

6강. 자바스크립트 기초 문법



JavaScript 작성하기

- 자바스크립트 언어

- 정의: 웹 브라우저에서 실행되는 대표적인 스크립트 언어.
- 역할: 웹 페이지에 ****동적 기능(Interactive)****을 추가하는 데 사용.
(HTML → 구조, CSS → 디자인, JavaScript → 동작)

- 역사

1995년: Netscape에서 Brendan Eich가 10일 만에 JavaScript 개발

1997년: ECMAScript로 표준화

2008년: 구글 V8 엔진, Chrome → 성능 혁명

2009년: **Node.js** → 서버에서도 활용

2015년 이후: **ES6(ES2015)** 등장, 현대적 문법 확립, 프레임워크 시대 개막

JavaScript 작성하기

- **특징**

- 인터프리터 언어**

- 별도의 컴파일 과정 없이, 작성하면 바로 실행됩니다.

- 브라우저 내장**

- Chrome, Edge, Firefox, Safari 등 모든 브라우저에서 지원합니다.

- 객체 기반 언어**

- 객체(Object)를 활용하여 데이터와 기능을 효율적으로 관리할 수 있습니다.

- 동적 타이핑**

- 변수 선언 시 타입 지정이 필요 없고, 실행 시점에 타입이 결정됩니다.

- 이벤트 기반 프로그래밍**

- 버튼 클릭, 입력, 마우스 이동 같은 이벤트에 따라 코드 실행.

JavaScript 작성하기

- **활용 분야**

- 웹 프론트엔드**

- 버튼 클릭, 폼 입력 검증, 애니메이션, AJAX(비동기 데이터 처리)

- 웹 백엔드**

- Node.js 환경에서 서버 개발 가능

- 모바일 앱**

- React Native, Ionic 같은 프레임워크로 앱 개발

- 데스크탑 앱**

- Electron을 이용해 크로스 플랫폼 앱 개발 (VS Code도 Electron 기반)

- 게임 개발**

- HTML5 Canvas + JavaScript로 2D/3D 게임 제작

JavaScript 작성하기

- 자바스크립트 태그 사용 - 내부 스크립트
 - `<script>~</script>` 태그 안에 작성

자바 스크립트 예제

클릭

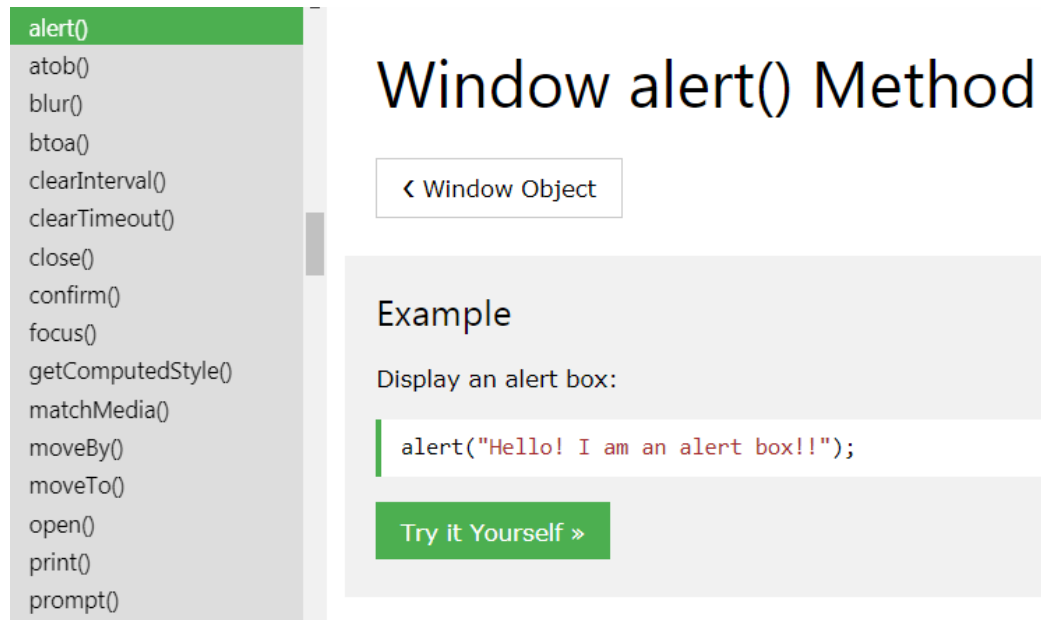
```
<h1 id="title">자바 스크립트 예제</h1>
<button onclick="sayHello()">클릭</button>

<script>
  function sayHello() {
    alert("Hello~ Javascript!")
  }
</script>
```

자바스크립트 Reference

alert() 함수 – 알림창을 띄우는 함수

- w3schools 사이트
- JavaScript Reference > Window Reference > Window > Window Object Methods > **alert()** 함수



The screenshot shows the W3Schools JavaScript Reference page for the `alert()` method. On the left, a sidebar lists various Window Object methods, with `alert()` highlighted at the top. The main content area is titled "Window alert() Method" and includes a navigation link "< Window Object". Below this, an "Example" section shows the code to display an alert box: `alert("Hello! I am an alert box!!");`. A green button labeled "Try it Yourself »" is positioned below the code.

alert()

atob()
blur()
btoa()
clearInterval()
clearTimeout()
close()
confirm()
focus()
getComputedStyle()
matchMedia()
moveBy()
moveTo()
open()
print()
prompt()

