



# 클라우드보안 과정:

## 자바스크립트 조건문

성결대학교 정보통신공학과

정복래

brjung@sungkyul.ac.kr

# Contents

## 학습목표

- if 조건문의 기본 사용 방법을 익힙니다.
- if 조건문과 논리 연산자를 함께 사용하는 방법을 익힙니다.
- switch 조건문을 이해합니다.

## 내용

- if 조건문
- 삼항 연산자
- if else 조건문
- 짧은 초기화 조건문
- 중첩 조건문
- 조금 더 나아가기
- if else if 조건문
- switch 조건문

# 1. if 조건문

## ■ 기본 형태

```
if (불_표현식) {  
  
}
```

- 불 표현식이 true이면 문장을 실행, false이면 문장을 무시함

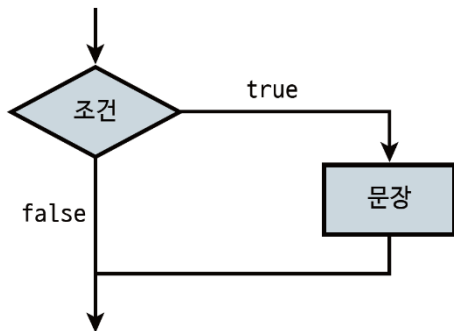


그림 3-1 if 조건문

# 1. if 조건문

## ■ [예제 3-1] 홀수 짝수 구분(1)

코드 3-1

홀수와 짝수 구분 (1)

conditionBasicA.js

```
let input = 32;

if (input % 2 == 0) {
  console.log("짝수입니다!");
}

if (input % 2 == 1) {
  console.log("홀수입니다!");
}
```

실행 결과

짝수입니다!

# 1. if 조건문

- [예제 3-2] 오전 오후 구분(1)
  - ❶ 현재 시간 구하기

코드 3-2 현재 시간 구하기

```
let date = new Date();

console.log(date.getFullYear());
console.log(date.getMonth());
console.log(date.getDay());
console.log(date.getHours());
console.log(date.getMinutes());
console.log(date.getSeconds());
```

# 1. if 조건문

- ② 오전과 오후 구분하기

코드 3-3

오전과 오후 구분 (1)

conditionBasicB.js

```
let date = new Date();

if (date.getHours() < 12) {
  console.log("오전입니다.");
}

if (12 <= date.getHours()) {
  console.log("오후입니다.");
}
```

실행 결과

오후입니다.

## 2. if else 조건문

### ■ 기본 형태

```
if (불_표현식) {  
    // 불_표현식이 참일 때 실행할 문장  
} else {  
    // 불_표현식이 거짓일 때 실행할 문장  
}
```

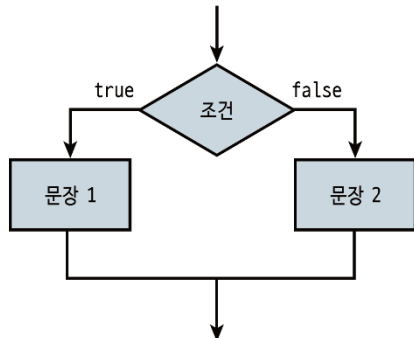


그림 3-2 if else 조건문

## 2. if else 조건문

### ■ [예제 3-3] 홀수 짝수 구분(2)

코드 3-4

홀수와 짝수 구분 (2)

conditionBasicC.js

```
let input = 32;

if (input % 2 == 0) {
  console.log("짝수입니다!");
} else {
  console.log("홀수입니다!");
}
```

실행 결과

짝수입니다!



## 2. if else 조건문

### ■ [예제 3-4] 오전 오후 구분(2)

코드 3-5

오전과 오후 구분 (2)

conditionBasicD.js

```
let date = new Date();

if (date.getHours() < 12) {
  console.log("오전입니다.");
} else {
  console.log("오후입니다.");
}
```

실행 결과

오후입니다.

### 3. 중첩 조건문

#### ■ 기본 형태

```
if (불_표현식) {  
    if (불_표현식) {  
        문장;  
    } else {  
        문장;  
    }  
} else {  
    if (불_표현식) {  
        문장;  
    } else {  
        문장;  
    }  
}
```

### 3. 중첩 조건문

- [예제 3-5] 중첩 조건문
  - `DateTime.Now.Hour < 11` 조건을 비교
  - `false`이면 `DateTime.Now.Hour < 15` 조건을 한 번 더 비교

코드 3-6

중첩 조건문

nestedCondition.js

```
let date = new Date();
let hours = date.getHours();

if (hours < 11) {
    console.log("아침 먹을 시간입니다.");
} else {
    if (hours < 15) {
        console.log("점심 먹을 시간입니다.");
    }
    else {
        console.log("저녁 먹을 시간입니다.");
    }
}
```

