

# 01

## 조건문





## 학습내용

- 01 제어문이란
- 02 if 조건문
- 03 switch~case 조건문

## 학습목표

- 제어문의 개념을 이해하고, 프로그램의 흐름을 조건에 따라 제어하는 방법을 설명할 수 있다.
- if 조건문을 활용하여 다양한 조건 분기 상황을 자바스크립트 코드로 구현할 수 있다.
- switch ~ case 조건문을 이해하고, 여러 경우에 따른 분기 처리를 효율적으로 구현할 수 있다.

# 지난 주차 복습

## 데이터 준비와 연산 명령어로 기본 프로그램을 만들기

### 변수와 연산자

#### 1교시

##### 프로그래밍 실행 원리 & 메모리

- 프로그램의 실행 원리
- 메모리 구조
- 변수란?
- 변수 선언법
- var/let/cons

#### 2교시

##### 자료형

- 자료형의 필요성
- 자료형의 종류
- 자료형 변환

#### 3교시

##### 연산자

- 종류
  - 산술, 대입, 비교, 논리, 비트, ??, 삼항연산자 등
- 사용법
  - 우선순위, 결합규칙

## 생각 해보기

### Q

아래의 예시 코드처럼 Java나 C와 같은 전통적인 프로그래밍에서는 원칙적으로 변수 타입을 정적으로 처리하는 방식을 제공하며, 이와 달리 JavaScript는 실행 도중, 명시적이든 자동이든 정해진 타입을 바꿀 수 있습니다. 동적 타입 언어인 자바스크립트에서 변수 타입이 자동으로 바뀌는 특징은 어떠한 장점이나 단점이 발생할 수 있는지, 각각 한가지씩 생각해 보고, 여러분의 생각을 자유롭게 게시판에 올려주세요.

// Java (정적 타입)

```
int num = 10;
```

```
num = "hello"; // 오류 발생
```

// JavaScript (동적 타입)

```
let num = 10;
```

```
num = "hello"; // 오류 없음
```



# 01

## 제어문이란



## 제어문

프로그램의 순차적 수행 흐름(순서)을  
조건이나 반복에 따라 다르게 실행되도록 만드는 명령어

예 “비가 올 것 같으면 우산을 챙겨서 가고, 아니면 그냥 간다”

- ➡ 조건에 따라 행동이 달라지는 논리 구조를 가짐
- ➡ 프로그래밍에서는 조건문으로 구현함

## ● 왜 제어문이 필요할까?



반복적인 명령어를 반복적으로 작성할 경우,  
오류 발생률이 높아지며 비효율적임



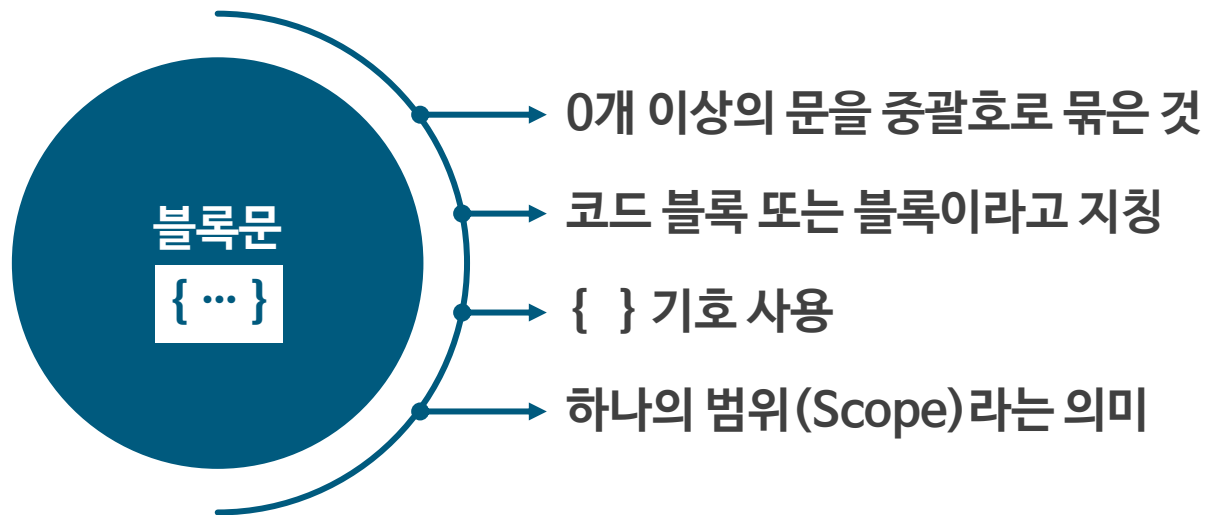
주어지는 데이터(상황)에 따라 동적으로 결과를 처리할 수  
있어야 유연한 프로그램이 될 수 있음