Window alert() Method

[< Window Object](#)

Example

Display an alert box:

```
alert("Hello! I am an alert box!!");
```

[Try it Yourself »](#)

JavaScript 작성하기

자바스크립트 출력 - 외부 스크립트

`<script src="js파일경로"></script>`

```
<h1 id="title">자바 스크립트 예제</h1>
```

```
<h3>console.log() 함수 사용</h3>
```

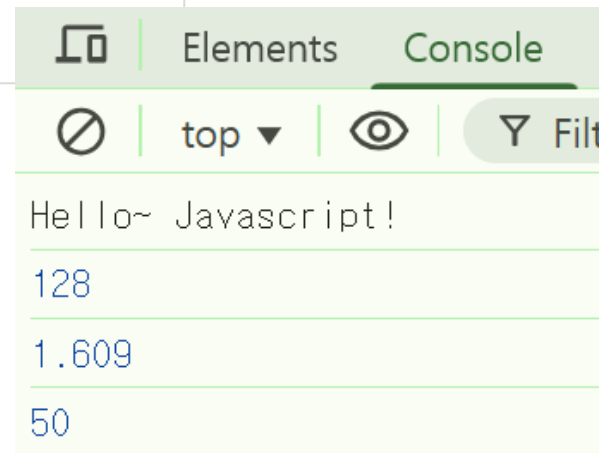
```
<script src="js/output.js"></script>
```

데이터 출력

콘솔창을 이용한 출력

- 크롬 브라우저 > F12(개발자도구) > Console
- **console.log() 메서드**

```
console.log("Hello~ Javascript!");  
console.log(128);  
console.log(1.609);  
console.log(20+30); //50
```



자바스크립트 해석 엔진

웹 브라우저에는 HTML 분석기(Parser), CSS 분석기, 자바스크립트 해석기(Interpreter)가 포함되어 있음

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>Javascript</title>
  <style>
    body{
      text-align: center;
    }
  </style>
</head>
<body>
  <script>
    document.write("Hello~ Javascript");
  </script>
</body>
</html>
```

- ① HTML분석기는 HTML의 순서와 포함관계를 확인한다.
- ② CSS 분석기는 HTML 분석기가 태그 분석을 끝낸 다음 <style> 태그 사이의 스타일 정보를 분석한다.
- ③ 마지막으로 자바스트립트 해석기(기계어로 번역-인터프리터)가 <script>태그 사이의 소스 코드를 해석한다.

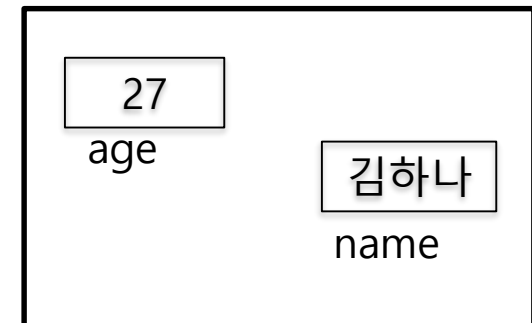
변수(Variable)

■ 변수란?

- 프로그램에서 사용되는 자료를 저장하기 위한 메모리 공간(영역)
- 할당받은 메모리의 주소 대신 부르는 이름 (메모리 주소: ac13bf00)
- 프로그램 실행 중에 값 변경 가능, variable 이라 함

■ 변수의 선언 및 초기화

- 변수 선언은 어떤 타입의 데이터를 저장할 것인지 그리고 변수이름은 무엇인지를 결정한다. **var, let** 키워드를 사용한다.
- (자료형) 변수이름; `var age; let name;`
- (자료형) 변수이름 = 초기값;
`var age = 27, let name = "김하나"`



변수(Variable)

- 변수의 선언과 초기화

```
let 변수명 = 값
```

- 변수명 작성시 주의

- 변수 이름은 숫자로 시작할 수 없고, 공백 불허

틀린 예) var 3box; let sky 1234

- 변수 이름은 영문, 한글, 특수문자, 숫자의 결합으로 만든다.
- 영어 대소문자를 구별하며 **예약어**는 쓸 수 없다.

예약어) if, for, while 등

변수(Variable)

- 변수와 상수

```
<script>
  let name = "이정후";
  let age = 26;
  let isStudent = false;

  console.log("Name:", name);
  console.log("Age:", age);
  console.log("Is Student:", isStudent);

  // 값 변경(재할당)
  age = 27;
  console.log("Updated Age:", age);

  // Constant variable(상수)
  const birthYear = 1988;

  //birthYear = 1994; // Error: 재할당할 수 없음.
  console.log("Birth Year:", birthYear); //1998
</script>
```

variable.html

상수(Constant)

- 상수란?

한번 설정해 두면 그 프로그램이 종료 될 때까지 변경될 수 없는 값

- 상수의 선언과 초기화

```
const 변수명 = 값
```

- 상수 작성시 주의

- 값을 변경(재할당) 할 수 없다. - 오류 발생
- 관례상 대문자 사용

호이스팅(hoisting)

- **var 변수와 호이스팅**

- ES6 이전에는 변수를 선언할 때 var 예약어로 사용, 지금도 사용함
- **호이스팅**: '끌어올린다'는 의미로 자바스크립트 해석기가 변수를 기억해 두어 선언한 효과를 가짐.

즉, 선언하지 않고도 사용할 수 있고, 오류도 발생하지 않음

