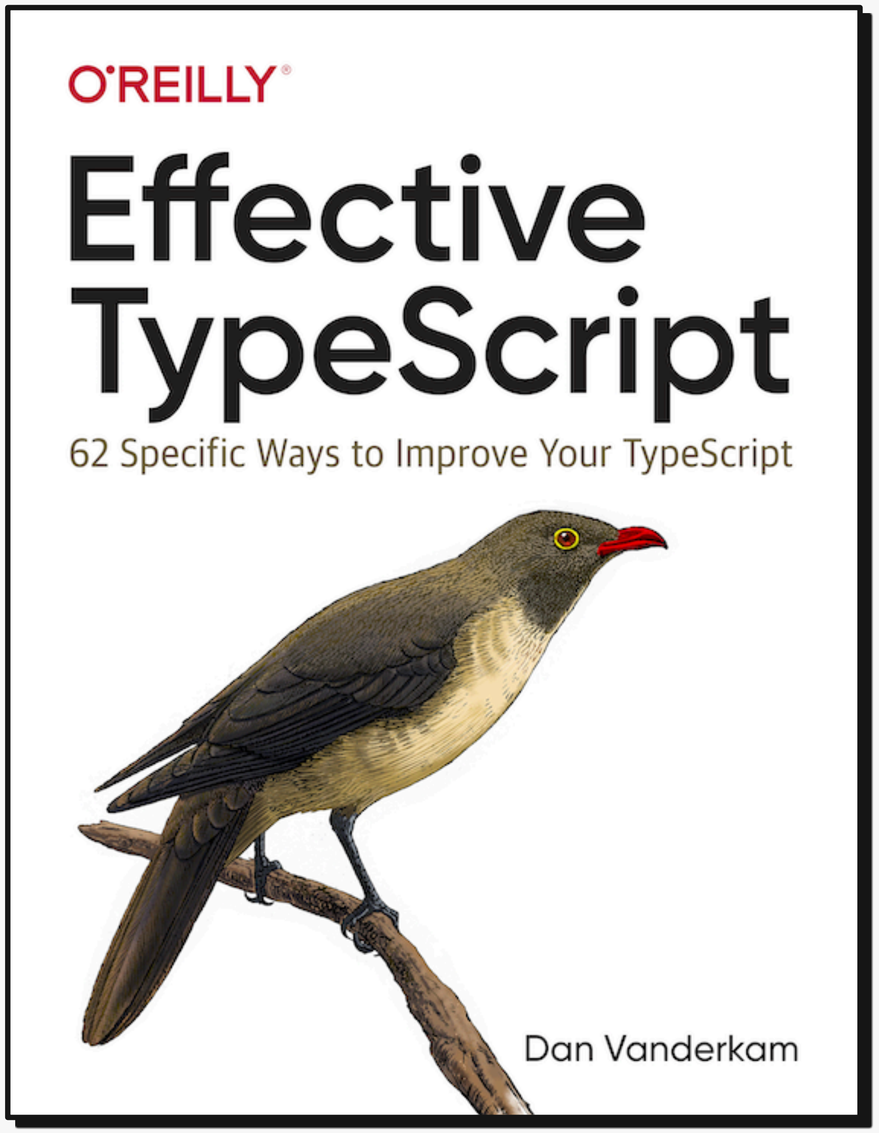
Effective TypeScript

Item 55

DOM 계층 구조 이해하기

@Bori-github

목 차



01 DOM 계층 구조

- DOM 노드는 종류에 따라 각각 다른 프로퍼티를 지원
- Element와 EventTarget에 달려있는 노드의 구체적인 타입을 안다면?

