

삼육대학교 SW융합교육원 김진호

자바스크립트 기초교육

09. strict mode (엄격 모드)

9-1. strict mode

9-2. example

❖ 9. strict mode

9-1. strict mode (엄격 모드)

9-1-1. strict mode

```
function test() {  
    // 암묵적으로 전역 변수가 된다.  
    x = 10;  
}  
  
test();  
  
console.log(x); // 10
```

- 개발자의 의도와 상관 없이 발생한 암묵적 전역은 오류를 발생시키는 원인이 될 수 있으므로 반드시 var, let, const 키워드를 사용하여 변수를 선언한 다음 사용해야 한다.
- 이와 같은 잠재적 오류 발생을 근본적으로 막기 위해 ES5부터 strict mode(엄격 모드)가 추가되었다.
- ES6에서 도입된 클래스와 모듈은 기본적으로 strict mode가 적용된다.

❖ 9. strict mode

9-1. strict mode (엄격 모드)

9-1-2. strict mode apply (엄격 모드 적용)

```
// 전역의 선두에 추가하면 스크립트 전체에 strict mode가 적용된다.  
// 'use strict';  
  
function test() {  
    // 함수 몸체의 선두에 추가하면 해당 함수와 중첩 함수에 strict mode가 적용된다.  
    // 'use strict';  
  
    x = 10; // ReferenceError: x is not defined  
  
    // 코드의 선두에 위치시키지 않으면 strict mode가 제대로 동작하지 않는다.  
    // 'use strict';  
}  
  
test();
```

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

❖ 9. strict mode

9-1. strict mode (엄격 모드)

9-1-2. strict mode apply (엄격 모드 적용)

```
// 서드파티 라이브러리가 non-strict mode인 경우 즉시 실행 함수로  
스크립트 전체를 감싸서 스코프를 구분하고 즉시 실행 함수의 선두에  
strict mode를 적용한다.
```

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

- 함수 단위로 strict mode를 적용하면 strict mode와 non-strict mode의 혼용이 되어 바람직하지 않고 모든 함수에 일일이 적용하는 것도 번거롭다.
- 즉시 실행 함수로 감싼 스크립트 단위로 적용하는 것이 바람직하다.

❖ 9. strict mode

9-2. example

9-2-1. error case (에러 상황)

암묵적 전역

```
(function(){  
    'use strict';  
  
    // 선언하지 않은 변수를 참조하면 에러가 발생한다.  
    x = 1;           // ReferenceError: x is not defined  
    console.log(x);  
})();
```

❖ 9. strict mode

9-2. example

9-2-1. error case (에러 상황)

변수, 함수, 매개변수의 삭제

```
(function(){  
    'use strict';  
  
    var x = 1;  
    // delete 연산자로 변수, 함수, 매개변수를 삭제하면 문법 에러가 발생한다.  
    delete x; // strict 모드에서는 식별자에 대해 'delete'를 호출할 수 없습니다.  
})();
```

❖ 9. strict mode

9-2. example

9-2-1. error case (에러 상황)

매개변수 이름의 중복

```
(function(){  
    'use strict';  
  
    function test(x, x) { // SyntaxError: Duplicate parameter  
                           name not allowed in this context  
        return x + x;  
    }  
  
    console.log(test(1, 2));  
})();
```


❖ 9. strict mode

9-2. example

9-2-1. error case (에러 상황)

with 문의 사용

```
(function(){  
    'use strict';  
  
    // with문은 전달된 객체를 스코프 체인에 추가한다.  
    // 객체 이름을 생략할 수 있어 코드가 간단해지는 효과가 있지만 성능과 가독성이  
    나빠지는 문제가 있어 사용하지 않는 것이 좋다.  
    with({ x : 1 }) {    // 'with' 문은 strict 모드에서 사용할 수 없습니다.  
        console.log(x);  
    }  
})();
```

❖ 9. strict mode

9-2. example

9-2-2. change case (변화 상황)

일반 함수의 this

```
(function(){  
    'use strict';  
  
    function test() {  
        console.log(this);  
    }  
  
    test();           // undefined  
  
    new test();       // test {}  
})();
```

- 생성자 함수가 아닌 일반 함수 내부에서는 this를 사용할 필요가 없기 때문에 strict mode에서는 함수를 일반 함수로서 호출하면 this에 undefined가 바인딩 된다.

❖ 9. strict mode

9-2. example

9-2-2. change case (변화 상황)

arguments 객체

```
(function(x){  
    'use strict';  
  
    x = 2;  
  
    console.log(arguments);  
})(1);
```

- strict mode에서는 매개변수에 전달된 인수를 재할당하여 변경해도 arguments 객체에 반영되지 않는다.