



4강 - 제어문

조건문

조건문 if

else 절

다중 분기 조건문 else if 절

반복문

반복문 while

반복문 for

중첩 반복문

반복문 빠져나오기 (강제 종료)

break

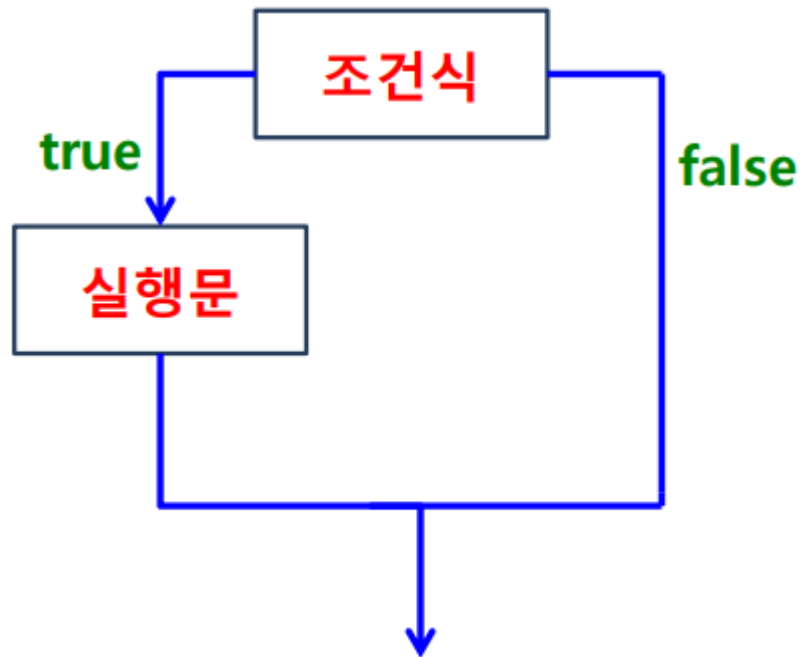
무한 루프

다음 반복으로 넘어가기 (반복문 건너뛰기)