```
<script>
  var x = 10;
  var sum = x + y; //y 변수는 선언 전에 참조되었지만 에러가 발생하지 않음
  var y = 20;
  console.log(sum);
</script>
```

호이스팅(hoisting)

- 변수 비교

구 분	선언하지 않고 사용	재선언	재할당
let	오류 발생	X	O
const	오류 발생	X	X
var	오류 없음(호이스팅)	O	O

자료형(Type)

- 자료형(type)

프로그램에서 처리할 데이터의 형태

사용할 데이터의 종류에 따라 메모리 공간을 적절하게 설정해 주는 것.

구분	자료형	예시
숫자(정수, 실수)	number	let age = 16, let unit = 2.54
문자(문자열)	string	let season = "winter"
불리언(true/false)	boolean	let isMarried = true
배열	object	let seasons = ["봄", "여름", "가을", "겨울"]
객체	object	let date = new Date()

자료형(type)

■ 숫자형

- 정수와 실수를 함께 묶어서 '숫자형' 이라고 한다.
- typeof(숫자) 확인하면 - number

■ 문자형

- 작은 따옴표나 큰 따옴표로 묶은 자료형이다.
- typeof(문자) 확인하면 - string

■ 불리언형

- true / false 값을 가지는 자료형이다.
- typeof(문자) 확인하면 - boolean

자료형(type)

- **undefined**

자료형이 정의되지 않았을 때의 데이터 상태 : 변수를 선언할 때
아직 값이 할당되지 않은 경우

- **null**

데이터 값이 유효하지 않은 경우 : 변수에 값이 할당되었는데 잘못된
(유효하지 않은) 값인 경우

```
> let x
```

```
< undefined
```

```
> age = prompt("나이 입력: ") → (취소한 경우 null이 들어감)
```

```
< null
```

자료형(type)

- 배열

한 개의 변수에 여러 개의 값을 저장하는 자료형이다.

대괄호를 사용하고 인덱스(위치)로 접근함

- 객체

한 개의 변수에 다양한 자료가 포함되어 있는 자료형이다.

키와 값의 쌍으로 구성됨

중괄호를 사용함

- 템플릿 리터럴

문자열과 변수, 식을 섞어서 하나의 문자열을 만드는 표현 형식이다.

백틱(`)을 사용해 문자열을 만든다.

자료형(type)

- 자료형 예제

```
<title>자료형 예제</title>  
</head>  
<body>  
  <script src="js/type.js"></script>  
</body>  
</html>
```

자료형(type)

■ 자료형 예제

```
//1. 숫자형 (Number)
let age = 26; //정수형 (Integer)
let temperature = 24.5; //실수형 (Float)

console.log("나이:" + age + "세");
console.log("온도:", temperature + "도");
console.log(typeof(age)); //number
console.log(typeof(temperature)); //number

//2. 문자열형 (String)
let player = "이정후"; //큰따옴표 사용
let city = '샌프란시스코'; //작은따옴표 사용
console.log("선수이름:", player + ", 도시: " + city);
console.log('선수이름:', player, ', 도시:', city);
console.log(typeof(player)); //string
//백틱(`) 사용 (템플릿 문자열)
console.log(`선수이름: ${player}, 도시: ${city}`);
```

type.js

자료형(type)

■ 자료형 예제

```
//3. 불린형 (Boolean)
let isStudent = false; //불리언형(Boolean), true or false
let hasLicense = true;
console.log("Is Student:", isStudent);
console.log("Has License:", hasLicense);
console.log(typeof(hasLicense)); //boolean

//4. null과 undefined
let emptyValue = null; //값이 없음
let notAssigned; //값이 할당되지 않음
console.log("Empty Value:", emptyValue); //null
console.log("Not Assigned:", notAssigned); //undefined
console.log(typeof(emptyValue)); //object (null은 object로 나옴)
console.log(typeof(notAssigned)); //undefined
```

자료형(type)

■ 자료형 예제

//4. 배열형 (Array)

```
let fruits = ["Apple", "Banana", "Cherry"]; //배열 (Array)
console.log(fruits); //["Apple", "Banana", "Cherry"]
console.log(fruits[0]); //Apple
console.log(fruits[1]); //Banana
console.log(fruits[2]); //Cherry
console.log("배열의 크기", fruits.length); //3
```

//5. 객체형 (Object)

```
let person = {
  name: "우상혁",
  age: 29,
  isStudent: false
};
console.log(person); //{name: '우상혁', age: 29, isStudent: false}
console.log("이름:", person.name); //우상혁
console.log("나이:", person.age); //29
console.log("학생여부:", person.isStudent); //false
console.log(typeof(person)); //object
console.log(person.name + "은 " + person.age + "세 입니다.");
console.log(`${person.name}은 ${person.age}세 입니다.`); //템플릿 문자열
```

형 변환(Type Conversion)

- 형변환이란 숫자형을 문자형으로, 문자형을 숫자형 데이터로 변경하는 것

연산자	기능
Number()	부울형(true/false)과 날짜를 숫자로 변환
parseInt()	문자형을 정수형으로 변환
parseFloat()	문자형을 실수형으로 변환
toString()	자료형을 문자열 데이터로 변환

자료형(type)

■ 형 변환

