

# 자바스크립트 (Javascript)

✓ 원리를 알면 IT가 맛있다

**JavaScript for Beginners**



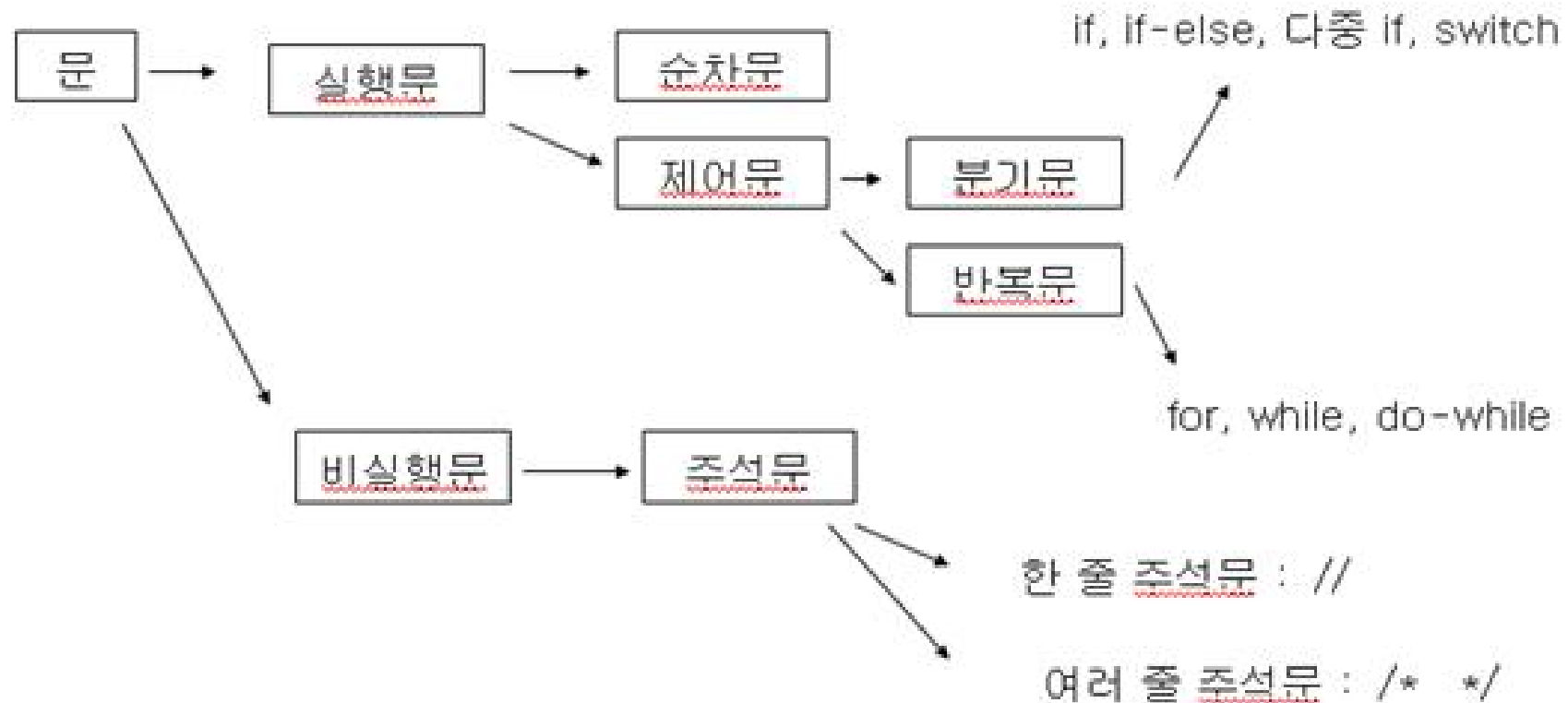
chapter 04.

---

문장

- 자바스크립트에서 사용 가능한 문장의 종류에 관하여 학습한다.
- 실행문과 비실행문에 관하여 학습한다.
- 조건에 따라서 실행되는 if문, if~else문, 다중 if문, switch 문에 관하여 학습한다.
- 조건에 따라서 반복적으로 수행되는 while문, do~while문, for문에 관하여 학습한다.
- break와 continue을 이용한 제어문 활용법을 학습한다.