continue

조건문

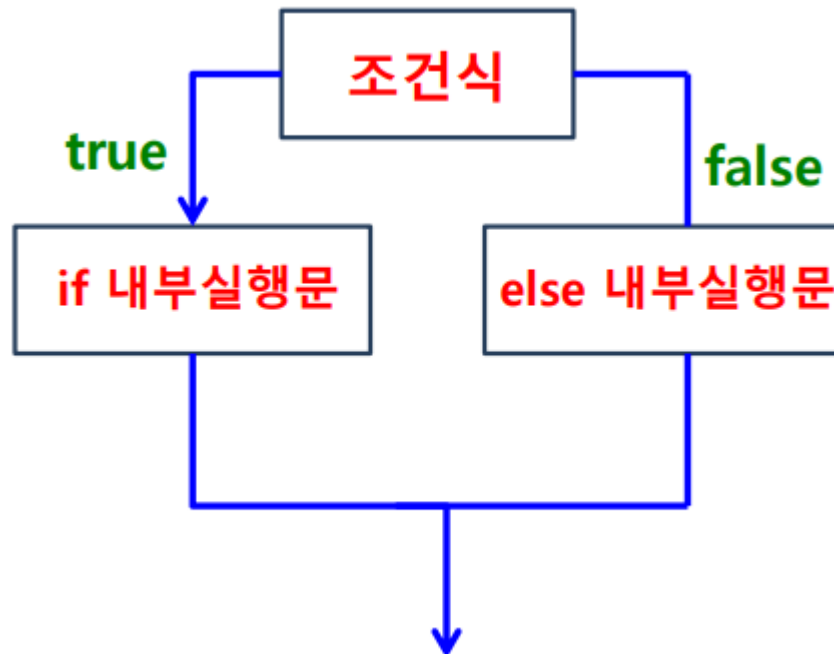
조건문 if



- 조건문은 프로그램에서 조건식의 참, 거짓에 따라 코드를 다르게 실행하게 하는 분기점을 만드는 제어문입니다.
- if는 조건식의 논리결과가 참일 경우 블록 내의 코드를 실행하며 거짓일 경우 코드를 실행하지 않습니다.
- if블록 내부의 코드가 단 한 줄일 경우 블록(중괄호)을 생략할 수 있습니다.

```
let year = prompt('ECMAScript-2015 명세는 몇 년도에 출판되었을까요?', '');  
  
if (year == 2015) {  
    alert( "정답입니다!" );  
    alert( "아주 똑똑하시네요!" );  
}
```

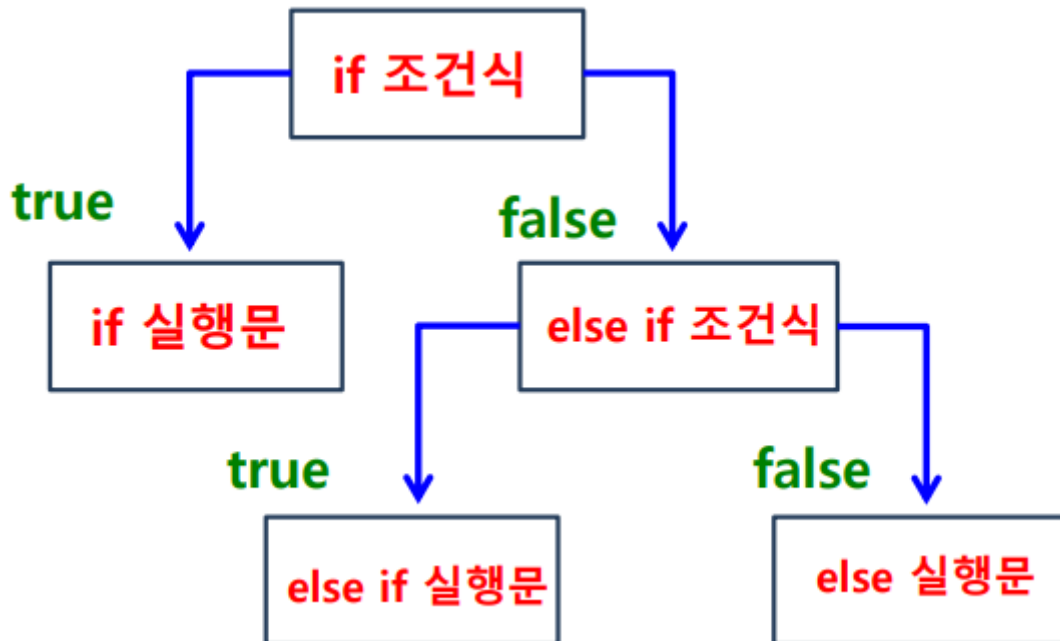
else 절



- `if` 문엔 `else` 절을 붙일 수 있습니다. `else` 뒤에 이어지는 코드 블록은 조건이 거짓일 때 실행됩니다.
- `else` 키워드는 `if`문과 반드시 함께 사용해야 하며 단독으로 사용할 수 없습니다.
- `else` 블록에도 실행문이 단 한 문장이라면 블록을 생략할 수 있습니다.

```
var year = prompt('ECMAScript-2015 명세는 몇 년도에 출판되었을까요?', '');  
  
if (year == 2015) {  
    alert( '정답입니다!' );  
} else {  
    alert( '오답입니다!' ); // 2015 이외의 값을 입력한 경우  
}
```