```
<!-- 자료형 변환 -->  
<script src="js/conversion.js"></script>
```

```
//Number() 함수 사용  
let numString = "123";  
let convertedNumber = Number(numString); //문자열 -> 숫자  
console.log(numString, convertedNumber); //123 123  
console.log(typeof(numString), typeof(convertedNumber));  
  
//String() 함수 사용  
let strNumber = 456;  
let convertedString = String(strNumber); //숫자 -> 문자열  
console.log(strNumber, convertedString); //456 "456"  
console.log(typeof(strNumber), typeof(convertedString));
```

conversion.js

자료형(type)

■ 형 변환

```
//toString() 메서드 사용
let num = 789;
let str = num.toString(); //숫자 -> 문자열
console.log(num, str); //789 "789"
console.log(typeof(num), typeof(str)); //number string

//parseInt() 함수 사용
let floatString = "123.45";
let intNumber = parseInt(floatString); //문자열 -> 정수
console.log(floatString, intNumber); //123.45 123
console.log(typeof(floatString), typeof(intNumber)); //string number

//parseFloat() 함수 사용
let floatNumber = parseFloat(floatString); //문자열 -> 실수
console.log(floatString, floatNumber); //123.45 123.45
console.log(typeof(floatString), typeof(floatNumber)); //string number
```

자료형(type)

■ 형 변환

```
//prompt() 함수로 입력받은 값은 항상 문자열임
let age = prompt("나이를 입력하세요:"); //string
console.log(age, typeof age); //string

//나이를 숫자로 변환
//let numericAge = Number(age);
let numericAge = parseInt(age); //정수로 변환
console.log(numericAge, typeof numericAge); //number

if(numericAge >= 20){
  console.log("성인입니다.");
} else {
  console.log("미성년자입니다.");
}

//isNaN() 함수 사용
if(isNaN(numericAge)){ //NaN이면 true, 숫자이면 false
  console.log("숫자가 아닙니다.");
} else {
  console.log("숫자입니다.");
}
```

conversion.js

항과 연산자

■ 항(operand)

- 연산에 사용되는 값

■ 연산자(operator)

- 연산에 사용되는 기호

예) $3 + 7$ (3과 7은 항, '+'는 연산자)



■ 항의 개수에 따른 연산자 구분

연산자	설명	연산 예
단항 연산자	항이 한 개인 연산자	$\text{num}++$
이항 연산자	항이 두 개인 연산자	$\text{num1} * \text{num2}$
삼항 연산자	항이 세 개인 연산자	$(5 > 3) ? 'T' : 'F'$

대입 연산자

◆ 대입(할당) 연산자

- 오른쪽의 값을 왼쪽의 변수에 대입(기억)
- '=' 연산자를 사용.

```
let number = 20
```

```
let userId = "abc123"
```

산술 및 증감 연산자

◆ 산술 및 증감 연산자

연산자	연산 작업	설 명
+	result = n1 + n2	두 수를 더하기
-	result = n1 - n2	두 수를 빼기
*	result = n1 * n2	두 수를 곱하기
/	result = n1 / n2	두 수를 나누기
%	result = n1 % n2	두 수의 나머지
++	n++ (n = n + 1)	1 증가하기
--	n1-- (n = n - 1)	1 감소하기

연산자

◆ 복합 대입 연산자

연산자	연산 작업	연산 작업
<code>+=</code>	<code>val += 10</code>	<code>val = val + 10</code>
<code>-=</code>	<code>val -= 10</code>	<code>val = val - 10</code>
<code>*=</code>	<code>val *= 10</code>	<code>val = val * 10</code>
<code>/=</code>	<code>val /= 10</code>	<code>val = val / 10</code>
<code>%=</code>	<code>val %= 10</code>	<code>val = val % 10</code>

산술 및 증감 연산자

◆ 산술 및 증감, 복합 대입 연산자

```
// 산술 연산자
let x = 10;
let y = 3;
console.log("x + y = " + (x + y)); // 13
console.log("x - y = " + (x - y)); // 7
console.log("x * y = " + (x * y)); // 30
console.log("x / y = " + (x / y)); // 3.3333...
console.log("x % y = " + (x % y)); // 1
console.log("x ** y = " + (x ** y)); // 1000

//증감 연산자
let b = 5;
console.log("b = " + b); // 5
console.log("++b = " + (++b)); // 6 (전위 증가)
console.log("b++ = " + (b++)); // 6 (후위 증가)
console.log("b = " + b); // 7

let c = 5;
console.log("c = " + c); // 5
console.log("--c = " + (--c)); // 4 (전위 감소)
console.log("c-- = " + (c--)); // 4 (후위 감소)
console.log("c = " + c); // 3
```


산술 및 증감 연산자

◆ 산술 및 증감, 복합 대입 연산자

```
// 복합 대입 연산자
let val = 10;
val += 5; // val = val + 5
console.log("val += 5 => " + val); // 15
val -= 3; // val = val - 3
console.log("val -= 3 => " + val); // 12
val *= 2; // val = val * 2
console.log("val *= 2 => " + val); // 24
val /= 4; // val = val / 4
console.log("val /= 4 => " + val); // 6
val %= 4; // val = val % 4
console.log("val %= 4 => " + val); // 2
val **= 3; // val = val ** 3
console.log("val **= 3 => " + val); // 8
```

비교 연산자

◆ 비교 연산자

연산자	연산 작업	결과
>	10 > 5	
>=	10 >= 5	
<	10 < 5	
<=	10 <= 5	
==	10 == 10	true
!=	10 != 10	false
==	10 == '10'	true
!=	10 != '10'	false
===	10 === '10'	false
!==	10 !== '10'	true

'=='는 문자와 계산시
오류 발생

'===' 와 '!==' 사용

연산자

◆ 논리 연산자

연산자	기능
&&	두 항이 모두 참인 경우에만 결과값이 참.
 	두 항 중 하나의 항만 참이면 결과값이 참.
!	단항이 참이면 결과값은 거짓, 거짓이면 참

비교 연산자

◆ 비교 및 논리 연산자

