

처음부터 시작하는 웹 프론트엔드 기초

5. Javascript : 제어문 , DOM

복습

- Javascript 사용법
- 출력문
- 변수
- 자료형
 - Number , String, Boolean , Object , Array ...
- 함수

Javascript 문법 : 제어문

- 제어문 : 프로그램의 순차적인 흐름을 제어해야 할 때 사용하는 실행문
 - 조건문
 - 반복문

Javascript 문법 : 조건문

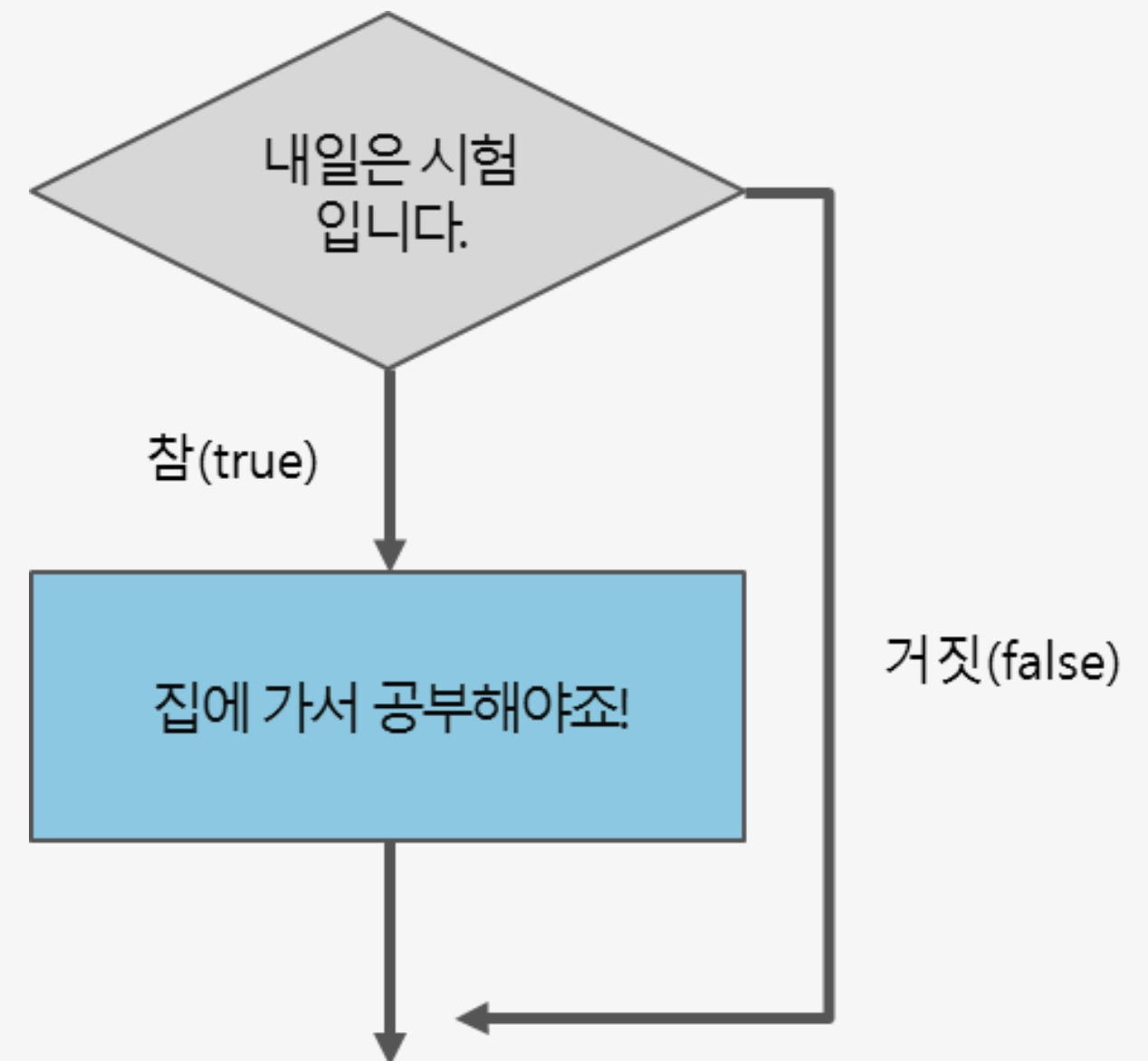
- 조건문 : 주어진 코드의 결과에 따라 명령을 수행하도록 제어하는 실행문
 - if 문
 - if / else 문
 - if / else if / else 문
 - switch 문

Javascript 문법 : 조건문

- if 문

- 표현식의 결과가 참(true)이면 실행문을 실행
- 거짓(false)이면 아무것도 실행 X
- 표현식은 무조건 true , false의 Boolean!

```
if (표현식) {  
    실행문;  
}
```



Javascript 문법 : 조건문

- 비교 연산자
 - 값 ,표현식의 관계를 비교하여 true 또는 false 을 반환하는 연산자

$a < b$	a가 b보다 작다
$a > b$	a가 b보다 크다
$a == b$	a와 b가 같다
$a != b$	a와 b가 같지 않다
$a \geq b$	a가 b보다 크거나 같다
$a \leq b$	a가 b보다 작거나 같다

Javascript 문법 : 조건문

- if 문 : 예제

```
let score = 90;

if (score >= 90) {
    alert('A')
}
if (score < 90) {
    alert('B')
}
```

Javascript 문법 : 조건문

- 논리 연산자
 - Boolean 값의 논리적 관계를 세밀하게 표현하기 위해 사용하는 연산자
 - 그리고 , 또는 , ~가 아닌 ... 등등을 표현하기 위해!

