

DOM, 이벤트

5 / DOM, 이벤트



목차

1. DOM 이란?
2. 이벤트(Event) 란? (feat. 이벤트 핸들러)
3. 구조분해할당(Destructuring)
4. 스코프

01

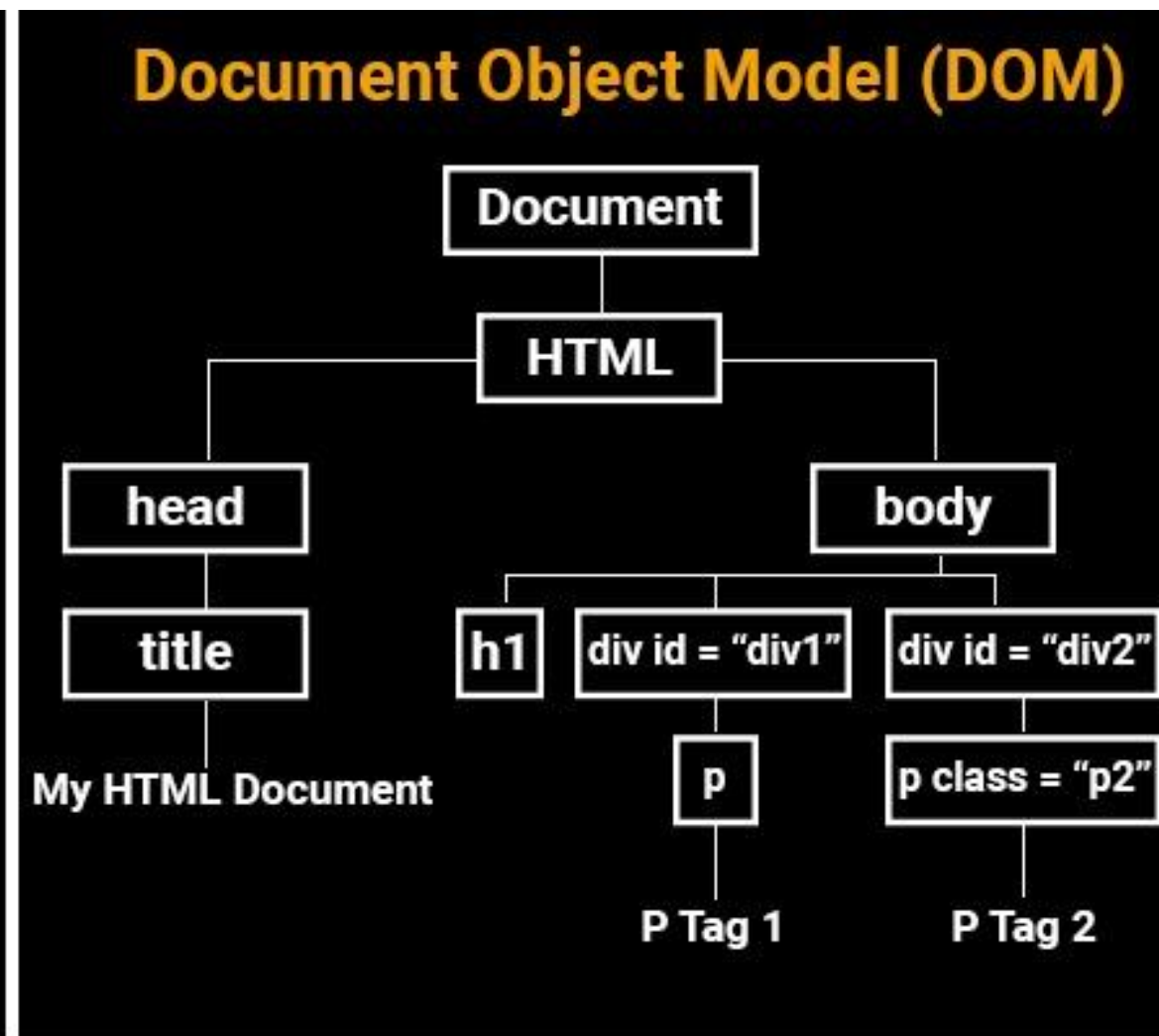
DOM 이란?