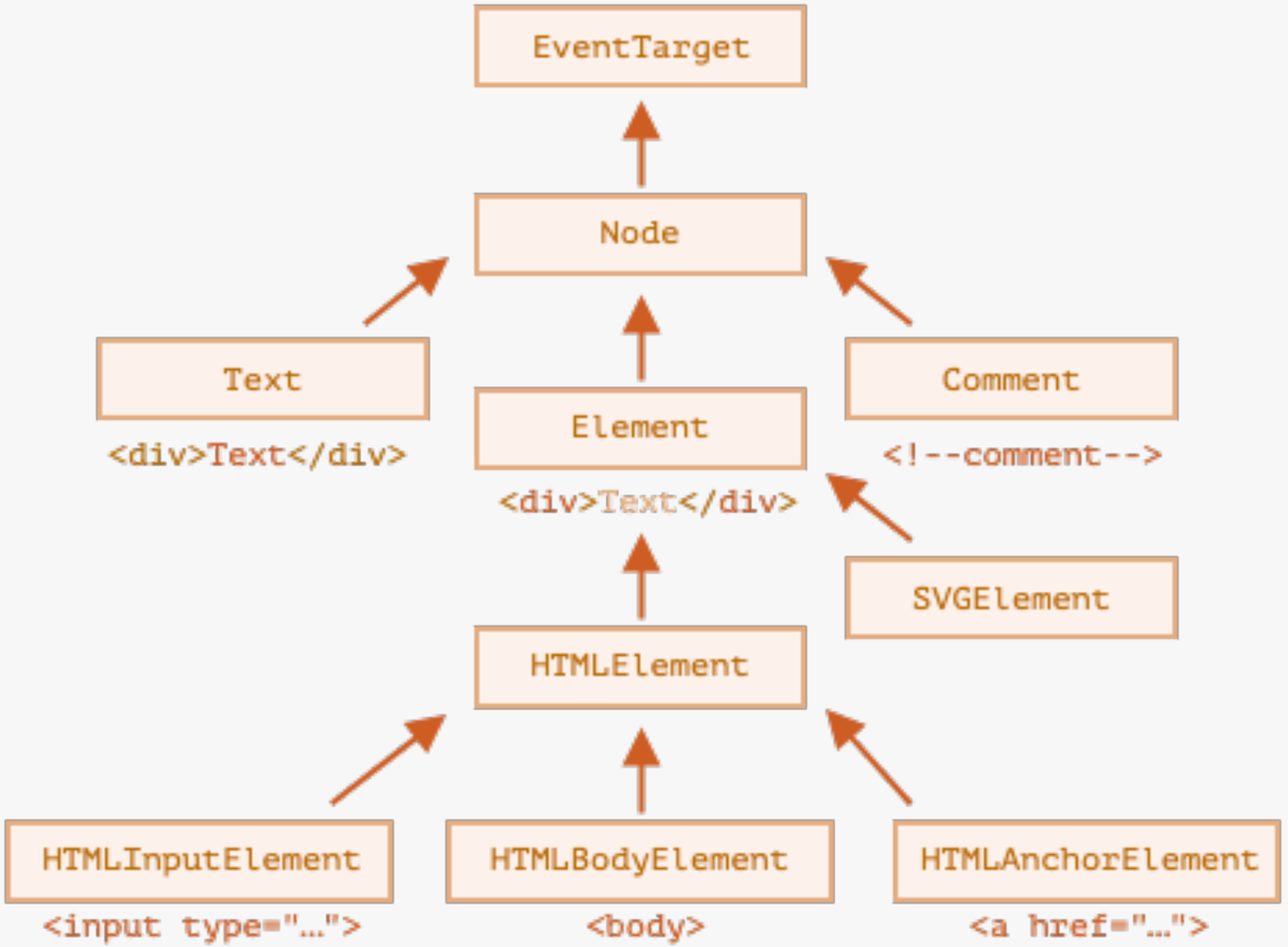
  ⇒ 타입 오류를 디버깅 할 수 있다.

  ⇒ 타입 단언을 사용 시점을 알 수 있다.