&&	AND 연산자
	OR 연산자
!	NOT 연산자

Javascript 문법 : 조건문

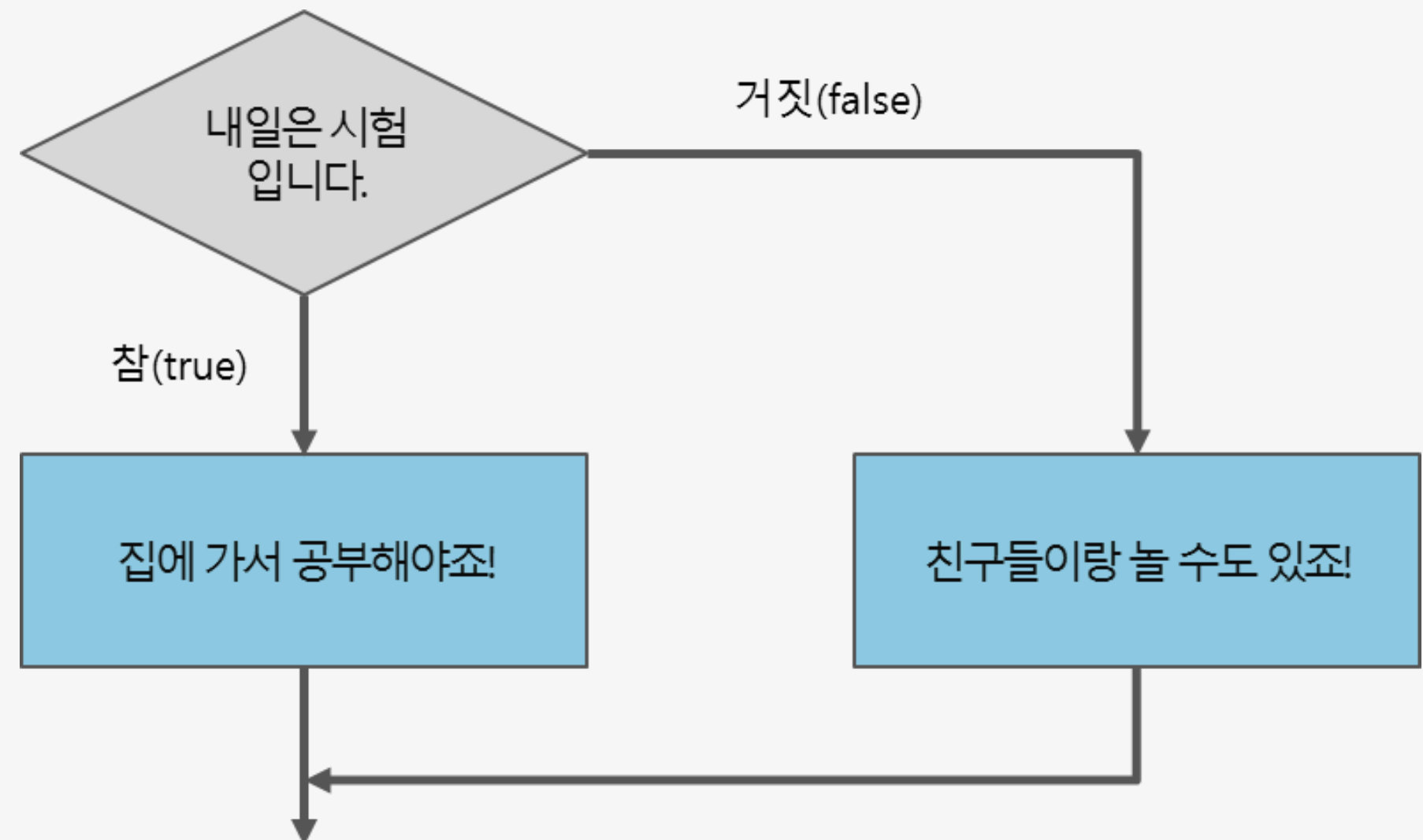
- if 문 : 예제

```
let score = 90;  
let gender = "male";  
  
if (score >= 90 && gender == "male") {  
    alert('Chanhyuk')  
}  
if (score >= 90 && gender == "female") {  
    alert('Soohyun')  
}
```

Javascript 문법 : 조건문

- if / else 문
 - else 문은 if 문의 표현식 결과가 false일 때 주어진 실행문을 실행

```
if (표현식) {  
    실행문1;  
}  
else {  
    실행문2;  
}
```