☑ DOM

- DOM 이란??
 - 브라우저가 이해할 수 있는 자료구조 (Document Object Model)
 - HTML 문서를 파싱한 결과물

HTML Document

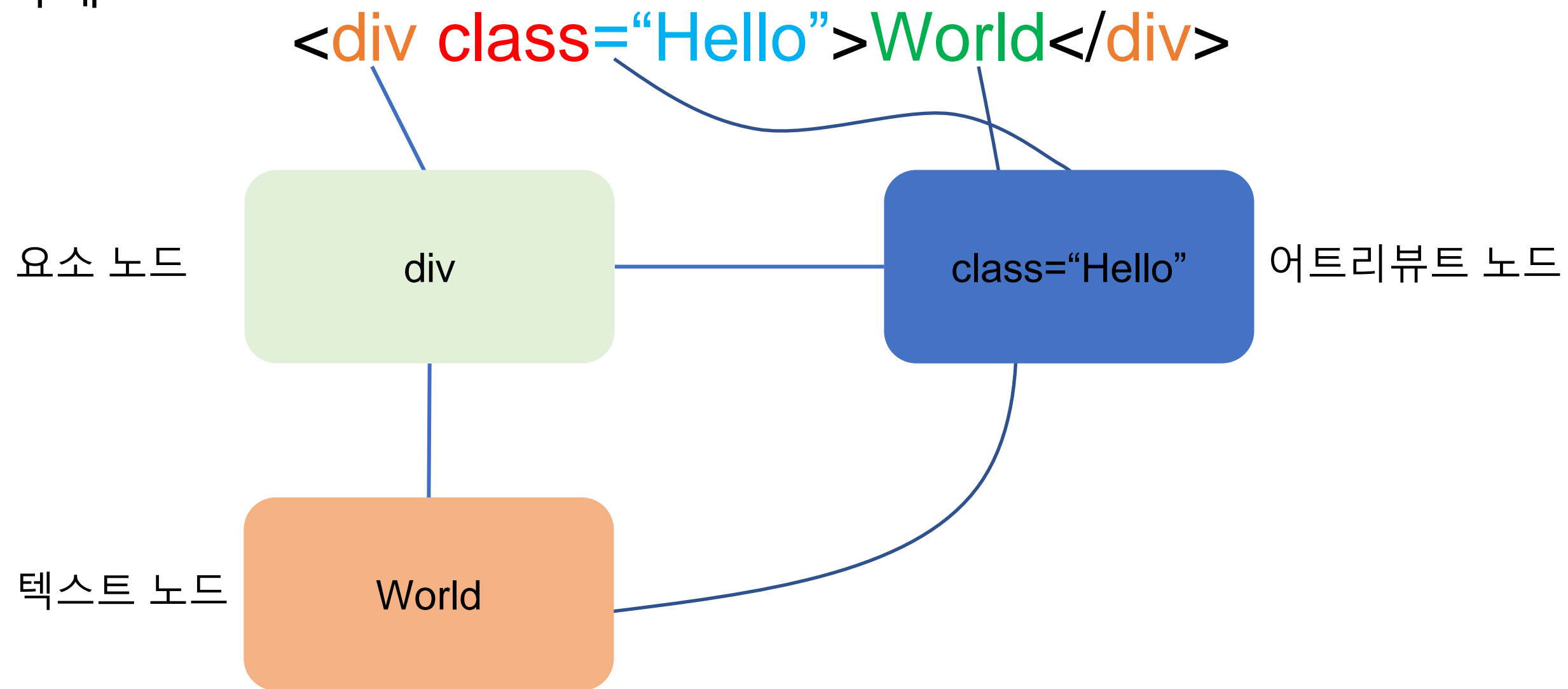
```
index.html x
1 <html>
2   <head>
3     <title>My HTML Document</title>
4   </head>
5
6   <body>
7     <h1>Heading</h1>
8     <div id="div1">
9       <p>P Tag 1</p>
10    </div>
11    <div id="div2">
12      <p class="p2">P Tag 2</p>
13    </div>
14  </body>
15 </html>
```



자바스크립트 문법

☑ DOM

- DOM의 구성요소
 - 노드 객체



☑ DOM

- 노드 객체 타입 (총 12개의 노드 타입이 있음)
 - 문서 노드 (document node)
 - 요소 노드 (element node)
 - 어트리뷰트 노드 (attribute node)
 - 텍스트 노드 (text node)
 - 주석 노드 (comment node)
 - 그 외의 노드 (etc..)

