

Conditional branching: if, '?'

☑ if와 '?'를 사용한 조건 처리

조건에 따라 다른 행동을 취해야 할 때가 있다.

이럴 땐, if문과 '?' operator라고도 불리는 조건부 operator ?를 사용하면 된다.

❖ 'if' statement

if(...)문은 괄호 안에 들어가는 조건을 평가하는데, 그 결과가 true이면 code block이 실행된다.

[예시]

```
let year = prompt('ECMAScript-2015 명세는 몇 년도에 출판되었을까?', '');
```

```
if (year == 2015) alert( '정답입니다!' );
```

위 예시에선 조건(year == 2015)이 간단한 경우만 다뤘는데, 조건문은 더 복잡할 수도 있다.

조건이 true일 때 복수의 문을 실행하고 싶다면 curly brace(중괄호)로 code block을 감싸야 한다.

```
if (year == 2015) {  
    alert( "정답입니다!" );  
    alert( "아주 똑똑하시네요!" );  
}
```

if 문을 쓸 때는 조건이 true일 경우 실행되는 구문이 단 한 줄이더라도 중괄호 {}를 사용해 code를 block으로 감싸는 것을 추천한다. 이렇게 하면 코드 가독성이 증가한다.

❖ Boolean conversion

if (...) 문은 괄호 안의 표현식을 평가하고 그 결과를 boolean(불린값)으로 변환한다.

☞ number 0, empty string "", null, undefined, NaN은 불린형으로 변환 시 모두 false가 된다. 이런 값들은 'falsy(거짓 같은)' 값이라고 부른다.

☞ 이 외의 값은 불린형으로 변환 시 true가 되므로 'truthy(참 같은)' 값이라고 부른다.

이 규칙에 따르면 아래 예시의 code block은 절대 실행되지 않는다.

```
if (0) {           // 0은 falsy 이다.
  ...
}
```

아래 예시의 코드 블록은 항상 실행된다.

```
if (1) {           // 1은 truthy이다.
  ...
}
```

아래와 같이 평가를 통해 확정된 불린값을 if 문에 전달할 수도 있다.

```
let cond = (year == 2015); // equals comparison을 통해 true/false 여부를 결정한다.

if (cond) {
  ...
}
```

❖ 'else' clause(절)

if 문엔 else 절을 붙일 수 있다. else 뒤에 이어지는 code block은 조건이 false 일 때 실행된다.

[예시]

```
let year = prompt('ECMAScript-2015 명세는 몇 년도에 출판되었을까?', '');

if (year == 2015) {
  alert( '정답입니다!' );
} else {
  alert( '오답입니다!' ); // 2015 이외의 값을 입력한 경우
}
```

❖ 'else if'로 복수 조건 처리하기

유사하지만 약간씩 차이가 있는 조건 여러 개를 처리해야 할 때가 있다. 이때 else if를 사용할 수 있다.

[예시]

```
let year = prompt('ECMAScript-2015 명세는 몇 년도에 출판되었을까?', '');

if (year < 2015) {
  alert( '숫자를 좀 더 올려보세요.' );
} else if (year > 2015) {
  alert( '숫자를 좀 더 내려보세요.' );
} else {
```

```
    alert( '정답입니다!' );  
}
```

위 예시에서, JavaScript는 조건 `year < 2015`를 먼저 확인한다. 이 조건이 `false` 이라면 다음 조건 `year > 2015`를 확인한다. 이 조건 또한 `false` 라면 `else` 절 내의 `alert`를 실행한다.

`else if` 블록을 더 많이 붙이는 것도 가능하다. 마지막에 붙는 `else`는 필수가 아닌 선택 사항이다.

❖ 조건부 operator '?'

조건에 따라 다른 값을 변수에 할당해줘야 할 때가 있다.

[예시]

```
let accessAllowed;  
let age = prompt('나이를 입력해 주세요.', '');  
  
if (age > 18) {  
    accessAllowed = true;  
} else {  
    accessAllowed = false;  
}  
  
alert(accessAllowed);
```

'물음표(question mark) operator'라고도 불리는 '조건부(conditional) operator'를 사용하면 위 예시를 더 짧고 간결하게 변형할 수 있다.