⬇

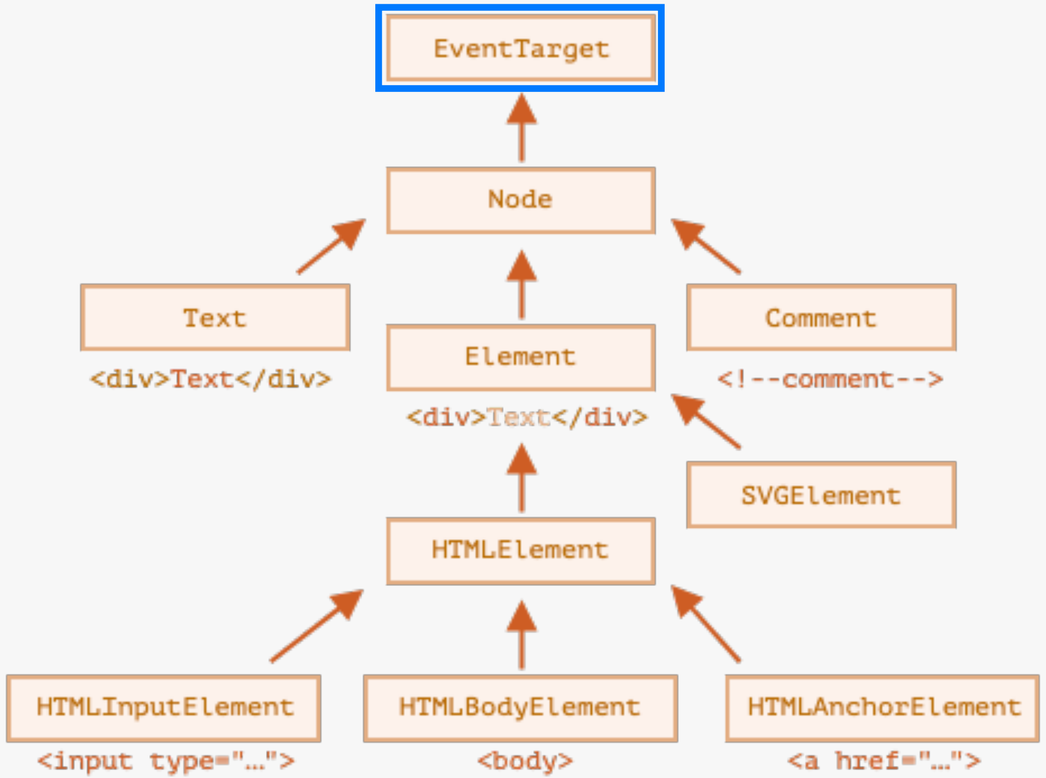
타입스크립트에서는 DOM 엘리먼트 계층 구조를 파악하기 용이

01 DOM 계층 구조


EventTarget
Node
Text
Text
Element
Text
Comment
<!--comment-->
SVGElement
HTMLElement
HTMLInputElement
<input type="...">
HTMLBodyElement
<body>
HTMLAnchorElement
<a href="...">
Text

