

JavaScript란?

『자바스크립트(JavaScript)는 웹 개발에서 가장 널리 사용되는 프로그래밍 언어 중 하나
HTML과 CSS와 함께 웹사이트를 구성하는 핵심 요소이며, 웹 페이지를 동적으로 만들고,
사용자와 상호작용 한다.』

웹 환경에서 JavaScript의 목적 》 HTML 조작과 변경

자바스크립트의 특징

☒ 인터프리터 언어

코드를 한 줄씩 읽고 실행하는 방식(컴파일 과정 필요 없음).
코드 수정 후 바로 실행 가능.

☒ 동적 타입(Dynamically Typed)

변수의 타입을 명시적으로 선언할 필요 없음.
실행 중에 변수의 타입이 변경될 수 있음.

☒ 객체 기반(Object-Oriented)

기본적으로 객체(Object) 단위로 구성.
객체를 생성하고 조작하는 것이 주요한 프로그래밍 방식.

☒ 이벤트 기반(Event-Driven)

버튼 클릭, 마우스 이동 등의 이벤트를 감지하고 실행 가능.
addEventListener() 등을 활용.

☒ 비동기 처리(Asynchronous)

콜백 함수(Callback), 프로미스(Promise), async/await 등을 이용해 비동기 작업 수행 가능.
AJAX, API 호출, 데이터베이스 연동 등에 활용.

언어	실행 환경	컴파일 방식	프레임워크/라이브러리
JavaScript	웹 브라우저(Chrome) + Node.js(백엔드)	인터프리터 방식(즉시 실행)	React, Vue, Node.js
C, C++	운영체제(window, Linux 등)에서 컴파일 후 실행	컴파일 후 실행(빠름)	Qt, Boost
Python	인터프리터로 실행(스크립트 언어, 웹/데이터 분석, AI)	인터프리터 방식(즉시 실행)	Django, Flask
Java	JVM(Java Virtual Machine) 위에서 실행 (멀티플랫폼)	바이트코드 변환 후 JVM에서 실행	Spring, Android SDK, Hibernate

1. 변수 선언 및 데이터 타입

JavaScript (동적 타입) : 변수 선언 시 타입을 지정하지 않아도 됨. 실행 중에 변경 가능.

```
let x =10; // 숫자
x ="hello"; // 문자열로 변경 가능 (타입이 동적으로 변함)
console.log(x); // hello
```

C/C++/Java (정적 타입) : 변수 타입을 명확히 선언해야 하며, 실행 중 변경 불가.

```
int x =10;
x ="hello"; // 컴파일 오류 발생 (타입 불일치)
```

Python (동적 타입) : JavaScript 유사

```
x =10
x ="hello" # 가능 (Python도 동적 타입)
print(x) # hello
```

2. 입출력 방식

JavaScript

```
console.log("Hello, JavaScript!");
```

C++

```
#include <iostream>
using namespace std;
int main() {
    cout <<"Hello, C++!"<<endl;
    return 0;
}
```

Python

```
print("Hello, Python!")
```

Java

```
public class Main {
    public static void main(String[] args){
        System.out.println("Hello, Java!");
    }
}
```

3. 조건문 (if-else)

JavaScript

```
let num =10;
if (num >0) {
    console.log("양수입니다.");
}else {
    console.log("음수입니다.");
}
```

C++

```
#include <iostream>
using namespace std;
int main() {
    int num =10;
    if (num >0) {
        cout <<"양수입니다."<<endl;
    }else {
        cout <<"음수입니다."<<endl;
    }
    return 0;
}
```

Python

```
num =10
if num >0:
    print("양수입니다.")
else:
    print("음수입니다.")
```

4. 반복문 (for)

JavaScript

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

C++

```
for (int i =0; i <5; i++) {
    cout <<i <<endl;
}
```

Python

```
for i in range(5):
    print(i)
```

5. 함수 선언 및 호출

JavaScript - 익명 함수 + 화살표 함수

```
// 기본 함수
function greet(name){
    return "Hello, "+name;
}

// 화살표 함수 (ES6+)
const greetArrow = (name) => `Hello, ${name}`;
console.log(greet("Alice"));
console.log(greetArrow("Bob"));
```

C++

```
#include <iostream>
using namespace std;
string greet(string name) {
    return "Hello, "+name;
}
int main() {
    cout << greet("Alice") << endl;
    return 0;
}
```

Python

```
def greet(name):
    return "Hello, "+name
print(greet("Alice"))
```

6. 객체지향 프로그래밍(OOP)