다중 분기 조건문 else if 절

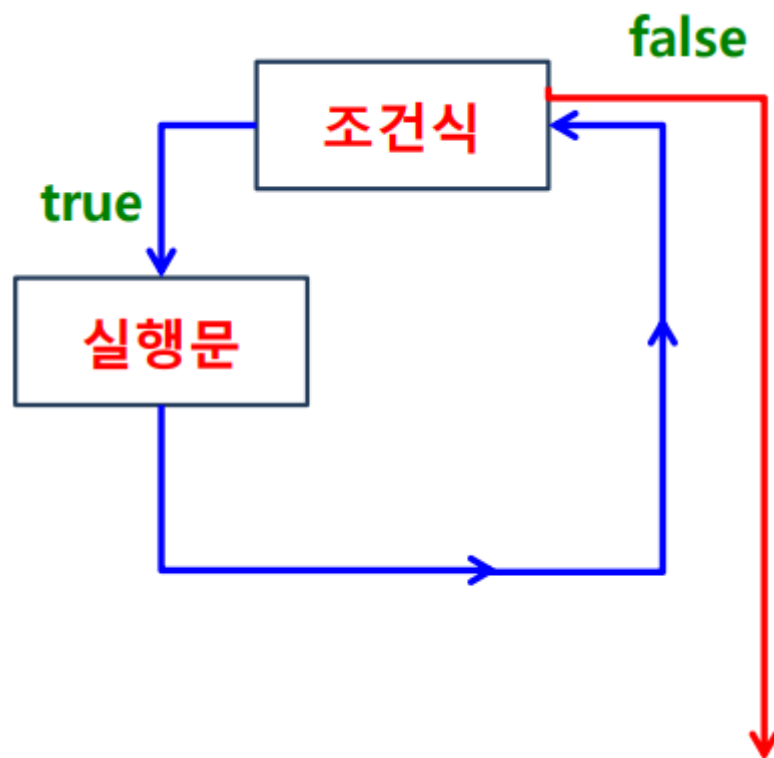


- if ~ else if문은 좀 더 많은 분기를 만들고 싶을 때 사용하는 조건문입니다.
- 다만 위에서부터 조건을 검색하면서 내려오기 때문에 범위 조건일 경우 상위 조건이 하위 조건을 포괄적으로 포함하지 않도록 주의해야 합니다.

```
var year = prompt('ECMAScript-2015 명세는 몇 년도에 출판되었을까요?', '');  
  
if (year < 2015) {  
    alert( '숫자를 좀 더 올려보세요.' );  
} else if (year > 2015) {  
    alert( '숫자를 좀 더 내려보세요.' );  
} else {  
    alert( '정답입니다!' );  
}
```

반복문

반복문 while



- while문은 조건식을 검사하여 참일 경우 블록 내부의 코드를 실행하며 실행이 끝날 때마다 반복적으로 조건식을 검사하여 false가 나올 때까지 반복합니다.
- while문도 if와 마찬가지로 논리값이 도출되는 조건식이나 함수를 사용합니다.
- while문은 반복 횟수를 제어하기 위한 증감식이나 탈출문이 필요합니다.