자바스크립트 문법

☑ DOM

- 요소 노드(element node)를 검색 및 생성할 수 있다!
- 먼저 구하는 방법부터 알아보자!
 - id를 이용한 요소 노드 구하기
 - 태그 이름을 이용한 요소 노드 구하기
 - class를 이용한 요소 노드 구하기
 - css 선택자를 이용한 요소 노드 구하기
 - `querySelector`
 - `querySelectorAll`

☑ DOM

- 앞에서 구한 노드를 옮겨다니면서 다른 노드를 탐색할 수도 있다!

- 자식 노드 탐색

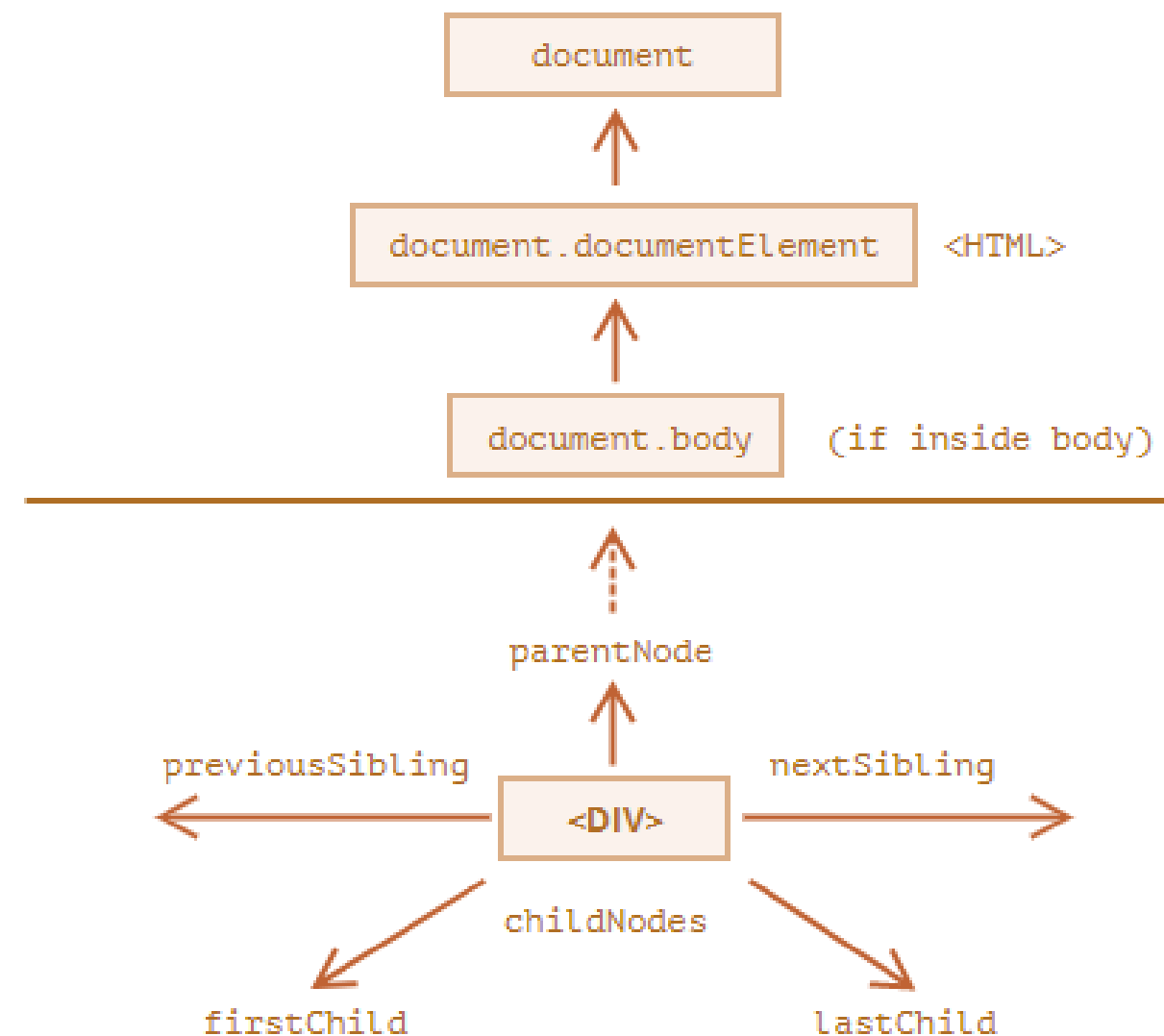
- `childNodes`
- `children`
- `firstChild`
- `lastChild`
- `firstElementChild`
- `lastElementChild`