JavaScript (프로토타입 기반 객체지향)

```
function Person(name,age){
    this.name =name;
    this.age =age;
}
Person.prototype.greet =function(){
    console.log(`Hello, my name is ${this.name}`);
};
let p1 =new Person("Alice",25);
p1.greet();
```

C++ (클래스 기반)

```
#include <iostream>
using namespace std;
class Person {
public:
    string name;
    int age;
    Person(string name,int age){
        this->name =name;
        this->age =age;
    }
    void greet(){
        cout <<"Hello, my name is "<<name <<endl;
    }
};
int main() {
    Person p1("Alice",25);
    p1.greet();
    return 0;
}
```

Python (클래스 기반)

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age
    def greet(self):
        print(f"Hello, my name is {self.name}")
p1 =Person("Alice",25)
p1.greet()
```

1. 동적 기능 (Dynamic Feature)

- 실행 시간(Runtime)에 결정되거나 변경되는 기능을 의미
- 동적 메모리 할당, 동적 타입, 동적 함수 생성 등이 포함
- 자바스크립트처럼 런타임에 코드 실행이 자유로운 언어가 대표적

```
function dynamicFunction(){  
    return function(){  
        console.log("This is a dynamically created function!");  
    };  
}  
  
let func = dynamicFunction(); // 실행 중에 새로운 함수 생성  
func(); // "This is a dynamically created function!" 출력
```

dynamicFunction()이 실행되면 새로운 함수가 만들어짐 → 동적 기능.

2. 정적 기능 (Static Feature)

- 컴파일 시간(Compile-time)에 결정되는 기능
- 변수, 데이터 타입, 함수 등이 고정
- C++처럼 정적 타입 언어가 대표적

```
#include <iostream>  
using namespace std;  
class MyClass {  
public:  
    static void staticFunction() {  
        cout << "This is a static function!" << endl;  
    }  
};  
  
int main() {  
    MyClass::staticFunction(); // 클래스 이름을 통해 정적 함수 호출  
    return 0;  
}
```

staticFunction()은 클래스에 소속된 정적 함수 → 실행 중에 변경되지 않음.

HTML과 출력 하는 연동 과정 예시

-코드 예시 / 랜덤 명언 생성기 (Html + CSS + JavaScript)

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>랜덤 명언 생성기</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      margin-top: 50px;
      background-color: #f4f4f4;
    }
    #quote-box {
      font-size: 24px;
      margin: 20px;
      padding: 20px;
      background: white;
      display: inline-block;
      border-radius: 10px;
      box-shadow: 0px 4px 8px rgba(0,0,0,0.2);
    }
    button {
      font-size: 18px;
      padding: 10px 20px;
      border: none;
      background: #007bff;
      color: white;
      cursor: pointer;
      border-radius: 5px;
      margin-top: 20px;
    }
    button:hover {
      background: #0056b3;
    }
  </style>
</head>
<body>
  <h1> 랜덤 명언 생성기 </h1>
  <div id="quote-box">버튼을 눌러 명언을 확인하세요!</div>
  <br>
  <button onclick="generateQuote()">새로운 명언 보기</button>
  <script>
    function generateQuote(){
      const quotes = [
        "삶이 있는 한 희망은 있다. - 키케로",
        "최고에 도달하려면 최저에서 시작하라. - P.시루스",
        "행복은 습관이다. 그것을 몸에 지너라. - 허버드",
        "길을 잃는다는 것은 곧 길을 알게 된다는 것이다. - 동아프리카 속담",
        "문제점을 찾지 말고 해결책을 찾으라. - 헨리 포드",
        "인생을 다시 살다면 다음번에는 더 많은 실수를 저지르리라. - 나딘 스테어",
        "한 번의 실패와 영원한 실패를 혼동하지 마라. - F.스콧 피츠제럴드"
      ];
      // 랜덤으로 명언 선택
      let randomIndex =Math.floor(Math.random() *quotes.length);
      let randomQuote =quotes[randomIndex];
      // HTML 요소 업데이트
      document.getElementById("quote-box").innerHTML =randomQuote;
    }
  </script>
</body>
</html>
```

분석 -

HTML (구조)

```
<h1> 랜덤 명언 생성기 </h1>
<div id="quote-box">버튼을 눌러 명언을 확인하세요!</div>
<br>
<button onclick="generateQuote()">새로운 명언 보기</button>
```

`<h1>` : 제목 표시.

`<div id="quote-box">` : 명언을 표시하는 영역.

`<button>` : 버튼을 누르면 `generateQuote()` 함수를 실행.