```
//비교 연산자
console.log(10 > 5); //true
console.log(10 < 5); //false
console.log(10 >= 5); //true
console.log(10 <= 5); //false
console.log(10 == 5); //false
console.log(10 != 5); //true
console.log(10 == '10'); //true (값만 비교)
console.log(10 === '10'); //false (값과 타입 모두 비교)
console.log(10 !== '10'); //true (값과 타입 모두 비교)

//논리 연산자
let a = 10;
console.log(a > 5 && a < 15); //true (AND)
console.log(a > 5 && a > 15); //false
console.log(a > 5 || a > 15); //true (OR)
console.log(a < 5 || a > 15); //false
console.log(!(a > 5)); //false (NOT)
console.log(!(a < 5)); //true (NOT)
```

operator2.js

조건 연산자

◆ 조건 연산자

(조건식) ? 조건식이 true이면 실행 : 조건식이 false이면 실행

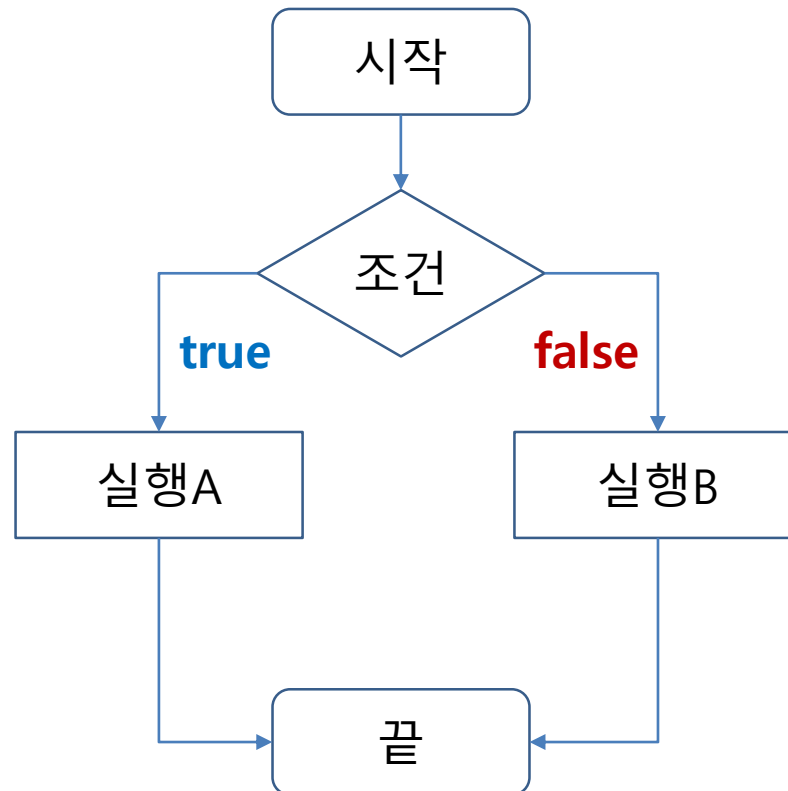
```
let age = 20;
let result = (age >= 18) ? "성인" : "미성년자";
//console.log("age = " + age + ", result = " + result); // age = 20, result = 성인
console.log(`age = ${age}, result = ${result}`); // age = 20, result = 성인

let result2 = (age < 13) ? "어린이" : (age < 20) ? "청소년" : "성인";
console.log(`age = ${age}, result2 = ${result2}`); // age = 20, result2 = 성인
```

조건문

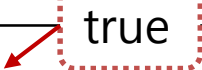
조건문

- 특정한 조건에 의해서 프로그램 진행이 분기되는 문장
- if문, switch문이 대표적임.



if 문

❖ if 문



```
if(조건식){  
    실행문 //조건식이 true이면 실행  
}
```

❖ if ~ else문

```
if(조건식){  
    실행문1 //조건식이 true이면 실행  
}else{  
    실행문2 //조건식이 false이면 실행  
}
```

If ~ else 문

❖ 짝수/홀수를 판별하는 프로그램

if문

12는(은) 짝수입니다.

```
<!-- if ~ else -->
<h2>if문</h2>
<div id="result"></div>

<script src="js/if1.js"></script>
```


If ~ else 문

❖ 짝수/홀수를 판별하는 프로그램

```
const result = document.getElementById("result");

let num = 11;
if(num % 2 == 0){
    //console.log(num + "은(는) 짝수입니다.");
    result.textContent = `${num}은 짝수입니다.`;
} else {
    //console.log(num + "은(는) 홀수입니다.");
    result.textContent = `${num}은 홀수입니다.`;
}
```

If ~ else 문

❖ 짝수/홀수를 판별하는 프로그램

