

이 책의 학습 목표

▪ CHAPTER 01: 자바스크립트 개요와 개발환경 설정

- 자바스크립트 개발환경 설치와 자바스크립트 프로그래밍 기본 용어 학습

▪ CHAPTER 02: 자료와 변수

- 프로그램 개발의 첫걸음. 자료형과 변수 학습

▪ CHAPTER 03: 조건문

- 프로그램의 흐름을 변화시키는 요소. 조건문의 종류를 알아보고 사용 방법을 이해

▪ CHAPTER 04: 반복문

- 배열의 개념과 문법을 익혀 while 반복문과 for 반복문 학습

▪ CHAPTER 05: 함수

- 다양한 형태의 함수를 만들기과 매개변수를 다루는 방법 이해

▪ CHAPTER 06: 객체

- 객체의 속성과 메소드, 생성, 관리하는 기본 문법 학습

▪ CHAPTER 07: 문서 객체 모델

- DOMContentLoaded 이벤트를 사용한 문서 객체 조작과 다양한 이벤트의 사용 방법 이해

▪ CHAPTER 08: 예외 처리

- 구문 오류와 예외를 구분하고, 예외 처리의 필요성과 예외를 강제로 발생시키는 방법을 이해

▪ CHAPTER 09: 클래스

- 객체 지향을 이해하고 클래스의 개념과 문법 학습

▪ CHAPTER 10: 리액트 라이브러리

- 리액트 라이브러리 사용 방법과 간단한 애플리케이션을 만드는 방법 학습

Contents

- CHAPTER 03: 조건문

SECTION 3-1 if 조건문

SECTION 3-2 switch 조건문과 짧은 조건문



CHAPTER 03 조건문

프로그램의 흐름을 변화시키는 요소. 조건문의 종류를 알아보고 사용 방법을 이해

SECTION 3-1 if 조건문(1)

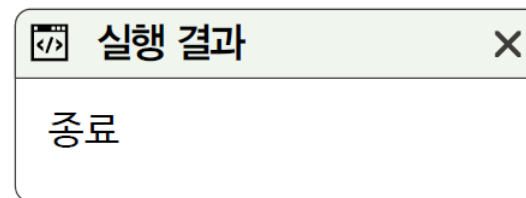
◦ if 조건문

- 불 표현식의 값이 true면 중괄호 안의 문장을 실행하고 false면 문장을 무시

```
if(불 값이 나오는 표현식) {  
    불 값이 참일 때 실행할 문장  
}
```

- if 조건문 사용하기 (소스 코드 3-1-1.html)

```
01 <script>  
02 // if 조건문  
03 if (273 < 100) {  
04     // 표현식 273 < 100이 참일 때 실행합니다.  
05     alert('273 < 100 => true')  
06 }  
07  
08 // 프로그램 종료  
09 alert('종료')  
10 </script>
```



SECTION 3-1 if 조건문(2)

- if 조건문

- 현재 시각에 따라 오전과 오후를 구분하는 프로그램
 - 현재 시각 구하기 (Chapter 7에서 학습)

```
> const date = new Date()  
undefined  
> date.getFullYear()  
2020  
> date.getMonth() + 1  
6  
> date.getDate()  
4  
> date.getHours()  
15  
> date.getMinutes()  
5  
> date.getSeconds()  
7
```

SECTION 3-1 if 조건문(3)

◦ if 조건문

- 현재 시각에 따라 오전과 오후를 구분하는 프로그램
 - 오전과 오후 구분하기 (소스 코드 3-1-2.html)

```
01 <script>
02  // 변수를 선언
03  const date = new Date()
04  const hour = date.getHours()
05
06  // if 조건문
07  if (hour < 12) {
08    // 표현식 hour < 12가 참일 때 실행
09    alert('오전입니다.');
```

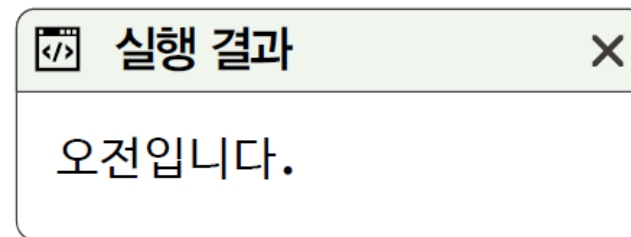
→ 현재 날짜와 시간을 갖는 객체 생성

→ 현재시간을 0~23 사이의 값으로 출력하는 메소드

```
10  }
11
12  if (hour >= 12) {
13    // 표현식 hour >= 12가 참일 때 실행
14    alert('오후입니다.')
```

→

```
15  }
16 </script>
```



SECTION 3-1 if 조건문(4)

- if else 조건문

- 서로 반대되는 상황을 표현하는 구문

```
if(불 값이 나오는 표현식){  
    불 값이 참일 때 실행할 문장  
} else {  
    불 값이 거짓일 때 실행할 문장  
}
```

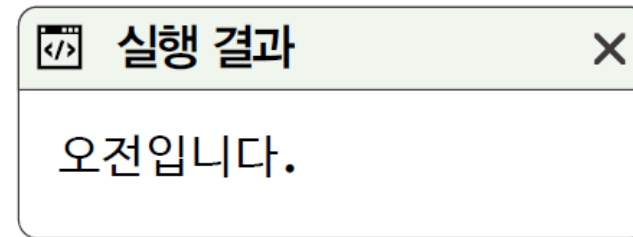
- if else 조건문을 사용해 현재 시간 구하기 소스 코드 3-1-3.html

SECTION 3-1 if 조건문(5)

