



Chap20 Strict mode

[20-1 strict mode란?](#)

[20-2,3,4 strict mode의 적용](#)

[20-5 strict mode가 발생하는 에러](#)

[20-6 strict mode 적용에 의한 변화](#)

20-1 strict mode란?

▼ strict mode

자바스크립트의 문법을 좀 더 엄격하게 적용하여 오류를 발생시킬 가능성을 높이거나 자바스크립트 엔진의 최적화 작업에 문제를 일으킬 수 있는 코드에 대해 명시적인 에러를 발생시킨다.

ES6에서 도입된 클래스와 모듈은 기본적으로 `strict mode`가 적용된다.

▼ 암묵적 전역

변수로 선언된 것이 아니라, 전역 객체의 프로퍼티가 된 것

▼ ESLint

strict mode와 유사한 효과를 얻을 수 있다.

린트도구는 정적분석(static analysis)기능을 통해 소스코드를 실행하기 전에 소스코드를 스캔하여 문법적 오류만 아니라 잠재적 오류까지 찾아내고 오류의 원인을 리포팅해주는 유용한 도구다.

린트도구는 `strict mode`가 제한하는 오류는 물론 코딩 컨벤션을 설정 파일 형태로 정의하고 강제할 수 있기 때문에 더욱 강력한 효과를 얻을 수 있다.

20-2,3,4 strict mode의 적용

▼ 적용방법

- strict mode를 적용하려면 전역선두 또는 함수몸체의 선두에 `use strict`를 추가한다.

전역의 선두에 추가하면 스크립트 전체에 strict mode가 적용된다.

함수 몸체의 선두에 추가하면 해당 함수와 중첩함수에 strict mode가 적용된다.

코드의 선에 구문을 위치시키지 않으면 strict mode가 제대로 동작하지 않는다.

▼ 주의사항

1. 전역에 strict mode 설정은 피하자

Why?

`strict mode` 스크립트와 `non-strict mode` 스크립트를 혼용하는 것은 오류를 발생시킬 수 있다.

특히 외부 서드파티 라이브러리를 사용하는 경우 라이브러리가 non-strict mode인 경우도 있기 때문에 전역에 `strict mode`를 적용하는 것은 바람직하지 않다.

2. 함수 단위로 strict mode 설정도 피하지자

Why?

함수 단위로 `strict mode`를 적용하게 되면 모든 함수마다 일일이 strict mode를 적용해야 하는데 이는 매우 번거로운 일이다.

그리고 strict mode가 적용된 함수가 참조할 함수 외부의 컨텍스트에 strict mode를 적용하지 않는다면 문제가 발생할 수 있다.

```
(function () {  
  // non-strict mode  
  var let = 10; // 에러가 발생하지 않는다.  
  
  function foo() {  
    'use strict';  
  
    let = 20; // SyntaxError: Unexpected strict mode reserved word  
  }  
  foo();  
})();
```

▼ 해결방법

즉시 실행 함수 이용

즉시 실행 함수란?

→ 정의되자마자 즉시 실행되는 함수를 말한다.

```
// 즉시 실행 함수의 선두에 strict mode 적용
(function () {
    'use strict';

    // Do something...
})();
```

20-5 strict mode가 발생하는 예러

▼ 암묵적 전역

선언하지 않은 변수를 참조하면 ReferenceError가 발생한다.

▼ 변수, 함수, 매개변수의 삭제

delete 연산자로 변수, 함수, 매개변수를 삭제하면 SyntaxError를 발생시킨다.

▼ 매개변수 이름의 중복

중복된 매개변수 이름을 사용하면 SyntaxError를 발생시킨다.

▼ with문의 사용

with문을 사용하면 SyntaxError를 발생시킨다.

```
(function () {
    'use strict';

    // SyntaxError: Strict mode code may not include a with
    with({ x: 1 }) {
        console.log(x);
    }
})();
```

▼ with문

반복된 동일한 객체의 요소들에 접근할때 반복되는 부모 객체를 적는것을
축약 하기 위해서 사용한다.

```
with(반복될 객체){
    뒤에 오게될 메서드 또는 객체요소
}
```

단, with문 사용을 지양한다. 왜냐하면 동일한 객체의 프로퍼티를 반복해서 사용할 때 이름을 생략할 수 있어서 코드가 간단해지는 효과는 있지만 성능과 가독성이 나빠지는 문제가 있다.

20-6 strict mode 적용에 의한 변화

▼ 일반 함수의 this

strict mode에서 함수를 일반 함수로서 호출하면 this에 undefined가 바인딩된다. 생성자 함수가 아닌 일반 함수 내부에서는 this를 사용할 필요가 없기 때문이다. 이는 예러는 발생하지 않는다.

```
(function () {
    'use strict';

    function foo() {
        console.log(this); // undefined
    }
    foo();

    function Foo() {
        console.log(this); // Foo
    }
    new Foo();
})();
```

▼ arguments 객체

strict mode에서는 매개변수에 전달된 인수는 재할당하여 변경해도 arguments 객체에 반영되지 않는다.

```
(function (a) {
    'use strict';
    // 매개변수에 전달된 인수를 재할당하여 변경
```

```
a = 2;

// 변경된 인수가 arguments 객체에 반영되지 않는다.
console.log(arguments); // { 0: 1, length: 1 }
}(1));
```

▼ react도 strictmode가 실행되서 2번씩 실행되는 이유?