Javascript 문법 : 조건문

- if / else 문 : 예제

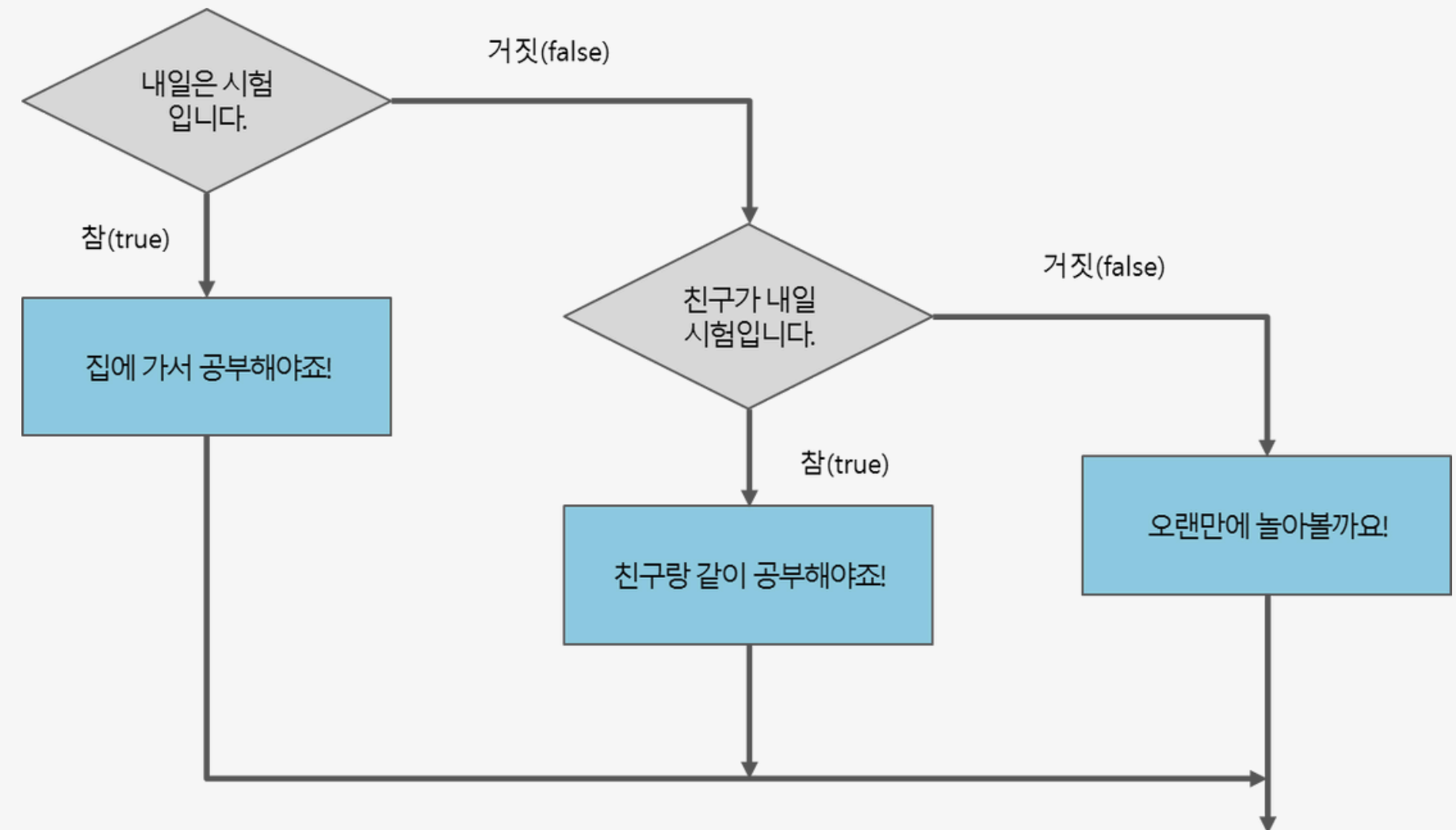
```
let score = 90;
```

```
if (score >= 90) {  
    alert('A')  
} else {  
    alert('B')  
}
```

Javascript 문법 : 조건문

- if / else if / else 문
 - else if 문은 if 문처럼 표현식을 설정 가능
 - if 문과 else 문은 한 번만 사용
 - else if 문은 여러 번 사용 가능

```
if (표현식1) {  
    실행문1;  
}  
else if (표현식2) {  
    실행문2;  
}  
else {  
    실행문3;  
}
```



Javascript 문법 : 조건문

- if / else if / else 문 : 예제

```
let score = 90;

if (score >= 90) {
    alert('A');
} else if (score >= 80) {
    alert('B');
} else {
    alert('C');
}
```

Javascript 문법 : 조건문

- switch 문

```
switch (조건 값) {  
  
    case value1:  
  
        조건 값이 value1일 때 실행하고자 하는 실행문;  
  
        break;  
  
    case 값2:  
  
        조건 값이 value2일 때 실행하고자 하는 실행문;  
  
        break;  
  
    ...  
  
    default:  
  
        조건 값이 어떠한 case 절에도 해당하지 않을 때 실행하고자 하는 실행문;  
  
        break;  
  
}
```