- if else 조건문

- if else 조건문을 사용해 현재 시간 구하기 소스 코드 3-1-3.html

```
01 <script>
02  // 변수를 선언
03  const date = new Date()
04  const hour = date.getHours()
05
06  // if 조건문
07  if (hour < 12) {
08    // 표현식 hour < 12가 참일 때 실행
09    alert('오전입니다.')
10  } else {
11    // 표현식 hour < 12가 거짓일 때 실행
12    alert('오후입니다.')
13  }
14 </script>
```



SECTION 3-1 if 조건문(6)

◦ 중첩 조건문

- 조건문 안에 조건문을 중첩해 사용

```
if (불 값이 나오는 표현식 1) {  
    if (불 값이 나오는 표현식 2) {  
        표현식 2가 참일 때 실행할 문장  
    } else {  
        표현식 2가 거짓일 때 실행할 문장  
    }  
} else {  
    if (불 값이 나오는 표현식 3) {  
        표현식 3이 참일 때 실행할 문장  
    } else {  
        표현식 3이 거짓일 때 실행할 문장  
    }  
}
```

표현식 1이 참이면 실행

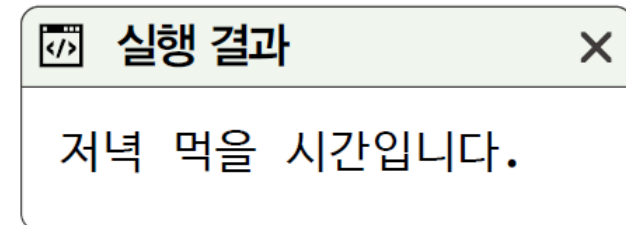
표현식 1이 거짓이면 실행

SECTION 3-1 if 조건문(7)

◦ 중첩 조건문

- 중첩 조건문으로 시간 파악하기 (소스 코드 3-1-4.html)

```
01 <script>
02  // 변수를 선언
03  const date = new Date()
04  const hour = date.getHours()
05
06  // 중첩 조건문
07  if (hour < 11) {
08    // 표현식 hour < 11이 참일 때 실행
09    alert( ' 아침 먹을 시간입니다. ' )
10  } else {
11    // 표현식 hour < 11이 거짓일 때 실행
12    if (hour < 15) {
13      // 표현식 hour < 15가 참일 때 실행
14      alert('점심 먹을 시간입니다.')
15    } else {
16      // 표현식 hour < 15가 거짓일 때 실행
17      alert('저녁 먹을 시간입니다.')
18    }
19  }
20 </script>
```



SECTION 3-1 if 조건문(8)

- if else if 조건문

- 중첩 조건문에서 중괄호를 생략한 형태

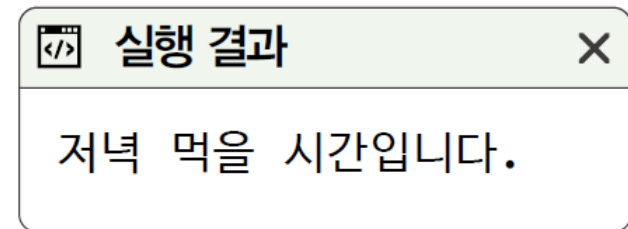
```
if (불 표현식) {  
    문장  
} else if (불 표현식) {  
    문장  
} else if (불 표현식) {  
    문장  
} else {  
    문장  
}  
}
```

- 중첩 조건문으로 만들었던 예제를 if else if 조건문의 형태로 바꾸는 것은 매우 간단하여, 한 쌍의 중괄호를 지우면 됨

SECTION 3-1 if 조건문(9)

- if else if 조건문
 - if else if 조건문으로 시간 파악하기 (소스 코드 3-1-5.html)

```
01 <script>
02  // 변수를 선언
03  const date = new Date()
04  const hour = date.getHours()
05
06  // if else if 조건문
07  if (hour < 11) {
08    // 표현식 hour < 11이 참일 때 실행
09    alert('아침 먹을 시간입니다.')
10  } else if (hour < 15) {
11    // 표현식 hour < 11이 거짓이고 표현식 hour < 15가 참일 때 실행
12    alert('점심 먹을 시간입니다.')
13  } else {
14    // 표현식 hour < 15가 거짓일 때 실행
15    alert('저녁 먹을 시간입니다.')
16  }
17 </script>
```



[마무리①]

◦ 4가지 키워드로 정리하는 핵심 포인트

- if 조건문은 조건에 따라 코드를 실행하거나 실행하지 않도록 하기 위해 사용하는 구문
- else 구문은 if 조건문 뒤에 사용하며, if 조건문이 거짓일 때 사용
- 중첩 조건문은 조건문을 중첩해서 사용하는 경우를 의미
- if else if 조건문은 중첩 조건문에서 중괄호를 생략한 형태로, 겹치지 않는 3가지 이상의 조건으로 나눌 때 사용

◦ 확인 문제

1. 다음 예제 중에서 '참입니다'를 출력하는 것은?

