

1. JavaScript 소개

1.1 JavaScript란?

JavaScript는 웹 페이지에 동적인 기능을 추가하는 데 사용되는 스크립팅 언어입니다. HTML과 CSS가 웹 페이지의 구조와 스타일을 정의한다면, JavaScript는 사용자와 상호작용하고, 데이터를 처리하며, 페이지의 동작을 제어할 수 있게 해줍니다.

1.2 JavaScript 실행 환경

JavaScript는 웹 브라우저 내에서 실행됩니다. 모든 최신 브라우저는 JavaScript 엔진을 포함하고 있어, 웹 페이지에서 JavaScript를 직접 실행할 수 있습니다.

1.3 JavaScript 코드 포함 방법

1. HTML 파일에 직접 포함하기:

```
<html>
<head>
  <title>JavaScript 예제</title>
</head>
<body>
  <h1>JavaScript 테스트</h1>
  <script>
    alert('Hello, JavaScript!');
  </script>
</body>
</html>
```

- `<script>` 태그 안에 JavaScript 코드를 직접 작성할 수 있습니다.

2. 외부 파일로 포함하기:

```
<html>
<head>
  <title>JavaScript 예제</title>
</head>
<body>
  <h1>JavaScript 테스트</h1>
  <script src="script.js"></script>
</body>
</html>
```

- `script.js` 파일을 만들어 JavaScript 코드를 외부 파일로 분리할 수 있습니다.

2. JavaScript 기본 문법

2.1 변수

변수는 데이터를 저장하는 데 사용됩니다. JavaScript에서는 `var`, `let`, `const` 를 사용하여 변수를 선언합니다.

```
let message = "Hello, World!";  
const PI = 3.14;  
var age = 30;
```

- `let` : 블록 스코프 변수를 선언할 때 사용합니다.
- `const` : 값이 변경되지 않는 상수를 선언할 때 사용합니다.
- `var` : 함수 스코프 변수를 선언할 때 사용합니다.

2.2 데이터 타입

JavaScript에는 다양한 데이터 타입이 있습니다.

- 문자열(String): `"Hello"`
- 숫자(Number): `42`
- 불리언(Boolean): `true` 또는 `false`
- 배열(Array): `[1, 2, 3]`
- 객체(Object): `{ name: "John", age: 30 }`
- null: `null`
- undefined: `undefined`


```
let name = "Alice";  
let age = 25;  
let isStudent = true;  
let hobbies = ["reading", "gaming", "coding"];  
let person = { name: "Bob", age: 30 };
```

2.3 연산자

JavaScript는 다양한 연산자를 제공합니다.

- 산술 연산자: `+`, `-`, `*`, `/`, `%`
- 비교 연산자: `==`, `===`, `!=`, `!==`, `>`, `<`, `>=`, `<=`
- 논리 연산자: `&&`, `||`, `!`

```
let sum = 10 + 5; // 15
let isEqual = (10 === 5); // false
let isAdult = (age >= 18); // true
let isLoggedIn = true && false; // false
```

2.4 조건문

조건문을 사용하여 특정 조건에 따라 코드가 실행되도록 할 수 있습니다.

```
let age = 20;

if (age >= 18) {
    console.log("성인입니다.");
} else {
    console.log("미성년자입니다.");
}
```

2.5 반복문

반복문을 사용하여 코드 블록을 여러 번 실행할 수 있습니다.

- for문:

```
for (let i = 0; i < 5; i++) {  
    console.log("숫자: " + i);  
}
```

- while문:

```
let i = 0;
while (i < 5) {
  console.log("숫자: " + i);
  i++;
}
```

2.6 함수

함수는 재사용 가능한 코드 블록입니다. `function` 키워드를 사용하여 함수를 정의할 수 있습니다.

```
function greet(name) {  
    return "Hello, " + name;  
}  
  
let message = greet("Alice");  
console.log(message); // "Hello, Alice"
```

