



조건 반복문

- 분기문(**if**)에 대하여 기본적인 코드 구조를 알고 활용할 수 있다.
- 반복문(**for, while, do while**)의 구조를 알고, 각 반복문의 특징을 파악하여 활용할 수 있다.
- 반복문과 분기문을 적절하게 조화하여 원하는 데이터를 가져올 수 있다.
- 반복문에 쓰이는 **break, continue**문을 구분하여 활용할 수 있다.
- 배열이란 무엇인지 알고, 기본적인 **code**를 쓸 수 있다.
- 반복문과 배열에서 쓰이는 **keyword**를 효과적으로 활용할 수 있다.



생각해봅시다! :

- 프로그래밍에서 조건은 어떤 경우에 쓰일 것인가?
 - 효과적인 조건문을 쓰기 위해서 어떤 연습일 필요할 것인가?
 - 반복문은 단순한 기계적으로 **list**만 하는 작업일까?
 - 배열과 반복문은 어떤 관계에 있을까?
-



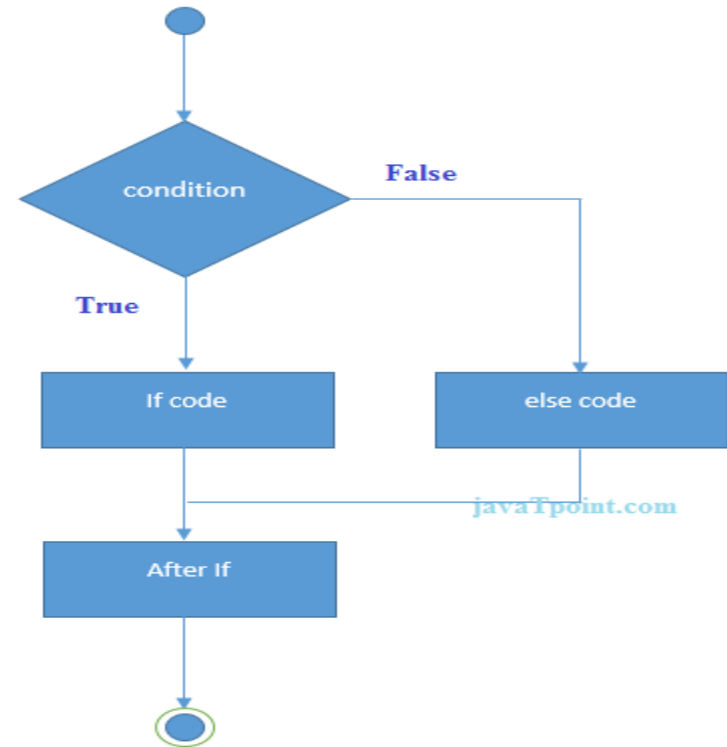
자바스크립트 현재 어디즈음:

- 개요
 - 기본문법
 - **조건 반복문**
 - 함수
 - 객체
 - 생성자 함수
 - 기본 내장 객체
 - 브라우저 객체 모델
 - 문서 객체 모델
 - 이벤트
 - 예외처리
-



if조건문:

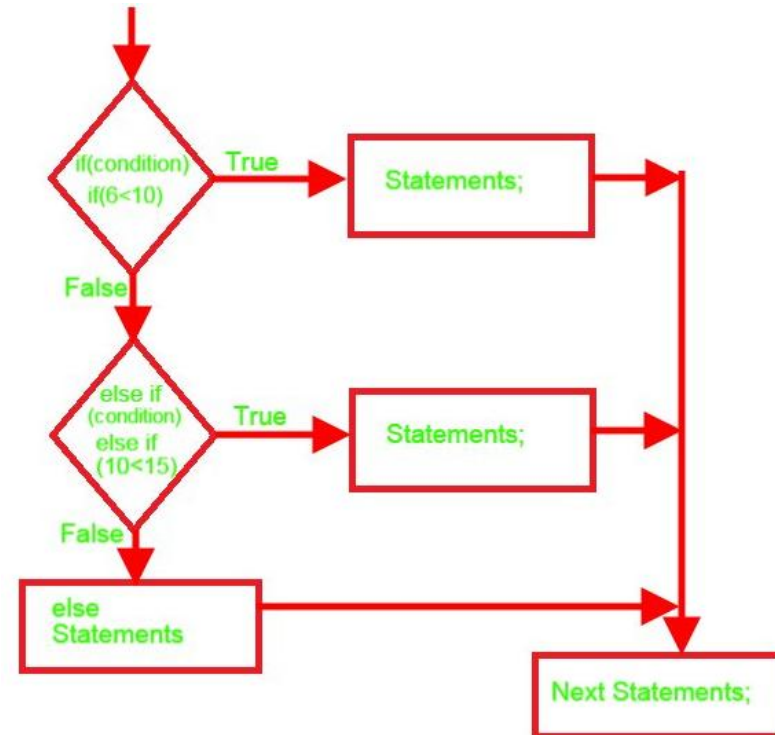
- **if(불 표현식){**
 해당 조건에 처리되는 내용
}
- **if(불 표현식){**
 해당 조건이 참일 때,
}else{
 해당 조건이 거짓일 때,
}





if else if 조건문 :