01 DOM 계층 구조

📌 EventTarget

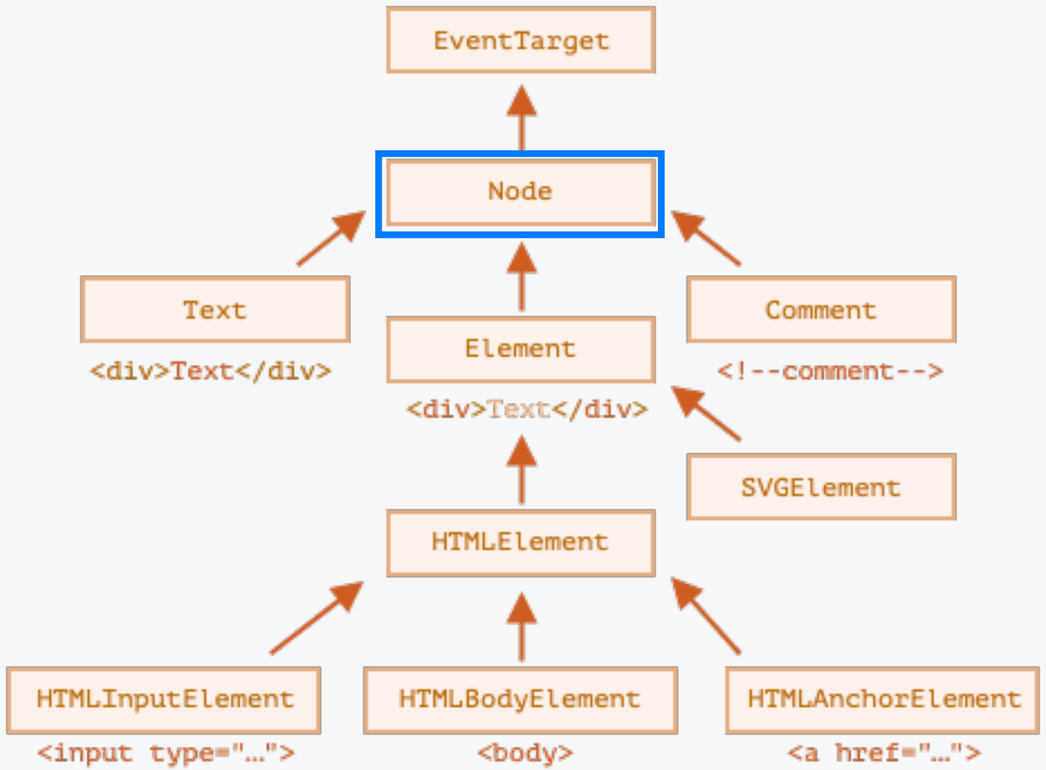


- DOM 타입 중 가장 추상화된 타입
- 이벤트 리스너를 추가/제거 및 이벤트를 보내는 것 가능
- 루트에 있는 EventTarget이 <u>모든 DOM 노드의 기초</u>

  ⇒ 이를 통해 DOM 노드에서 이벤트를 사용 가능

01 DOM 계층 구조

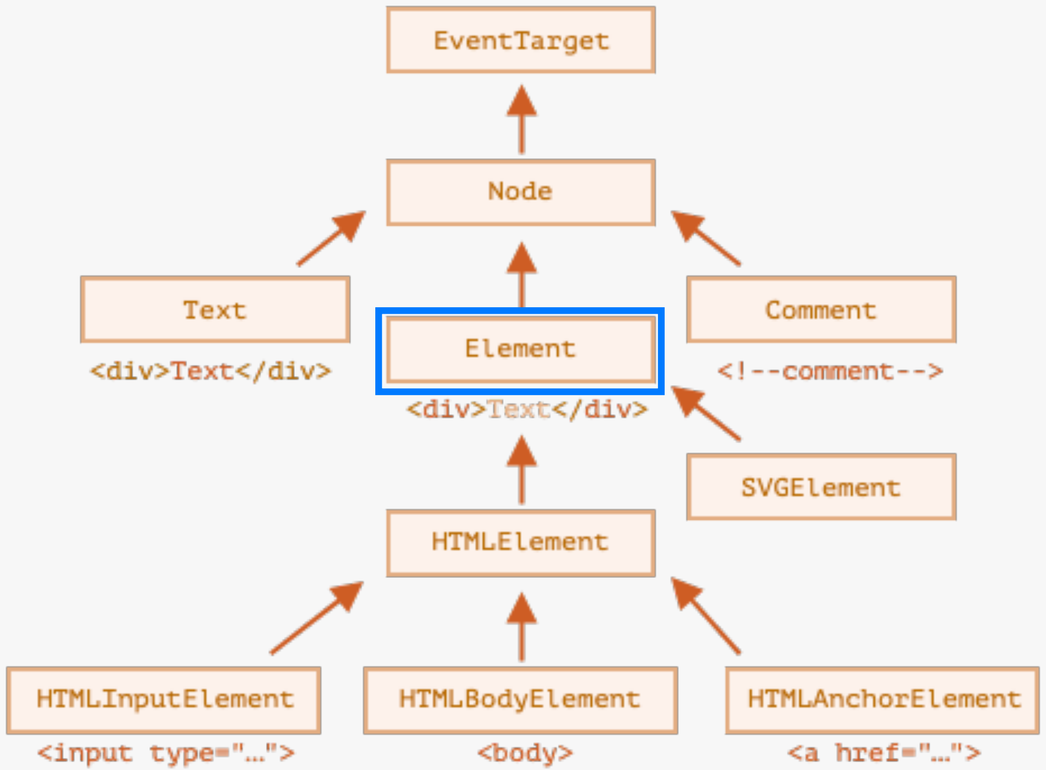
📌 Node



- DOM 노드의 베이스 역할
- parentNode, nextSibling, childNodes 등의 주요 트리 탐색 기능을 제공
- 다음 타입은 Node 타입을 상속 받음
  - 텍스트 노드를 위한 Text
  - 요소 노드를 위한 Element
  - 주석 노드를 위한 Comment