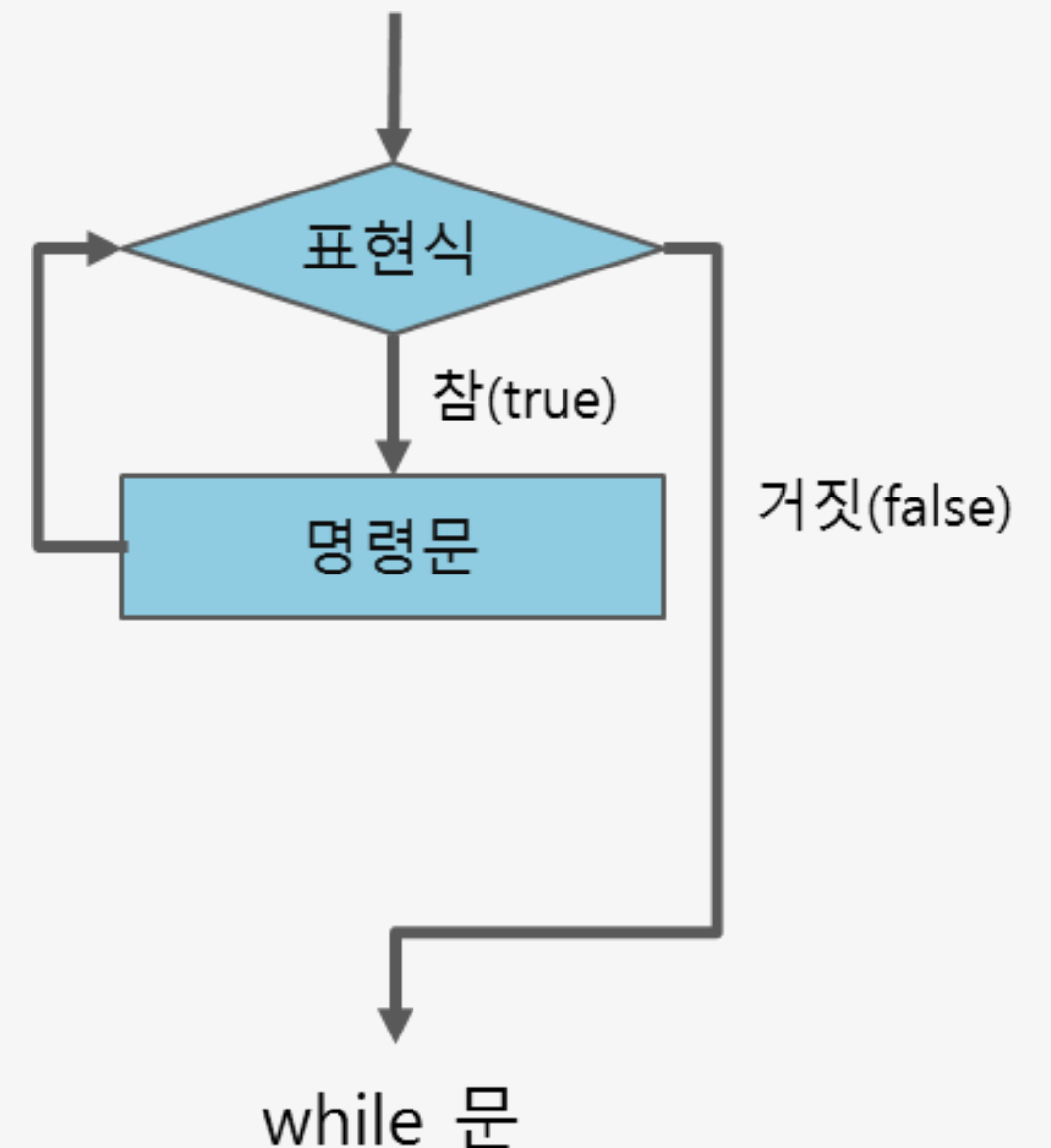
Javascript 문법 : 반복문

- 반복문 : 똑같은 명령을 일정 횟수만큼 반복하여 수행하도록 하는 실행문
 - while 문
 - do / while 문
 - for 문

Javascript 문법 : 반복문

- while 문
 - while 문은 특정 조건을 만족할 때까지 계속해서 실행문을 반복 실행
 - 표현식엔 true,false 의 Boolean !
 - 표현식이 무한loop에 빠지지 않도록 설계

```
while (표현식) {  
    실행문;  
}
```