- ‘문(statement)’은 프로그램을 개발하기 위해서 자바스크립트 소스코드에 입력시키는 문장을 의미한다.



### ○ 주석문

- 실제 프로그램에 영향을 주지 않으며 단지 소스코드의 기능이나 동작을 설명하기 위해 사용되는 문장이다.

[표 3-1] 주석문의 종류

주석 종류	의미	설명
// 주석문	단행 주석처리	현재 행에서 //의 뒷문장부터 주석으로 처리된다.
/* 주석문 */	다행 주석처리	/*에서 */ 사이의 문장이 주석으로 처리된다.

### ○ 순차문

- 소스코드내의 문장 중에서 순차적으로 실행되는 문장을 의미한다.
- 반드시 ;(세미콜론)으로 끝나며, 소스코드의 대부분이 순차문에 해당된다.

### ○ 제어문

- 프로그램의 흐름에 영향을 주고 때에 따라 제어가 가능하도록 하는 것이 바로 '제어문'이다.
- 모든 제어문은 중첩이 가능하다.

### ○ 제어문의 종류

#### ■ 분기문 (비교문)

: 주어진 조건의 결과에 따라 실행 문장을 다르게 하여 전혀 다른 결과를 얻기 위해 사용되는 제어문이다.

- 단일 if문, if~else문 , 다중 if 문 , switch문

#### ■ 반복문

: 특정한 문장을 정해진 규칙에 따라 반복처리하기 위한 제어문이다.

- for문, while문, do~while문

- break문 : 반복문 내에서 쓰이며 반복문을 빠져나갈 때 쓰이는 제어문이다.  
switch문에서 사용시 switch블럭을 빠져나간다.

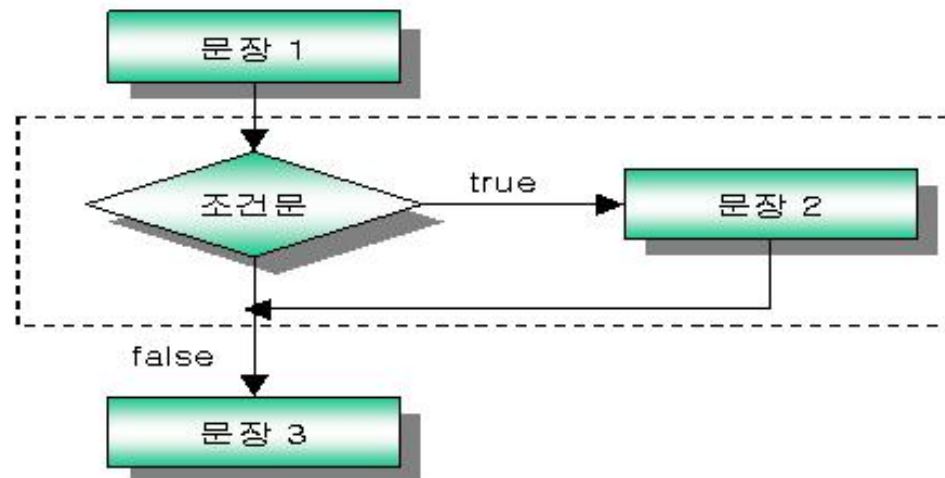
- continue문 : 현재 진행되는 반복 회차를 포기하고 다음 회차로 이동 한다.

### ○ 단일 if 문

주어진 조건을 만족하는 경우에만 특정 문장을 수행하도록 제어하는 문이다.

문법:

```
문장1;  
if(조건식){  
  문장2;  
}  
문장3;
```





### ○ 특이사항

자바스크립트 변수는 블록단위로 관리되지 않고 함수단위 스코프로 동작된다.

