

웹 개발자 부트캠프 과정

CodingOn

JavaScript

JS





귀여운 - 형용사



미니언이 - 명사



쑤쑤다 - 동사



JavaScript

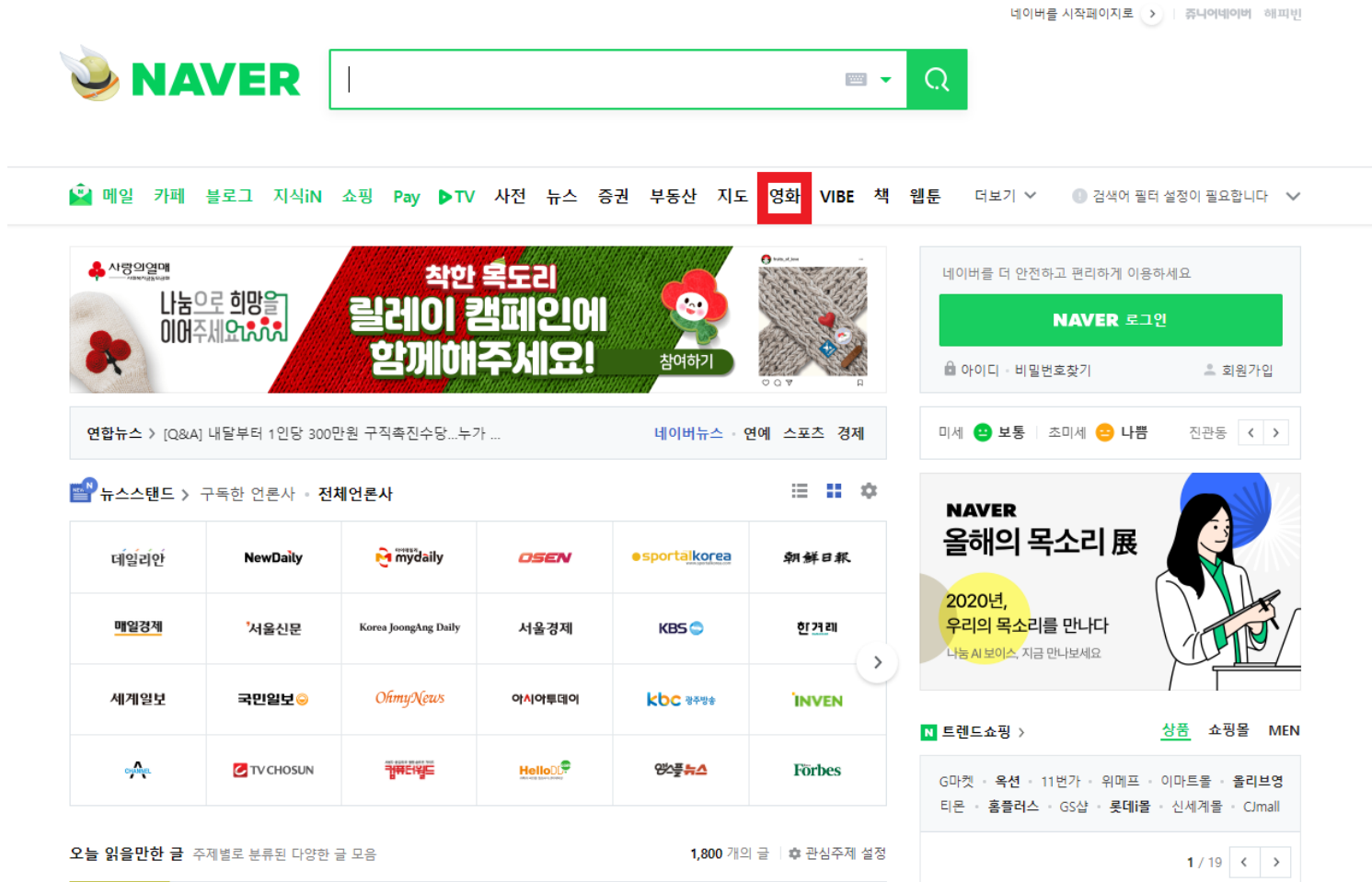


JavaScript

웹 페이지에서 **복잡한 기능**을 구현할 수 있도록 하는
스크립팅 언어 또는 프로그래밍 언어



JavaScript 사용



동적기능

동적처리

이벤트 처리

슬라이드 메뉴

...

JavaScript 참조 방식

내장
방식

?

링크
방식

내장 방식!

```
<script>  
    alert("헤드 그런데 js 파일 링크 위");  
</script>
```

- 위치는 어디서나 사용이 가능합니다!
 - Head 태그 내부
 - Body 태그 내부
 - Head 와 body 사이
 - Body 아래 등

링크 방식!

- Java Script 파일을 따로 만들어서 링크하는 방식 like CSS

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

- 위치는 어디서나 사용이 가능합니다!
 - Head 태그 내부
 - Body 태그 내부
 - Head 와 body 사이
 - Body 아래 등
-

각각의 장단점이 존재 하겠죠?

- 내장 방식
 - 간단하게 만들 수 있음
 - 특정 페이지에서만 작동하는 기능일 경우 내장 방식으로만 따로 구현 가능
- 링크 방식
 - JS 코드의 양이 많아지면 파일로 관리하는 편이 편함
 - 같은 기능을 다른 페이지에서 사용하고 싶을 때 JS 파일 링크만 걸어서 사용 가능
 - 유지 보수 용이성이 편리

일단 해봅시다!