①

```
<script>
  const x = 1
  if (x > 4) {
    console.log('참입니다')
  }
</script>
```

②

```
<script>
  const x = 0
  if (x > 4) {
    console.log('참입니다')
  }
</script>
```

③

```
<script>
  const x = 10
  if (x > 4) {
    console.log('참입니다')
  }
</script>
```

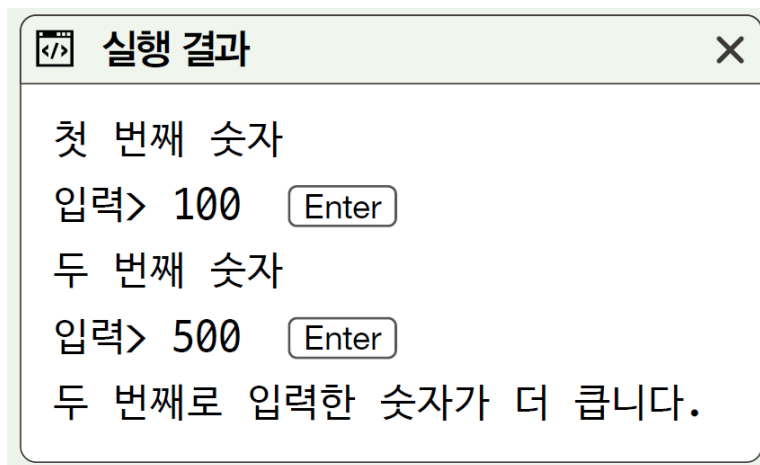
[마무리②]

◦ 확인 문제

2. 사용자로부터 숫자 2개를 입력받아 첫 번째 입력받은 숫자가 큰지, 두 번째 입력받은 숫자가 큰지를 구하는 프로그램을 다음 빈칸을 채워 완성하기

```
<script>
const a =  (prompt('첫 번째 숫자', ''))
const b =  (prompt('두 번째 숫자', ''))

if (  ) {
    alert('첫 번째로 입력한 숫자가 더 큼니다.')
} else if (  ) {
    alert('두 숫자가 같습니다.')
} else {
    alert('두 번째로 입력한 숫자가 더 큼니다.')
}
</script>
```



[마무리③]

◦ 확인 문제

3. 중첩 조건문은 2장에서 배운 논리 연산자를 적용해 하나의 if 조건문으로 만들 수 있다. 빈칸에 어떤 논리 연산자가 들어가야 할까?

```
if (x > 10) {  
  if (x < 20) {  
    console.log('조건에 맞습니다.')  
  }  
}
```

```
if (x > 10  x < 20) {  
  console.log('조건에 맞습니다.')  
}
```

[힌트] 두 조건이 모두 true일 때 코드를 실행해야 한다면 어떤 논리 연산자를 사용해야 할까?
두 코드의 실행 결과는 ex3-1-3-1.html과 ex3-1-3.html에서 확인

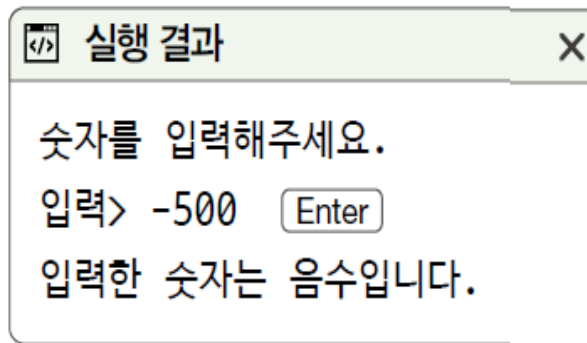
[마무리④]

◦ 확인 문제

4. 사용자에게 숫자를 입력받아 양수, 0, 음수를 구분하는 프로그램 만들기

```
<script>  
const a = Number(prompt('숫자를 입력해주세요.', ''))
```

```
</script>
```

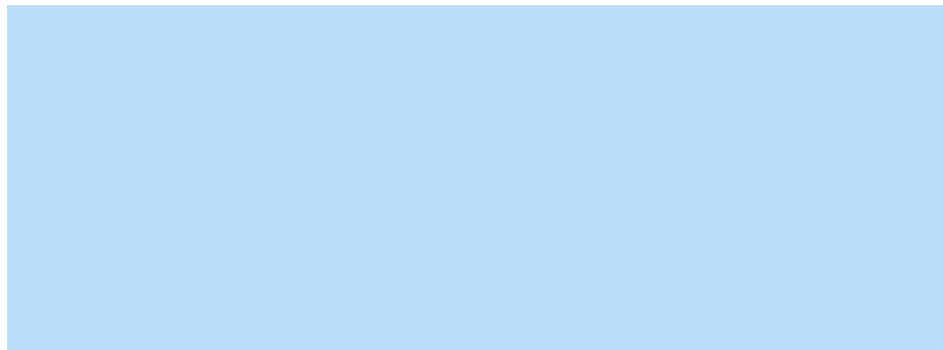


[마무리⑤]

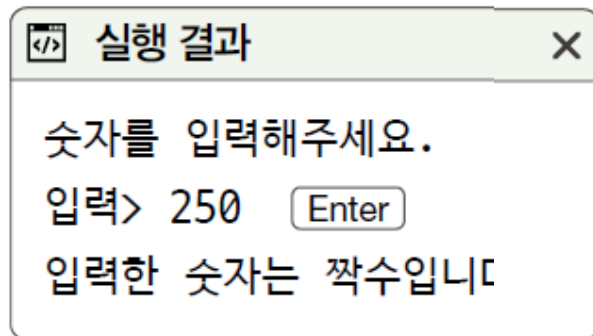
◦ 확인 문제

5. 사용자에게 숫자를 입력받아 홀수와 짝수를 구분하는 프로그램 만들기

```
<script>  
const a = Number(prompt('숫자를 입력해주세요.', ''))
```



```
</script>
```



[힌트] 홀수와 짝수를 어떻게 구분해야 할까?