## 4. if else if 조건문

### ■ if else if 조건문

- 중복되지 않는 세 가지 이상의 조건을 구분할 때 사용
- 기본 형태

```
if (불_표현식) {  
  
} else if (불_표현식) {  
  
} else if (불_표현식) {  
  
} else {  
  
}
```

## 4. if else if 조건문

### ■ [예제 3-6] if else if 조건문

코드 3-7

if else if 조건문

conditionBasicE.js

```
let date = new Date();  
let hours = date.getHours();  
  
if (hours < 11) {  
    console.log("아침 먹을 시간입니다.");  
} else if (hours < 15) {  
    console.log("점심 먹을 시간입니다.");  
} else {  
    console.log("저녁 먹을 시간입니다.");  
}
```

실행 결과

점심 먹을 시간입니다.

# 5. switch 조건문

## ■ 기본 형태

```
switch (비교할_값) {  
    case 값:  
        문장  
        break;  
    case 값:  
        문장  
        break;  
    default:  
        문장  
        break;  
}
```

## 5. switch 조건문

- [예제 3-7] switch 조건문
  - 홀수와 짝수 구분

코드 3-10

switch 조건문

conditionBasicF.js

```
let input = 32;

switch (input % 2) {
  case 0:

    console.log("짝수입니다.");
    break;
  case 1:
    console.log("홀수입니다.");
    break;
}
```

# 5. switch 조건문

## ■ switch 조건문

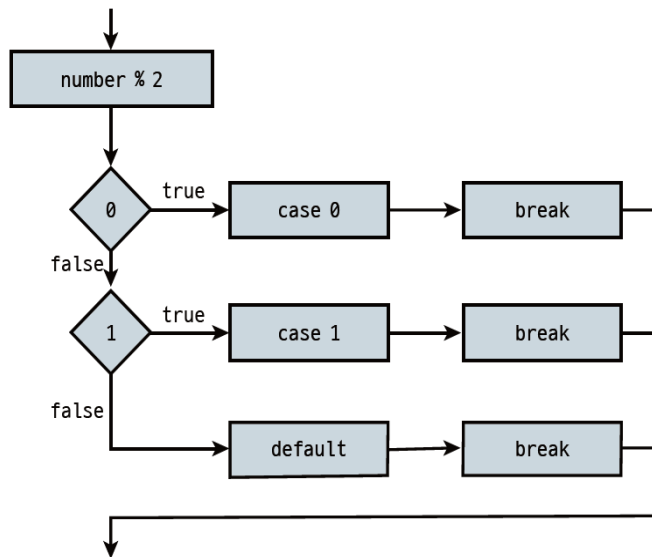


그림 3-3 switch 조건문



## 5. switch 조건문

- [예제 3-8] break 키워드를 사용하지 않는 switch 조건문

코드 3-11

break 키워드를 사용하지 않는 switch 조건문

switchWithoutBreak.js

```
let date = new Date();

switch (date.getMonth() + 1) {
  case 12:
  case 1:
  case 2:
    console.log("겨울입니다.");
    break;
  case 3:
  case 4:
  case 5:
    console.log("봄입니다.");
    break;
```

## 5. switch 조건문

```
case 6:  
case 7:  
case 8:  
    console.log("여름입니다.");  
    break;  
case 9:  
case 10:  
case 11:  
    console.log("가을입니다.");  
    break;  
default:  
    console.log("대체 어떤 행성에 살고 계신가요?");  
    break;  
}
```

실행 결과

가을입니다.

## 6. 조건 연산자

### ■ 기본 형태

불\_표현식 ? 참일\_때\_실행하고\_밖으로\_낼\_값 : 거짓일\_때\_실행하고\_밖으로\_낼\_값

- [예제 3-9] 조건 연산자를 활용한 변수 초기화
  - 변수가 undefined일 때 새로운 값으로 초기화

코드 3-12 조건 연산자를 활용한 변수 초기화

switchWithoutBreak.js

```
// 변수를 선언합니다.
let test;

// 삼항 연산자로 해당 변수가 undefined인지 확인하고 초기화합니다.
test = typeof(test) !== 'undefined' ? test : "초기화_1";
console.log(test)

// 삼항 연산자로 해당 변수가 undefined인지 확인하고 초기화합니다.
test = typeof(test) !== 'undefined' ? test : "초기화_2";
console.log(test)
```

undefined인지 확인합니다.

실행 결과

초기화\_1

초기화\_1

## 6. 조건 연산자

- 조건문으로 변경한 코드
- 조건 연산자를 활용하는 것이 더 간단하므로, 일반적으로 조건 연산자 사용

코드 3-13

```
let test;

if (typeof(test) !== 'undefined') {
  test = "초기화_1"
}
console.log(test)

if (typeof(test) !== 'undefined') {
  test = "초기화_2"
}
console.log(test)
```