- `console.log("TEXT");`
 - 우리의 디버깅 친구!
 - 물론 이걸 쓰는 건 안 좋은 방식입니다!
- `alert("TEXT");`
 - 화면에 뭐든 떠야 재미 있으니까!
 - 자매품 `confirm("TEXT");`
 - 애는 취소 버튼이 있어요! → 값을 리턴한다 & IF문에 사용 가능!

읽기 순서?

- CSS의 방식 또는 선언 위치에 따라서 읽기 순서가 달라졌지요?
- JS는 어떤지 확인해 봅시다!

```
<!DOCTYPE html>
<html lang="ko">
<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>
  <script>
    alert("헤드 그런데 js 파일 링크 위")
  </script>
  <script src="./index.js"></script>
  <script>
    alert("헤드 그런데 js 파일 링크 아래")
  </script>
</head>
<script>
  alert("헤드랑 바디 사이");
</script>
<body>
  <h1>hello, Js World!</h1>
  <script>
    alert("바디 안쪽");
  </script>
</body>
<script>
  alert("바디 아래");
</script>
</html>
```

index.html

```
alert("링크방식");
```

index.js

읽기 순서!

- 말 그대로 읽히는 순서에 따라서 작동 합니다!!

JS 기초!

표기법

dash-case (kebab-case)

snake_case

camelCase

PascalCase

tomakeyoufeelmyloveknowmetoowell

dash-case(kebab-case)

to-make-you-feel-my-love-know-me-too-well



snake_case

to_make_you_feel_my_love_know_me_too_well



camelCase

toMakeYouFeelMyLoveKnowMeTooWell



PascalCase

ToMakeYouFeelMyLoveKnowMeTooWell

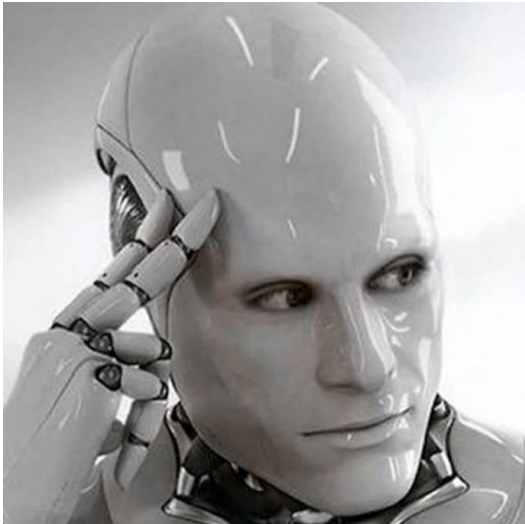


Zero-based Numbering

- 프로그래밍에서는 0 기반으로 번호를 매긴다!
- 특수한 경우를 제외하고는 0부터 숫자를 시작함



1 2 3 4 5 6 7 8 9 10 ~



0 1 2 3 4 5 6 7 8 9 ~

```
let fruits = ['Apple', 'Banana', 'Cherry'];  
  
console.log(fruits[0]); // 'Apple'  
console.log(fruits[1]); // 'Banana'  
console.log(fruits[2]); // 'Cherry'
```


JavaScript 주석

```
// 한 줄 메모  
/* 한 줄 메모 */  
  
/**  
 * 여러 줄  
 * 메모1  
 * 메모2  
 */
```

JavaScript 변수

JavaScript 변수

Identifier

Memory

myNumber



Address	Value
0012CCGWH80	23

JavaScript 변수

1) 선언



JavaScript ▾

```
let 변수이름;
```

```
var 변수이름;
```

2) 할당

```
const 변수이름 = 값;
```

```
let 변수이름 = 값;
```

```
var 변수이름 = 값;
```

var & let & const

```
var 변수이름;
```

- 1) 선언 단계와 초기화가 동시에 이루어지며 아무것도 할당하지 않으면 자동으로 **undefined** 가 할당
- 2) **중복 선언** 가능. **재선언** 가능

var & let & const

```
let 변수이름;
```

- 1) 변수 **중복 선언이 불가능**하지만, **재할당 가능**
- 2) var 과 마찬가지로 선언을 하지 않으면 자동으로 undefined가 들어간다.

var & let & const

```
const 변수이름 = 값;
```

- 1) 초반에 선언할 때 반드시 초기화를 동시에 진행해야 한다.
- 2) 재선언 불가능, 재할당 불가능

잠깐 퀴즈!

1. 다음 중 올바른 변수 이름은?

A) my_variable

B) 3rd_number

C) \$money

D) function

2. 다음 중 올바른 변수 이름은?

A) user-name

B) _userAge

C) 123var

D) return

3. 다음 중 올바른 변수 이름은?

A) class

B) my_variable_1

C) special!char

D) true_false

변수명 Tip!

- 변수 이름은 읽기 쉽도록 센스 있게!

```
// 변수 이름은 읽기 쉽도록 센스 있게!  
let a = 1;  
let b = "길동";  
  
let userNumber = 1;  
let userName = "길동";
```

변수명 Tip2!

- 상수는 대문자로 선언해서 다른 개발자도 알 수 있게 하기!

```
// 상수는 대문자로 선언해서 다른 개발자도 알 수 있게 해주세요
```

```
const WIDTH = 1100;  
const CODINGON = "codingon";
```

이름에도 규칙이 있다?

알고 쓰는 네이밍 컨벤션 A to Z

프로그래밍 코딩컨벤션 네이밍컨벤션

프로그래머가 가장 힘들어하는 일



Quora 및 Ubuntu 포럼에서 진행된 토론 스레드에 따르면 토론에 응답한 개발자 49%가 이름 짓는 걸 가장 어려운 작업으로 꼽았다고 해요. ©Phil Johnson/ITworld (번역)