`conditional operator`는 물음표 `?`로 표시한다. `operand`가 세 개이기 때문에 `conditional operator`를 '삼항(ternary) operator'라고 부르는 사람도 있다. 참고로, JavaScript에서 `operand`가 3개나 받는 operator는 `conditional operator`가 유일하다.

[문법]

```
let result = condition ? value1 : value2;  
평가 대상인 condition이 truthy라면 value1이, 그렇지 않으면 value2가 반환된다.
```

[예시]

```
let accessAllowed = (age > 18) ? true : false;
```

`age > 18` 주위의 괄호는 생략 가능하다. 물음표 operator는 우선순위가 낮으므로, `comparison operator`

> 가 실행되고 난 뒤에 실행된다.

아래 예시는 위 예시와 동일하게 동작한다.

```
// operator 우선순위 규칙에 따라, 비교 연산 'age > 18'이 먼저 실행된다.  
// (조건문을 괄호로 감쌀 필요가 없다.)  
let accessAllowed = age > 18 ? true : false;
```

괄호가 있으나 없으나 차이는 없지만, 코드의 가독성 향상을 위해 괄호를 사용할 것을 권장한다.

주의할 점은 comparison operator 자체가 true나 false를 반환하기 때문에 위 예시에서 물음표 operator를 사용하지 않아도 된다.

```
// 동일하게 동작함  
let accessAllowed = age > 18;
```

❖ Multiple '?'

물음표 operator를 여러 개 연결하면 복수의 조건을 처리할 수 있다.

[예시]

```
let age = prompt('나이를 입력해주세요.', 18);  
  
let message = (age < 3) ? '아기야 안녕?' :  
  (age < 18) ? '안녕!' :  
  (age < 100) ? '환영합니다!' :  
  '나이가 아주 많으시거나, 나이가 아닌 값을 입력 하셨군요!';  
  
alert( message );
```

물음표 operator를 이런 방식으로 쓰는 걸 처음 본 분이라면 이 코드가 어떻게 동작하는지 파악하기 힘들 수 있다. 그러나 주의를 집중하고 보면, 단순히 여러 조건을 나열한 코드임에 불과하다는 것을 알 수 있다.

- ① 첫 번째 물음표에선 조건문 `age < 3`을 검사한다.
- ② 그 결과가 true 이면 '아기야 안녕?'를 반환한다. 그렇지 않다면 첫 번째 콜론 ":"에 이어지는 조건문 `age < 18`을 검사한다.
- ③ 그 결과가 true이면 '안녕!'를 반환한다. 그렇지 않다면 다음 콜론 ":"에 이어지는 조건문 `age < 100`을 검사한다.
- ④ 그 결과가 true 이면 '환영합니다!'를 반환한다. 그렇지 않다면 마지막 콜론 ":" 이후의 표현식인 '나이가 아주 많으시거나, 나이가 아닌 값을 입력 하셨군요!'를 반환한다.

if..else를 사용하면 위 예시를 아래와 같이 변형할 수 있다.

```
if (age < 3) {  
    message = '아기야 안녕?';  
} else if (age < 18) {  
    message = '안녕!';  
} else if (age < 100) {  
    message = '환영합니다!';  
} else {  
    message = '나이가 아주 많으시거나, 나이가 아닌 값을 입력 하셨습니다!';  
}
```

❖ 부적절한 '?'

물음표 ?를 if 대용으로 쓰는 경우가 종종 있다.

```
let company = prompt('JavaScript는 어떤 회사가 만들었을까요?', '');  
  
(company == 'Netscape') ?  
    alert('정답입니다!') : alert('오답입니다!');
```

조건 company == 'Netscape'의 검사 결과에 따라 ? 뒤에 이어지는 첫 번째 혹은 두 번째 표현식이 실행되어 alert 창이 뜬다.

위 예시에선 평가 결과를 변수에 할당하지 않고, 결과에 따라 실행되는 표현식이 달라지도록 하였다. 그런데 이런 식으로 물음표 operator를 사용하는 것은 좋지 않다. 개발자 입장에서 if문을 사용할 때 보다 코드 길이가 짧아진다는 점 때문에 물음표를 if 대용으로 쓰는 게 매력적일 순 있다. 그러나 이렇게 코드를 작성하면 가독성이 떨어진다.