01 DOM 계층 구조

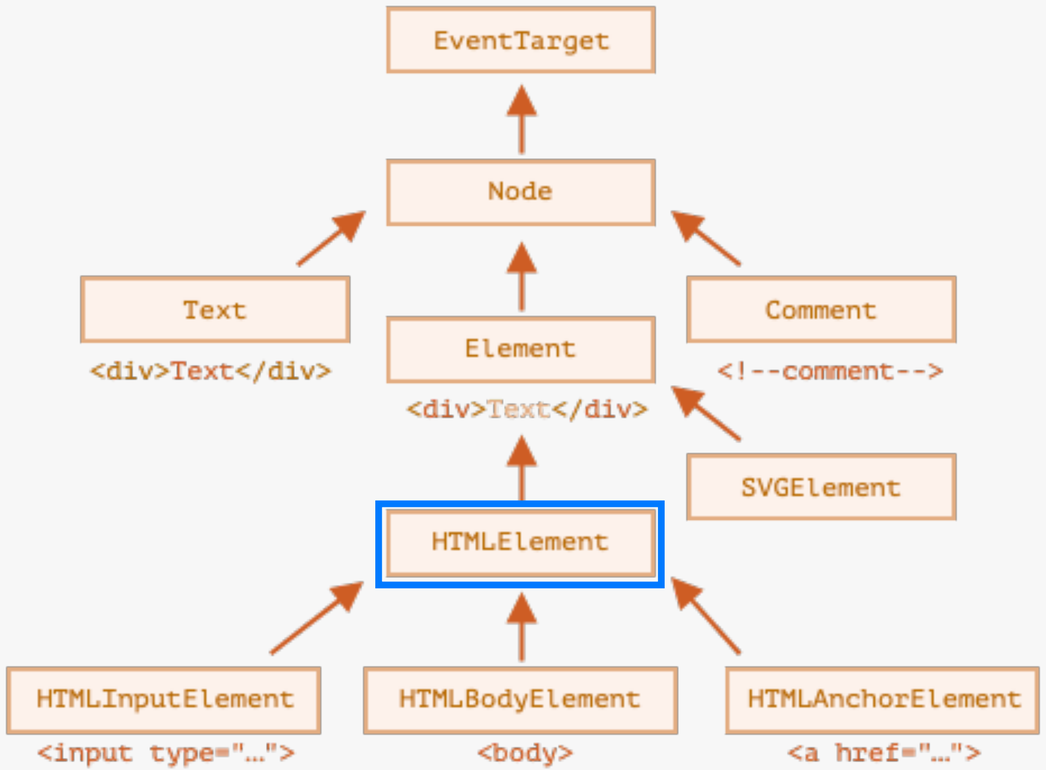
📌 Element



- DOM 요소를 위한 베이스
- children, getElementsByTagName, querySelector 같이 요소 전용 탐색을 도와주는 프로퍼티나 메서드가 Element를 기반으로 함
- 브라우저는 HTML뿐만 아니라 XML, SVG도 지원
  - Element는 이와 관련된 SVGElement, XMLElement, HTMLElement 의 베이스 역할