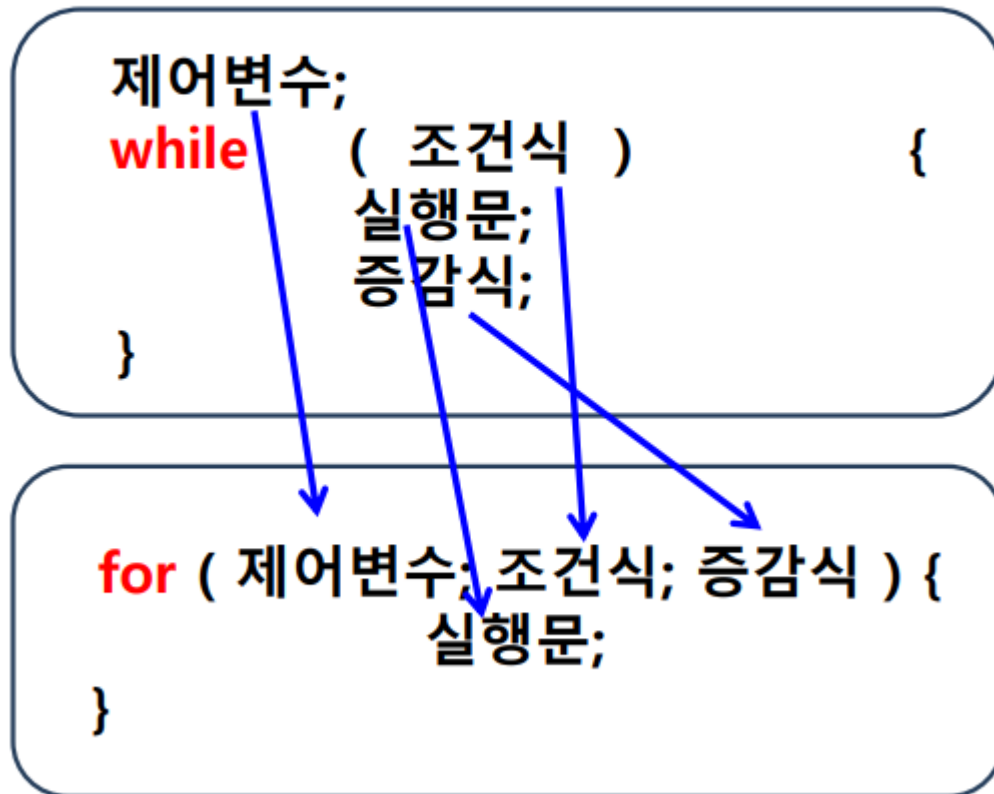
```
while (condition) {  
  // 코드  
  // '반복문 본문(body)'이라 불림  
}
```

```
var i = 0;  
while (i < 3) { // 0, 1, 2가 출력됩니다.  
  alert( i );  
  i++;  
}
```

- 반복문 본문이 한 번 실행되는 것을 *반복(iteration, 이터레이션)* 이라고 부릅니다. 위 예시에선 반복문이 세 번의 이터레이션을 만듭니다.

반복문 for

< while문과 for문 비교 >



- for문은 반복 제어조건을 한꺼번에 지정한다는 점이 다른 반복문과는 다릅니다.
- 따라서 정확한 반복 횟수를 알고 있을 때는 for문이 while문보다 가독성이 좋습니다.

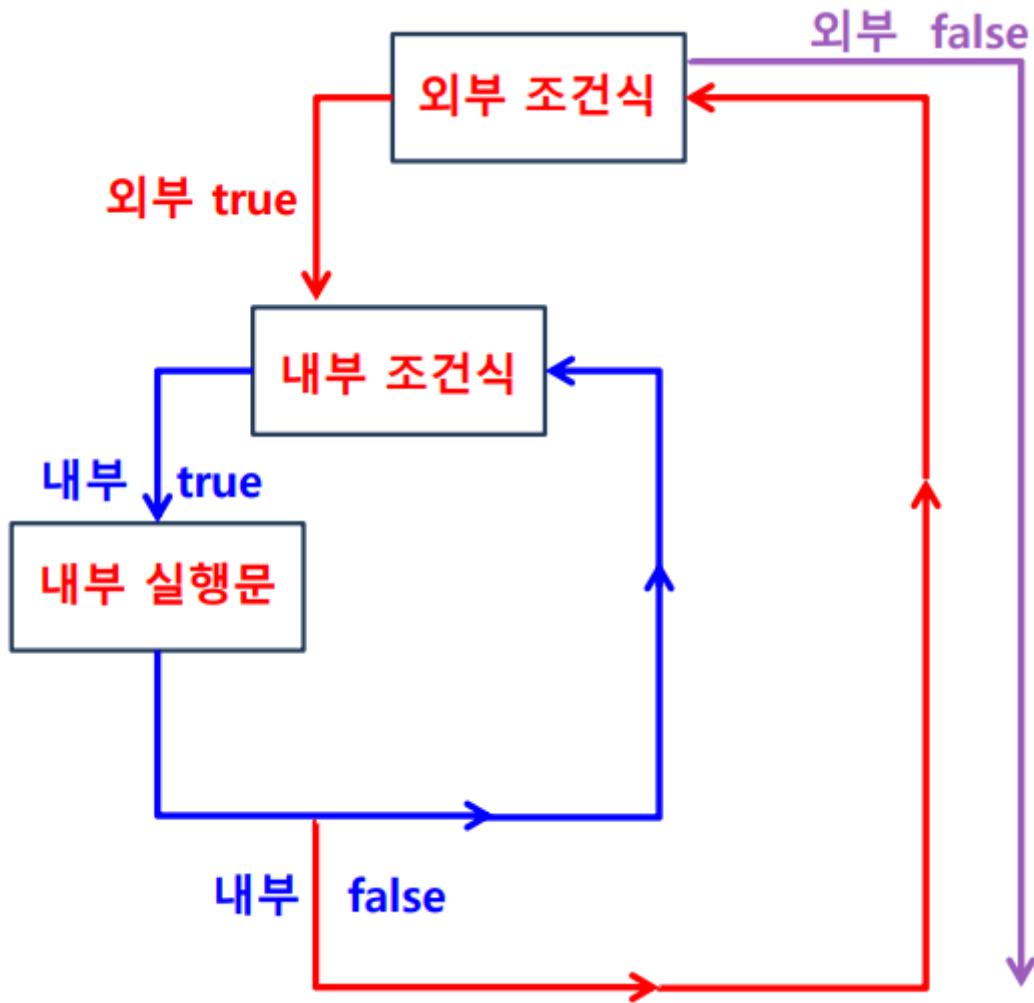
```
for (begin; condition; step) {
    // ... 반복문 본문 ...
}
```