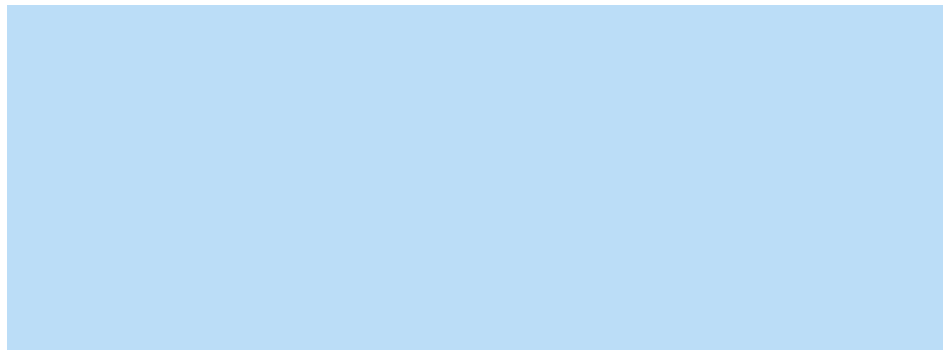
직접 만들어본 뒤에 「짝수와 홀수 구분하기 누적 예제」를 참조

[마무리⑥]

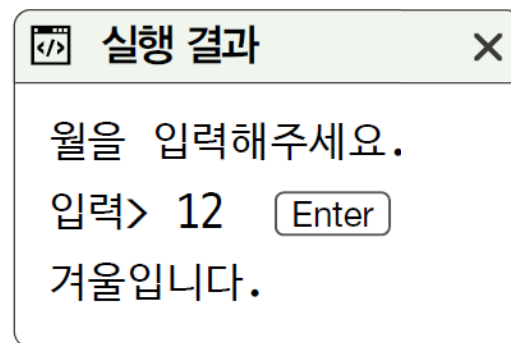
◦ 확인 문제

6. 현재가 몇 월인지 확인하고, 계절을 구분하는 프로그램 만들기

```
<script>  
const a = Number(prompt('월을 입력해주세요.', ''))
```



```
</script>
```



[힌트] 홀수와 짝수를 어떻게 구분해야 할까?

직접 만들어본 뒤에 「짝수와 홀수 구분하기 누적 예제」를 참조

SECTION 3-2 switch 조건문과 짧은 조건문(1)

- switch 조건문

- switch 조건문의 기본 형태. default 키워드는 생략 가능

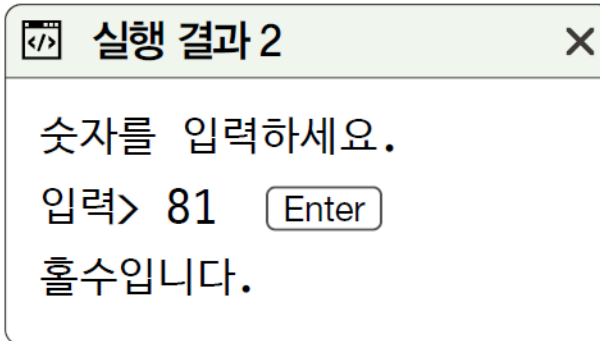
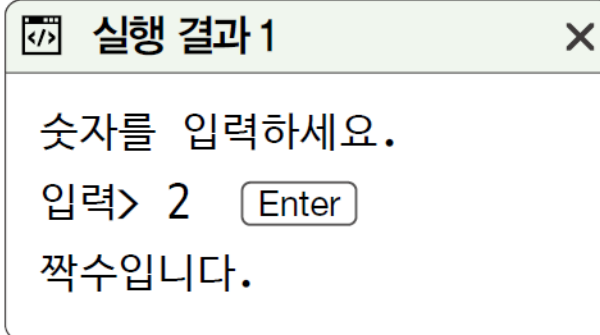
```
switch (자료) {  
  case 조건 A:  
    break  
  case 조건 B:  
    break  
  default: → 생략할 수 있음  
    break  
}
```

SECTION 3-2 switch 조건문과 짧은 조건문(2)

- switch 조건문

- switch 조건문 사용하기 (소스 코드 3-2-1.html)