Javascript 문법 : 반복문

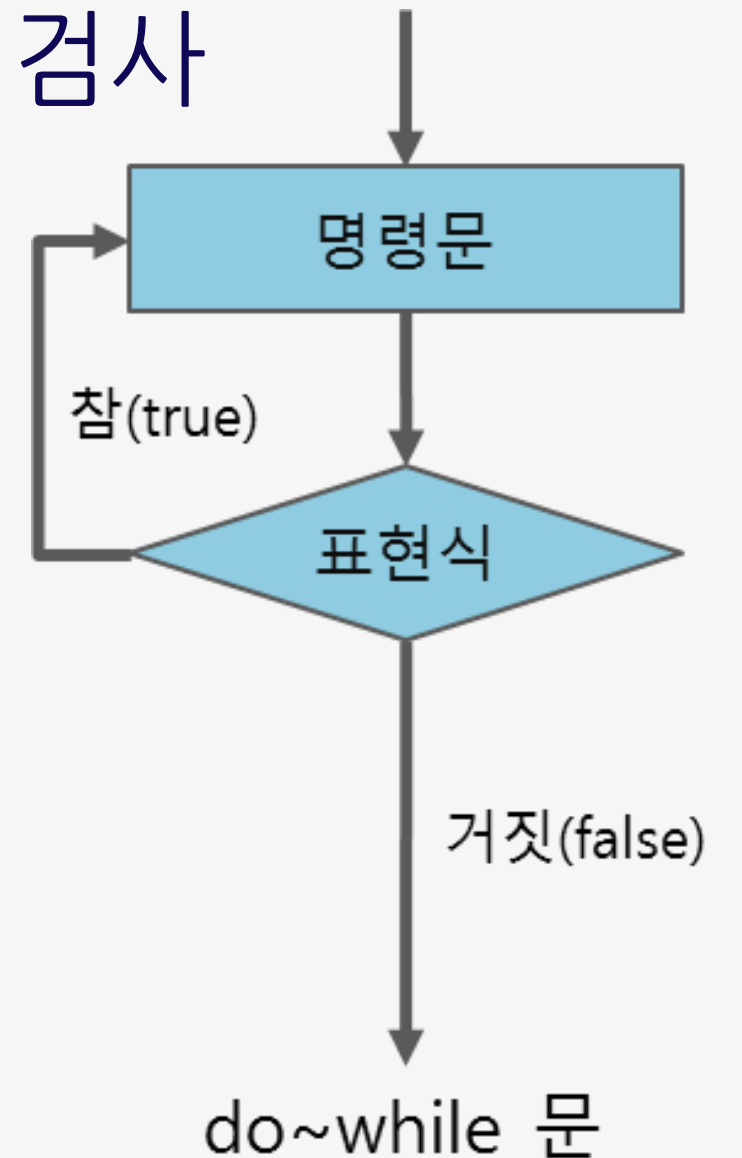
- while 문 : 예제

```
const colors = ['red', 'blue', 'green'];  
let i=0;  
while (colors[i] != null) {  
    console.log( colors[i] );  
    i = i+1;  
}
```

Javascript 문법 : 반복문

- do / while 문
 - while 문은 loop에 진입하기 전에 먼저 표현식부터 검사
 - do / while 문은 실행문을 한번 실행한 후에 표현식을 검사
 - 표현식의 결과와 상관없이 무조건 한 번은 실행

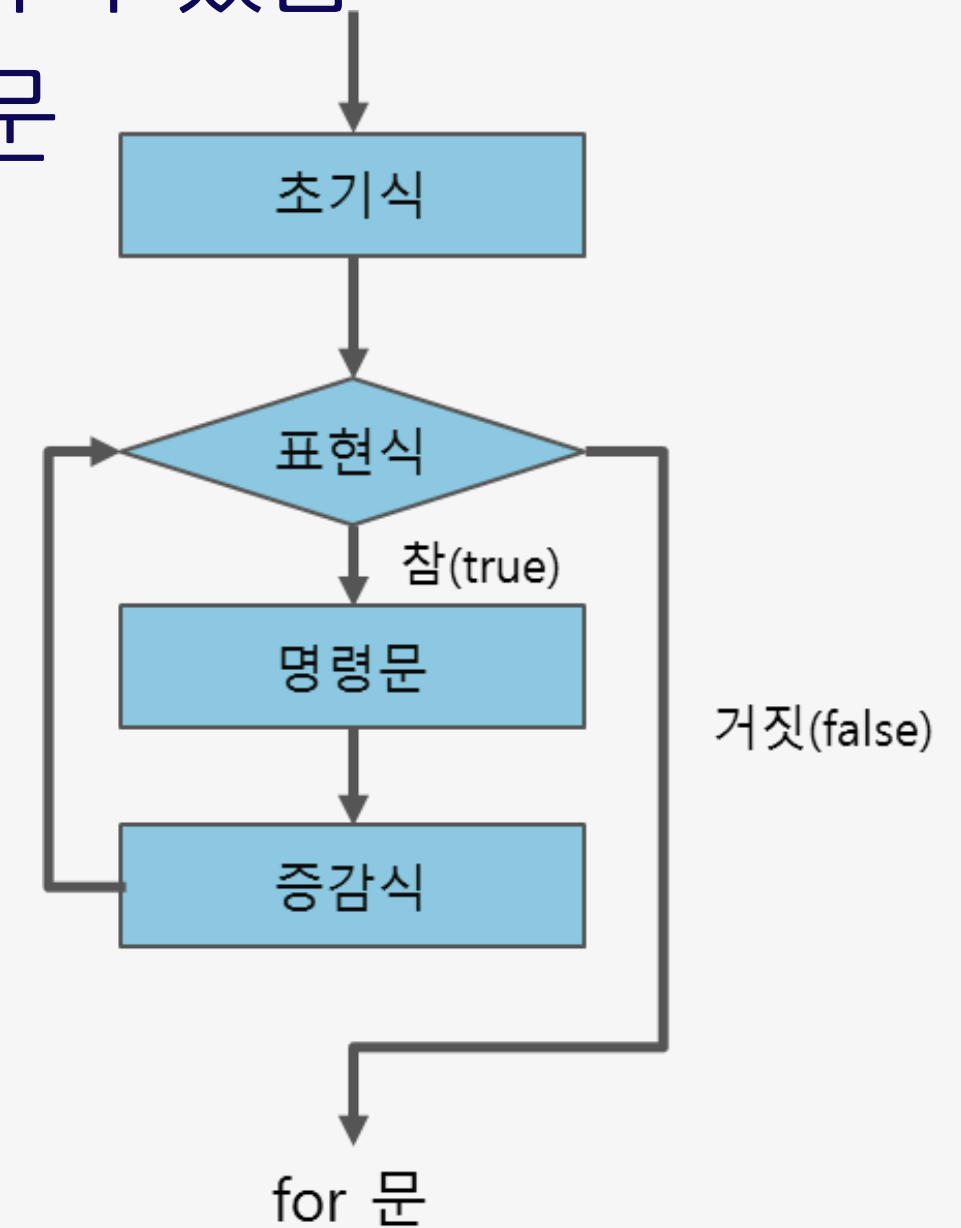
```
do {  
    실행문;  
} while (표현식);
```