“ 프로그램의 실행 흐름을 원하는 대로  
조절하여 반복, 선택, 분기를 가능하게  
해주는 핵심 도구 ”



## ● 블록문 (Block Statement/Compound Statement)



```
// 블록  
{  
    var foo = 10;  
}
```

```
// 제어문  
var x = 1;  
if (x < 10) {  
    x++;  
}
```

- 블록문 (Block Statement/Compound Statement)

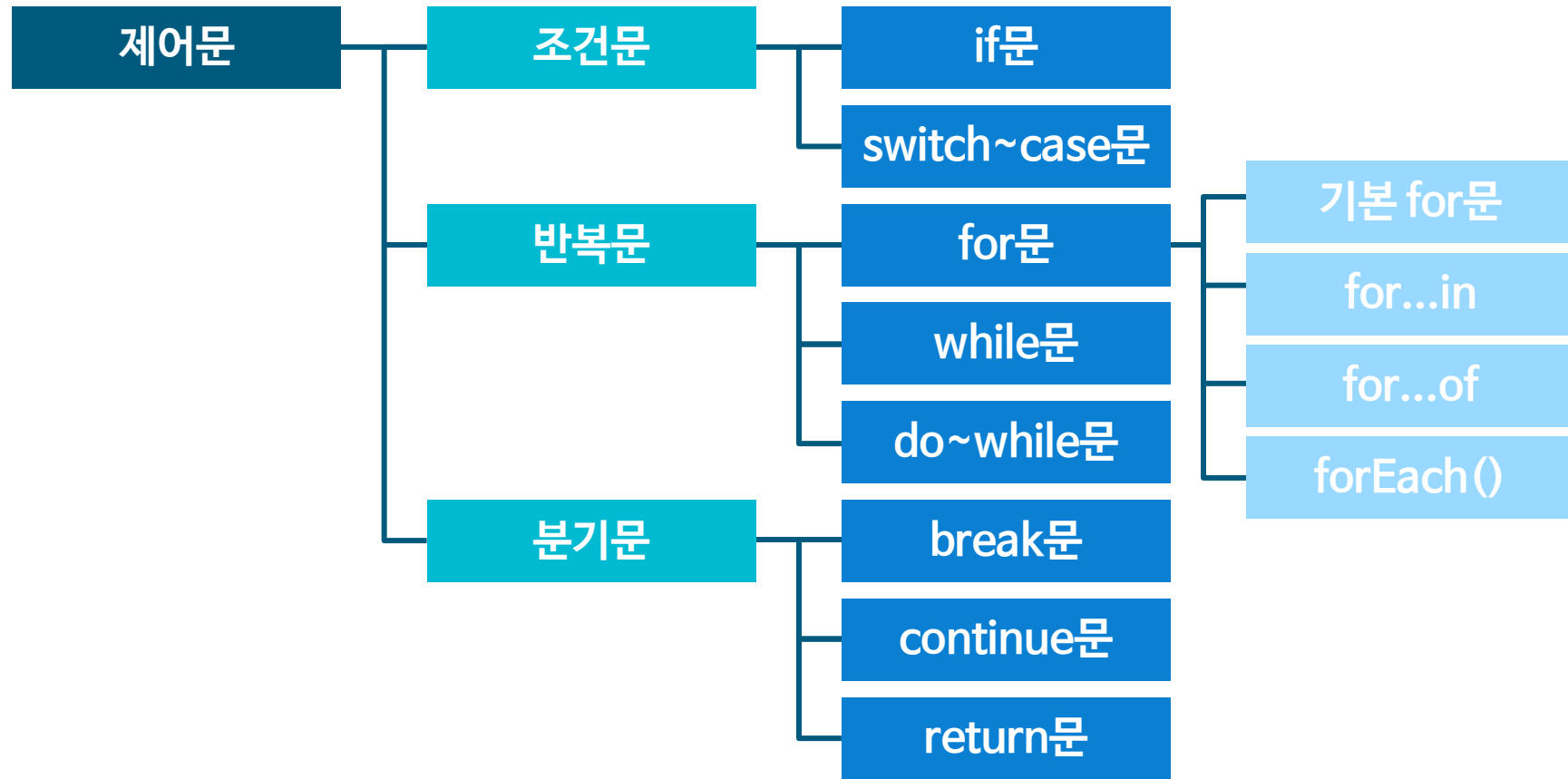


JavaScript는 블록문을 하나의 실행 단위로 취급함



블록문은 언제나 문의 종료를 의미하는 자체 종결성을 갖고 있기 때문에 블록문의 끝에는 세미콜론을 붙이지 않음

