

JavaScript

Part I

목차

1. JavaScript 기본 문법

- 변수와 상수
- 데이터 타입
- 연산자
- 조건문과 반복문
- 함수
- 객체와 배열

2. DOM과 이벤트 처리

- Document Object Model 개념
- DOM 조작하기
- 이벤트 처리방법

3. JavaScript 실습

- 따라하기
- 아무거나 만들기

Web = HTML + CSS + JavaScript

JavaScript

웹 페이지를 동적으로 만들고, 사용자와 상호작용할 수 있는 다양한 기능을 제공하기 위해 만들어진 언어로, 웹 표준으로 정해진 언어이므로 **사실상 웹 프론트엔드 개발의 표준 언어**

웹 페이지의 요소들을 조작하고 변경할 수 있는 기능을 제공하며, Node.js 런타임을 통해 서버 측에서도 사용될 수 있음

자바스크립트는 객체 기반의 언어로, 함수형 프로그래밍과 객체 지향 프로그래밍 모두를 지원함. 정확히는 프로토타입 기반 언어로 클래스 기반 언어가 아니지만, 클래스 기반의 언어처럼 사용할 수 있도록 문법을 지원함

ECMAScript

ECMAScript는 JavaScript의 표준 규격이며, JavaScript는 ECMAScript를 구현한 프로그래밍 언어 중 하나

TypeScript

Microsoft에서 개발한 자바스크립트의 상위 집합 언어로, 정적 타입 시스템을 제공하여 코드의 안정성과 가독성을 높인 언어

WASM

브라우저에서 실행되는 소프트웨어의 성능과 기능성을 대폭 향상시키는 새로운 기술로, C/C++ 등의 언어로 작성된 코드를 브라우저에서 실행할 수 있게 해주는 기술

변수와 상수

변수 : let

상수 : const

```
// 변수, 상수
let myVariable = 123;
const myConstant = "Hello, world!";

console.log(myVariable); // 123
console.log(myConstant); // Hello, world!
```

데이터 타입

```
// 데이터 타입
let myNumber = 123;
let myString = "Hello, world!";
let myBoolean = true;
let myNull = null;
let myUndefined = undefined;
let myObject = { name: "John", age: 30 };
let mySymbol = Symbol("mySymbol");

console.log(typeof myNumber); // number
console.log(typeof myString); // string
console.log(typeof myBoolean); // boolean
console.log(typeof myNull); // object (이는 JavaScript의 버그)
console.log(typeof myUndefined); // undefined
console.log(typeof myObject); // object
console.log(typeof mySymbol); // symbol
```

연산자

```
// 연산자
let x = 10;
let y = 5;

console.log(x + y); // 15
console.log(x - y); // 5
console.log(x * y); // 50
console.log(x / y); // 2
console.log(x % y); // 0

console.log(x > y); // true
console.log(x < y); // false
console.log(x >= y); // true
console.log(x <= y); // false
console.log(x === y); // false
console.log(x !== y); // true

let a = true;
let b = false;

console.log(a && b); // false
console.log(a || b); // true
console.log(!a); // false
```


1 == "1" ➡ True

1 === "1" ➡ False

조건문: if

```
// 조건문
let age = 20;

if (age >= 18) {
  console.log("You are an adult.");
} else if (age >= 13) {
  console.log("You are a teenager.");
} else {
  console.log("You are a child.");
}
```

조건문: switch

```
// switch문
let fruit = "banana";

switch (fruit) {
  case "apple":
    console.log("This is an apple.");
    break;
  case "banana":
    console.log("This is a banana.");
  case "orange":
    console.log("This is an orange.");
    break;
  default:
    console.log("Unknown fruit.");
}
```

반복문

```
// for문  
for (let i = 0; i < 5; i++) {  
  console.log(i);  
}
```

```
// while문  
let i = 0;  
  
while (i < 5) {  
  console.log(i);  
  i++;  
}
```