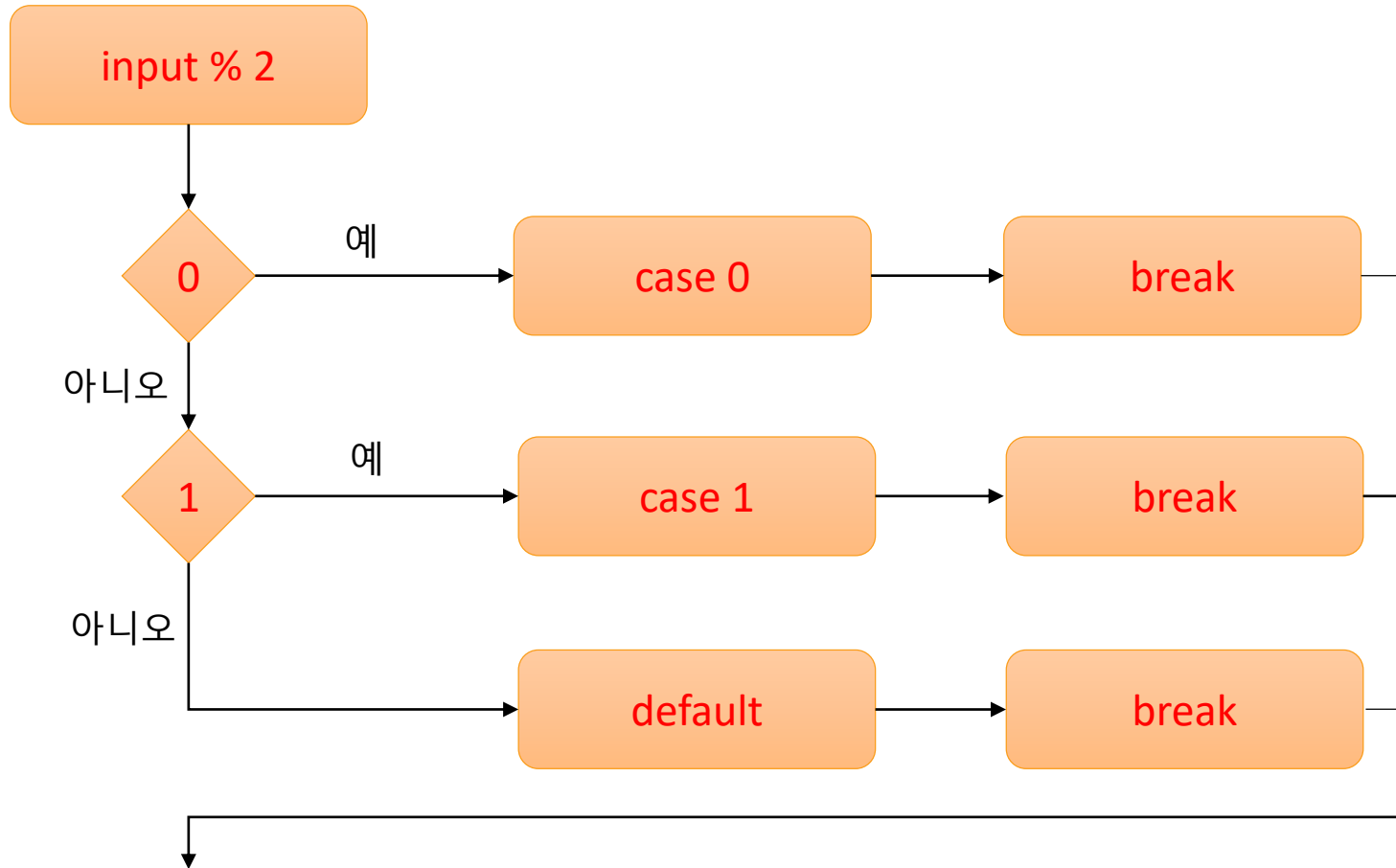
```
01 <script>
02 // 변수를 선언
03 const input = Number(prompt('숫자를 입력하세요.', '숫자'))
04
05 // 조건문
06 switch (input % 2) { → 나머지 연산자를 사용하여 홀수와 짝수를 구분
07   case 0:
08     alert('짝수입니다.')
09     break
10   case 1:
11     alert('홀수입니다.')
12     break
13   default:
14     alert('숫자가 아닙니다.')
15     break
16 }
17 </script>
```



SECTION 3-2 switch 조건문과 짧은 조건문(3)

- switch 조건문

- break: switch 조건문이나 반복문을 빠져나가기 위해 사용하는 키워드
- switch 조건문의 괄호 안에는 비교할 값을 입력

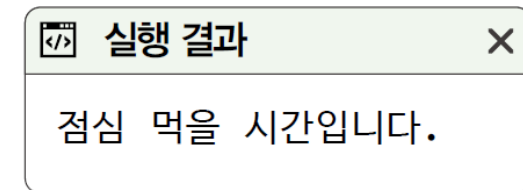


SECTION 3-2 switch 조건문과 짧은 조건문(4)

- switch 조건문

- switch 조건문을 if 조건문으로 변환하기 (소스 코드 3-2-2.html)

```
01 <script>
02  // 변수를 선언
03  const date = new Date()
04  const hour = date.getHours()
05
06  // 조건문
07  switch (true) {
08    case hour < 11:
09      // 표현식 hour < 11이 참일 때 실행
10      alert( ' 아침 먹을 시간입니다. ' )
11      break
12    case hour < 15:
13      // 표현식 hour < 11이 거짓이고 표현식 hour < 15가 참일 때 실행
14      alert('점심 먹을 시간입니다.')
15      break
16    default:
17      // 위의 모든 것이 거짓일 때 실행
18      alert('저녁 먹을 시간입니다.')
19      break
20  }
21 </script>
```



SECTION 3-2 switch 조건문과 짧은 조건문(5)

◦ 조건부 연산자

- 기본 형태

불 표현식 ? 참일 때의 결과 : 거짓일 때의 결과

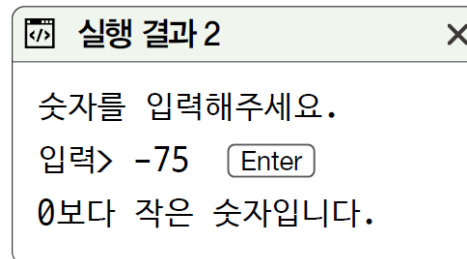
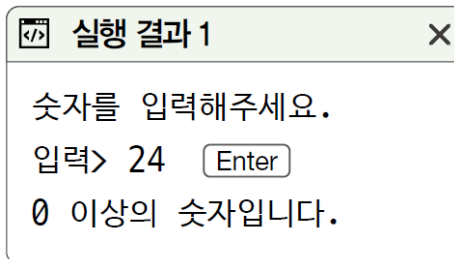
- 자바스크립트에서 항을 3개 갖는 연산자는 조건부 연산자가 유일해서 삼항 연산자라고 부르기도 함

- 조건부 연산자 사용하기 (소스 코드 3-2-3.html)

```
01 <script>
02 // 변수를 선언
03 const input = prompt('숫자를 입력해주세요.', '')
04 const number = Number(input)
05
06 // 조건문
07 const result = (number >= 0) ? 0 이상의 숫자입니다.' : 0보다 작은 숫자입니다.'
08 alert(result)
09 </script>
```

(number >= 0)이 true면 이 값이 할당

(number >= 0)이 false면 이 값이 할당



SECTION 3-2 switch 조건문과 짧은 조건문(6)