01 DOM 계층 구조

📌 HTMLElement

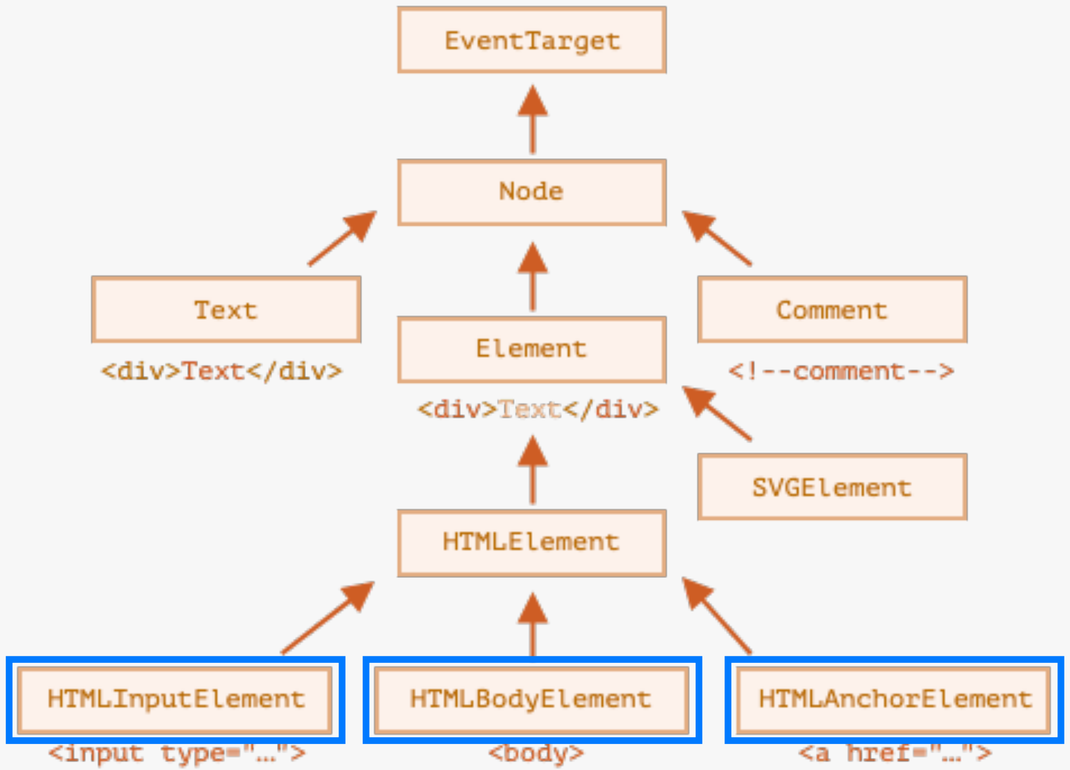


- HTML 요소 노드의 베이스 역할
- HTMLxxxElement는 <u>실제 HTML 요소에 대응</u>하고 **HTMLElement를 상속 받음**

  예) HTMLInputElement는 <input> 요소에 대응

01 DOM 계층 구조

📌 HTMLxxxElement



- HTMLxxxElement 형태의 특정 엘리먼트들은

  자신만의 고유한 속성을 가짐

  예) HTMLImageElement 에는 src 속성,

  HTMLInputElement 에는 value 속성 등이 존재

- 고유한 속성에 접근하려면, 타입 정보가 실제 엘리먼트

  타입이어야 하기 때문에 구체적인 타입 지정이 필요

01 DOM 계층 구조

보통 HTML 태그 값에 해당하는 리터럴 값을 사용하면

DOM에 대한 정확한 타입을 얻을 수 있음

```
1  document.getElementsByTagName('p')[0];  // HTMLParagraphElement
2  document.createElement('button');  // HTMLButtonElement
3  document.querySelector('div');  // HTMLDivElement
```

01 DOM 계층 구조

다음과 같이 요소의 정보를 알 수 없는 경우, 정확한 타입을 얻을 수 없음

```
1  document.getElementById('my-div');  // HTMLElement
```

DOM 관련해서는 타입스크립트보다 개발자가 더 정확히 알고 있음

⇒ 타입 단언문을 사용 가능

```
1  document.getElementById('my-div') as HTMLDivElement;
```

01 DOM 계층 구조

strictNullChecks가 설정되어 있다면,

document.getElementById의 null 체크 필요

```
1  const div = document.getElementById('my-div'); // HTMLElement | null
2
3  if (div !== null) {
4    // ...
5  }
```

02

Event 타입 계층 구조

📌 Event

- 가장 추상화된 이벤트로 모든 이벤트에 공통된 속성과 메서드를 가짐
- DOM 요소는 이벤트를 "수신"하고, 받은 이벤트를 "처리"

