



# Chap8 제어문

[8-1 블록문](#)

[8-2 조건문](#)

[8-3 반복문](#)

[8-4 break문](#)

[8-5 continue문](#)

## 8-1 블록문

- 0개 이상의 문을 중괄호로 묶은 것.
- 코드블록, 블록이라고 부르기도 한다.
- 단독으로 사용하기보단 제어문이나 함수 정의할 때 사용
- 블록문은 자체 종결성을 갖기 때문에 끝에 세미콜론을 붙이지 않는다.

## 8-2 조건문

- 주어진 조건식(불리언 값으로 평가되는 표현식)의 평가 결과에 따라 블록문의 실행을 결정한다.
- 종류는 if~else문과 switch문이 있다.

### ▼ if~else

- 조건식은 불리언 값으로 평가되어야한다. 만약 다른 값으로 평가된다면 암묵적 타입변환으로 불리언 타입으로 강제 변환되어 실행할 코드 블록이 결정된다.

```
if( 조건식 ){  
    // 조건식이 참일 때 실행  
}else{
```

```
// 조건식이 거짓일 때 실행
}
```

- 조건을 더 추가하고 싶다면 else if문 사용하기

#### ▼ switch

- 주어진 표현식을 평가하여 평가 값과 일치하는 case문으로 실행 흐름을 옮긴다.
- case문은 상황을 의미하는 표현식을 적고 콜론으로 마무리한다. 그리고 그 뒤에 실행할 문들을 위치시킨다.
- 일치하는 case문이 없다면 default문으로 이동할 수 있다.
- default문은 선택사항인데 설정하지 않았는데 일치하는 case문이 없다면 아무것도 출력되지 않고 그냥 종료된다.
- break 키워드는 코드에서 탈출하는 역할을 한다. 적지 않을시 일치하지 않은 case 문이어도 실행 흐름을 옮긴다.
- default문은 break를 생략한다.

```
let fruit = 'banana';

switch (fruit) {
  case 'apple':
    console.log('Apple is selected');
    break;
  case 'banana':
    console.log('Banana is selected');
    break;
  case 'orange':
    console.log('Orange is selected');
    break;
  default:
    console.log('Sorry, we are out of ' + fruit + '.');
}
```

#### ▼ 폴스루

- `switch` 문에서 하나의 case 블록이 실행된 후에 다음 case 블록으로 이동하는 현상을 말한다.
- break 생략

보통은 break 키워드를 적어주는 경우가 더 많지만 폴스루가 더 유용할 때가 있다.

**ex** 윤년판별하여 2월 일수 구하기

## 8-3 반복문

- 조건식의 평가 결과가 참인 경우 코드 블록 실행
- 반복문 종류
  - for문
  - while문
  - do...while문

### ▼ for문

- 조건식이 거짓으로 평가될 때까지 코드 블록을 반복 실행한다.
- 반복 횟수가 명확할 때 사용

```
for( 변수 선언문 또는 할당문; 조건식; 증감식){  
    조건식이 참일 경우 실행될 문;  
}
```

- 선언문, 조건식, 증감식 모두 옵션

### ▼ 사용안할시

무한루프가된다

```
for(;;){...} //무한루프
```

- for문 안에 for문 중첩 가능

### ▼ while문

- 조건식의 평가 결과가 참이면 코드 블록 반복 실행
- 반복 횟수가 불명확할 때 사용
- 조건식의 평가값이 불리언이 아니면 불리언으로 암묵적 타입 변환을 한다.
- 조건식이 언제나 참이면 무한루프가된다.

```
while(true){...}
```

### ▼ do...while문

- 조건식이 아닌 코드 블록을 먼저 실행 → 코드 블록 무조건 한번 이상 실행됨

## 8-4 break문

- 레이블 문, 반복문, switch문을 탈출하기 위해 사용(이외에 코드 블록에서 사용하면 SyntaxError문법 에러가 발생한다.)

### ▼ 레이블 문

```
label:  
    statement;
```

label

JavaScript에서 사용할 수 있는 식별자면 모두 가능하다.

statement

break는 모든 레이블 구문에서 사용될 수 있으며, continue는 반복 레이블 구문에 서만 사용할 수 있다.

## 8-5 continue문

- 반복문의 코드 블록 실행을 현재 시점에서 중단하고 반복문의 증감식으로 실행을 옮긴 다.

```
for(var i=0; i<string.length; i++){  
    if(string[i]=== search){  
        count++;  
    }  
}
```

위 예제와 같이 if문 안에 실행해야할 문장이 한개라면 continue를 안쓰는 것이 좋지만 여러 문장이라면 쓰는 것이 가독성 면에서 좋다.