- 부모 노드 탐색

- `parentNode`

- 형제 노드 탐색

- `previousSibling`
- `nextSibling`
- `previousElementSibling`
- `nextElementSibling`



☑ DOM

- HTMLCollection과 NodeList
 - DOM API가 여러 개의 결과값을 반환하기 위한 DOM 컬렉션 객체
 - 유사 배열 객체이면서 이터러블
 - **노드 객체의 상태 변화를 실시간으로 반영하는(live한) 객체**

☑ DOM

- HTMLCollection
 - 노드 객체의 상태 변화를 실시간으로 반영하는(live) 객체
- NodeList
 - 노드 객체의 상태 변화를 실시간으로 반영하지 않는(non-live) 객체
 - 하지만 childNodes 프로퍼티가 반환하는 NodeList는 HTMLCollection 처럼 실시간으로 노드의 상태를 live하게 반영함
- **권장사항**
 - 상태 변경과 상관 없이 안전하게 DOM 컬렉션을 사용하려면 HTMLCollection, NodeList를 배열로 변환하여 사용하자!

☑ DOM

- DOM 조작

- 아래와 같은 방법으로 새로운 노드를 생성하여 DOM에 추가하거나 기존 노드를 삭제 또는 교체할 수 있다
 - HTML 마크업(HTML 내용) 변경
 - innerHTML
 - insertAdjacentHTML
 - 노드 생성 및 추가
 - **createElement**, createTextNode, createDocumentFragment
 - **appendChild**, insertBefore
 - 노드 복사
 - cloneNode
 - 노드 교체
 - replaceWith, replaceChild
 - 노드 삭제
 - HTML 요소 삭제: remove
 - 노드 삭제: removeChild

자바스크립트 문법

④ DOM

- 요소 노드 텍스트 조작
 - `nodeValue`: 사용 시 `firstChild` 를 참조해서 수정해야 함
 - `textContent`: 해당 요소 노드의 텍스트와 자식 노드의 텍스트를 모두 취득
- `nodeValue`를 사용해서 수정하면 번거로우니 `textContent` 를 쓰자

자바스크립트 문법

☑ DOM

- 스타일 조작
 - 인라인 스타일 조작
 - 클래스 조작
 - className
 - classList
 - add(...className)
 - remove(...className)
 - item(index)
 - contains(className), replace(oldClassName, newClassName), toggle(className)...
- CSS 스타일 참조: window.getComputedStyle

