[i].style.fontSize = `${(i + 1) * 10}px`;
      divs[i].innerHTML = `${i}번째 div`;
    }
  </script>
</body>
</html>
```



자스 재밌어요

0번째 div

1번째 div

2번째 div

3번째 div

4번째 div

5번째 div

6번째 div

7번째 div

style = "10px" 과 같이 string으로 대입 해야한다

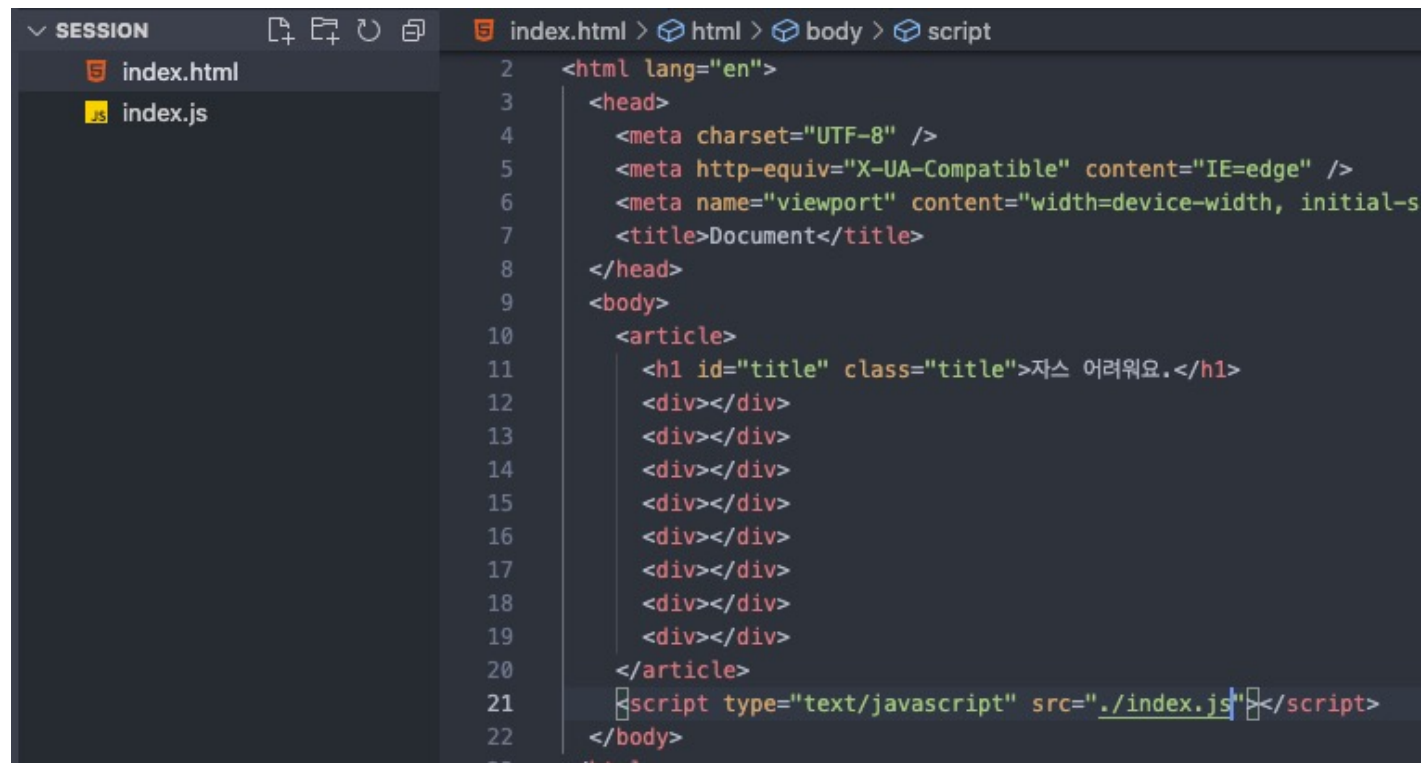


# Script는 항상 html안에 있어야 됨?

CSS와 마찬가지로 불러올 수 있습니다.

index.js 파일 생성 후,  
아까 작성했던 script 파일을 복사

Css와 같이 같은 결과를 얻을 수 있다



```
SESSION [index.html] [index.js]
index.html > html > body > script
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6   <meta name="viewport" content="width=device-width, initial-s
7   <title>Document</title>
8 </head>
9 <body>
10   <article>
11     <h1 id="title" class="title">자스 어려워요.</h1>
12     <div></div>
13     <div></div>
14     <div></div>
15     <div></div>
16     <div></div>
17     <div></div>
18     <div></div>
19     <div></div>
20   </article>
21   <script type="text/javascript" src="./index.js"></script>
22 </body>
```

# 그럼 장고도 되겠네?

이제 Error 메시지는 h1 tag 대신 "alert"로 띄워줍시다.

```
{% if error %}
<script type="text/javascript">
    alert('{{error}}')
</script>
{% endif %}
```

위와 같이 장고 템플릿을 통해 넘어온 변수도 처리 가능

장고에서 불러오려면 <script type="text/javascript" src="{% static 'index.js' %}"> </script> 이런식으로 하면 됩니다.  
해커톤 때 적용하고 싶으면 뭐,, 알아서 공부하시길

## 과제 1

100ms마다 body의 색이 무작위로 변하게

```
</article>
<script type="text/javascript">
  const getRandomHexaColor = () => {
    const hexa = '0123456789abcdef';
    // random color를 반환하는 로직 (힌트는 hexa 코드)
  };

  // setInterval은 100ms당 해당 함수를 실행하는 함수
  setInterval(() => {
    document.querySelector('body').style.backgroundColor =
      getRandomHexaColor();
  }, 100);
</script>
```

## 과제 2

# 시계 만들기

제한 조건

0000년 00월 00 일 00시 00분 00초 (ex. 2021년 05월 20일 03시 21분 09초)

\* 자리수 잘못추기

```
<body>
  <h1 class="now"></h1>
  <script type="text/javascript">
    const clockContent = document.querySelector('.now');

    const getCurrentTime = () => {
      // 현재 시간을 반환하는 객체 Date
      const date = new Date();
      console.log(date); //F12 console창에서 찍어보면 알 수 있다.

      // 시간을 계산해서 clockContent에 표시해줍니다.
    };

    const initClock = () => {
      getCurrentTime();
      setInterval(getCurrentTime, 1000);
    };

    initClock();
  </script>
</body>
```