◦ 짧은 조건문

- 짧은 조건문은 논리 연산자의 특성을 조건문으로 사용

- 논리합 연산자를 사용한 짧은 조건문

불 표현식 || 불 표현식이 거짓일 때 실행할 문장

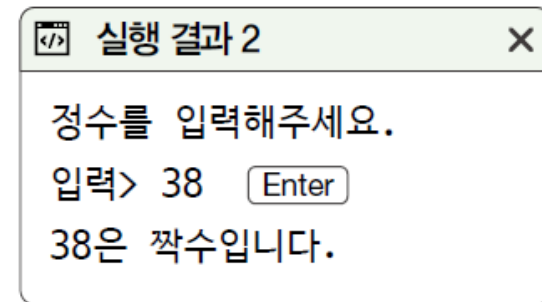
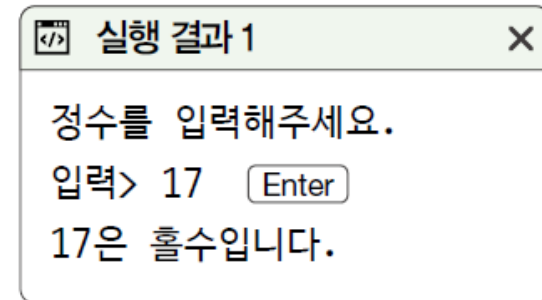
- 논리곱 연산자를 사용한 짧은 조건문

결과가 거짓인 불 표현식 && 불 표현식이 참일 때 실행할 문장

SECTION 3-2 switch 조건문과 짧은 조건문(7)

- 짝수와 홀수 구분하기(누적 예제)
 - if else 조건문으로 짝수와 홀수 구분하기(1) (소스 코드 3-2-4.html)

```
01 <script>
02 // 입력이 문자열이므로 다음과 같은 코드를 사용할 수 있음
03 const 입력 = prompt('정수를 입력해주세요.');"
04 const 끝자리 = 입력 [입력.length - 1]
05
06 // 끝자리를 비교
07 if (끝자리 === "0" ||
08     끝자리 === "2" ||
09     끝자리 === "4" ||
10     끝자리 === "6" ||
11     끝자리 === "8") {
12     alert(`${입력}은 짝수입니다.`)
13 } else {
14     alert(`${입력}은 홀수입니다.`)
15 }
16 </script>
```



SECTION 3-2 switch 조건문과 짧은 조건문(8)

- 짝수와 홀수 구분하기(누적 예제)

- if else 조건문으로 짝수와 홀수 구분하기(2) (소스 코드 3-2-5.html)

```
01 <script>
02 const 입력 = prompt('정수를 입력해주세요,')
03 const 숫자 = Number(입력)
04
05 if (숫자 % 2 === 0) {
06   alert(`${입력}은 짝수입니다.`)
07 } else {
08   alert(`${입력}은 홀수입니다.`)
09 }
10 </script>
```

실행 결과 1

정수를 입력해주세요.
입력> 17
17은 홀수입니다.

실행 결과 2

정수를 입력해주세요.
입력> 38
38은 짝수입니다.

SECTION 3-2 switch 조건문과 짧은 조건문(9)

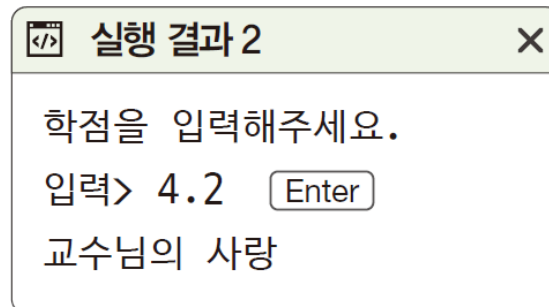
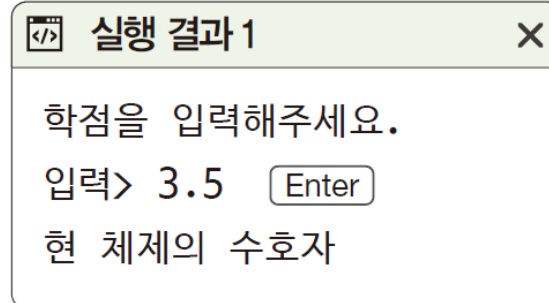
- 학점을 기반으로 별명 붙여주기(누적 예제)
 - 인터넷에서 학점을 학생들이 재미있게 표현한 유머를 이를 조건문으로 구현하고 출력

조건	설명(학생 평가)
4.5	신
4.2~4.5	교수님의 사랑
3.5~4.2	현 체제의 수호자
2.8~3.5	일반인
2.3~2.8	일탈을 꿈꾸는 소시민
1.75~2.3	오락문화의 선구자
1.0~1.75	불가촉천민
0.5~1.0	자벌레
0~0.5	플랑크톤
0	시대를 앞서가는 혁명의 씨앗

SECTION 3-2 switch 조건문과 짧은 조건문(10)

- 학점을 기반으로 별명 붙여주기(누적 예제)
 - 중첩 조건문 사용하기(1) (소스 코드 3-2-6.html)

```
01 <script>
02  const score = Number(prompt('학점을 입력해주세요.', '학점'))
03  if (score === 4.5) {
04    alert('신')
05  } else if (4.2 <= score && score < 4.5) {
06    alert('교수님의 사랑')
07  } else if (3.5 <= score && score < 4.2) {
08    alert('현 체제의 수호자')
09  } else if (2.8 <= score && score < 3.5) {
10    alert('일반인')
11  } else if (2.3 <= score && score < 2.8) {
12    alert('일탈을 꿈꾸는 소시민')
13  } else if (1.75 <= score && score < 2.3) {
14    alert('오락문화의 선구자')
15  } else if (1.0 <= score && score < 1.75) {
16    alert('불가촉천민')
17  } else if (0.5 <= score && score < 1.0) {
18    alert('자벌레')
19  } else if (0 < score && score < 0.5) {
20    alert('플랑크톤')
21  } else {
22    alert('시대를 앞서가는 혁명의 씨앗')
23  }
24 </script>
```



