



# JavaScript 101 - (3) 조건문과 반복문

## JavaScript의 truthy와 falsy

JavaScript에서 truthy란 Boolean 컨텍스트에서 true로 간주되는 값입니다. 반대로 falsy란 false로 간주되는 값입니다. JavaScript는 Boolean 컨텍스트에서 타입 변환을 사용합니다.

JavaScript에서 falsy한 값은 다음과 같습니다:

- `false`
- `0`
- `-0`
- `0n` (BigInt 타입의 `0`)
- `""` (빈 문자열)
  - (주의) `'''` 와 다릅니다.
- `null`
- `undefined`
- `NaN` (Number 타입으로 변환된 숫자가 아닌 무언가).
  - `typeof NaN === "number" // true`

이 외의 모든 값은 truthy합니다.

예를 들어, 다음과 같은 코드를 보세요.

```
if (value) {
  // value가 truthy하면 실행됩니다.
} else {
  // value가 falsy하면 실행됩니다.
}
```

value에 다양한 값을 대입해보고 결과를 확인해보세요.

예시:

```
let value = "Hello"; // truthy한 값

if (value) {
  console.log("Truthy"); // 이 문장이 출력됩니다.
} else {
  console.log("Falsy");
}
```

```
let value = ""; // falsy한 값

if (value) {
  console.log("Truthy");
} else {
  console.log("Falsy"); // 이 문장이 출력됩니다.
}
```

## JavaScript의 조건문

조건문이란 특정 조건에 따라 다른 코드를 실행하는 문장입니다<sup>1</sup>. JavaScript에서는 다음과 같은 조건문을 사용할 수 있습니다:

- `if`: 만약 조건이 참이면 코드 블록을 실행합니다.
- `else`: 만약 조건이 거짓이면 코드 블록을 실행합니다.
- `else if`: 만약 첫 번째 조건이 거짓이고 다른 조건이 참이면 코드 블록을 실행합니다.

- 삼항 연산자: 조건과 물음표(?) 그리고 참일 때와 거짓일 때의 표현식으로 구성된 연산자입니다.

예를 들어, 다음과 같은 코드를 보세요.

```
let score = 85; // 점수

if (score >= 90) {
  // 점수가 90 이상이면
  console.log("A"); // A 등급
} else if (score >= 80) {
  // 점수가 80 이상이면
  console.log("B"); // B 등급
} else if (score >= 70) {
  // 점수가 70 이상이면
  console.log("C"); // C 등급
} else {
  // 그 외의 경우
  console.log("F"); // F 등급
}
```

위의 코드는 점수에 따라 다른 등급을 출력하는 예시입니다. 점수가 순서대로 각 조건에 맞는지 확인하고, 맞으면 해당 코드 블록을 실행하고 종료합니다. 위의 예시에서는 `score` 가 `85` 이므로 첫 번째 조건은 거짓이고, 두 번째 조건은 참입니다. 따라서 `"B"` 를 출력하고 종료합니다.

삼항 연산자는 다음과 같은 형식으로 사용할 수 있습니다.

```
condition ? expressionIfTrue : expressionIfFalse;
```

예를 들어, 다음과 같은 코드를 보세요.

```
let age = 18; // 나이

let message = age >= 19 ? "성인입니다." : "미성년자입니다."; // 삼항 연산자

console.log(message); // 결과 출력
```

위의 코드는 나이에 따라 다른 메시지를 출력하는 예시입니다. 나이가 19 이상이면 "성인입니다."라는 표현식을 `message` 에 할당하고, 그렇지 않으면 "미성년자입니다."라는 표현식을 `message` 에 할당합니다. 위의 예시에서는 `age` 가 18이므로 `message` 에 "미성년자입니다."가 할당되고, 이것을 출력합니다.

## JavaScript의 반복문

반복문이란 특정 조건에 따라 같은 코드를 여러 번 실행하는 문장입니다. JavaScript에서는 다음과 같은 반복문을 사용할 수 있습니다:

- `for`: 정해진 횟수만큼 코드 블록을 실행합니다.
- `for/in`: 객체의 속성들을 순회하면서 코드 블록을 실행합니다.
- `for/of`: 반복 가능한 객체의 값들을 순회하면서 코드 블록을 실행합니다.
- `while`: 조건이 참인 동안 코드 블록을 실행합니다.
- `do/while`: 코드 블록을 한 번 실행하고, 조건이 참인 동안 계속 반복합니다.