begin	<code>i = 0</code>	반복문에 진입할 때 단 한 번 실행됩니다.
condition	<code>i < 3</code>	반복마다 해당 조건이 확인됩니다. false이면 반복문을 멈춥니다.
body	<code>alert(i)</code>	condition이 truthy일 동안 계속해서 실행됩니다.
step	<code>i++</code>	각 반복의 body가 실행된 이후에 실행됩니다.

```
var total = 0;  
for (var ①n = 1; n ②<= 10; n③++) {  
    total += n; ④  
}
```

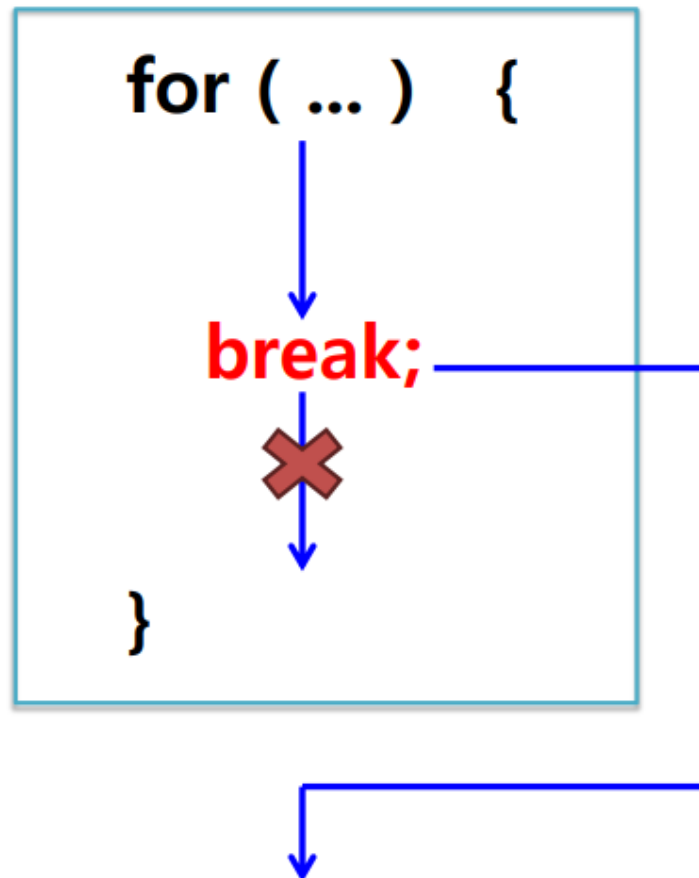
- 실행순서는 ①제어변수 선언 -> ②조건식 판단 -> ④실행문 -> ③증감식 -> ② 조건식 판단 -> ④실행문 -> ③증감식 순서로 진행되므로 순서에 주의하세요!