예>

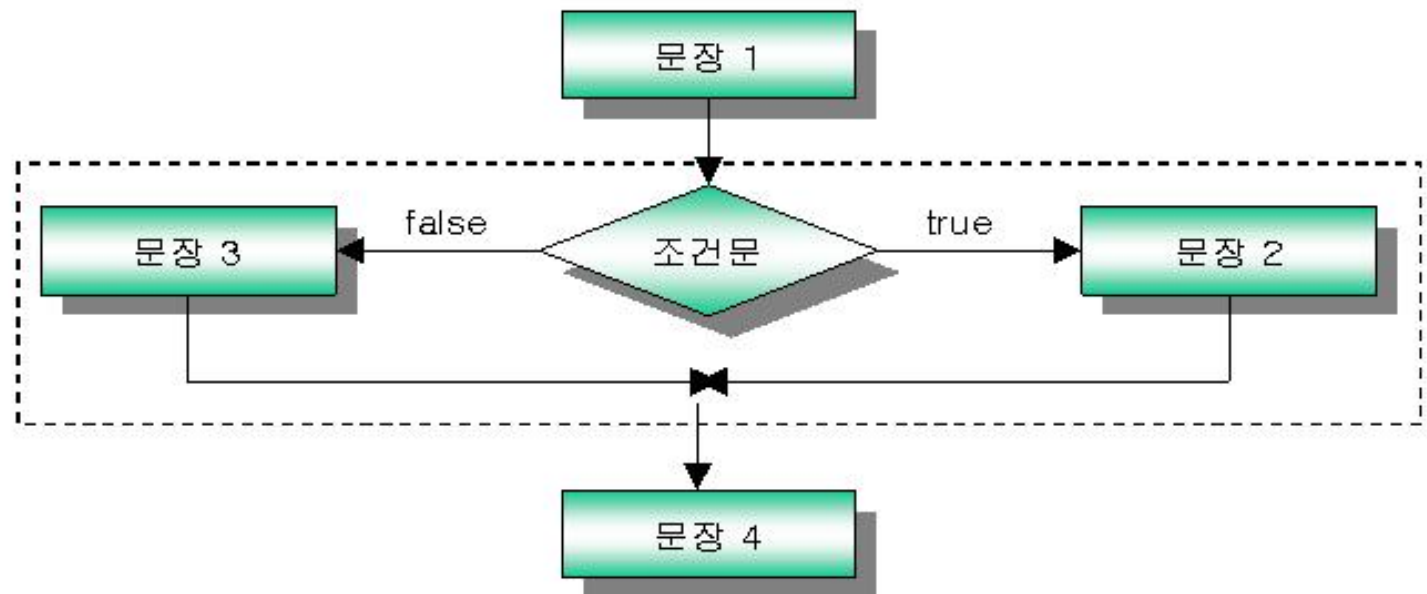
```
if( 3 > 2){  
    var local = 10;  
}  
console.log(local); // 10
```

### ○ if ~ else 문

조건식의 결과에 따라서 실행되는 문장이 서로 다른 경우에 사용한다.