## 2) 제어문의 종류





# 02

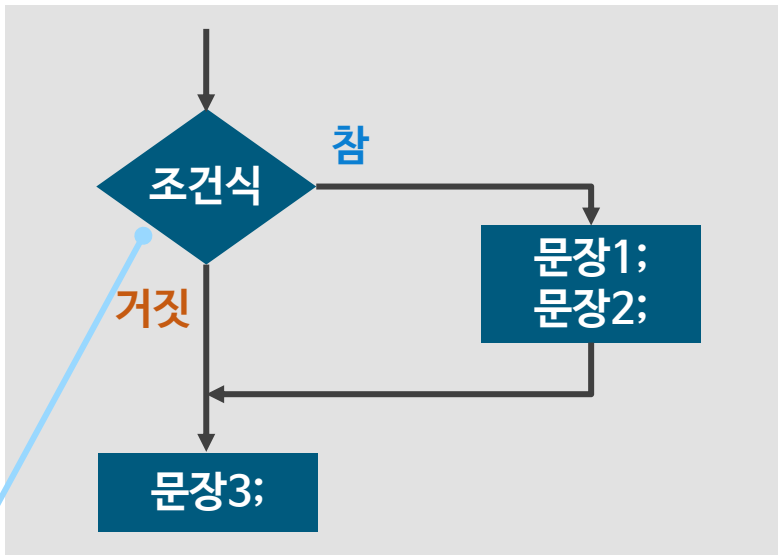
## if 조건문



# 1) 단순 if문



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



## 조건식의 결과

- 참(true)일 경우 : { } 내의 문장을 수행한 후 문장3을 수행함
- 거짓(false)일 경우 : 아무 것도 수행하지 않고 a 다음 문장(문장3)으로 넘어감

# 1) 단순 if문



- 예시 : 0보다 큰 2의 배수를 출력하는 조건문

```
if((num>0) && (num%2==0))  
{  
    console.log("0보다 큰 양수입니다.");  
    console.log("짝수입니다.");  
}
```