중첩 반복문



반복문 빠져나오기 (강제 종료)

break



- break문은 for문이나 while문에서 사용되며 반복문이 실행 중 break를 만나는 순간 반복문을 해당 시점에서 종료합니다.
- break를 만나는 순간 이하의 반복문의 남은 코드는 모두 실행되지 않으며 블록을 탈출합니다.
- 대개 if문과 함께 사용되며 조건에 따라 반복문을 종료할 때 사용합니다.

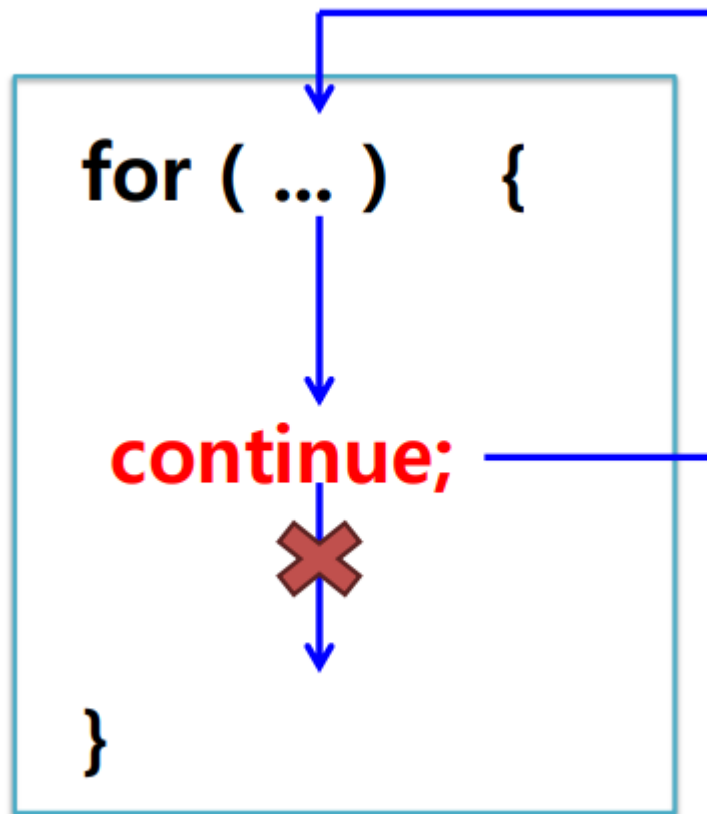
무한 루프

```
while ( true ) {  
  
    if( ... ) {  
  
        break;  
  
    }  
  
}
```

- 무한 루프는 반복문의 반복횟수를 개발자가 사전에 정확하게 인지하지 못하는 상황에서 사용하며 특정 조건 하에서 반복문을 강제로 종료하는 형태로 구성합니다.
- 프로그램이 중단되지 않게 유지할 때도 무한루프를 사용합니다.

다음 반복으로 넘어가기 (반복문 건너뛰기)

continue



- continue문은 for문이나 while문에서 사용되며 반복문이 실행 중 continue를 만나는 순간 for문의 경우 증감식, while문의 경우 조건식으로 이동합니다.
- continue는 break와 달리 반복문을 종료하지 않고 계속 수행합니다.
- 조건부로 특정 반복회차를 건너뛸 때 사용합니다.

reference:

if와 '?'를 사용한 조건 처리



<https://ko.javascript.info/ifelse>



JAVASCRIPT.INFO
The Modern JavaScript Tutorial

while과 for 반복문



<https://ko.javascript.info/while-for>



JAVASCRIPT.INFO
The Modern JavaScript Tutorial