



반복문(for,while)

or , while , do...while

for...of , for...in

이제 자바스크립트에서 **반복문**에 대해 배워보겠습니다. 반복문은 특정 코드를 여러 번 실행할 수 있게 해주는 구조입니다. 대표적인 반복문은 **for문**과 **while문**입니다. 각각의 반복문은 다양한 상황에 맞게 사용될 수 있으며, 반복 횟수나 조건에 따라 다르게 동작합니다.

1. for문

for문은 반복 횟수가 명확할 때 사용하는 반복문입니다. 즉, 몇 번 반복할지 알고 있는 경우에 자주 사용됩니다. for문은 **초기화식**, **조건식**, **증감식**으로 구성됩니다.

기본 구조:

```
for (초기화식; 조건식; 증감식) {  
    // 반복할 코드  
}
```

예시:

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

위 코드는 `i`가 0부터 4까지 1씩 증가하면서 "안녕"을 5번 출력합니다. `i++`는 `i`의 값을 1씩 증가시키는 증감식입니다.

2. while문

while문은 반복 횟수가 명확하지 않을 때 사용됩니다. 조건식이 참인 동안 반복을 계속합니다.

기본 구조:

```
while (조건식) {  
    // 반복할 코드  
}
```

예시:

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

위 예시는 `i`가 5보다 작을 동안 "안녕"을 출력하고, `i`가 1씩 증가하는 구조입니다.

3. for...of 문

for...of 문은 배열이나 순회 가능한 객체(iterable object)를 순회할 때 사용됩니다. 배열에 있는 각 요소를 하나씩 순서대로 반복하여 처리할 수 있습니다.

기본 구조:

```
for (let 변수 of 배열) {  
    // 반복할 코드  
}
```

예시:

```
let fruits = ["apple", "banana", "cherry"];  
  
for (let fruit of fruits) {  
    console.log(fruit);  
}
```

위 코드에서는 배열 `fruits`의 각 요소(`apple`, `banana`, `cherry`)가 한 번씩 출력됩니다.

4. for...in 문

for...in 문은 객체의 속성을 순회할 때 사용됩니다. 객체의 속성 이름(키)을 하나씩 가져와 반복할 수 있습니다.

기본 구조:

```
for (let key in 객체) {  
    // 반복할 코드  
}
```

예시:

```
let person = { name: "John", age: 30, city: "Seoul" };  
  
for (let key in person) {  
    console.log(key, person[key]);  
}
```

위 코드는 객체 `person`의 각 속성(`name`, `age`, `city`)을 순회하면서 속성 이름과 값을 출력합니다.

5. do...while문

do...while문은 조건을 나중에 검사하는 반복문입니다. 즉, 최소 한 번은 실행한 후 조건을 확인하여 반복을 결정합니다.

기본 구조:

```
do {  
    // 반복할 코드  
} while (조건식);
```

예시:

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

위 코드는 `i`가 5보다 작을 동안 `"i"`를 출력하지만, 최소 한 번은 반드시 실행됩니다.

요약

- **for문**: 반복 횟수가 명확할 때 사용.
- **while문**: 조건이 참인 동안 반복(반복 횟수 미정일 때).
- **for...of문**: 배열이나 iterable 객체를 순회할 때.
- **for...in문**: 객체의 속성을 순회할 때.
- **do...while문**: 최소 한 번은 실행 후 조건을 검사하는 반복문.

자바스크립트의 반복문을 사용하면 특정 작업을 여러 번 자동으로 실행할 수 있어 코드를 더 효율적으로 작성할 수 있습니다.

이번에는 **while문**과 **do...while문**에 대해 정리해보겠습니다.

while문

while문은 조건이 참인 동안 계속해서 코드를 반복 실행하는 반복문입니다. 주로 **반복 횟수가 명확하지 않을 때** 사용됩니다. 조건식이 참이면 코드를 반복하고, 조건식이 거짓이 되면 반복을 멈춥니다.

기본 구조:

```
while (조건식) {  
    // 반복할 코드  
}
```

예시:

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

위 코드는 `count`가 5보다 작을 동안, `count` 값을 출력하고 `count++`로 값을 증가시키면서 반복합니다. 결과적으로 0부터 4까지 출력됩니다.

do...while문

do...while문은 **최소 한 번은 실행한 후**, 조건식을 평가하여 반복을 결정하는 반복문입니다. 즉, 조건과 상관없이 코드를 **최소 한 번 실행한 후**, 조건을 평가해 반복 여부를 결정하는 구조입니다.

기본 구조:

```
do {  
    // 반복할 코드  
} while (조건식);
```

예시:

```
let count = 0;  
  
do {  
    console.log(count);  
    count++;  
} while (count < 5);
```

위 코드에서는 `count`가 5보다 작을 때까지 반복되며, 0부터 4까지 출력됩니다. **do...while문**의 경우, 조건이 처음부터 거짓이라도 코드를 최소 한 번 실행한다는 점에서 **while문**과 차이가 있습니다.

while문과 for문 차이

- **for문**은 반복 횟수가 **명확할 때** 사용하며, 초기화식, 조건식, 증감식을 한 줄에 모두 포함할 수 있습니다.
- **while문**은 반복 횟수가 **명확하지 않을 때** 사용하며, 조건식만 소괄호 안에 작성하고, 초기화식과 증감식은 반복문 외부에서 처리합니다.
- **do...while문**은 **최소 한 번은 실행한 후** 조건을 평가하는 반복문입니다.

예시 비교:

```
// for문 예시  
for (let i = 0; i < 5; i++) {  
    console.log(i); // 0, 1, 2, 3, 4  
}
```

```

// while문 예시
let count = 0;
while (count < 5) {
    console.log(count); // 0, 1, 2, 3, 4
    count++;
}

// do...while문 예시
let count2 = 0;
do {
    console.log(count2); // 0, 1, 2, 3, 4
    count2++;
} while (count2 < 5);

```

위 예시들을 통해 for문, while문, do...while문의 차이와 사용 용도를 잘 이해할 수 있습니다. **while문**은 조건을 먼저 평가하고, **do...while문**은 실행 후 조건을 평가하는 특징을 잘 기억하세요!