반복문: 배열 순회

```
// 배열 순회
const arr = [1, 2, 3, 4, 5];

for (let i = 0; i < arr.length; i++) {
  console.log(arr[i]);
}

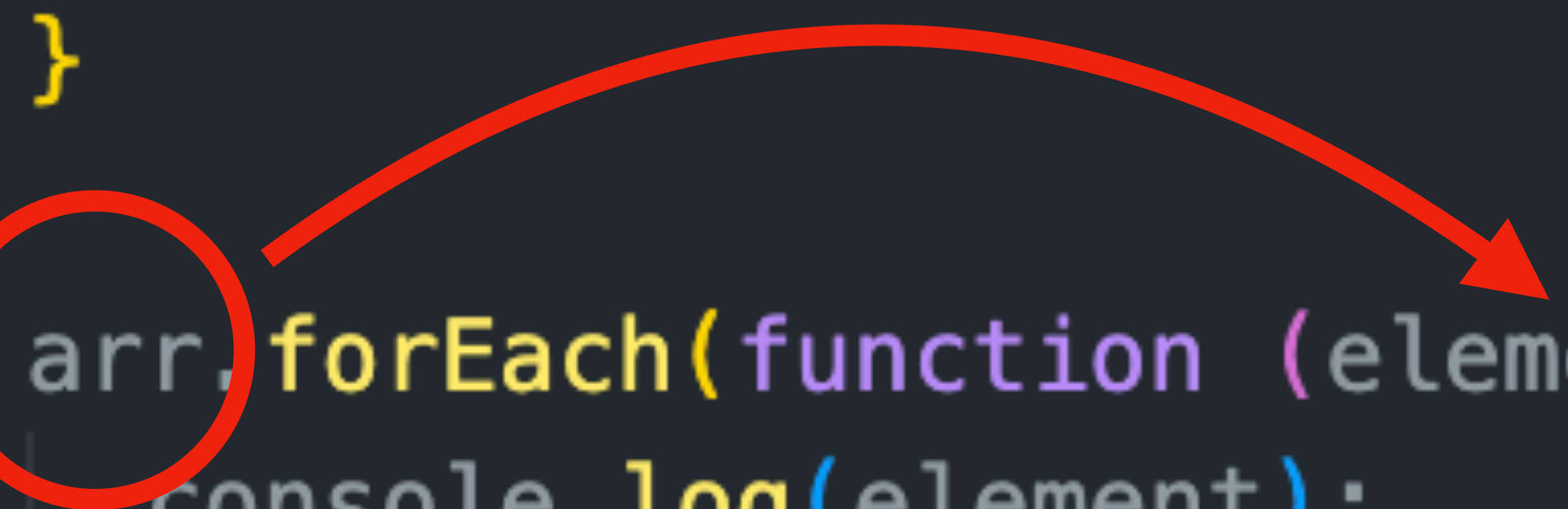
arr.forEach(function (element) {
  console.log(element);
});
```

반복문: 배열 순회

```
// 배열 순회
const arr = [1, 2, 3, 4, 5];

for (let i = 0; i < arr.length; i++) {
  console.log(arr[i]);
}

arr.forEach(function (element) {
  console.log(element);
});
```



함수

```
// 함수  
function add(a, b) {  
  return a + b;  
}  
  
let result = add(1, 2);  
  
console.log(result); // 3
```

객체

```
// 객체
```

```
let myObject2 = {  
  name: "John",  
  age: 30,  
  isMarried: false,  
};
```

) **JSON**

```
console.log(myObject2.name); // John  
console.log(myObject2.age); // 30  
console.log(myObject2.isMarried); // false
```


배열

```
// 배열  
let myArray = ["apple", "banana", "orange"];  
  
console.log(myArray[0]); // apple  
console.log(myArray[1]); // banana  
console.log(myArray[2]); // orange
```

DOM 핸들링

```
// HTML 요소 선택하기
const myParagraph = document.getElementById("my-paragraph");
const myButton = document.getElementById("my-button");

// HTML 요소 속성 변경하기
myParagraph.innerHTML = "This is a new paragraph.";
myParagraph.style.color = "red";

// 이벤트 처리 방법
myButton.addEventListener("click", function () {
    alert("Button clicked!");

    console.log(event.target); // 이벤트가 발생한 HTML 요소
    console.log(event.clientX); // 마우스 클릭 지점의 x 좌표
    console.log(event.clientY); // 마우스 클릭 지점의 y 좌표
});
```

DOM 핸들링

```
// HTML 요소 선택하기
const myParagraph = document.getElementById("my-paragraph");
const myButton = document.getElementById("my-button");

// HTML 요소 속성 변경하기
myParagraph.innerHTML = "This is a new paragraph.";
myParagraph.style.color = "red";

// 이벤트 처리 방법
myButton.addEventListener("click",
    alert("Button clicked!"));

console.log(event.target); // 이벤트가 발생한 HTML 요소
console.log(event.clientX); // 마우스 클릭 지점의 x 좌표
console.log(event.clientY); // 마우스 클릭 지점의 y 좌표
});
```



dom.html:24
<button id="my-button">Click me!</button>

57 dom.html:25

178 dom.html:26

DOM = Document Object Model

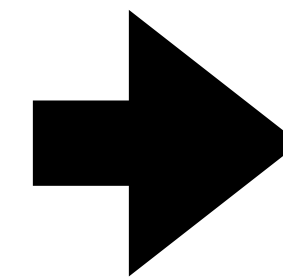
DOM

DOM(Document Object Model)이란, HTML, XML 등의 문서를 계층 구조로 표현한 모델. 즉, HTML 문서의 모든 요소들을 객체로 나타낸 것.