{ 문장이 여러 문장일 경우, “{}”을 생략할 수 없음 }

## 2) if~else문



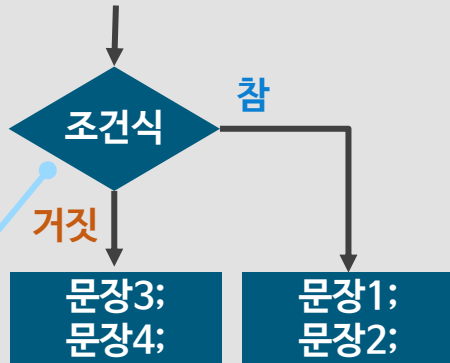
### 조건이 '참'인 경우

if 블록{}이 실행

### 조건이 '거짓'인 경우

else 블록{}이 반드시 실행

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



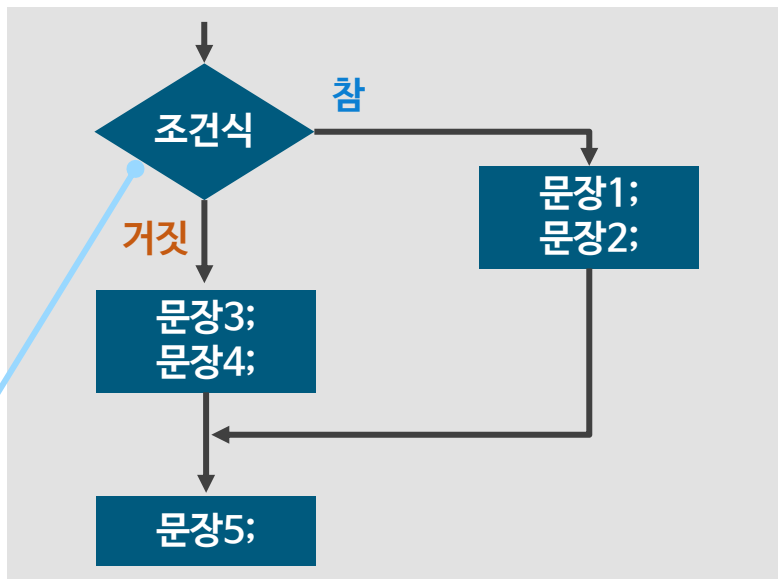
조건식의 결과에 따라 수행 명령문을 지정

- 참(true)일 경우 : if{ }내에 있는 문장1, 2를 수행함
- 거짓(false)일 경우 : else{ } 블록 내 문장3, 4를 수행함

## 2) if~else문

{ 조건에 따라 해당 블록을 수행 후, “문장5”는 모두 수행 }

```
if ( 조건식 )  
{  
    문장1;  
    문장2;  
  
} else {  
    문장3;  
    문장4;  
}  
문장5;
```



조건식을 만족할 경우는 문장1, 문장2를 수행하며,  
조건식을 만족하지 않을 경우, else 뒤에 지정된 블록인 문장3, 문장4를 수행,  
문장5는 조건식의 결과와 상관없이 수행함



## 2) if~else문



```
if(num%3==0) {  
    console.log("3의 배수입니다.");  
}  
else {  
    console.log("3의 배수가 아닙니다.");  
}  
console.log("종료합니다.");
```