자바스크립트 문법

☑ DOM

- 그 외 기능
 - HTML 어트리뷰트를 수정

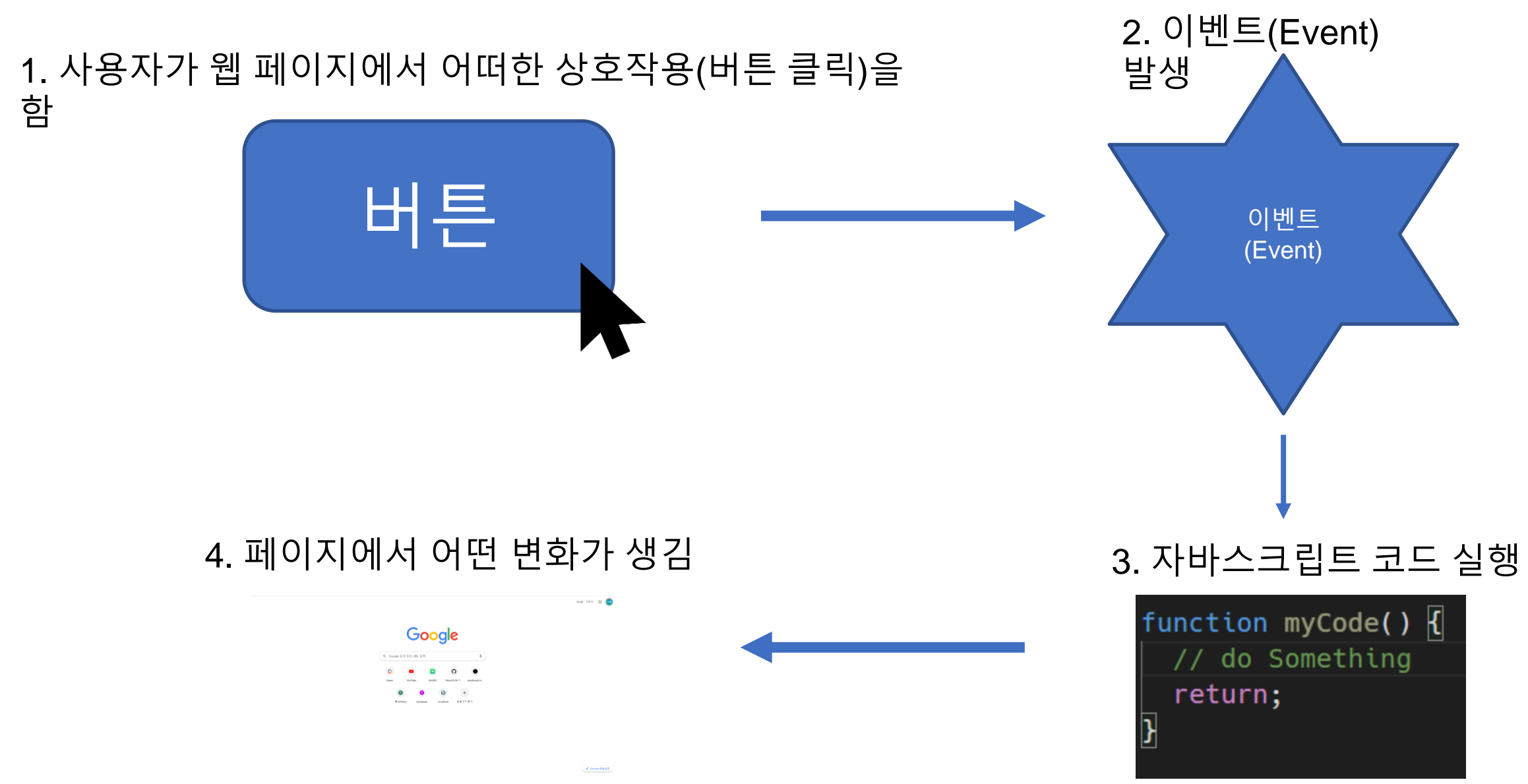
02

이벤트(Event) 란? (feat. 이벤트 핸들러)

④ 이벤트

- 이벤트(Event) 정의
 - 사전적 정의: 월드컵, 올림픽 등의 행사를 의미함
 - 프로그램상 정의: 우리가 프로그래밍하고 있는 시스템에서 일어나는 사건(action) 혹은 발생(occurrence)
 - 브라우저에서 사건이란 마우스 클릭, 스크롤링, 텍스트 박스 입력 등등 사용자가 브라우저와 상호작용 하면서 일어나는 사건을 의미

이벤트



자바스크립트 문법

④ 이벤트

- 이벤트 핸들러(Event Handler) 정의
 - 각각의 이벤트들은 이벤트 핸들러(event handler)를 가지고 있음
 - 이벤트 핸들러란 이벤트가 발생되면 실행되는 코드 블록 (JavaScript 함수)

index.html

```
<button>Change color</button>
```

예제) 버튼 클릭시 색 변경



index.js

```
const btn = document.querySelector('button');

function random(number) {
  return Math.floor(Math.random() * (number+1));
}

// 이벤트 핸들러 함수
btn.onclick = function() {
  const rndCol = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) + ')';
  document.body.style.backgroundColor = rndCol;
}
```


