



함수선언식과표현식

함수 선언식과 함수 표현식

1. 함수 선언식 (Function Declaration)

- **형태:** `function` 키워드를 사용해 함수를 정의합니다.
- **호이스팅:** 함수 선언식은 코드에서 정의된 위치와 상관없이, 자바스크립트가 실행될 때 해당 함수의 선언이 코드의 최상단으로 끌어올려집니다. 즉, 함수가 정의되기 전에 호출해도 정상적으로 실행됩니다.

예시:

```
hello(); // 정상적으로 실행됨: "Hello, World!"

function hello() {
  console.log("Hello, World!");
}
```

위 코드에서는 함수 `hello` 가 선언되기 전에 호출되었지만, **호이스팅** 덕분에 오류 없이 실행됩니다.

2. 함수 표현식 (Function Expression)

- **형태:** 함수를 변수에 할당하는 방식으로 정의합니다. 함수는 이름이 없을 수도 있으며, 이를 **익명 함수**라고 합니다.
- **호이스팅:** 함수 표현식은 호이스팅이 되지 않으며, 함수가 정의된 이후에만 호출할 수 있습니다. 즉, 선언 전에 함수를 호출하면 오류가 발생합니다.

예시:

```
hello(); // 오류 발생: Uncaught ReferenceError: Cannot access 'hello' before initialization

const hello = function() {
  console.log("Hello, World!");
};
```

이 경우에는 함수가 정의되기 전에 호출했기 때문에 **ReferenceError**가 발생합니다.

호이스팅 (Hoisting)

호이스팅은 자바스크립트에서 변수와 함수 선언이 코드 실행 전에 **위로 끌어올려지는 동작**을 말합니다. 함수 선언식은 호이스팅되지만, 함수 표현식은 호이스팅되지 않기 때문에, 함수 선언식은 정의된 위치에 상관없이 어디서든 호출할 수 있지만, 함수 표현식은 선언 이후에만 호출할 수 있습니다.

함수 선언식과 함수 표현식의 차이점 정리

- 함수 선언식은 코드 어디에서든 호출 가능 (호이스팅 적용).
- 함수 표현식은 선언 이후에만 호출 가능 (호이스팅 적용되지 않음).

권장 사항

- * `const` *나 ** `let` *을 사용하여 함수 표현식을 정의하는 것이 좋습니다. `var` 는 호이스팅 때문에 혼란을 초래할 수 있으므로 권장되지 않습니다.

핵심 개념 정리

1. 자바스크립트는 위에서 아래로 순차적으로 실행되는 언어입니다.
2. **호이스팅**은 함수 선언식에 적용되어 함수가 선언되기 전에 호출할 수 있게 만듭니다.
3. 함수 표현식은 호이스팅되지 않으므로, 선언 후에만 호출할 수 있습니다.
4. 일반적으로 코드를 깔끔하고 예측 가능하게 작성하기 위해 ** `const` *로 함수 표현식을 작성하는 것이 권장됩니다.