SECTION 3-2 switch 조건문과 짧은 조건문(11)

- 학점을 기반으로 별명 붙여주기(누적 예제)
 - 중첩 조건문 사용하기(2) (소스 코드 3-2-7.html)

```
01 <script>
02  const score = Number(prompt('학점을 입력해주세요.', '학점'))
03  if (score === 4.5) {
04    alert('신')
05  } else if (4.2 <= score) {
06    alert('교수님의 사랑')
07  } else if (3.5 <= score) {
08    alert('현 체제의 수호자')
09  } else if (2.8 <= score) {
10    alert('일반인')
11  } else if (2.3 <= score) {
12    alert('일탈을 꿈꾸는 소시민')
13  } else if (1.75 <= score) {
14    alert('오락문화의 선구자')
15  } else if (1.0 <= score) {
16    alert('불가촉천민')
17  } else if (0.5 <= score) {
18    alert('자벌레')
19  } else if (0 < score) {
20    alert('플랑크톤')
21  } else {
22    alert('시대를 앞서가는 혁명의 씨앗')
23  }
24 </script>
```

- if 조건문은 위에서 아래로 흐르고 else 구문은 이전의 조건이 맞지 않을 때 넘어오는 부분. 따라서 앞에서 이미 제외된 조건을 한 번 더 검사할 필요는 없음
- 3행에서 score가 4.5인지는 검사했으므로 이를 생략. 이렇게 조건식을 바꾸면 조건 비교를 절반만 하게 되고 코드도 훨씬 쉽게 읽을 수 있음

```
else if (4.2 <= score && score < 4.5)
```

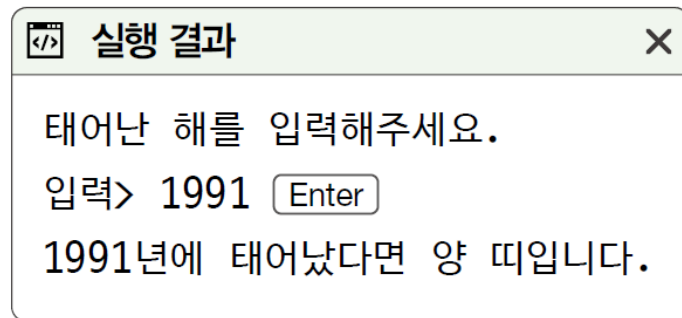


```
else if (4.2 <= score)
```

SECTION 3-2 switch 조건문과 짧은 조건문(12)

- 태어난 연도를 입력받아 띠 출력하기(누적 예제)
 - if else if 조건문 사용해보기 (소스 코드 3-2-8.html)

```
01 <script>
02  const rawInput = prompt('태어난 해를 입력해주세요.', '')
03  const year = Number(rawInput)
04  const e = year % 12
05
06  let result
07  if (e === 0) { result = '원숭이' }
08  else if (e === 1) { result = '닭' }
09  else if (e === 2) { result = '개' }
10  else if (e === 3) { result = '돼지' }
11  else if (e === 4) { result = '쥐' }
12  else if (e === 5) { result = '소' }
13  else if (e === 6) { result = '호랑이' }
14  else if (e === 7) { result = '토끼' }
15  else if (e === 8) { result = '용' }
16  else if (e === 9) { result = '뱀' }
17  else if (e === 10) { result = '말' }
18  else if (e === 11) { result = '양' }
19  alert(`${year}년에 태어났다면 ${result} 띠입니다.`)
20 </script>
```



SECTION 3-2 switch 조건문과 짧은 조건문(13)

- 태어난 연도를 입력받아 띠 출력하기(누적 예제)
 - split로 문자열을 잘라 사용하기 (소스 코드 3-2-9.html)

```
01 <script>
02  const rawInput = prompt('태어난 해를 입력해주세요.');"
03  const year = Number(rawInput)
04  const tti = '원숭이,닭,개,돼지,쥐,소,호랑이,토끼,용,뱀,말,양'.split(',')
05
06  alert(`${year}년에 태어났다면 ${tti[year % 12]} 띠입니다.`)
07 </script>
```

[노트] ' 문자열A '.split(' 문자열B ') 메소드는 문자열A를 문자열B로 잘라서 배열을 만들어내는 메소드. 배열과 관련된 내용은 04장에서 학습

위의 코드 '원숭이,닭,개,돼지,쥐,소,호랑이,토끼,용,뱀,말,양'.split(',')에서는 원숭이,닭,개,돼지,쥐,소,호랑이,토끼,용,뱀,말,양을 ' , ' 로 잘랐으므로, ['원숭이', '닭', '개', '돼지', '쥐', '소', '호랑이', '토끼', '용', '뱀', '말', '양']라는 배열이 만들어짐

[마무리①]