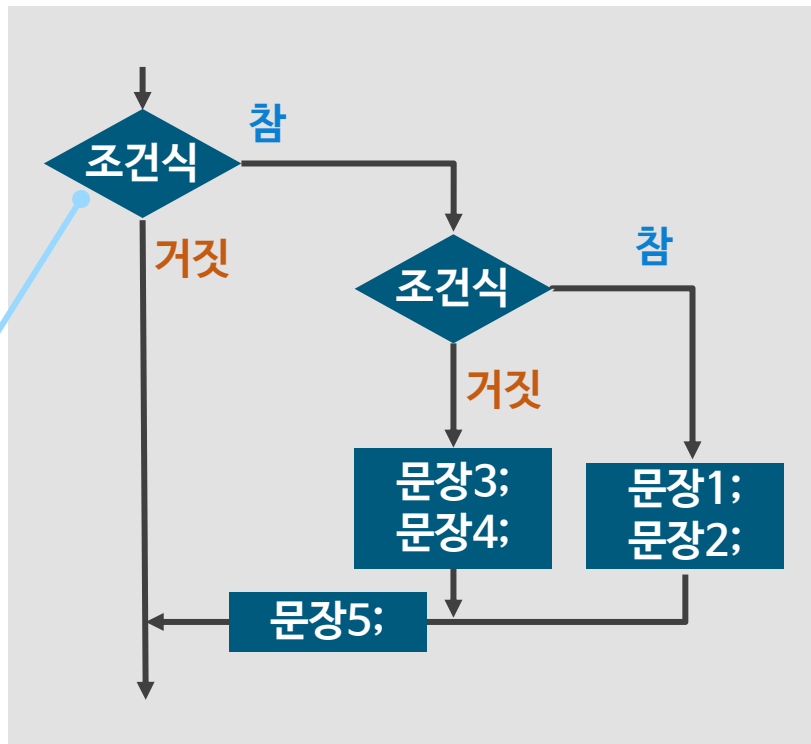
➡ 조건식의 결과가 참일 경우 실행

➡ 조건식의 결과가 거짓일 경우 실행

➡ 조건식과 상관없이 수행

### 3) 중첩 if문

```
if (조건식)
{
    if(조건식) {
        문장1;
        문장2;
    }else{
        문장3;
        문장4;
    }
    문장5;
}
```

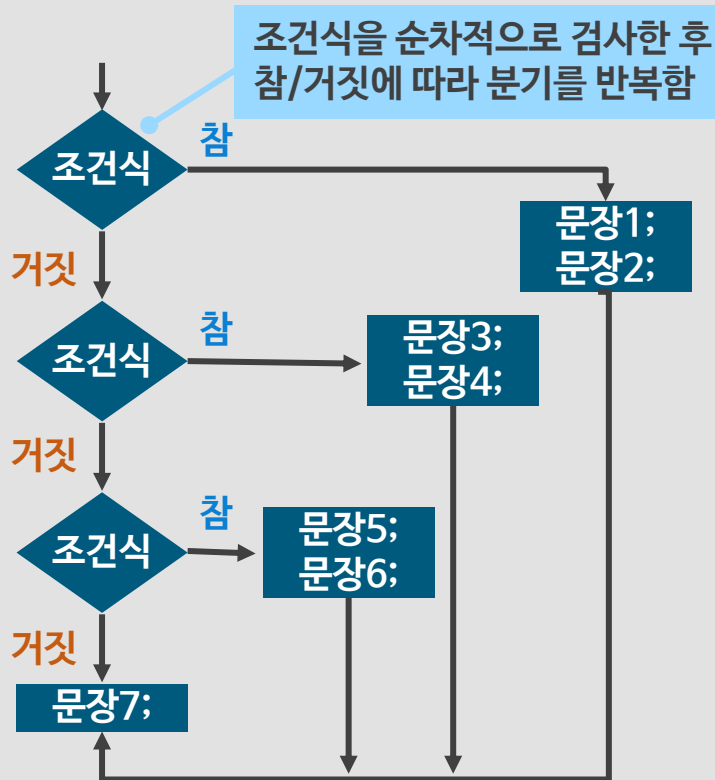


중첩된 if문을 사용해 다중 선택 논리를 표현할 수 있음

## 4) 다중 if문



```
if (조건식)
{
    문장1;
    문장2;
} else if(조건식) {
    문장3;
    문장4;
    ...
} else if(조건식) {
    문장5;
    문장6;
} else {
    문장7;
}
```