문법:

```
문장1;  
if(조건식){  
  문장2;  
}else{  
  문장3;  
}  
문장4;
```

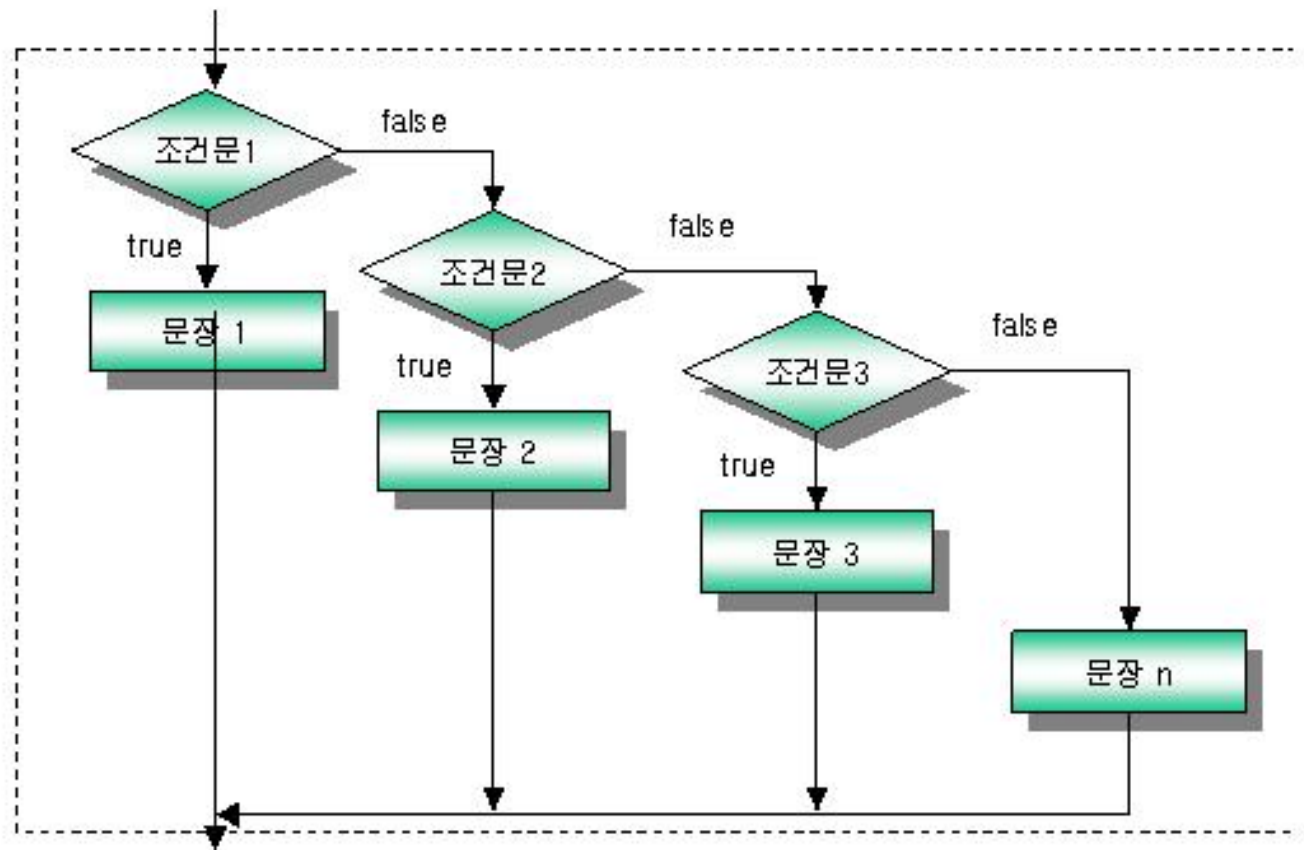


### ○ 다중 if ~ else 문

비교할 조건식이 여러 개인 경우에 사용된다.

문법:

```
if(조건식1){  
    문장1;  
}else if(조건식2){  
    문장2;  
}else if(조건식3){  
    문장3;  
}else{  
    문장n;  
}
```



- 다중 if ~ else 문과 비슷한 용도로 사용된다.

```
switch(인자값) {  
  
  case 조건값1 :  
      수행문; break;  
  
  case 조건값2 :  
      수행문; break;  
  
  case 조건값3 :  
      수행문; break;  
  
  default :   수행문;  
}
```

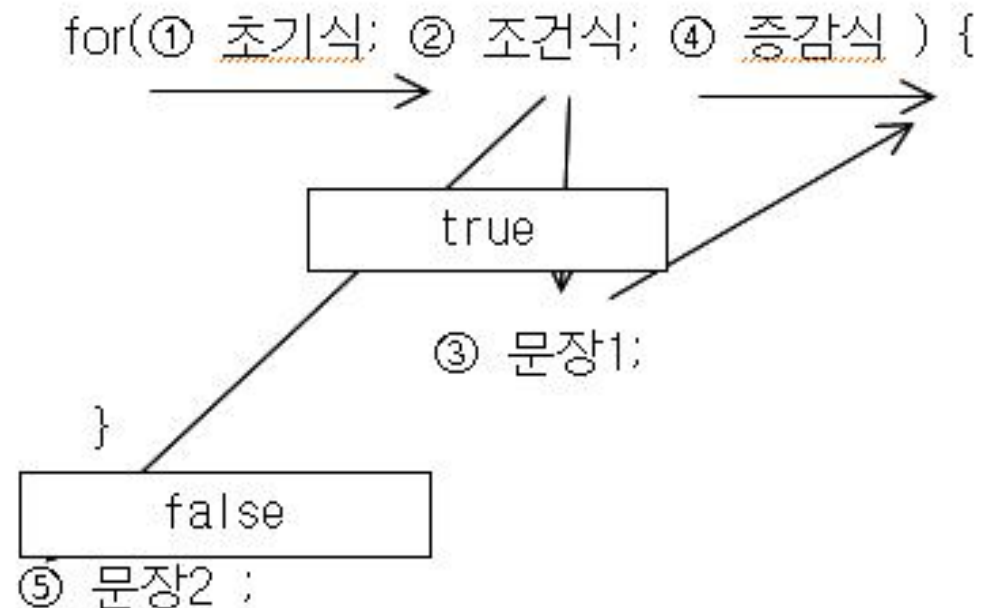
## ○ 지정된 횟수만큼 반복 처리하는 제어문이다.