```
let num = prompt("숫자를 입력하세요.", "1");

if(num == null || isNaN(num)){
    result.textContent = `입력이 취소되었습니다.`;
}else{
    num = parseInt(num); //문자열 -> 숫자 변환

    if(num % 2 == 0){
        result.textContent = `${num}는(은) 짝수입니다.`;
    } else {
        result.textContent = `${num}는(은) 홀수입니다.`;
    }
}
```

if 조건문

❖ if ~ else if ~ else문

```
if(조건식1){  
    실행문1; //조건식1이 true이면 실행  
}else if(조건식2){  
    실행문2; //조건식2가 true이면 실행  
}else{  
    실행문3; //조건식1,2가 모두 false이면 실행  
}
```

if 조건문

◆ 학점 계산 프로그램

점수	학점
90점 ~ 100점	A
80점 ~ 90점	B
70점 ~ 80점	C
60점 ~ 70점	D
60점 미만	F

학점 계산 프로그램

점수: 85, 학점: B입니다.

if 조건문

◆ 학점 계산 프로그램

학점 계산 프로그램

점수: 85, 학점: B입니다.

```
<h2>학점 계산 프로그램</h2>  
<div id="result"></div>  
  
<script src="js/score.js"></script>
```

if 조건문

◆ 학점 계산 프로그램

```
const result = document.getElementById("result");
let score = prompt("점수를 입력하세요.", "80");

if(score == null || isNaN(score)) {
    result.textContent = "유효한 숫자를 입력하세요.";
}else {
    score = parseInt(score); //문자열 -> 숫자 변환
    let grade = "";
    if(score >= 90 && score <= 100){
        grade = 'A';
    } else if(score >= 80){
        grade = 'B';
    } else if(score >= 70){
        grade = 'C';
    } else if(score >= 60){
        grade = 'D';
    } else {
        grade = 'F';
    }
    result.innerHTML = `점수: ${score}, 학점: <span>${grade}</span>입니다.`;
}
```

switch문

- switch ~ case문

조건식의 결과가 정수 또는 문자 값이고 그 값에 따라 수행문이 결정될때 사용

```
switch(변수값){  
  case 조건1:  
    조건1이 true이면 실행  
    break;  
  case 조건2:  
    조건2가 true이면 실행  
    break;  
  default:  
    모든 조건이 false이면  
    실행  
    break;  
}
```

switch문

- switch ~ case문

엘리베이터 타기

3층을 눌렀습니다.

```
<style>
  span{color: ■#f00; font-size: 1.2em;}
</style>
</head>
<body>
  <!-- switch ~ case문 -->
  <h2>엘리베이터 타기</h2>
  <div id="result"></div>

  <script src="js/switch1.js"></script>
```


switch문

▪ switch ~ case문

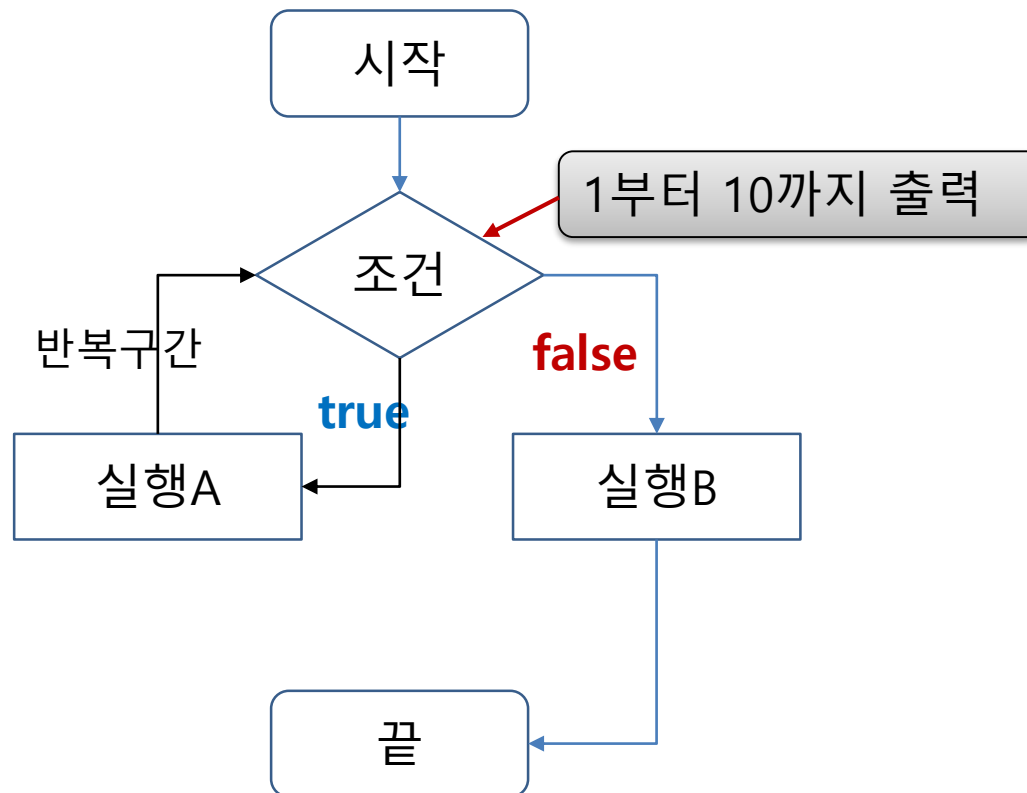
```
const result = document.getElementById("result");
let floor = prompt("몇 층으로 가실건가요?", "1~3");
console.log(typeof(floor)); //string

switch(floor){
  case '1':
    result.innerHTML = `<span>1층</span>을 눌렀습니다.`;
    break;
  case '2':
    result.innerHTML = `<span>2층</span>을 눌렀습니다.`;
    break;
  case '3':
    result.innerHTML = `<span>3층</span>을 눌렀습니다.`;
    break;
  default:
    result.innerHTML = `건물에 없는 층입니다.`;
    break;
}
```

반복문

반복문

- 주어진 조건이 만족할 때까지 실행문을 반복적으로 수행



반복문

- while문