## 4) 다중 if문



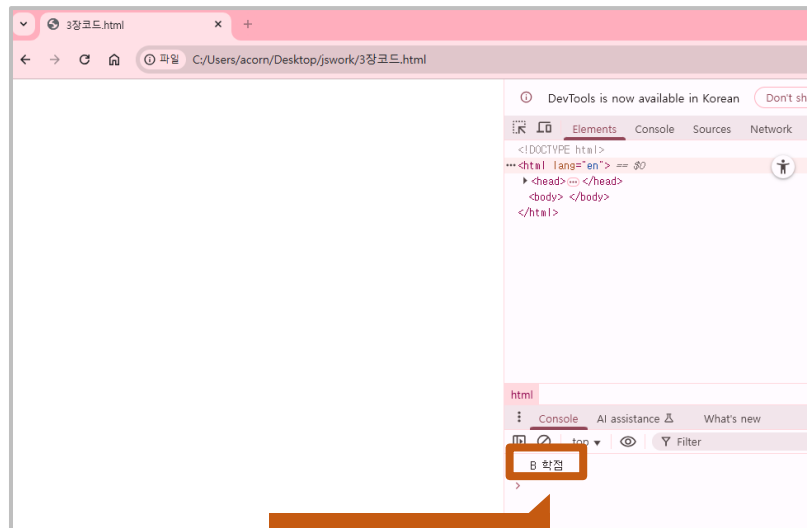
### ● 예시

// 점수에 따라 등급을 출력하는 예제

```
let score = 85;
```

```
if (score >= 90) {  
    console.log("A 학점");  
} else if (score >= 80) {  
    console.log("B 학점");  
} else {  
    console.log("C 학점 이하");  
}
```

결과 화면



B 학점



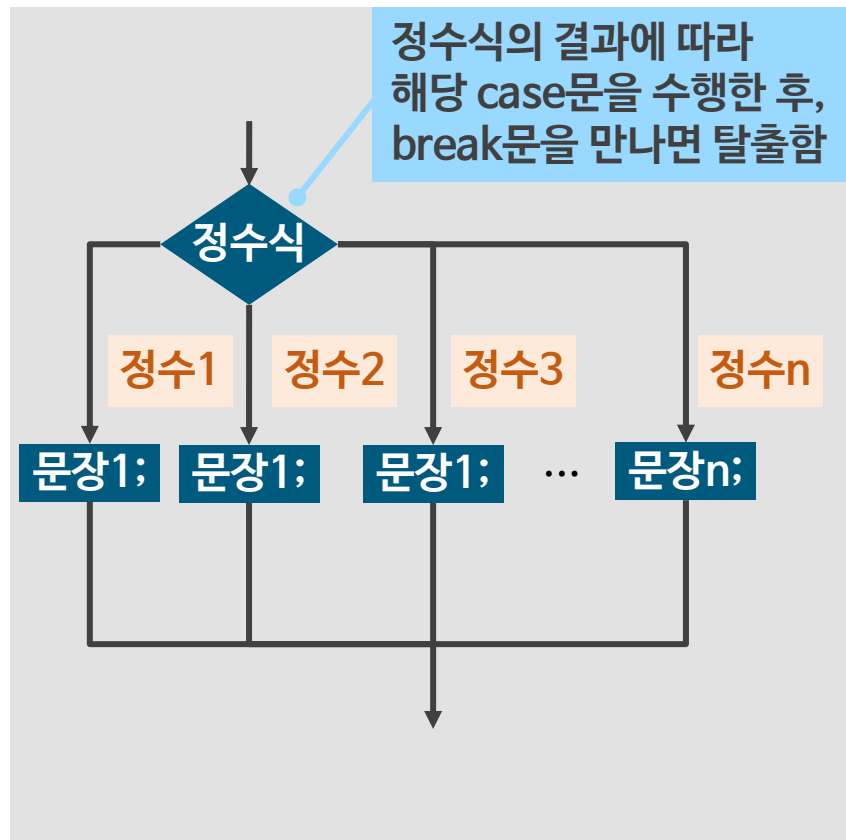
# 03

## switch~case 조건문



# 1) 기본 사용법

```
switch ( 정수식 )  
{  
  case 정수1:  
    문장1;  
    break;  
  case 정수2:  
    문장2;  
    break;  
  case 정수3:  
    문장3;  
    break;  
}
```