초기식, 조건식, 증감식으로 구성된다.

일반적으로 반복횟수가 예측 가능할 때 주로 사용된다.

문법:

```
for(초기식;조건식;증감식){
    문장1;
}
문장2;
```



### ○ 특이사항

자바스크립트 변수는 블록단위로 관리되지 않고 함수단위 스코프로 동작된다.

예>

```
function f( count ){  
    alert( i );    // undefined  
    for( var i = 0 ; i < count ; i++ ){  
        ..  
    }  
    return i;  
}  
f(100);    // 100
```

- for문과 문법적인 형태만 다르며 동일한 방식으로 동작한다. 차이점은 for문은 초기식, 조건식, 증감식이 지정된 위치가 존재하지만, while문은 조건식만 정해져 있기 때문에 초기식과 증감식은 적당한 위치에 알아서 지정해야 된다. 따라서 반복횟수 예측이 힘들기 때문에 예측 불가능한 형태의 반복문에 주로 사용된다.

문법:

```
    초기식;  
while(조건식){  
    문장;  
    증감식;  
}
```

- while문과 비슷하지만, 차이점은 조건식이 일치하지 않더라도 반드시 한번은 문장이 실행된다.

따라서 조건이 일치하지 않더라도 적어도 한번은 꼭 수행되어야 하는 경우에 사용 가능하다.

문법:

```
    초기식;  
do{  
    문장;  
    증감식;  
}while(조건식);
```



- 반복문내에 또다른 반복문을 중첩으로 지정하여 처리하는 문이다.  
for문, while문, do~while문 모두 사용 가능하다.

```
for(초기식1 ; 조건식1 ; 증감식1) {
```

```
    for(초기식2 ; 조건식2 ; 증감식2){  
        명령어2;  
    }
```

```
}  
명령어3;
```

- 배열이나 객체를 쉽게 처리할 수 있도록 지원하는 반복문.

문법:

```
for (가변수 in 배열/객체) {  
    루프 내에서 실행하는 명령(군)  
}
```

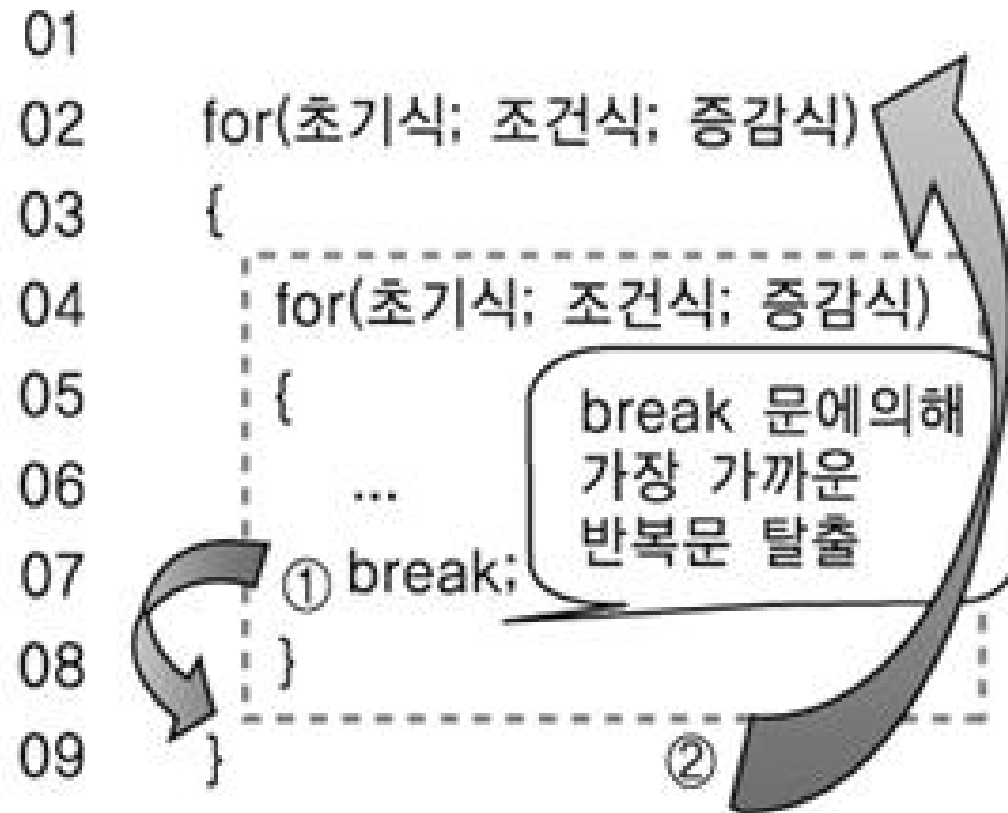
가변수에는 배열/연상배열이나 객체로부터 얻은 **요소의 인덱스값이나 연상배열의 키값, 멤버명이 저장되어** for in 블록안에서 요소값을 참조할 때 사용할 수 있다.

