

Ch8 제어문

블록문

- 제어문은 조건에 따라 코드를 실행 혹은 반복 실행할때 사용한다.
- 블록문은 0개 이상의 문을 중괄호로 묶은 것
- 언제나 문의 종료를 의미하는 자체 종결성을 갖기 때문에 **세미콜론을 붙이지 않는다.**

조건문

조건문은 주어진 조건식의 평가 결과에 따라 코드 블록의 실행을 결정한다.

if ... else 문

조건식의 평가 결과 true일 경우 **if문**의 코드 블록이 실행되고 false일 경우 **else 문**의 코드 블록이 실행된다.

```
if (조건식1) {  
    //조건식1이 참이면 이 코드 블록이 실행된다.  
}  
else if (조건식2) {  
    //조건식2가 참이면 이 코드 블록이 실행된다.  
}  
else {  
    //조건식1과 조건식2가 모두 거짓이면 이 코드 블록이 실행된다.  
}
```

switch 문

주어진 표현식을 평가하여 그 값과 일치하는 표현식을 **case 문**으로 실행 흐름을 옮긴다.

break 문은 코드 블록에서 탈출하는 역할로 break 문이 없으면 case 문의 표현식과 일치하지 않더라도 실행 흐름이 다음 case 문으로 이동한다.

default 문에는 break 문을 생략한다.

```
switch (표현식) {  
    case 표현식1:  
        switch 문의 표현식과 표현식1이 일치하면 실행될 문;  
        break;  
    case 표현식2:  
        switch 문의 표현식과 표현식2가 일치하면 실행될 문;  
        break;  
    default:  
        switch 문의 표현식과 일치하는 case 문이 없을 때 실행될 문;  
}
```

반복문

조건식의 평가 결과가 참인 경우 코드 블록을 실행하며 조건식이 **거짓일 때까지 반복**한다.

- **for 문**
 - 조건식이 거짓으로 평가될 때까지 코드 블록을 반복 실행한다.

```
for (var i = 1; i >= 0; i--) {  
    console.log(i);  
}
```

```
}
```

- for 문의 변수 선언문, 조건식, 증감식은 모두 옵션으로 어떤 식도 선언하지 않으면 무한루프가 된다.

```
// 무한루프
for (;;) { ... }
```

- for 문 내에 for 문을 중첩해 사용할 수 있음

• while 문

- 주어진 조건식의 평가 결과가 참이면 코드 블록을 계속해서 반복 실행한다. 반복 횟수가 불명확할 때 주로 사용한다.
- 조건식의 평가 결과가 언제나 참이면 무한루프가 되고 무한루프에서 탈출하기 위해서는 if 문으로 탈출 조건을 만들고 break 문으로 코드 블록을 탈출한다.

```
var count = 0;

// 무한루프
while (true) {
  console.log(count);
  count++;
  //count가 3이면 코드 블록을 탈출한다.
  if (count === 3) break;
} // 0 1 2
```

• do ... while 문

- 코드 블록을 먼저 실행하고 조건식을 평가한다. 따라서 코드 블록은 무조건 한 번 이상 실행된다.

```
var count = 0;

// count가 3보다 작을 때까지 코드 블록을 계속 반복 실행한다.
do {
  console.log(count); // 0 1 2
  count++;
} while (count < 3);
```

break 문

코드 블록을 탈출한다.

```
// foo라는 식별자가 붙은 레이블 블록문
foo: {
  console.log(1);
  break foo; // foo 레이블 블록문을 탈출한다.
  console.log(2);
}

console.log('Done!');
```

continue 문

continue 문은 반복문의 코드 블록 실행을 현 지점에서 중단하고 반복문의 증감식으로 실행 흐름을 이동시킨다. break 처럼 반복문을 탈출하지 않는다.

```
var string = 'Hello World.';
var search = 'l';
var count = 0;

// 문자열은 유사배열이므로 for 문으로 순회할 수 있다.
for (var i = 0; i < string.length; i++) {
  // 'l'이 아니면 현 지점에서 실행을 중단하고 반복문의 증감식으로 이동한다.
  if (string[i] !== search) continue;
  count++; // continue 문이 실행되면 이 문은 실행되지 않는다.
}

console.log(count); // 3

// 참고로 String.prototype.match 메서드를 사용해도 같은 동작을 한다.
const regexp = new RegExp(search, 'g');
console.log(string.match(regexp).length); // 3
```

✓ if 문 내에서 실행해야 할 코드가 길다면 들여쓰기가 한 단계 더 깊어지므로 **continue 문**을 사용하는 편이 가독성이 더 좋다.