2.7 이벤트 처리

JavaScript는 사용자와의 상호작용을 처리하기 위해 이벤트를 다룰 수 있습니다. 예를 들어, 버튼을 클릭했을 때 특정 작업을 수행할 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
  <title>이벤트 처리 예제</title>
</head>
<body>

<button onclick="sayHello()">Click me</button>

<script>
  function sayHello() {
    alert("Hello, World!");
  }
</script>

</body>
</html>
```


2.8 DOM 조작

JavaScript는 문서 객체 모델(DOM)을 조작하여 웹 페이지의 내용을 동적으로 변경할 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
  <title>DOM 조작 예제</title>
</head>
<body>

  <p id="demo">이 문장을 변경할 것입니다.</p>
  <button onclick="changeText()">Change Text</button>

  <script>
    function changeText() {
      document.getElementById("demo").innerHTML = "텍스트가 변경되었습니다!";
    }
  </script>

</body>
</html>
```

2.9 배열과 객체

배열과 객체는 데이터를 구조화하는 데 사용됩니다.

- 배열:

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

- 객체:

```
let person = {  
  firstName: "John",  
  lastName: "Doe",  
  age: 30  
};  
console.log(person.firstName); // "John"
```

2.10 JSON

JSON(JavaScript Object Notation)은 데이터를 표현하기 위한 경량 포맷입니다. 객체와 유사한 구조를 가지고 있으며, 주로 서버와의 데이터 교환에 사용됩니다.

```
let jsonData = '{"name": "Alice", "age": 25}';  
let obj = JSON.parse(jsonData); // JSON 문자열을 객체로 변환  
console.log(obj.name); // "Alice"
```

3. 예제 프로젝트: 간단한 계산기 만들기

이제 간단한 예제를 통해 JavaScript의 기본 개념을 적용해 보겠습니다. HTML, CSS, JavaScript를 사용하여 간단한 계산기를 만들어보겠습니다.

3.1 HTML 파일

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>간단한 계산기</title>
</head>
<body>

  <h1>간단한 계산기</h1>

  <input type="text" id="num1" placeholder="숫자 1">
  <input type="text" id="num2" placeholder="숫자 2">

  <button onclick="add()">더하기</button>
  <button onclick="subtract()">빼기</button>
  <button onclick="multiply()">곱하기</button>
  <button onclick="divide()">나누기</button>

  <p id="result"></p>

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

</body>
</html>
```

3.2 JavaScript 파일 (calculator.js)

```
function add() {
    let num1 = parseFloat(document.getElementById("num1").value);
    let num2 = parseFloat(document.getElementById("num2").value);
    let result = num1 + num2;
    document.getElementById("result").innerHTML = "결과: " + result;
}

function subtract() {
    let num1 = parseFloat(document.getElementById("num1").value);
    let num2 = parseFloat(document.getElementById("num2").value);
    let result = num1 - num2;
    document.getElementById("result").innerHTML = "결과: " + result;
}

function multiply() {
    let num1 = parseFloat(document.getElementById("num1").value);
    let num2 = parseFloat(document.getElementById("num2").value);
    let result = num1 * num2;
    document.getElementById("result").innerHTML = "결과: " + result;
}

function divide() {
    let num1 = parseFloat(document.getElementById("num1").value);
    let num2 = parseFloat(document.getElementById("num2").value);
    if (num2 === 0) {
        document.getElementById("result").innerHTML = "0으로 나눌 수 없습니다.";
    } else {
        let result = num1 / num2;
        document.getElementById("result").innerHTML = "결과: " + result;
    }
}
```

4. 결론

이 튜토리얼에서는 JavaScript의 기본 개념과 간단한 예제 프로젝트를 다뤘습니다. JavaScript는 매우 강력한 언어이며, 이 기본기를 다진 후 더 복잡한 기능들을 추가해가며 실력을 향상시킬 수 있습니다. 추가 학습을 통해 이벤트 처리, DOM 조작, 비동기 프로그래밍, API 연동 등을 더 깊이 이해할 수 있습니다. JavaScript로 웹 페이지를 더 동적이고 상호작용적으로 만들어 보세요!