

# 자바스크립트란?

2학년 권도윤



JavaScript

# 목 차

- 자바스크립트 역할
- 자바스크립트의 변수선언
- 자바스크립트의 메소드
- 자바스크립트만의 등가비교 및 함수
- 이벤트

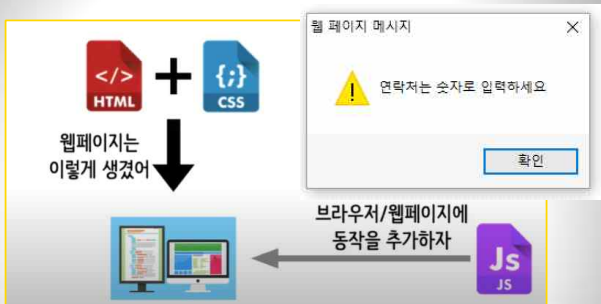


# 자바스크립트란?

- 자바스크립트(**java**script)는 프로그래밍언어
- js는 서버 개발, 어플리케이션 개발 등 다양한 목적으로 사용가능
- 하지만, 주로 **웹 개발**에 사용
- 필요한 태그는 **<script></script>**



# 자바스크립트의 역할



웹 브라우저가 가진 기능을 실행 시키거나 HTML/CSS를 통해  
렌더링된 화면을 조작할 수 있다.

# 자바스크립트 변수선언

## □ var, let, const가 존재

```
var name = 'bathingape'  
console.log(name) // bathingape  
  
var name = 'javascript'  
console.log(name) // javascript
```

□ var는 같은 변수 선언을 할 때, 값을 다르게 하면 각각 다른 값이 출력

□ 코드량이 많아 진다면 어디에서 어떻게 사용 될지도 파악하기 힘들뿐

더러 값이 바뀔 수도 있음



# 자바스크립트 변수선언

```
let name = 'bathingape'  
console.log(name) // bathingape  
  
let name = 'javascript'  
console.log(name)  
// Uncaught SyntaxError: Identifier 'name' has already been declared  
  
name = 'react'  
console.log(name) //react
```

```
const name = 'bathingape'  
console.log(name) // bathingape  
  
const name = 'javascript'  
console.log(name)  
// Uncaught SyntaxError: Identifier 'name' has already been declared  
  
name = 'react'  
console.log(name)  
//Uncaught TypeError: Assignment to constant variable.
```



# 메소드-window.prompt()

□ 사용자로부터 문자열을 입력받을 수 있는 다이얼로그 박스를

열어주는 메소드(메소드에서 window생략 가능)

□ console.log()로 콘솔창에 결과값 출력

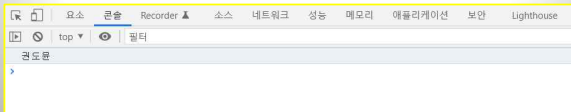
```
<script>  
  const promptMessage = "당신의 이름은 무엇입니까?";  
  const result = prompt(promptMessage);  
  console.log(result);  
</script>
```



# 예시

127.0.0.1:5500 내용:

당신의 이름은 무엇입니까?





# 메소드-window.document

□ 현재 브라우저에 렌더링되고 있는 문서를 의미하며, 이 속성

을 이용하면 해당 문서에 접근가능

□ 대표적인 document메소드 :

`querySelector, getElementById, textContent`



# document.querySelector

- html의 선택자를 인자로 전달받아, 그 선택자와 일치하는 문서 내 첫번째 요소를 반환한다.



# 예시

```
<body>
  <h1>큰 제목 h1</h1>
  <p>문단 p</p>
  <p id="text">아이디 문단 ID : text</p>
  <script>
    console.log(document.querySelector("h1"));
    console.log(document.querySelector("p"));
    console.log(document.querySelector("#text"));
  </script>
</body>
```

큰 제목 h1

문단 p

아이디 문단 ID : text

요소 콘솔 Recorder 소스

top 필터

```
<h1>큰 제목 h1</h1>
<p>문단 p</p>
<p id="text">아이디 문단 ID : text</p>
```



# textContent

- 해당 노드가 포함하고 있는 텍스트 콘텐츠를 표현하는 속성
- textContent를 통해 요소가 포함한 텍스트를 읽을 수도, 변경할 수도있다.

```
// p 요소를 반환받아 상수로 이름을 붙인다!  
const p = document.querySelector("p");  
  
// p 요소의 textContent 속성을 콘솔에 출력해보자!  
console.log(p.textContent)  
  
// p 요소의 textContent 값을 변경해보자!  
p.textContent = "텍스트를 이걸로 바꿔!"
```



# js의 등가비교

자바스크립트의 반복문(for, while문), 조건문(if)는 c언어형태와 같다.

| 의미        | 연산자 | 사용 형태   |
|-----------|-----|---------|
| 같다        | ==  | A == B  |
| 같지 않다     | !=  | A != B  |
| 완전히 같다    | === | A === B |
| 완전히 같지 않다 | !== | A !== B |

```
console.log('1' == 1) // true  
console.log('1' === 1) // false
```



# js의 함수구조

## □ 함수 선언식

```
function 함수명(){  
}
```

## □ 함수 표현식

```
const 함수명 = function(){  
}
```



# 이벤트

- 웹페이지에서 마우스를 클릭했을 때
  - 키를 입력했을 때
  - 특정요소에 포커스가 이동되었을 때 등
- >어떤 사건을 발생시키는 것



# 이벤트-기본 형태

- 이벤트가 발생할 수 있는 타겟을 선택하고, 이벤트 핸들러(자가 만든 함수) 속성에 이벤트 핸들러를 대입한다.

타겟.on이벤트명 = 이벤트핸들러함수



button.onclick = handClick



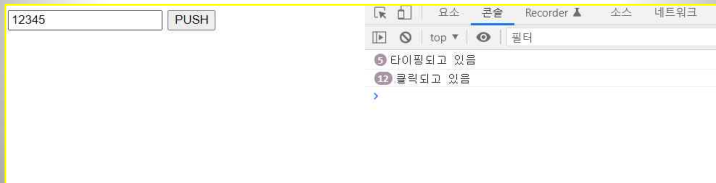


# 실습

```
<body>
  <!--텍스트입력칸 생성-->
  <input type="text" id="typing" />
  <!--버튼생성-->
  <input type="button" id="push" value="PUSH" />
  <script>
    const inputType = document.querySelector("#typing"); //입력값을 받을 inputType변수생성
    const inputClick = document.querySelector("#push"); //입력값을 받을 inputClick변수생성
    const handleTyping = function () {
      console.log("타이핑되고 있음");
    }; //Typing핸들링함수 생성
    const handleClick = function () {
      console.log("클릭되고 있음");
    }; //Click핸들링함수 생성
    inputType.onkeydown = handleTyping; //키가 눌리면 Typing핸들링함수실행
    inputClick.onclick = handleClick; //버튼이 눌리면 Click핸들링함수실행
  </script>
</body>
```



# 결과



# 앞으로의 계획 및 Q & A