Javascript 문법 : 반복문

- for 문
 - while 문은 따로 초기 변수 및 loop 설계를 설정해줘야 했음
 - 초기식, 표현식, 증감식을 모두 포함하고 있는 반복문

```
for ( 초기식 ; 표현식 ; 증감식 ) {  
    실행문 ;  
}
```



Javascript 문법 : 반복문

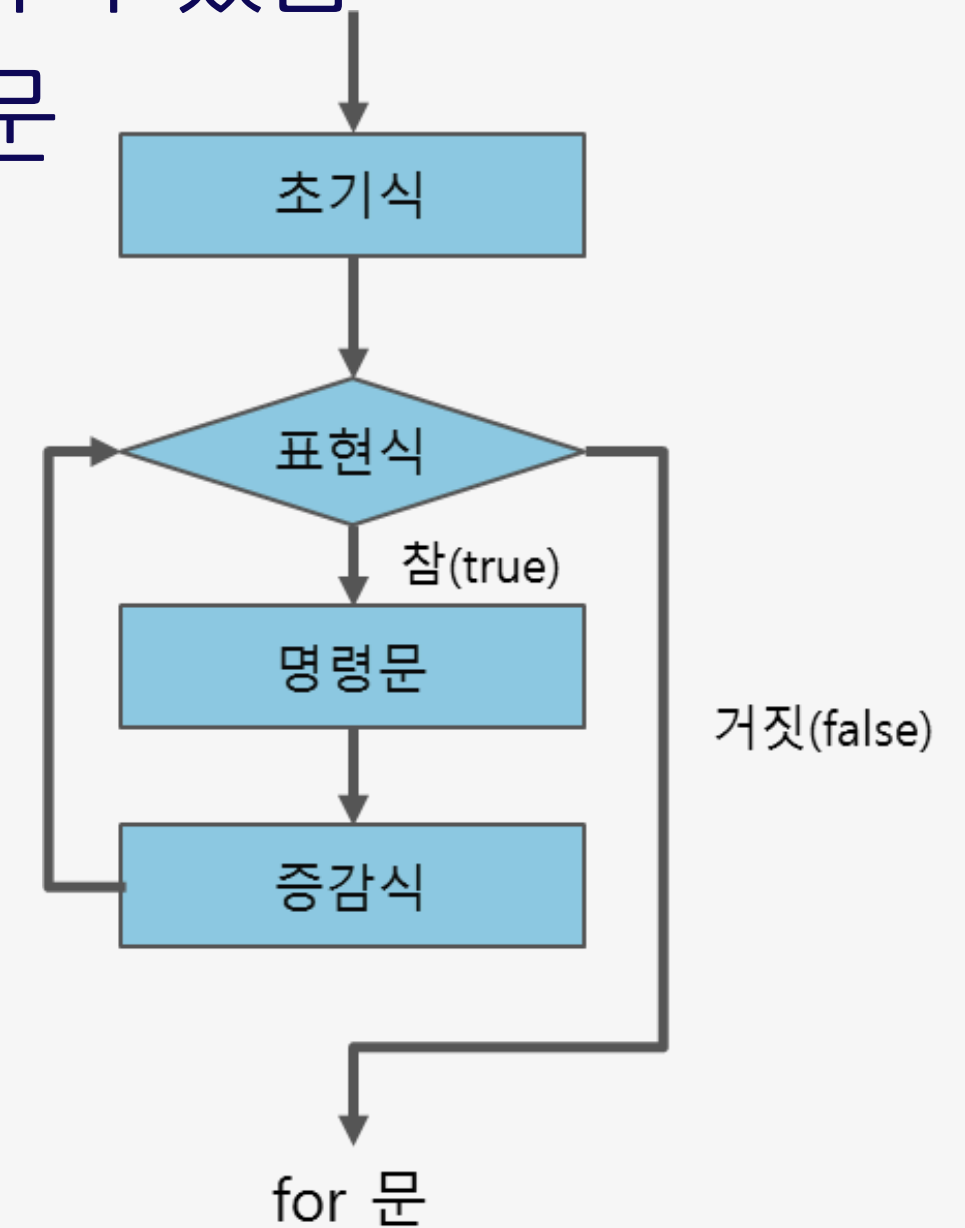
- for 문 : 예제

```
let colors = ['red', 'blue', 'green'];  
for (let i = 0; i < 3; i = i+1) {  
    console.log( colors[i] );  
}
```

Javascript 문법 : 반복문

- for 문
 - while 문은 따로 초기 변수 및 loop 설계를 설정해줘야 했음
 - 초기식, 표현식, 증감식을 모두 포함하고 있는 반복문

```
for ( 초기식 ; 표현식 ; 증감식 ) {  
    실행문 ;  
}
```



Javascript 문법 : 반복문

- for / of 문 , for / in 문
 - 많이 쓰지 않음 (세션장은)
 - Array 와 같은 객체 자료형의 요소에 하나씩 접근하며 실행하는 반복문

```
for (변수 in 객체) {  
    실행문;  
}
```

```
for (변수 of 객체) {  
    실행문;  
}
```