JavaScript 동작 흐름 분석

```
function generateQuote(){
  const quotes = [
    "삶이 있는 한 희망은 있다. - 키케로",
    "최고에 도달하려면 최저에서 시작하라. - P.시루스",
    "행복은 습관이다. 그것을 몸에 지녀라. - 허버드",
    "길을 잃는다는 것은 곧 길을 알게 된다는 것이다. - 동아프리카 속담",
    "문제점을 찾지 말고 해결책을 찾으라. - 헨리 포드",
    "인생을 다시 산다면 다음번에는 더 많은 실수를 저지르리라. - 나딘 스테어",
    "한 번의 실패와 영원한 실패를 혼동하지 마라. - F.스콧 피츠제럴드"
  ];
  // 랜덤으로 명언 선택
  let randomIndex = Math.floor(Math.random() * quotes.length);
  let randomQuote = quotes[randomIndex];
  // HTML 요소 업데이트
  document.getElementById("quote-box").innerHTML = randomQuote;
}
```

1. `generateQuote()` 함수가 호출됨 (버튼을 클릭할 때 실행됨).

2. 명언 배열 `quotes`를 생성

3. 랜덤 인덱스 생성

```
let randomIndex = Math.floor(Math.random() * quotes.length);
```

`quotes.length` → 배열의 길이 (현재는 7개).

`Math.random()` → 0 이상 1 미만의 랜덤 숫자 생성.

`Math.random() * quotes.length` → 0~6 사이의 랜덤 실수 생성.

`Math.floor()` → 소수점 이하를 버리고 정수로 변환 (0~6 중 하나가 선택됨).

```
let randomQuote = quotes[randomIndex];
```

`randomIndex`에 해당하는 배열 요소를 가져옴.

```
document.getElementById("quote-box").innerHTML = randomQuote;
```

`document.getElementById("quote-box")` → `<div id="quote-box">` 요소를 찾음.

`.innerHTML = randomQuote;` → 선택된 명언을 `<div>` 안에 표시함.

자바스크립트 강의

JS1.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>

  <h2 id="hello">안녕하세요</h2>
  <script>
    document.getElementById("hello").innerHTML = '안녕';
    //document : 웹 문서
    //document.getElementById >> . >> ~의, ~안에 있는
    //getElementById : HTML 요소를 가져와주세요
    //getElementById("hello") : ID hello인 HTML 요소 가져와주세요
    //innerHTML : 내부글자
  //해석 : 문서 안에 있는 ID hello 인 것을 가져와요 , 그다음에, 그 안에 있는 HTML 내용을 '안녕'으로 바꿔줘
    document.getElementById("hello").style.color = 'red';
    //style.color : 스타일 컬러
  //해석 : 문서 안에 있는 ID hello 인 것을 가져오고, 그다음에, 그안에 있는 style color를 'red'로 바꿔줘
    document.getElementById("???").???='???';
  // ★★★빈칸만 잘 채우면 HTML의 거의 모든 요소를 뭐든 원하는데로 변경가능 -> JavaScript★★★
  // ??? 포시는 바꿀 수 있는 것이 수백개가 있다 -> 구글검색필요!!!!

  </script>
</body>
</html>
```

document = 웹 문서

.getElementById : HTML 요소를 가져와주세요

.getElementById("hello") : ID hello인 HTML 요소 가져와주세요

.innerHTML = 내부글자

```
document.getElementById("hello").innerHTML = '안녕';
```

해석 : 문서 안에 있는 ID hello 인 것을 가져와요, 그 다음에 그 안에 있는 HTML 내용을 '안녕'으로 바꿔줘요

```
document.getElementById("hello").style.color = 'red';
```

해석 : 문서 안에 있는 ID hello 인 것을 가져오고, 그다음에, 그안에 있는 style color를 'red'로 바꿔줘

```
document.getElementById("???").???='???';
```

???의 빈칸만 잘 채우면 HTML의 거의 모든 요소를 뭐든 원하는데로 변경가능
바꿀 수 있는 수가 수백 개가 넘어서 구글 검색이 필요하다

JS2.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>

  <h2 id="hello">안녕하세요</h2><!-- 글자 크기를 30px로 늘려보세요! -->
  <h2 id="hi">올때메로나</h2><!-- 올때바밤바로 변경해보세요 -->
  <script>
    document.getElementById("hello").innerHTML ='안녕';
    document.getElementById("hello").style.fontSize ="30px";//Size 때문자 !!
    document.getElementById("hi").innerHTML ='올때바밤바';//정답!
    // !!
    // = 은 '같다' 가 아니라 '대입' 이라는 뜻 -> 오른쪽 것을 왼쪽 것으로 대입해주세요! 모든 언어가 동일!!
    // document.getElementById("hello") -> 셀렉터 selector 라고 부른다.

  </script>
</body>
</html>
```