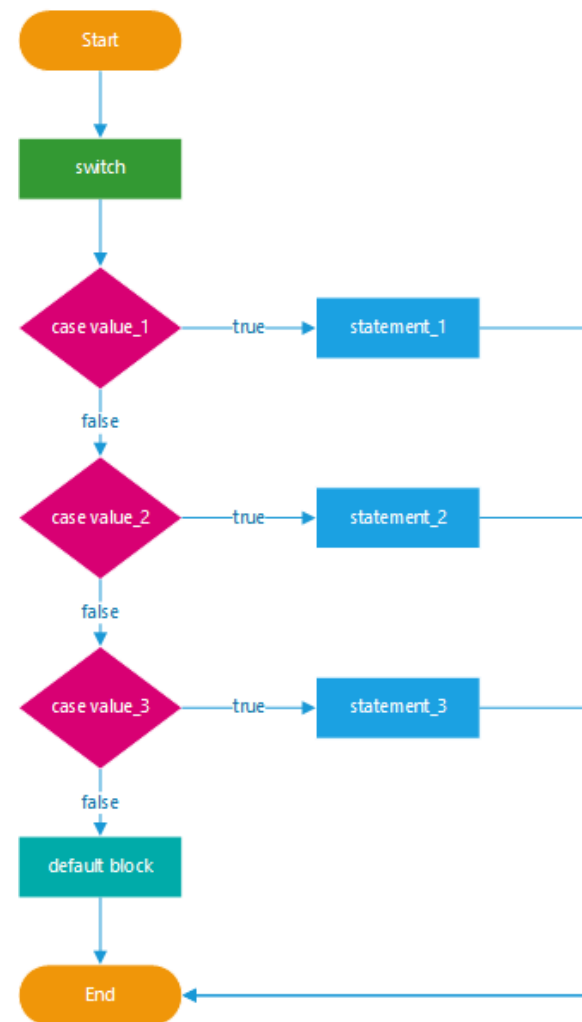
- **if(불 표현식){**
 해당 조건이 참일 때,
}else if(불 표현식){
 상위 조건을 제외한 조건식
}else{
 위 상위 if조건 제외
}





switch문 :

- **switch(비교 값){**
 case 값 : 문장 break;
 case 값 : 문장 break;
 default : 문장 ;
- **break;**가 없을 때?





삼항 연산자 :

- (불 표현식)?참일 때 실행하는 문장:거짓일 때 실행하는 문장;

$x = (y > 0) ? 10 : 20;$

y가 0보다 크면 10을,
크지않다면 20을 반환해라!

$x = (y > 0) ? a * b : a / b$

y가 0보다 크면 $a * b$ 의 결과를,
아니면 a / b 의 결과를 반환해라!



- 문자열을 반복해서 처리할 때.
 - `alert("안녕하세요");`
 - `alert("안녕하세요");`
 - `alert("안녕하세요");`
- 효과적으로 반복하는 하는 방법이 있을까?
- `for(초기값 ;한계범위;증감연산자){`
 반복할 데이터
 }
 `for(var cnt=1;cnt<=100;cnt++){`
 `console.log(cnt+"번째 인사!");`

■ 배열이란?

- 여러 개의 변수를 한꺼번에 다룰 수 있는 자료형
- 선언과 할당
 - `var array=[3000,2000,1000,5000];`
- 배열 접근
 - 배열명[index번호-0부터 시작]
 - ex) `array[0]` : 첫번째 데이터
- 배열의 크기 : 배열명.length
- 배열 추가
 - 배열명[index]="해당데이터"
 - 배열명.push(입력할 데이터)



배열과 loop:

- for문에서 배열 활용

- 배열의 크기: 배열.length

- index번호 : 0부터 ~ 배열크기-1

```
for(var idx=0;idx<배열.length;idx++){  
    배열[idx]  
}
```



while 반복문 :

- 무한반복의 처리
 - while(조건)
 - 반복한 프로세스
- 우선 한번은 실행! 그 다음 반복
 - do{
 - 1번이상 실행할 문
 - }while(조건);



for in 반복문 :

- `for(단위데이터 in 배열){`
 단위데이터가 어떤 형태이냐에 따라 처리가
 바뀜.
}
- `var list = ["사과", "바나나", "딸기"];`
`for(var fruitIdx in list){`
 `alert(list[fruitIdx]);`
}



중첩반복문 :

- 반복문을 중첩으로 쓰여서, 효과적인 로직을 처리하는 것을 말한다.
- `for(초기;한계치;증감연산자){ // 상위 for문`
// 상위for문에 쓰인 변수 하위 for문에 영향
`for(초기;한계치;증감연산자){ // 하위 for문`
이중 처리 반복문.
}
}



break 키워드 :

- switch 조건문이나 반복문을 벗어날 때 사용하는 키워드
- 반복문은 무한반복하는데, 이를 벗어날 때 사용하는 키워드

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



continue 키워드 :

- 반복문 내에서 현재 반복을 멈추고 다음 반복을 진행시키는 키워드
- 반복문 가운데 특정 단계에서는 그 단계에서만 반복을 pass시키는 경우를 말한다.

```
for( var grade=1;grade<=10;grade++){  
    if(grade==5){ //해당 단계만 하위내용처리X  
        continue;  
    }  
    console.log("현재 단계"+grade)  
}
```



정리 및 과제 :

- 반복문은 크게 어떤 종류들이 있는가?
 - 각 반복문의 특징이 무엇인지 구별할 수 있고, 어떤 상황에서 쓰이는가?
 - 조건문과 **continue**, **break**문은 어떤 경우에 쓰이는가?
 - 배열에서 쓰이는 주요 키워드는 무엇이며, 반복문에서 어떻게 활용되는가?
 - 배열에서 정렬과 이중반복문과의 처리 로직을 찾아보세요.
 - 이중반복문을 활용한 **star(*)**예제 확인해보세요.
-



다음 장은 함수 입니다 !