### for 문

예를 들어, 다음과 같은 코드를 보세요.

```
for (let i = 0; i < 5; i++) {
  // i가 0부터 시작해서 5보다 작은 동안
  console.log(i); // i를 출력하고
} // i를 1 증가시킵니다.
```

위의 코드는 `for` 문을 사용하여 0부터 4까지의 숫자를 출력하는 예시입니다. `for` 문은 괄호 안에 세 가지 부분으로 구성됩니다:

- 초기화: 반복문이 시작되기 전에 한 번만 실행되는 부분입니다. 보통 변수를 선언하고 초기값을 할당합니다. 예시에서는 `let i = 0` 입니다.

- 조건식: 반복문이 계속되는지 확인하는 부분입니다. 참이면 코드 블록을 실행하고, 거짓이면 반복문을 종료합니다. 예시에서는 `i < 5` 입니다.
- 최종식: 반복문의 각 사이클마다 실행되는 부분입니다. 보통 변수의 값을 변경하여 조건식에 영향을 줍니다. 예시에서는 `i++` 입니다.

## for/in 문

`for/in` 문은 객체의 속성들에 대해 반복할 수 있습니다. 예를 들어, 다음과 같은 코드를 보세요.

```
let person = {name: "Alice", age: 20, hobby: "coding"}; // 객체

for (let prop in person) {
  // prop에 person의 속성들이 순서대로 할당됩니다.
  console.log(prop + ": " + person[prop]); // 속성과 값 출력
}
```

위의 코드는 `for/in` 문을 사용하여 `person` 객체의 속성과 값을 출력하는 예시입니다. `for/in` 문은 괄호 안에 변수와 객체로 구성됩니다. 변수에는 객체의 속성들이 순서대로 할당되고, `객체[prop]` 형식으로 값을 접근할 수 있습니다.

## for/of 문

`for/of` 문은 배열, 문자열 등 반복 가능한 객체의 값들에 대해 반복할 수 있습니다<sup>14</sup>. 예를 들어, 다음과 같은 코드를 보세요.

```
let fruits = ["apple", "banana", "cherry"]; // 배열

for (let fruit of fruits) {
  // fruit에 fruits의 값들이 순서대로 할당됩니다.
  console.log(fruit); // 값 출력
}
```

위의 코드는 `for/of` 문을 사용하여 `fruits` 배열의 값을 출력하는 예시입니다. `for/of` 문은 괄호 안에 변수와 반복 가능한 객체로 구성됩니다. 변수에는 객체의 값들이 순서대로 할당됩니다.

## while 문

`while` 문은 조건식이 참인 동안 계속해서 코드 블록을 실행합니다. 예를 들어, 다음과 같은 코드를 보세요.

```
let n = 0; // 변수 선언 및 초기화

while (n < 3) {
  // n이 3보다 작은 동안
  console.log(n); // n을 출력하고
  n++; // n을 1 증가시킵니다.
}
```

위의 코드는 `while` 문을 사용하여 0부터 2까지의 숫자를 출력하는 예시입니다. `while` 문은 괄호 안에 조건식만 있습니다. 조건식이 참이면 코드 블록을 실행하고, 거짓이면 반복문을 종료합니다. 코드 블록 안에서 변수의 값을 변경하여 조건식에 영향을 줄 수 있습니다.

## do/while 문

`do/while` 문은 `while` 문과 비슷하지만, 코드 블록을 한 번 실행한 후에 조건식을 확인합니다. 예를 들어, 다음과 같은 코드를 보세요.

```
let m = 0; // 변수 선언 및 초기화

do {
  console.log(m); // m을 출력하고
  m++; // m을 1 증가시킵니다.
} while (m < 3); // m이 3보다 작은 동안 반복합니다.
```

위의 코드는 `do/while` 문을 사용하여 0부터 2까지의 숫자를 출력하는 예시입니다. `do/while` 문은 `do` 키워드 다음에 코드 블록이 오고, `while` 키워드 다음에 괄호 안에 조건식이 옵니다. 코드 블록은 최소 한 번은 실행되고, 조건식이 참이면 계속 반복됩니다.