HTML의 h2 id="hello"라는 class를 통해 script 문자열 자료에 대입

```
document.getElementById("hello").innerHTML ='안녕';
```

JS3_Alert_PJ.html // 알람 버튼 만들기 프로젝트

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <link rel="stylesheet" href="JS3.css"><!-- main.css 파일로 링크한다. -->
  <!-- rel="stylesheet"은 CSS 파일을 올바르게 불러오기 위한 필수 속성 값이다. -->
</head>
<body>

  <div class="alert-box" id="alert">
    <p>Alert 박스</p>
    <button onclick="document.getElementById('alert').style.display = 'none';">닫기</button>
  </div>
  <button onclick="document.getElementById('alert').style.display = 'block';">버튼</button>
  <!-- id = "alert" 요소 가져와 display : block 으로 바꿔 기존의 display : none 임-->
  <!-- 실제로는 길게 쓰지 않음. 이럴 때는 function을 사용한다 -->
  <!-- JS조작을 HTML요소 밑에서 해야한다. 위에서 하면 에러가 남
이유는 HTML을 먼저 읽어야 JS로 조작이 가능하기 때문!-->
  <script>

  </script>
</body>
</html>
```

```
.alert-box {
  background:rgb(178,178,255);
  color:rgb(143,143,221);
  padding:20px;
  border-radius:5px;
  display: none;
}
```

JS4_function.html //function의 기능

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <link rel="stylesheet" href="JS4.css"><!-- main.css 파일로 링크한다. -->
  <!-- rel="stylesheet"은 CSS 파일을 올바르게 불러오기 위한 필수 속성 값이다. -->
</head>
<body>

  <div class="alert-box" id="alert">
    <p>Alert 박스</p>
    <button onclick="알림창닫기()">닫기</button>
  </div>
  <button onclick="알림창열기()">버튼</button>
  <script>
    function 알림창열기(){//추약하는기능
      document.getElementById('alert').style.display = 'block';
    }//이제 쓸 때마다 안에 있는 긴 코드가 실행된다. 어떻게 쓰냐 -> 알림창열기()
    //위치 & 선정 예러 null
    // 오타 null 예러
    function 알림창닫기(){
      document.getElementById('alert').style.display = 'none';
    }

  </script>
</body>
</html>
```

```
.alert-box {
  background:rgb(178,178,255);
  color:rgb(143,143,221);
  padding:20px;
  border-radius:5px;
  display: none;
}
```

JS5_parameter.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <link rel="stylesheet" href="JS5.css"><!-- main.css 파일로 링크한다. -->
  <!-- rel="stylesheet"은 CSS 파일을 올바르게 불러오기 위한 필수 속성 값이다. -->
</head>
<body>

  <div class="alert-box" id="alert">
    <p>Alert 박스</p>
    <button onclick="알림창열고닫기('none')">닫기</button>
  </div>
  <button onclick="알림창열고닫기('block')">버튼</button>
  <script>
    function 알림창열고닫기(파라미터){//축약하는기능 //파라미터 사용해서 더 축약가능
      document.getElementById('alert').style.display =파라미터;
    }
    /* 알림창열기(123); 알림창 열기(123)을 실행해주세요 !!파라미터 자리에 숫자 123을 넣어서 실행해주세요!!*/
    //알림창열기('block');
    //알림창닫기('none'); //파라미터 동일 타이틀을 이용해서 더 축약이 가능하다

    function plus(파라미터){//파라미터 동일 타이틀을 이용해서 더 축약이 가능하다
      return 2 + 파라미터;
    }
    plus(1);// 2 + 1 이 실행
    plus(2);// 2 + 2 이 실행
    plus(3);// 2 + 3 이 실행

    console.log(plus(1));// 결과: 3
    console.log(plus(2));// 결과: 4
    console.log(plus(3));// 결과: 5
    //함수를 여러개 만들고 싶다 -> 파라미터 안써도 상관없음.

  </script>
</body>
</html>
```

※ eventListener / react 추후

(Html + CSS + JavaScript) - 로그인 & 회원가입 동적 기능 구현

실습 : <https://www.youtube.com/watch?v=p1GmFCGuVjw>

1~2시간 소요 예정