**가변수에 할당되는 것은 요소값 자체가 아니라 배열의 경우에는 인덱스값이고 객체의 경우에는 프로퍼티명이 된다.**

### ○ break문

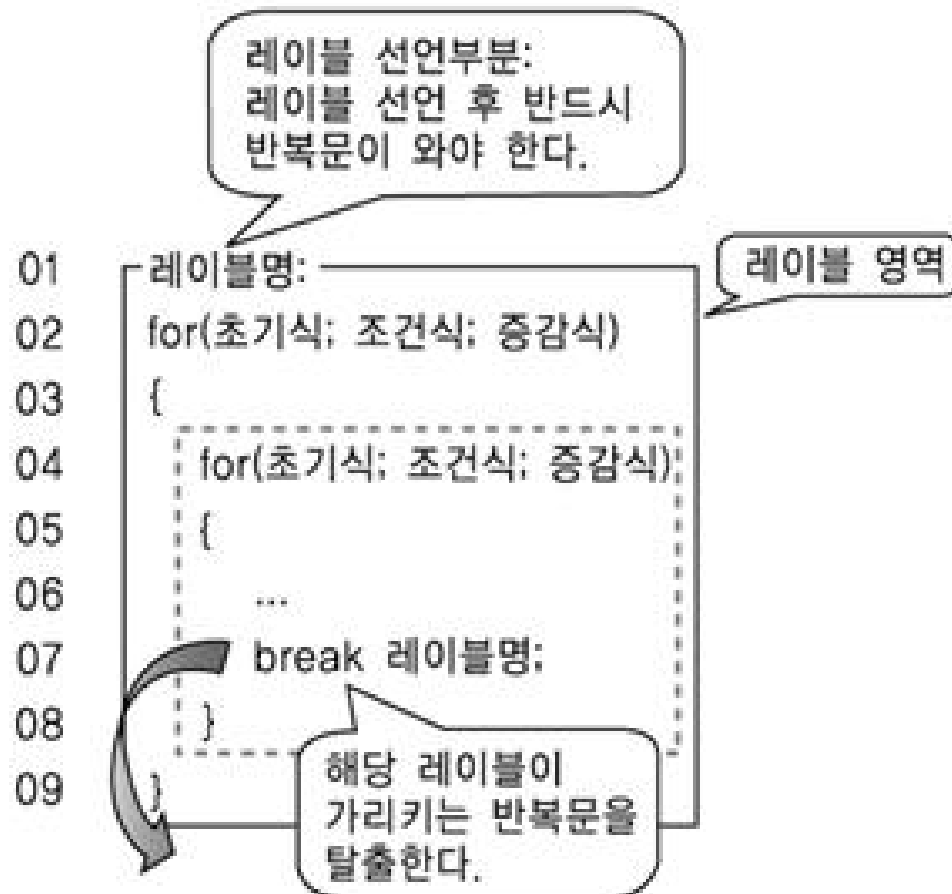
- 가장 가까운 반복문을 탈출할 때 쓰이는 제어문이다.

■



### ○ break label문

- break label은 break문과 같지만 다중 반복문에서 한번에 바깥쪽 반복문을 탈출할 때 많이 쓰이는 제어문이다.



