

## 8장 제어문

### 8.0 머릿말

제어문에는 블록문,  
조건문이 있다.

제어문 사용은 코드의 흐름을 이해하기 어렵게  
만들어 가독성을 해치는 단점이 있다.

forEach, map 등의 고차함수를 사용한 함수형  
프로그래밍 기법에서는 제어문의 사용을 억제하여  
복잡성을 해결하려고 노력한다.

### 8.1 블록문

0개 이상의 문을 중괄호로 묶은 것이며  
코드 블록, 블록 이라고 부른다.

js는 블록문을 하나의 실행  
단위로 취급한다.

단독으로도 사용이 가능하지만, 일반적으로  
제어문이나 함수를 정의할 때 사용

블록문 끝에는  
세미콜론을 붙이지  
않는다

자제 종결성을 갖기 때문에

### 8.2 조건문

주어진 조건식의 평가 결과에 따라  
코드 블록의 실행을 결정

조건식은 불리언 값으로  
평가될 수 있는 표현식이다.

#### 8.2.1 if ... else 문

조건식에 불리언 값이 없으면 암묵적  
타입 변환이 일어나기도 한다.

코드 블록 내 문이 하나라면  
중괄호를 생략할 수 있다.

대부분의 if else 문은 삼항 조건  
연산자로 바꿔 쓸 수 있다.

조건에 따라 단순히 값을 결정하여 변수에 할당하는  
경우 삼항 조건 연산자가 가독성이 좋다

조건에 따라 복잡한 여러 줄의 문이 필요하다면 if  
else 문을 사용하는 편이 가독성에 좋다.

#### 8.2.2 switch 문

Default 문은 선택사항이다.  
break는 안 넣는다.

Switch 문은 논리적 참, 거짓 보다는  
다양한 상황에 따라 실행할 코드  
블록을 결정할 때 사용한다.

폴 스루(fall through) 란? Switch 문을  
탈출하지 않고 default 문까지 계속해서  
실행하여 결과를 반환

각 break를 안써서 발생  
하지만 폴 스루를 이용할 수도 있다.  
이를 이용하여 연쇄조건을 달 수  
있다. Ex) 윤년 계산하기

If else 문으로 해결할 수 있다면 if 문을  
사용하는 것이 좋다. 하지만 조건이 너무  
 많다면 switch를 쓰자.

조건식의 평가 결과가 참일 경우  
코드 블록을 실행한다.

조건식이 거짓일 때까지 반복

**\*\* 반복문을 대체할 수 있는 다양한 기능**

forEach, for..in,  
for..of 와 같은 것들

간결해진다.

### 8.3 반복문

#### 8.3.1 for 문

for문은 반복 횟수가 명확할 때 주로  
사용하고 while문은 반복 횟수가  
불명확할 때 주로 사용한다.

#### 8.3.2 while 문

조건식 평가가 불리언이  
아니면 강제 변환

조건식이 항상 참일 땐, If 문으로 탈출 조건을 만들고 break  
문으로 코드 블록을 탈출한다.

#### 8.3.3 do...while 문

코드를 한 번 이상 무조건 실행

### 8.4 break 문

코드 블록을 탈출한다.

레이블 문, 반복문, switch  
문의 코드 블록을 탈출

레이블 문?

string.indexOf(찾고자하는 문자)  
: 해당 문자가 어느 인덱스에 있는지  
찾을 수 있는 String.prototype.indexOf 메서드

### 8.5 continue 문

Break 처럼 반복문 탈출 x

If 문 내에서 실행해야 할 코드가 한 줄이라면 continue 문을  
사용했을 때보다 간편하고 가독성도 좋다. 하지만 if 문 내에서 실행할  
코드가 길다면 들여쓰기가 한 단계 더 깊어지므로 continue 문을  
사용하는 편이 가독성에 더 좋다.