영인터내셔널 사내교육

(웹 프로그래밍)

반복문

자바스크립트

YOUNG

Contents

학습목표

- 배열을 생성하고 사용하는 방법을 익힙니다.
- while 반복문과 for 반복문을 이해합니다.

Contents

내용

- 배열
- while 반복문
- for 반복문

0. 반복문

- 붙여 넣기를 사용한 반복 - 1000번 출력하는 것은 무리임

```
console.log("출력");
console.log("출력");
console.log("출력");
console.log("출력");
console.log("출력");
```

- 반복문을 사용한 반복

```
for (let i = 0; i < 1000; i++) {
    console.log("출력");
}
```

1. 배열

배열 생성 방법

- 여러 개의 자료를 한꺼번에 다룰 수 있는 자료형
- 대괄호 내부의 각 자료는 쉼표로 구분
- 배열에는 여러 자료형이 섞여 있을 수 있음

let 이름 = [자료, 자료, 자료, 자료, 자료]

```
> let array = [52, 273, '아침밥', '점심밥', true, false]
undefined
> array
[ 52, 273, '아침밥', '점심밥', true, false ]
```

1. 배열

배열의 요소와 인덱스

- 요소(Element)

  배열 안에 들어 있는 각 자료

  배열의 요소에 접근할 때는 대괄호를 사용

- 인덱스(Index): 대괄호 안에 넣는 숫자

1. 배열

- 배열 생성하고 요소에 접근
  - 인덱스의 시작 숫자는 0 임에 주의

```
// 배열을 생성합니다.
let array = [52, 273, '아침밥', '점심밥', true, false]

// 배열의 요소를 변경합니다.
array[0] = 0

// 요소를 출력합니다.
console.log(array[0]);
console.log(array[1]);
console.log(array[2]);
console.log(array[3]);
console.log(array[4]);
```

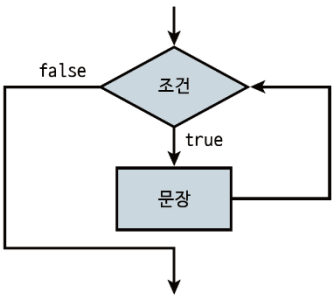
실행 결과

```
0
273
아침밥
점심밥
true
```

2. while 반복문

- 기본 형태

```
while (불_표현식) {
    // 불 표현식이 참인 동안 실행할 문장
}
```




```
while (true) {
    console.log("무한 반복");
}
```

2. while 반복문

- while 반복문 이용
  - 특정한 숫자를 증가시켜 불 표현식을 false로 만들어 반복문을 벗어남

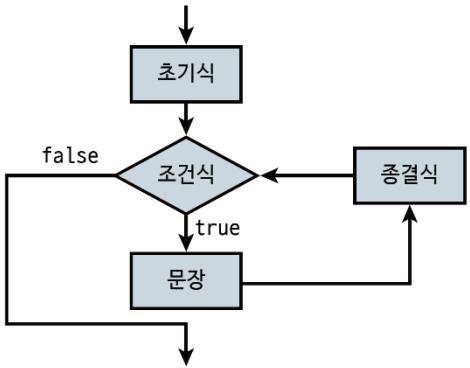
```
// 변수를 선언합니다.
let i = 0;
let array = [52, 273, 32, 65, 103];

// 반복을 수행합니다.
while (i < array.length) {
    // 출력합니다.
    console.log(i + "번째 출력:" + array[i]);

    // 반복문을 탈출하기 위해 변수를 더합니다.
    i++;
}
```

실행 결과

```
0번째 출력:52
1번째 출력:273
2번째 출력:32
3번째 출력:65
4번째 출력:103
```

3. for 반복문

- for 반복문의 각 단계
  - ❶ 초기식을 비교
  - ❷ 조건식을 비교
    조건이 false이면 반복문을
  - ❸ 문장을 실행
  - ❹ 종결식을 실행
  - ❺ 2단계로 이동



- 기본 형태

```
for (let i = 0; i < 반복_횟수; i++) {

}
```

3. for 반복문

- for 반복문을 이용한 덧셈
  - 0부터 100까지 더하기

```
// 변수를 선언합니다.
let output = 0;

// 반복을 수행합니다.
for (let i = 0; i <= 100; i++) {
    output += i;
}

// 출력합니다.
console.log(output);
```

실행 결과

```
5050
```

4. 중첩 반복문

- 반복문을 여러 번 중첩해서 사용

```
let output = '';

for (let i = 0; i < 10; i++) {
    for (let j = 0; j < i + 1; j++) {
        output += '*';
    }
    output += '\n';
}

console.log(output);
```

실행 결과

```
*
**
***
****
*****
******
*******
********
*********
**********
```

5. 스코프 Scope

스코프 Scope

- 변수를 사용할 수 있는 범위
- 스코프 == 블록

블록

- 중괄호로 둘러싸는 부분

5. 스코프 Scope

- 블록 내부에 선언된 변수는 해당 변수 내부에서만 사용가능

```
{
    let a = 10;
}

console.log(a);
```

- 반복문에 활용된 변수는 해당 블록에 있으므로 외부에서 활용할 수 없음

```
for (let i = 0; i < 3; i++) {
    console.log(i);
}

console.log(i);
```

변수 i는 해당 블록 외부에서 사용할 수 없습니다.

5. 스코프 Scope

- 스코프 내부에서 이름 중복

```
let a = 1;
let b = 1;
{
    let a = 2;
    {
        let a = 3;
        console.log(a);
    }
    console.log(b);
    console.log(a);
    console.log(b);
}
```

상위 블록에 있는 변수를 사용합니다.

같은 블록에 있는 변수를 사용합니다.

같은 블록에 있는 변수를 사용합니다.

같은 블록에 있는 변수를 사용합니다.