④ 이벤트

- 이벤트 종류
 - 마우스 이벤트: click, dbclick, mousedown, mouseup....
 - 키보드 이벤트: keydown, keypress, keyup
 - 포커스 이벤트: focus, blur, focusin, focusout
 - 폼 이벤트: submit, reset
 - 값 변경 이벤트: input, change, readystatechange
 - DOM 뮤테이션 이벤트: DOMContentLoaded
 - 뷰 이벤트: resize, scroll
 - 리소스 이벤트: load, unload, abort, error

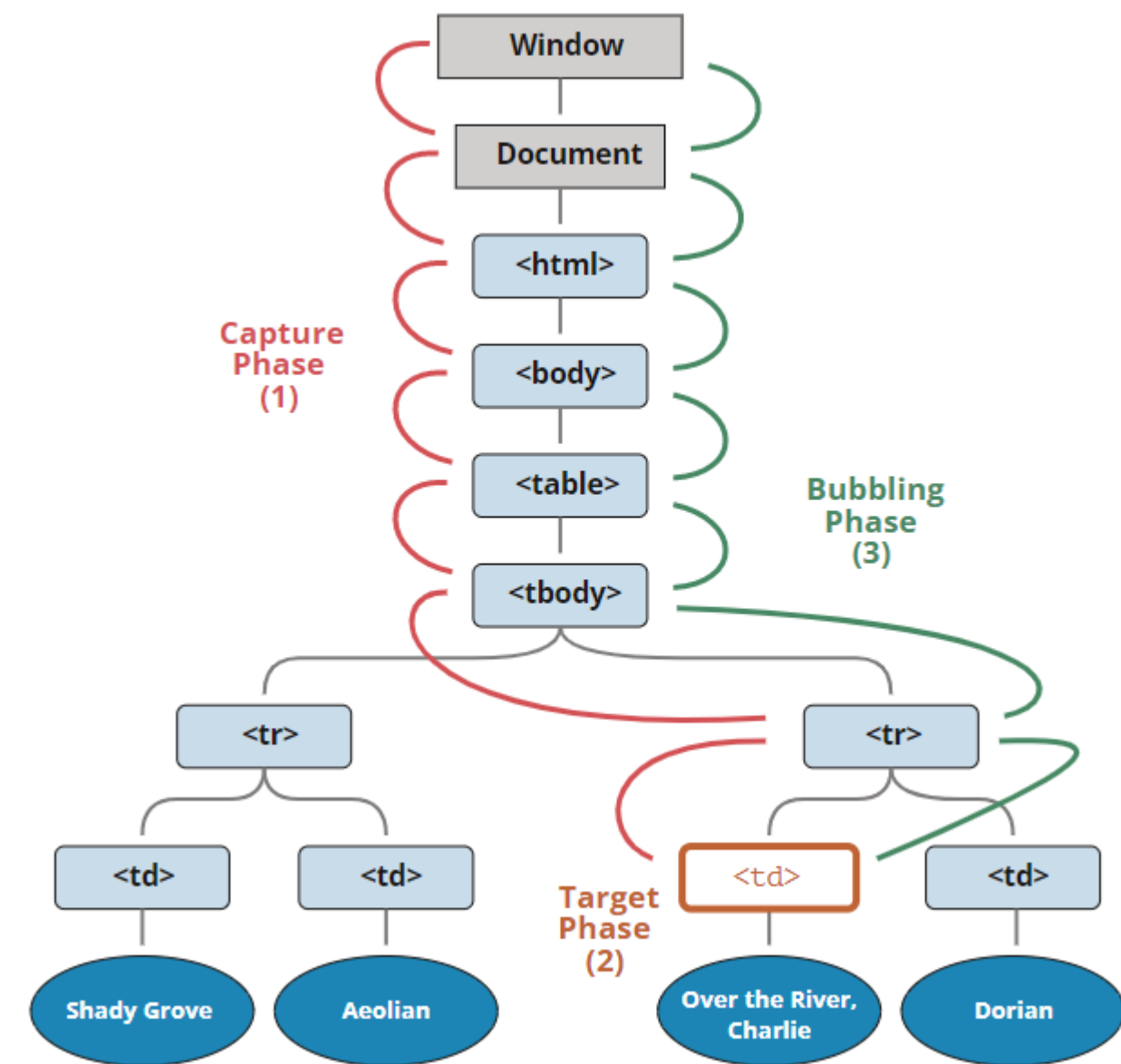
자바스크립트 문법

④ 이벤트

- 이벤트 핸들러 등록
 - 이벤트 핸들러 어트리뷰트 방식
 - 이벤트 핸들러 프로퍼티 방식
 - addEventListener
 - `EventTarget.addEventListener('eventType', yourFunction, [, useCapture]);`
- 이벤트 핸들러 제거
 - removeEventListener
 - `EventTarget.removeEventListener('eventType', yourFunction, [, useCapture]);`