```
let i = 초기값;  
while(조건식){  
    실행문;  
    증감값;  
}
```

```
<body>  
  <!-- while문 -->  
  <script src="js/while1.js"></script>  
</body>
```

while문

- while문

```
//1~10까지 출력  
let i = 0;  
while(i < 10){  
  i++;  
  console.log(i);  
}  
console.log("i =", i); //10
```

```
//1~10까지 짝수 출력  
let j = 0;  
while(j < 10){  
  j++;  
  if(j % 2 == 0)  
    console.log(j);  
}  
console.log("j =", j); //10
```

while문

● while문

```
//1~5까지 더하기  
let k = 0;  
let total = 0;  
while(k < 5){  
  k++;  
  total += k;  
  console.log(`k=${k}, total=${total}`)  
}  
console.log("total =", total); //15
```

k=1, total=1

k=2, total=3

k=3, total=6

k=4, total=10

k=5, total=15

total = 15

while ~ break 문

- while문 - if break문

반복문에서 break 문을 만나면 더 이상 반복을 수행하지 않고 반복문을 빠져 나옴

```
let n = 초기값;  
while(true){  
    if문  
        break;  
    실행문;  
    증감값;  
}
```

```
<body>  
    <!-- break문 -->  
    <script src="js/break.js"></script>  
</body>
```

while ~ break 문

- while문 - if break문

```
let i = 0;
while(true){
  i += 1;
  if (i > 10)
    break;
  console.log(i);
}
console.log("i:", i); //11
```

```
//1부터 10까지 더하기
let i = 0;
let sum = 0; //합계 초기화
while(true){
  i += 1;
  if(i > 10)
    break;
  sum += i;
  console.log(i)
}
console.log("합계:", sum);
```

while ~ break 문

while문 - break문

127.0.0.1:5500 내용:

계속 반복할까요?

127.0.0.1:5500 내용:

계속 반복합니다.

127.0.0.1:5500 내용:

반복을 중단합니다.

```
while(true){  
  let key = prompt("계속할까요?", 'y/n');  
  if(key == 'y' || key == 'Y'){  
    alert("반복합니다.");  
  }else if(key == 'n' || key == 'N'){  
    alert("중단합니다.");  
    break;  
  }else{  
    alert("잘못된 입력입니다.");  
  }  
}  
console.log("프로그램 종료")
```


for 문

- for문

```
for(초기값; 조건식; 증감값){  
    조건식이 true면 반복실행;  
}
```

```
var i;  
for(①i = 1; ②i <= 10; ④i++) {  
    ③console.log(i);  
}
```

```
//1부터 10까지 출력하기  
for(var i=1; i<=10; i++){  
    console.log(i);  
}
```

for 문

- for문 - 1부터 5까지 더하기

```
//1부터 10까지 출력
for(var i = 1; i <= 10; i++){
  console.log(i);
}

//1부터 5까지 더하기
let sum = 0;
for(var i=1; i<=5; i++){
  sum += i;
  console.log(`i=${i}, sum=${sum}`);
}
console.log(sum);
```

for 문

- for문 - 1부터 입력값까지 더하는 프로그램

합계 구하기

1부터 5까지의 합은 25입니다.

```
<h2>합계 구하기</h2>
```

```
<div id="result"></div>
```

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

for 문

- for문 - 1부터 입력값까지 더하는 프로그램

```
const result = document.getElementById("result");

let num = prompt("몇까지의 합을 계산할까요?", "1");
let sum = 0;

num = parseInt(num);
for(var i = 1; i <= num; i++){
    sum += num;
}
result.textContent = `1부터 ${num}까지의 합은 ${sum}입니다.`;
```

for 문

- for문 - 1부터 5까지 곱하기

5! = 5x4x3x2x1(팩토리알 : 계승)

```
let facto = 1;
for(let i = 1; i<=5; i++){
  facto *= i;
  console.log(`i=${i}, facto=${facto}`);
}
console.log(`${i-1}! = ${facto}`);
```

i=1, facto=1

i=2, facto=2

i=3, facto=6

i=4, facto=24

i=5, facto=120

5! = 120

for 문

- 구구단

구구단

7단

7 x 1 = 7

7 x 2 = 14

7 x 3 = 21

7 x 4 = 28

7 x 5 = 35

7 x 6 = 42

7 x 7 = 49

7 x 8 = 56

7 x 9 = 63

```
<h2>구구단</h2>
```

```
<div id="result"></div>
```

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

for 문

- 구구단

```
const result = document.getElementById("result");
const dan = 7;

result.innerHTML = `<h3>${dan}단</h3>`
for(var i=1; i<=9; i++){
  result.innerHTML += `${dan} x ${i} = ${dan * i}<br>`;
}
```

반복문

❖ 이중 for문

- 행, 열 구현하기