이 객체들은 계층 구조로 이루어져 있으며, 상위 요소에서 하위 요소로 내려갈수록 구체화되어 있음.

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>My Page</title>
  </head>
  <body>
    <h1>Welcome to my page!</h1>
    <p>This is a sample paragraph.</p>
    <ul>
      <li>Item 1</li>
      <li>Item 2</li>
      <li>Item 3</li>
    </ul>
  </body>
</html>
```



DOM

document

└─ html

└─ head

└─ meta

└─ title

└─ body

└─ h1

└─ p

└─ ul

└─ li

└─ li

└─ li

요소 선택

```
<html>
  <body>
    <div id="my-div" class="my-class">
      <p>This is a sample paragraph.</p>
    </div>
    <script>
      let myDiv = document.getElementById("my-div");
      let myClass = document.getElementsByClassName("my-class");
      let myTag = document.getElementsByTagName("p");

      console.log(myDiv); // <div id="my-div" class="my-class">...</div>
      console.log(myClass); // HTMLCollection [ <div id="my-div" class="my-class">...</div> ]
      console.log(myTag); // HTMLCollection [ <p>This is a sample paragraph.</p> ]
    </script>
  </body>
</html>
```

클래스 삭제 및 스타일 변경

```
<html>
  <body>
    <div id="my-div" class="my-class">
      <p>This is a sample paragraph.</p>
    </div>
    <script>
      let myDiv = document.getElementById("my-div");

      myDiv.setAttribute("title", "My Div");
      myDiv.removeAttribute("class");

      myDiv.style.color = "red";
      myDiv.style.backgroundColor = "yellow";
    </script>
  </body>
</html>
```


HTML 요소 추가 및 삭제

```
<html>
  <body>
    <div id="my-div" class="my-class">
      <p>This is a sample paragraph.</p>
      <span>This is a sample span.</span>
    </div>
    <script>
      let myDiv = document.getElementById("my-div");
      let mySpan1 = document.createElement("span");

      mySpan1.innerHTML = "This is a sample span 1.";
      myDiv.appendChild(mySpan1);

      let mySpan = myDiv.getElementsByTagName("span")[0];
      myDiv.removeChild(mySpan);
    </script>
  </body>
</html>
```

Event

이벤트(Event)란, 사용자가 버튼을 클릭하거나, 페이지가 로드될 때, 마우스 커서를 요소 위로 올릴 때 등 HTML 문서 내에서 발생하는 다양한 상황을 말함

이벤트의 종류에는 마우스 이벤트, 키보드 이벤트, 폼 이벤트, 터치 이벤트, 멀티미디어 이벤트 등 다양한 이벤트가 있음

Event

```
<html>
  <body>
    <button id="my-button">Click me!</button>

    <script>
      let myButton = document.getElementById("my-button");

      myButton.addEventListener("click", function () {
        console.log("Button clicked!");
      });
    </script>
  </body>
</html>
```

Event 종류

마우스 이벤트

- ✓ **click**: 요소 클릭
- ✓ **dblclick**: 요소 더블 클릭
- ✓ **mousedown**: 요소 위에서 마우스 버튼을 누르고 있을 때
- ✓ **mouseup**: 요소 위에서 마우스 버튼을 눌렀다 놓았을 때
- ✓ **mousemove**: 요소 위에서 마우스를 움직일 때

키보드 이벤트

- ✓ **keydown**: 키를 눌렀을 때
- ✓ **keyup**: 키를 눌렀다 놓았을 때
- ✓ **keypress**: ASCII 문자를 입력할 때

폼 이벤트

- ✓ **submit**: 폼을 제출할 때
- ✓ **reset**: 폼을 초기화할 때
- ✓ **change**: 폼 요소의 값이 변경될 때

실습

기본 문법 따라하기

전부 직접 타이핑 해서 따라 해 보고 결과를 본다. (인스펙터 혹은 터미널에서 실행)

DOM 예제 따라하기

<https://bit.ly/dom-practice-1>

각자 실습

1. 지난주에 만들었던 프로젝트 안에서 새 파일을 만들어서 시작한다
2. 구구단 프로그램을 최대한 AI의 도움 없이 만들어본다
3. 계산기를 AI와 함께 만들어본다
4. 작성한 파일을 커밋한 후 GitHub에 Push한다. (지난주와 동일한 레포에 업데이트)
5. 레포 주소 제출 🖱️ <https://github.com/SNU-WP/basecamp/issues/3>

* 프로그램을 만들고 나서, 꼭 한 번 이상 코드의 일부를 직접 수정하고 실행해본다.