

# Ch20 strict mode

## 20.1 strict mode란?

**암묵적 전역**은 변수 선언이 존재하지 않을 때 자바스크립트 엔진이 암묵적으로 전역 객체에 **프로퍼티**를 동적 생성해서 `ReferenceError`를 발생시키지 않고 **프로퍼티**를 전역 변수처럼 사용하는 현상을 말한다.

**! 암묵적 전역**은 의도와 상관없는 오류를 발생시킬 수 있기 때문에 반드시 `var`, `let`, `const` 키워드를 사용하여 변수를 선언한 다음 사용해야 한다.

ES5부터 추가된 **strict mode**는 자바스크립트 언어의 문법을 더 엄격히 적용하여 문제를 일으킬 수 있는 코드에 대해 명시적인 에러를 발생시킨다. (**ESLint**는 strict mode와 유사한 효과를 주지만 더 잠재적인 오류까지 찾아내준다.)

## 20.2 strict mode의 적용

전역의 선두 또는 함수 몸체의 선두에 `'use strict';`를 추가한다.

```
'use strict';

function foo() {
  x = 10; // Uncaught ReferenceError: x is not defined
}

foo()
```

## 20.3 전역에 strict mode를 적용하는 것은 피하자

외부 서드파티 라이브러리가 non-strict mode를 사용할 수도 있기 때문에 전역에 strict mode를 적용하면 오류를 발생시킬 수 있다.

## 20.4 함수 단위로 strict mode를 적용하는 것도 피하자

모든 함수에 일일이 **strict mode**를 적용하는 것은 번거롭고 놓칠 시에 문제가 발생할 수 있다.

→ strict mode는 **즉시 실행 함수**로 감싼 스크립트 단위로 적용하는 게 바람직하다.

## 20.5 strict mode가 발생시키는 에러

**strict mode**를 적용했을 때,

- 선언하지 않은 변수 참조 시 `ReferenceError` 발생
- `delete` 연산자로 변수, 함수, 매개변수 삭제 시 `SyntaxError` 발생
- 중복된 이름의 매개변수 사용 시 `SyntaxError` 발생
- `with` 문 사용 시 `SyntaxError` 발생
  - `with` 문은 전달된 객체를 스코프 체인에 추가한다
  - `with` 문은 코드를 간단하게 할 수 있지만 성능과 가독성이 나빠지는 문제가 있으므로 사용하지 않는 것이 좋다

## 20.6 strict mode 적용에 의한 변화

- strict mode에서 함수를 일반 함수로 호출하면 `this`에 `undefined`가 바인딩 된다. (생성자 함수가 아닌 일반 함수 내부에서는 `this`를 사용할 필요가 없기 때문)
- strict mode에서 매개변수에 전달된 인수를 재할당해도 `arguments` 객체에 반영되지 않는다.