Javascript 문법 : 실습

- 별찍기

- console 창에 다섯 줄의 별을 출력해보자!
- 줄바꿈이 되면 별이 하나씩 늘어나도록!
- Hint

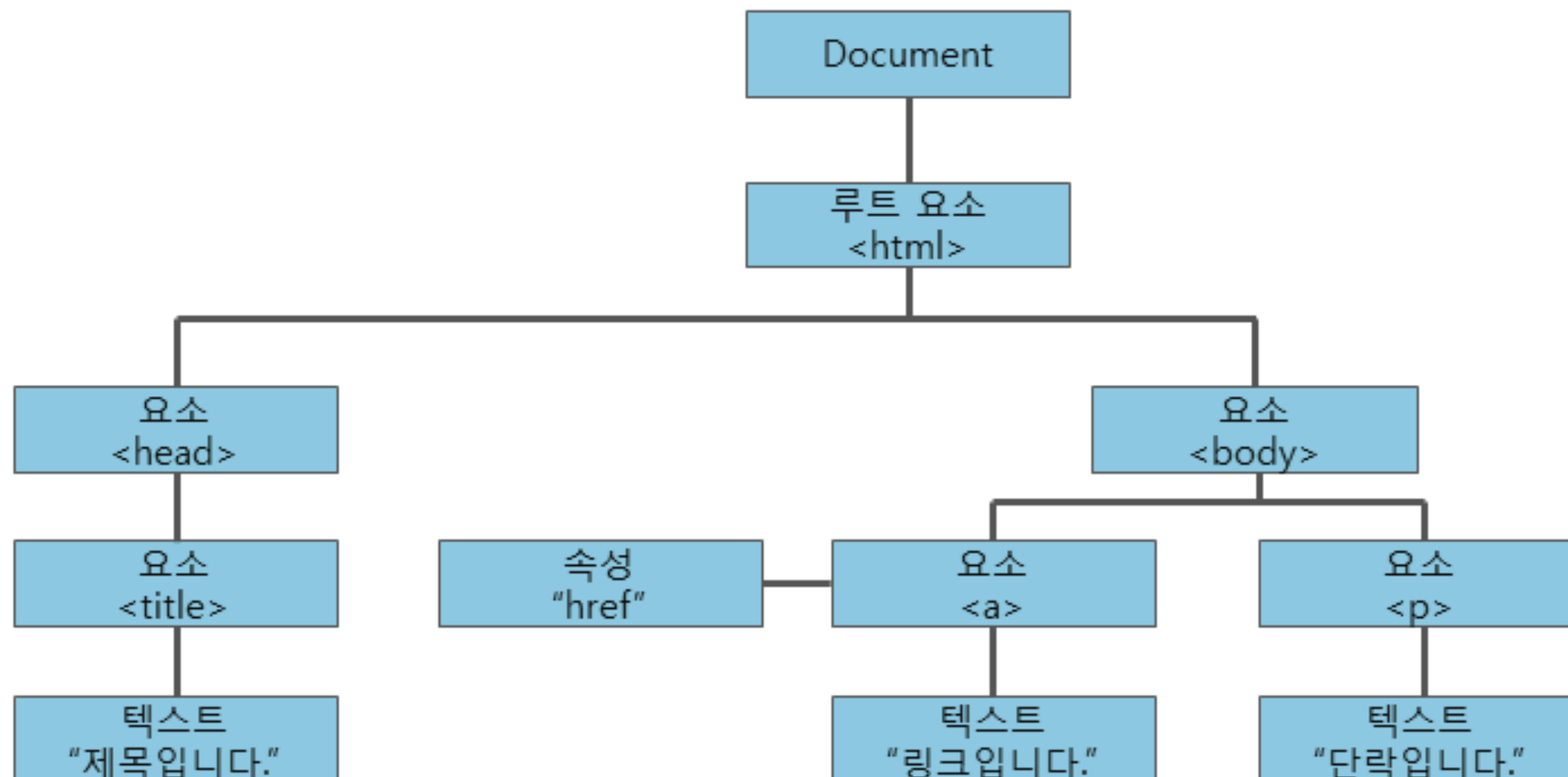
- 변수 = 변수 + 1;
- 빈 문자열 변수
- “\n”
- for문 안에 for문

```
*  
**  
***  
****  
*****
```

[example1.html:18](#)

Javascript DOM

- 문서 객체 모델 (DOM; Document Object Model)
 - 문서 내 모든 요소를 정의하고, 각 요소에 접근하는 방법을 제공하는 구조
 - HTML의 DOM 모델



Javascript DOM

- HTML 요소에 접근하기
 - 제어 대상(HTML 요소)를 찾기 위한 함수(메소드)
 - 대상을 정하고 함수 호출을 하면 객체를 반환

함수 이름	설명	반환값
getElementById()	해당 아이디의 요소를 선택함	객체(한개)
getElementsByClassName()	해당 클래스에 속한 요소를 모두 선택함.	배열(여러개)
getElementsByTagName()	해당 태그 이름의 요소를 모두 선택함	배열(여러개)
querySelector()	해당 셀렉터로 선택되는 요소 중에 첫번째 요소만 선택함	객체(한개)
querySelectorAll()	해당 선택자로 선택되는 요소를 모두 선택함	배열(여러개)