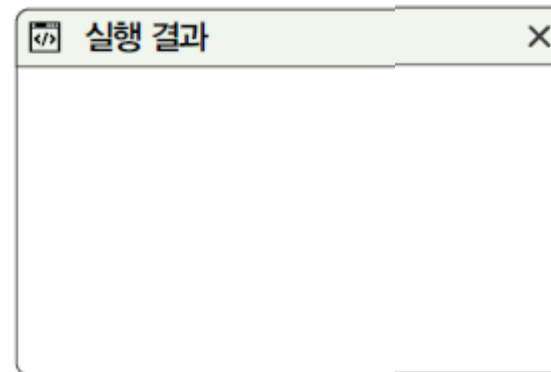
- 3가지 키워드로 정리하는 핵심 포인트

- switch 조건문은 값에 따라서 조건 분기를 걸어주는 조건문
- 조건부 연산자는 $A ? B : C$ 와 같은 형태로 피연산자 3개를 갖는 연산자. 조건 분기에 사용할 수 있음
- 짧은 조건문은 논리 연산자의 특이한 성질을 사용해서 조건 분기에 활용하는 코드

- 확인 문제

1. 다음 코드가 어떤 형태로 실행될지 예측하기

```
<script>
  const result = (100 > 200)
    ? prompt('값을 입력해주세요', '')
    : confirm('버튼을 클릭해주세요')
  alert(result)
</script>
```

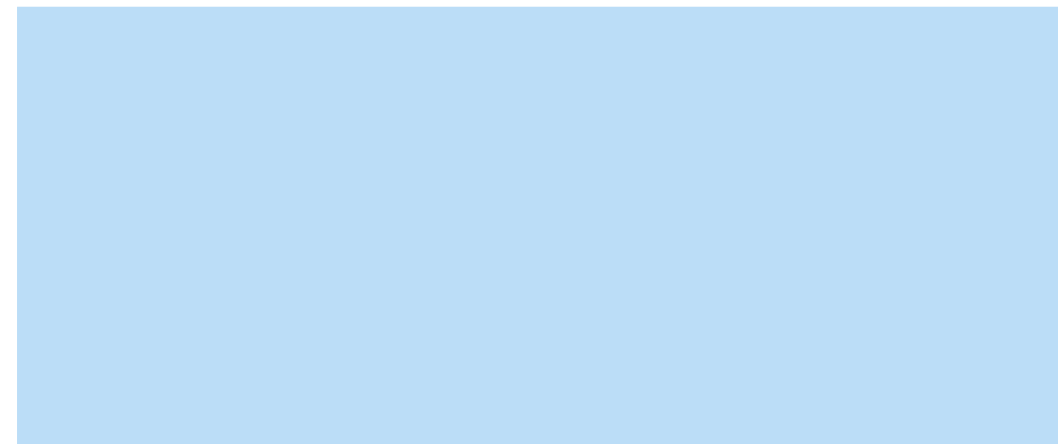


[마무리②]

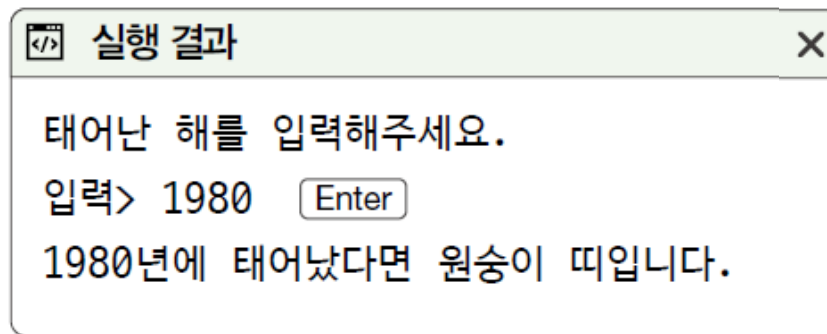
◦ 확인 문제

2. [누적 예제: 태어난 연도를 입력받아 띠 출력하기/] 예제(책 155쪽)에서 if 조건문을 switch 조건문으로 변경해서 구현해보기

```
<script>  
const rawInput = prompt('태어난 해를 입력해주세요.', '')  
const year = Number(rawInput)  
const e = year % 12  
let result
```



```
    alert(`${year}년에 태어났다면 ${result} 띠입니다.`)  
</script>
```



[마무리③]

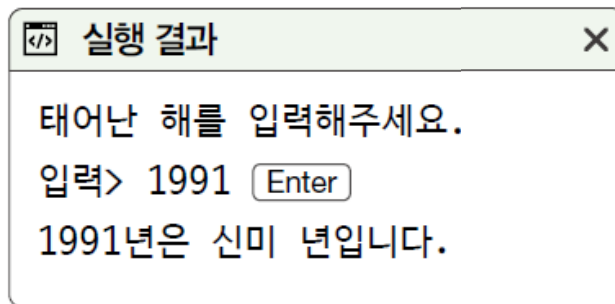
◦ 확인 문제

3. 태어난 연도를 입력받아 띠 출력하기' 예제(책 152쪽)에서 동물 이름을 쥘부터 '자, 축, 인, 묘, 진, 사, 오, 미, 신, 유, 술, 해'로 변경하고, 입력한 연도의 '갑, 을, 병, 정, 무, 기, 경, 신, 임, 계'를 계산합니다. 이 둘을 합쳐 다음과 같이 출력하는 프로그램을 만들기

```
<script>
const rawInput = prompt('태어난 해를 입력해주세요.', '')
const year = Number(rawInput)
let 간

let 띠

alert(`${year}년은 ${간}${띠} 년입니다.`)
</script>
```



[마무리④]

◦ 확인 문제

4. 다음 중에서 switch 조건문과 직접적인 관련이 없는 키워드는?

- ① switch ② break ③ default ④ else

5. 다음 중에서 다른 실행 결과를 내는 코드?

- ① true ? alert('출력A') : alert('출력B')
② false ? alert('출력B') : alert('출력A')
③ true || alert('출력A')
④ true && alert('출력A')