```
for(초기값;조건식;증감값){
```

```
    for(초기값;조건식;증감값){
```

```
        실행문;
```

```
    }
```

```
}
```

행

열

5행 5열

	열1	열2	열3	열4	열5
행1					
행2					
행3					
행4					
행5					

본문

❖ 이중 for문

행, 열 구현하기

가
가
가
가
가

```
for(var i = 1; i <= 5; i++){
    for(var j = 1; j <= 5; j++){
        document.write('가');
    }
    document.write('<br>');
}
```

반복문

❖ 별 찍기

*

**

```
//5행 5열의 별 출력
let i, j;

for(i = 1; i <= 5; i++){ //행이 5번 반복
    for(j = 1; j <= 5; j++){ //열이 5번 반복
        document.write("가")
    }
    document.write('<br>'); //행 바꿈
}

//직각삼각형 모양의 별
for(i = 1; i <= 5; i++){ //행이 5번 반복
    for(j = 1; j <= i; j++){ //열이 5번 반복
        document.write("*")
    }
    document.write('<br>'); //행 바꿈
}

/*
i = 1,  j = 1           *
i = 2,  j = 1, 2       **
i = 3,  j = 1, 2, 3    ***
i = 4,  j = 1, 2, 3, 4  ****
i = 5,  j = 1, 2, 3, 4, 5 *****
*/
```

반복문

❖ 연속으로 1증가하는 자연수를 출력하는 프로그램

```
1 2 3 4 5  
6 7 8 9 10  
11 12 13 14 15  
16 17 18 19 20  
21 22 23 24 25
```

```
for(let i = 0; i < 5; i++){  
  for(let j = 1; j <= 5; j++){  
    let num = i * 5 + j; //5는 열의 종료값  
    document.write(num + " ");  
  }  
  document.write("<br>") //다음 줄(행)  
}
```

for 문

- 구구단 전체

구구단 전체 (2 ~ 9단)

2단	3단	4단	5단	6단	7단	8단	9단
$2 \times 1 = 2$	$3 \times 1 = 3$	$4 \times 1 = 4$	$5 \times 1 = 5$	$6 \times 1 = 6$	$7 \times 1 = 7$	$8 \times 1 = 8$	$9 \times 1 = 9$
$2 \times 2 = 4$	$3 \times 2 = 6$	$4 \times 2 = 8$	$5 \times 2 = 10$	$6 \times 2 = 12$	$7 \times 2 = 14$	$8 \times 2 = 16$	$9 \times 2 = 18$
$2 \times 3 = 6$	$3 \times 3 = 9$	$4 \times 3 = 12$	$5 \times 3 = 15$	$6 \times 3 = 18$	$7 \times 3 = 21$	$8 \times 3 = 24$	$9 \times 3 = 27$
$2 \times 4 = 8$	$3 \times 4 = 12$	$4 \times 4 = 16$	$5 \times 4 = 20$	$6 \times 4 = 24$	$7 \times 4 = 28$	$8 \times 4 = 32$	$9 \times 4 = 36$
$2 \times 5 = 10$	$3 \times 5 = 15$	$4 \times 5 = 20$	$5 \times 5 = 25$	$6 \times 5 = 30$	$7 \times 5 = 35$	$8 \times 5 = 40$	$9 \times 5 = 45$
$2 \times 6 = 12$	$3 \times 6 = 18$	$4 \times 6 = 24$	$5 \times 6 = 30$	$6 \times 6 = 36$	$7 \times 6 = 42$	$8 \times 6 = 48$	$9 \times 6 = 54$
$2 \times 7 = 14$	$3 \times 7 = 21$	$4 \times 7 = 28$	$5 \times 7 = 35$	$6 \times 7 = 42$	$7 \times 7 = 49$	$8 \times 7 = 56$	$9 \times 7 = 63$
$2 \times 8 = 16$	$3 \times 8 = 24$	$4 \times 8 = 32$	$5 \times 8 = 40$	$6 \times 8 = 48$	$7 \times 8 = 56$	$8 \times 8 = 64$	$9 \times 8 = 72$
$2 \times 9 = 18$	$3 \times 9 = 27$	$4 \times 9 = 36$	$5 \times 9 = 45$	$6 \times 9 = 54$	$7 \times 9 = 63$	$8 \times 9 = 72$	$9 \times 9 = 81$

for 문

- 구구단 전체

```
<style>
  h2{padding-left: 20px;}
  div{
    display: inline-block;
    border: 1px solid #ccc;
    margin: 10px;
    padding: 10px 20px;
  }
  h3{text-align: center;}
</style>
</head>
<body>
  <h2>구구단 전체 (2 ~ 9단)</h2>
  <div id="result"></div>

  <script src="js/gugudan_full.js"></script>
</body>
```

for 문

- 구구단 전체

```
const result = document.getElementById("result");
let output = "";

for (let i = 2; i <= 9; i++) {
  output += `<h3>${i}단</h3>`;
  for (let j = 1; j <= 9; j++) {
    output += `${i} x ${j} = ${i * j}<br>`;
  }
  output += "<br>";
}
result.innerHTML = output;
```

for 문

- 구구단 전체 – 박스 디자인

```
let output = "";
for (let i = 2; i <= 9; i++) {
  output += "<div>"
  output += `<h3>${i}단</h3>`;
  for (let j = 1; j <= 9; j++) {
    output += `${i} x ${j} = ${i * j}<br>`;
  }
  output += "</div>";
}

result.innerHTML = output;
```