Javascript DOM

- HTML 요소 접근 예제

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>DOM 접근 예시</title>
  </head>
  <body>
    <div id="myElement">myElement id 요소</div>
    <div class="myClass">첫 번째 div 요소</div>
    <div class="myClass">두 번째 div 요소</div>
    <div class="myClass">세 번째 div 요소</div>
    <div>네 번째 div 요소</div>
    <script>
      let getId = document.getElementById("myElement");
      let getClass = document.getElementsByClassName("myClass");
      let getTagName = document.getElementsByTagName("div");
      let query = document.querySelector(".myClass");
      let queryAll = document.querySelectorAll(".myClass");

      console.log(getId);
      console.log(getClass);
      console.log(getTagName);
      console.log(query);
      console.log(queryAll);
    </script>
  </body>
</html>
```

Javascript DOM

- 접근한 객체 수정하기
 - querySelector를 이용해 찾은 객체에서 DOM을 제어할 수 있다.

속성	설명	예시
id	id값	if (myBox.id == "my-box")
className	class값	myBox.className = "box red-box"
tagName	태그의 이름	if (myBox.tagName == "div")
value	value값	nameInputArea = "홍길동"
style	style 오브젝트	myBox.style.color = "red"
innerHTML	태그 내부 HTML 텍스트값	myBox.innerHTML = "ok"

Javascript DOM

- 접근한 객체 수정하기
 - 접근한 태그의 attribute를 제어할 수 있는 함수

속성	설명	예시
getAttribute(name)	특정 attribute의 값을 반환	let value = element.getAttribute("id");
setAttribute(name, value)	특정 attribute의 값을 설정	element.setAttribute("class", "myClass");
hasAttribute(name)	특정 attribute가 존재하는지 여부를 확인. true 혹은 false를 반환	element.hasAttribute("id") // true , false
removeAttribute(name)	특정 attribute를 요소에서 제거	element.removeAttribute("class");

Javascript DOM

- HTML 태그의 attribute를 이용한 event 실행
 - 버튼을 클릭했을 때 자바스크립트 코드를 실행
 - onclick 속성은 클릭 되었을시 발생하는 이벤트를 의미

```
<button onclick="alert('Hello world!');">Click!</button>
```

```
<script>
```

```
function showAlert() {  
    alert('Hello world!');
```

```
}
```

```
</script>
```

```
<button onclick="showAlert()">Click!</button>
```

Javascript DOM

- innerHTML
 - 가져온 HTML 요소에서 내부 HTML 코드를 JavaScript 코드에서 변경

```
<body>
<p id="greeting">Hello world</p>
<button onclick="changeText()">Change</button>
<script>
    function changeText() {
        let target = document.getElementById("greeting");
        target.innerHTML = "Hi";
    }
</script>
</body>
```

Javascript DOM

- innerHTML
 - 가져온 HTML 요소에서 내부 HTML 코드를 JavaScript 코드에서 변경

```
<p>Hello world</p>
```

```
<p><button>Hi</button></p>
```

Javascript DOM

- style
 - 가져온 HTML 요소에서 자바스크립트로 style속성을 적용 가능
 - CSS에서 여러 단어로 이어진 속성은 - 로 구분되었는데
 - (ex. background-color)
 - JavaScript에서는 - 를 사용하지 않고 다음 단어를 대문자로 사용하며 (Camel Case) 접근
 - (ex. backgroundColor)

Javascript DOM

- style

```
<body>
  <p id="greeting">Hello world</p>
  <button onclick="changeColor()">Change</button>
  <script>
    function changeColor() {
      let target = document.getElementById("greeting");
      target.style.backgroundColor = "blue";
    }
  </script>
</body>
```

Javascript DOM : 실습

- 버튼을 누르면 태그가 생성되는 프로그램
 - 버튼을 누르면 태그가 누적되어 생성된다.
 - function : makeH1 , makeA

```
<body>
  <h1>태그 생성하기</h1>
  <button onclick="makeH1()">h1태그 생성</button>
  <button onclick="makeA()">a태그 생성</button>

  <div id="div1"></div>
  <script>

  </script>
</body>
```

태그 생성하기

h1태그 생성

a태그 생성

Javascript 문법 : 과제

- 별찍기
 - console 창에 피라미드 별을 출력해보자!
 - 5줄의 문자열. 첫 줄엔 별 하나 , 다음 줄엔 별이 두개씩 추가됨
 - Hint
 - 변수 = 변수 + 1;
 - 빈 문자열 변수
 - “\n”
 - 공백
 - for문 안에 for문

```
  *
 ***
*****
*****
*****
```

index.html:21

Javascript DOM : 과제

- 버튼을 누르면 span 태그의 숫자가 1씩 증가 되거나 감소되는 프로그램
 - Hint : html 태그에 있는 숫자는 문자열로 인식한다.
 - 0 이하로는 1감소 버튼을 눌러도 동작하지 않도록 하기!
 - function : plus , minus

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="ko">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>Document</title>
7    </head>
8    <body>
9      <span id="span1">0</span>
10     <br />
11     <button onclick="plus()">+1 증가</button>
12     <button onclick="minus()">-1 감소</button>
13     <script></script>
14   </body>
15 </html>
```