④ 이벤트

- 이벤트 전파
 - 앞서 배웠던 이벤트가 DOM 요소 노드에서 발생하면 DOM 트리를 거쳐서 전파가 된다
 - 캡처링 단계: 이벤트가 상위 요소에서 하위 요소로 전파
 - 타깃 단계: 이벤트가 해당 요소에 도달
 - 버블링 단계: 이벤트가 하위 요소에서 상위 요소로 전파



④ 이벤트

- 이벤트가 버블링을 통해 전파되지 않은 이벤트들
 - 포커스 이벤트: focus / blur
 - 리소스 이벤트: load / unload / abort / error
 - 마우스 이벤트: mouseenter / mouseleave

자바스크립트 문법

④ 이벤트

- 이벤트 위임
 - 여러 개의 하위 DOM 요소에 이벤트 핸들러를 등록하는 대신 상위 이벤트 DOM 요소에 이벤트 핸들러를 등록하는 방법
 - 100개의 버튼에 이벤트를 일일이 등록하는 대신 상위 요소에 한번만 등록해서 해결할 수 있다면 얼마나 편리한가?

④ 이벤트

- DOM 요소 기본 동작 조작
 - 앞서 배운 이벤트 전파가 원치 않게 이뤄지는 경우도 있다. 이를 위해 제공되는 기능들이 있다.
 - preventDefault: DOM 요소 기본 동작 중단
 - stopPropagation: 이벤트 전파 중지

03

구조 분해 할당

자바스크립트 문법

④ 구조 분해 할당

배열 또는 객체를 분해하여 1개 이상의 변수에 개별적으로 할당하는 것을 의미

- 사용방법

```
const arr = [1, 2, 3];

// ES6 배열 디스트럭처링 할당
// 변수 one, two, three를 선언하고 배열 arr을 디스트럭처링하여 할당한다.
// 이때 할당 기준은 배열의 인덱스다.
const [one, two, three] = arr;

console.log(one, two, three); // 1 2 3
```

배열: 인덱스 기준으로 할당함

```
const user = { firstName: 'Kitae', lastName: 'Park' };

// ES6 객체 디스트럭처링 할당
// 변수 lastName, firstName을 선언하고 user 객체를 디스트럭처링하여 할당한다.
// 이때 프로퍼티 키를 기준으로 디스트럭처링 할당이 이루어진다. 순서는 의미가 없다.
const { lastName, firstName } = user;

console.log(firstName, lastName); // Kitae Park
```

객체: 프로퍼티 키 기준으로 할당
선언된 변수 이름과 프로퍼티 키가
일치해야함

04

스코프

자바스크립트 문법

④ 스코프

스코프란?

- 식별자(변수 이름, 함수 이름, 클래스 이름 등)가 유효한 범위

```
function add(x, y) {  
  // 매개변수는 함수 몸체 내부에서만 참조할 수 있다.  
  // 즉, 매개변수의 스코프(유효범위)는 함수 몸체 내부다.  
  console.log(x, y); // 2 5  
  return x + y;  
}  
  
add(2, 5);  
  
// 매개변수는 함수 몸체 내부에서만 참조할 수 있다.  
console.log(x, y); // ReferenceError: x is not defined
```