📌 UIEvent

- 모든 종류의 사용자 인터페이스 이벤트
- MouseEvent, TouchEvent, WheelEvent, KeyboardEvent는 UIEvent를 상속 받음

02

Event 타입 계층 구조

📌 MouseEvent

- 사용자가 마우스와 같은 포인팅 장치를 사용해 상호작용할 때 발생하는 이벤트

📌 TouchEvent

- 모바일 기기와 같이 터치를 감지할 수 있는 표면과 접촉 상태가 변경될 때 발생하는 이벤트

📌 WheelEvent

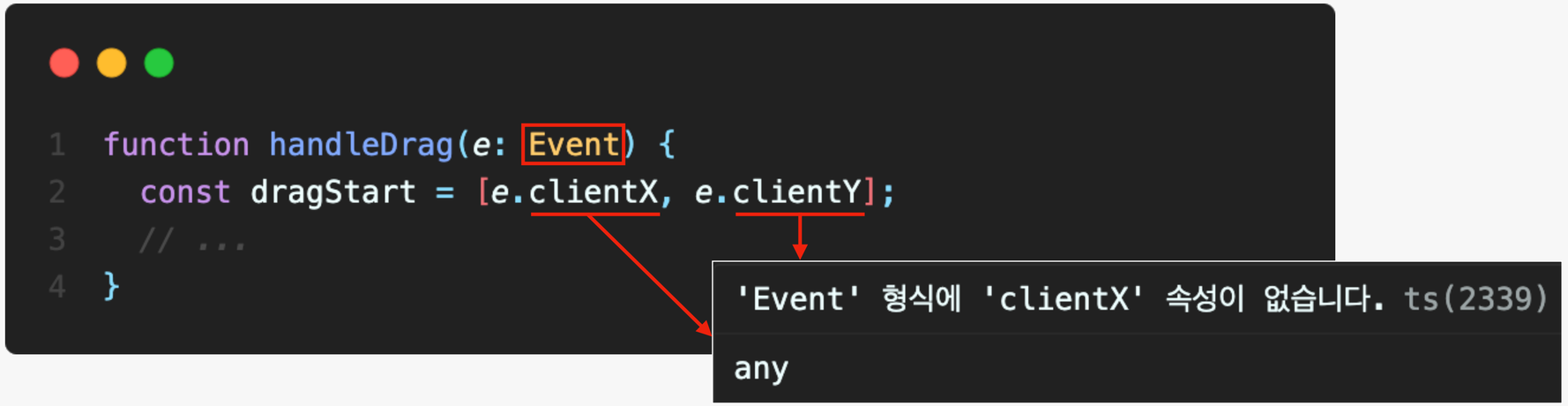
- 사용자가 마우스 휠이나 그와 비슷한 장치를 움직일 때 발생하는 이벤트

📌 KeyboardEvent

- 사용자가 키보드의 키를 눌렀을 때 발생하는 이벤트

02 Event 타입 계층 구조

☑ 이벤트 타입 지정하기



⬇

고유의 속성을 사용하기 위해 이벤트를 구체적으로 지정

02 Event 타입 계층 구조

✅ 구체적인 이벤트 타입 지정하기

```
function handleDrag(e: MouseEvent) {
  const dragStart = [e.clientX, e.clientY];
  // ...
}
```

⬇

구체적인 이벤트 타입을 지정하여 **고유의 속성을 에러없이 사용 가능**

05

마무리

- DOM에는 타입 계층 구조가 있고, 타입스크립트에서 DOM 타입은 중요한 정보입니다.
- DOM 계층 구조의 각 타입간의 차이점과 이벤트 타입의 차이점을 알아야 고유의 속성을 에러없이 사용할 수 있습니다.
- DOM 엘리먼트와 이벤트에는 구체적인 타입 정보를 사용하거나, 타입스크립트가 추론할 수 있도록 문맥 정보를 활용해야 합니다.

Thanks!

Do you have any questions?

qhflrnfl4324@gamil.com
https://github.com/Bori-github