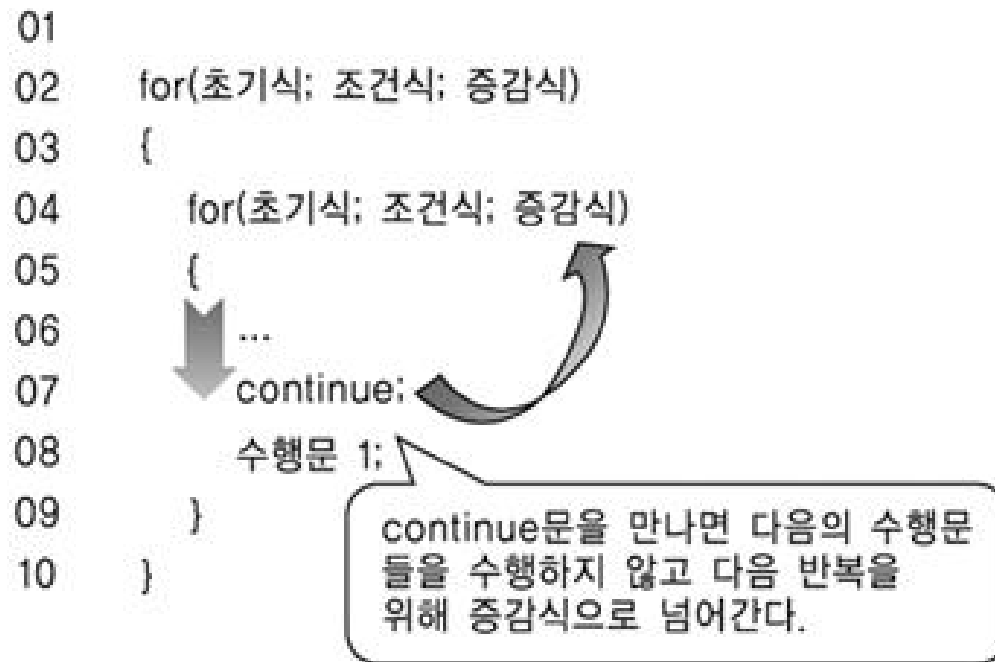
예>

OUTER:

```
for(var i = 0 ; i < 10 ; i++ ){
    for(var k = 0 ; k < 5 ; k++ ){
        console.log( i + " " + k);
        if( k == 2) break OUTER;
    }
}
```

### ○ continue문

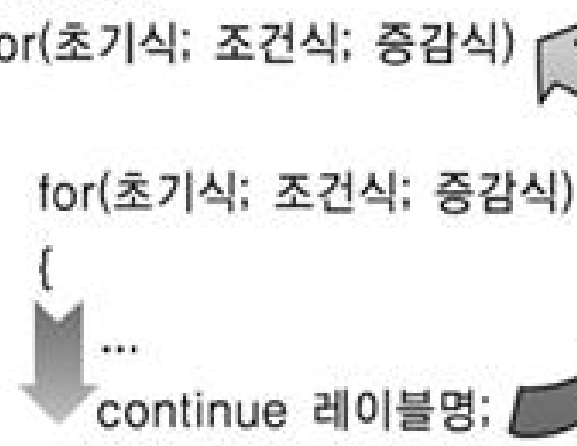
- 반복문을 탈출하기위해 사용되는 것이 아니라 continue문 이하의 수행문들을 포기하고 다음 회차의 반복을 수행하기위한 제어문이다.  
동작방식은 continu문을 만나면 반복문의 블록 끝(})으로 간다. 따라서 continue문 바로 아래 문장이 skip 되는 효과가 있다.



### ○ continue label문

- 레이블을 가지는 continue문은 레이블이 지칭하는 반복문의 조건식 또는 증감식으로 프로그램상 수행 시점(제어권)이 이동한다.

```
01   레이블명:
02   for(초기식; 조건식; 증감식)
03   {
04       for(초기식; 조건식; 증감식)
05       {
06           ...
07           continue 레이블명;
08           수행문 1;
09       }
10   }
```



내부 반복문을 중단하고 외부 반복문의 다음 반복회차를 수행하기 위한 제어문이다.  
continue문 아래의 수행문 1은 수행하지 못한다.

- 자바스크립트에서 사용 가능한 문장의 종류
- 실행문과 비실행문
- 조건문, 반복문
- 향상된 for 문
- break, continue 문



**Thank you**

---