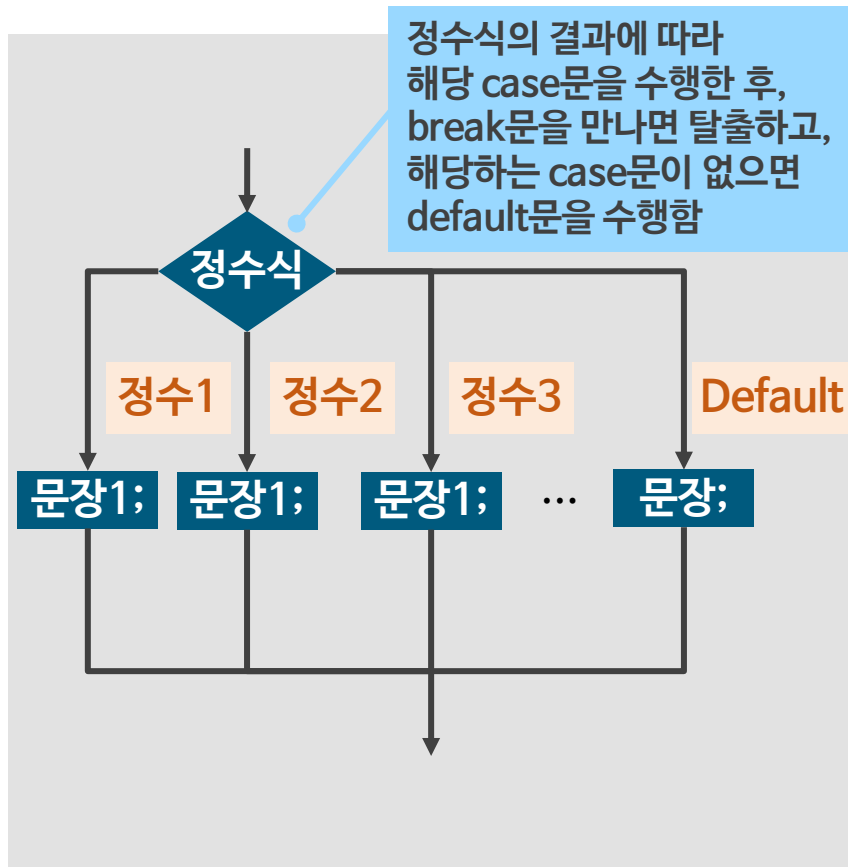
## ● switch 조건식에서 사용 가능한 다양한 자료형

예시	설명
switch (정수식)	<ul style="list-style-type: none"><li>정수 값을 기준으로 여러 경우(case)를 분기 처리</li><li>case 비교는 엄격 비교 (===)로 수행</li></ul>
switch (문자열)	<ul style="list-style-type: none"><li>문자열 값 비교 가능</li></ul>
switch (boolean)	<ul style="list-style-type: none"><li>true/false 조건 분기 가능</li></ul>
switch (표현식)	<ul style="list-style-type: none"><li>연산 결과로 판단 가능</li></ul>
switch (객체)	<ul style="list-style-type: none"><li>참조 비교 (엄격 비교 사용)</li></ul>

## 2) default문



```
switch ( 정수식 )  
{  
  case 정수1:  
    문장1;  
    break;  
  case 정수2:  
    문장2;  
    break;  
  case 정수3:  
    문장3;  
    break;  
  default:  
    문장n;  
}
```





### 3) switch~case문 예시



// 메뉴 번호에 따라 다른 음식을 출력하는 예제

let menu = 2;

switch (menu) {

case 1:

console.log("김치찌개");

break;

case 2:

console.log("된장찌개");

break;

//.....

//이어서 .....

case 3:

console.log("비빔밥");

break;

default:

console.log("메뉴를 다시 선택하세요");

}

### 3) switch~case문 예시



```
6 // 점수에 따라 등급을 출력하는 예제
7 let score = 85;
8
9 if (score >= 90) {
10   console.log("A 학점");
11 } else if (score >= 80) {
12   console.log("B 학점");
13 } else {
14   console.log("C 학점 이하");
15 }
16 //
17 let menu = 2;
18
19 switch (menu) {
20   case 1:
21     console.log("김치찌개");
22     break;
23   case 2:
24     console.log("원장찌개");
25     break;
26   case 3:
27     console.log("비빔밥");
28     break;
29 }
```

Console

B 학점  
원장찌개

B 학점  
원장찌개

03주. 흐름을 제어한다! 제어문

# 02

## 반복문