아래는 if를 사용해 변형한 코드이다. 어느 코드가 더 읽기 쉬운지 직접 비교해 보자.

```
let company = prompt('JavaScript는 어떤 회사가 만들었을까요?', '');  
  
if (company == 'Netscape') {  
    alert('정답입니다!');  
} else {  
    alert('오답입니다!');  
}
```

코드를 읽을 때 우리의 눈은 수직으로 움직인다. 수평으로 길게 늘어진 코드보단 여러 줄로 나뉘어 작성된 코드 블록이 더 읽기 쉽다.

물음표 operator?는 조건에 따라 반환 값을 달리하려는 목적으로 만들어졌다. 이런 목적에 부합하는 곳

에 물음표를 사용하도록 한다. 여러 분기를 만들어 처리할 때는 if를 사용한다.

[과제]

☞ if와 문자열 0

★ 중요도: 5

아래 코드에서 alert는 실행될까?

```
if ("0") {  
    alert( 'Hello' );  
}
```

[해답]

실행된다. 비어있는 string을 제외한 모든 string은 논리 평가 시 true를 반환한다. 문자 "0"은 비어있지 않은 string 이다. 직접 실행해서 그 결과를 확인해 보자.

```
if ("0") {  
    alert( 'Hello' );  
}
```

☞ JavaScript의 공식 이름

★ 중요도: 2

if..else 구조를 이용해 "JavaScript의 '공식' 이름은 무엇일까요?"라는 질문을 하는 코드를 작성해 보자. 사용자가 'ECMAScript'를 입력했다면 '정답입니다!', 아니라면 '모르셨나요? 정답은 ECMAScript입니다!'라는 메시지를 보여주자.

[해답]

```
<!DOCTYPE html>  
<html>  
<body>  
    <script>  
        'use strict';  
  
        let value = prompt('JavaScript의 "공식" 이름은 무엇일까요?', '');  
  
        if (value === 'ECMAScript') {  
            alert('정답입니다!');  
        } else {  
            alert("모르셨나요? 정답은 ECMAScript입니다!");  
        }  
    </script>  
</body>  
</html>
```

```
</script>
</body>
</html>
```

☞ 입력받은 숫자의 부호 표시하기

★ 중요도: 2

if..else와 프롬프트 대화상자를 사용해 사용자로 부터 숫자 하나를 입력받고, 아래 조건에 따라 그 결과를 alert 창에 출력해 보자.

- ☞ 입력받은 숫자가 0보다 큰 경우 1을 출력
- ☞ 입력받은 숫자가 0보다 작은 경우 -1을 출력
- ☞ 입력받은 숫자가 0인 경우 0을 출력

(사용자는 항상 숫자를 입력한다고 가정)

[해답]

```
let value = prompt('숫자를 입력하세요.', 0);

if (value > 0) {
  alert( 1 );
} else if (value < 0) {
  alert( -1 );
} else {
  alert( 0 );
}
```

☞ 'if'를 '?'로 교체하기

★ 중요도: 5

조건부 operator '?'를 이용해 if문이 사용된 아래 코드를 변형해보자. 동작 결과는 동일해야 한다.

```
let result;

if (a + b < 4) {
  result = '미만';
} else {
  result = '이상';
}
```

[해답]

```
let result = (a + b < 4) ? '미만' : '이상';
```

☞ 'if..else'를 '?'로 교체하기

★ 중요도: 5

조건부 operator '?'를 사용해 if..else 문이 사용된 아래 코드를 변형해자. 동작 결과는 동일해야 한다. 가독성을 위해 표현식을 여러 줄로 분할해 작성해 보자.

```
let message;

if (login === '직원') {
  message = '안녕하세요.';
} else if (login === '임원') {
  message = '환영합니다.';
} else if (login === '') {
  message = '로그인이 필요합니다.';
} else {
  message = '';
}
```

[해답]

```
let message = (login === '직원') ? '안녕하세요.' :
  (login === '임원') ? '환영합니다.' :
  (login === '') ? '로그인이 필요합니다.' :
  '';
```