☑ 스코프

스코프의 종류

구분	설명	스코프	변수
전역	코드의 가장 바깥 영역	전역 스코프	전역 변수
지역	함수 몸체 내부 또는 코드블록	지역 스코프	지역 변수

☑ 스코프

전역 스코프

```
var x = "global x";
var y = "global x";

function outer() {
  var z = "outer's local z";

  console.log(x); // ① global x
  console.log(y); // ② global y
  console.log(z); // ③ outer's local z

  function inner() {
    var x = "inner's local x";

    console.log(x); // ④ inner's local x
    console.log(y); // ⑤ global y
    console.log(z); // ⑥ outer's local z
  }

  inner();
}

outer();

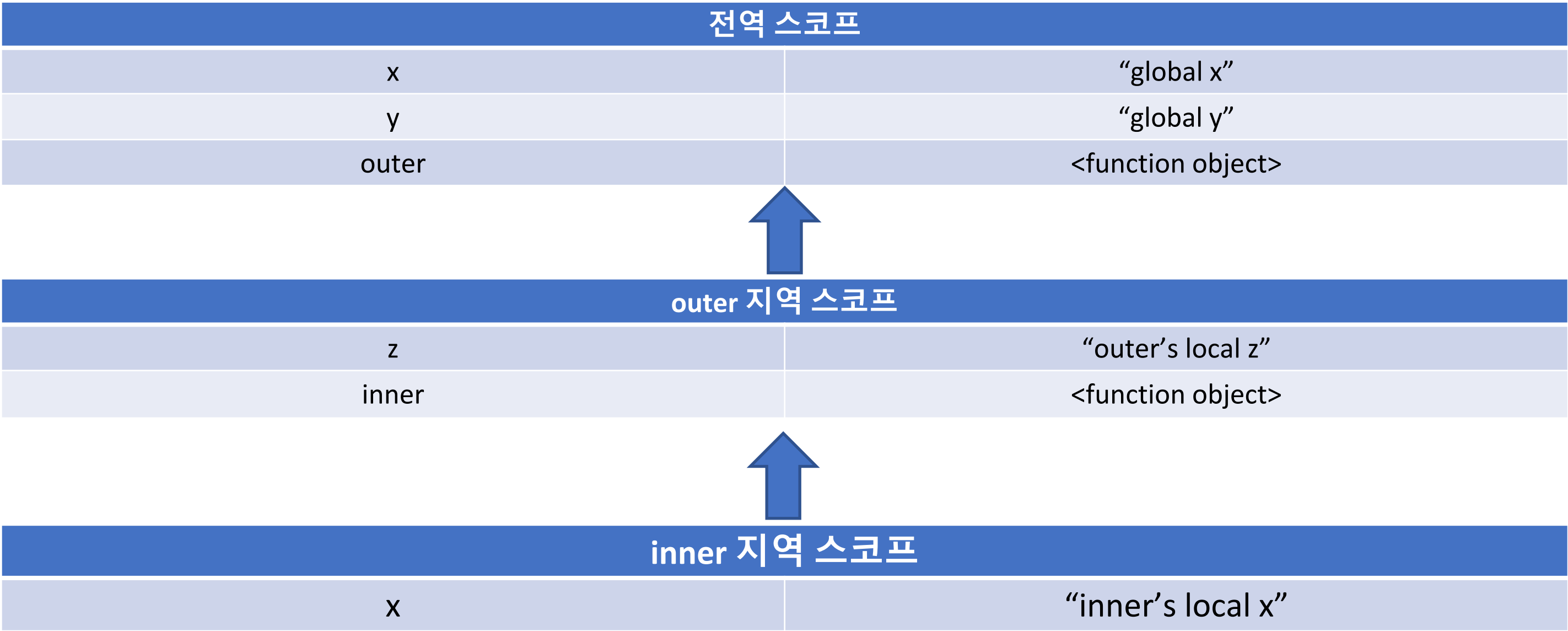
console.log(x); // ⑦ global x
console.log(z); // ⑧ ReferenceError: z is not defined
```

지역 스코프

지역 스코프

☑ 스코프

스코프 체인: 스코프가 계층적으로 연결된 것



자바스크립트 문법

☑ 스코프

렉시컬 스코프의 개념

- 함수를 어디서 정의했는지에 따라 함수의 상위 스코프를 결정하는 것

```
var x = 1;

function foo() {
  var x = 10;
  bar();
}

function bar() {
  console.log(x);
}

foo(); // ?
bar(); // ?
```

자바스크립트 문법

☑ 참고

- 모던 자바스크립트 Deep Dive
- 짐코딩 CODING GYM
- 코딩애플
- <https://ko.javascript.info/dom-navigation>