

20장 strict mode

strict mode란?

자바스크립트는 오류를 어느정도 무시하고 넘어감

-> 예상치 못한 오류가 발생할 수 있음

-> 문법을 엄격히 적용하여 오류를 발생시킬 가능성이 높거나
자바스크립트 엔진의 최적화 작업에 문제를 일으킬 수 있는 코드에
명시적인 에러 발생

```
> function foo() {  
    x = 10;  
}
```

```
< undefined
```

```
> foo();  
    console.log(x);
```

```
10
```

```
> 'use strict';  
function foo() {  
    x = 10;  
}  
foo();  
console.log(x);
```

```
✖ ▶ Uncaught ReferenceError: x is not defined  
    at foo (<anonymous>:3:7)  
    at <anonymous>:5:1
```

strict mode의 적용

strict mode는 전역이나 함수에 적용할 수 있다.

적용하려는 스코프의 선두에 'use strict';를 추가한다.

```
> 'use strict';
```

```
function foo() {  
  x = 10;  
}
```

```
y = 5;  
foo();
```

✖ ▶ Uncaught ReferenceError: y is not defined
at <anonymous>:7:3

```
> function foo() {  
  'use strict';  
  x = 10;  
}
```

```
y = 5;  
foo();
```

✖ ▶ Uncaught ReferenceError: x is not defined
at foo (<anonymous>:3:5)
at <anonymous>:7:1

전역의 선두에 추가

```
> function foo() {  
  x = 10;  
  'use strict';  
  z = 8;  
}
```

```
y = 5;  
foo();  
console.log(x, y, z);
```

```
10 5 8
```

함수의 선두에 추가

선두에 위치시키지
않으면 동작을 안함

strict mode의 적용

strict mode는 즉시 실행 함수로 감싼 스크립트 단위로 적용하는 것이 바람직하다

- 전역에 사용시 스크립트 단위로 적용됨
 - strict mode 스크립트와 non-strict mode 스크립트를 혼용하면 오류가 날 수 있음
 - 외부 라이브러리를 사용하는 경우 라이브러리가 non-strict mode일 수 있음

```
<script src="./dist/test.js"></script>
<script>
  (function() {
    'use strict';
    x = 'strict mode';
    console.log(x);
  })();
</script>
<script>
  y = 'non-strict mode';
  console.log(y);
</script>
```

external non-strict mode

✖ ▶ Uncaught ReferenceError: x is not defined
at [index.html:13](#)
at [index.html:15](#)

⚠ DevTools failed to load source map: Could not load source map for [chrome-extension://jblndlipeogpafnldhgmapcemaps/contentscript.js.map](#): HTTP error: status not supported: ERR_UNKNOWN_URL_SCHEME

non-strict mode

strict mode의 적용

strict mode는 즉시 실행 함수로 감싼 스크립트 단위로 적용하는 것이 바람직하다

- 함수 단위로 사용 시
 - 함수 별로 strict mode가 다른 것은 바람직하지 않음
 - 일일이 strict mode를 적용하는 것은 번거로움

```
> let = 5;

function foo() {
  'use strict';
  x = 10;
}

console.log(let);
foo();

5
```

✖ ▶ Uncaught ReferenceError: x is not defined
at foo (<anonymous>:5:5)
at <anonymous>:9:1

```
> (function(){
  'use strict';
  let = 5;

  function foo() {
    x = 10;
  }

  console.log(let);
  foo();
})();
```

✖ Uncaught SyntaxError: Unexpected strict mode reserved word

strict mode가 발생시키는 에러

1. 암묵적 전역

- 선언하지 않은 변수를 참조하면 ReferenceError

```
> (function(){  
  'use strict';  
  
  x = 10;  
  console.log(x);  
})();
```

```
✖ ▶ Uncaught ReferenceError: x is not defined  
   at <anonymous>:4:5  
   at <anonymous>:6:2
```

```
> (function(){  
  x = 10;  
  console.log(x);  
})();
```

```
10
```

2. 변수, 함수, 매개변수의 삭제

- delete 연산자로 변수, 함수, 매개변수를 삭제하면 SyntaxError (delete는 객체의 property를 삭제하는 연산자)

```
> (function(){  
  'use strict';  
  
  var x = 10;  
  delete x;  
  
  function foo(a) {  
    delete a;  
  }  
  delete foo;  
})();
```

```
✖ Uncaught SyntaxError: Delete of an unqualified  
  identifier in strict mode.
```

```
> (function(){  
  var x = 10;  
  delete x;  
  
  function foo(a) {  
    delete a;  
  }  
  delete foo;  
})();
```

```
<< undefined
```

```
>
```

strict mode가 발생시키는 에러

3. 매개변수 이름의 중복

- 중복된 매개변수 이름을 사용하면 SyntaxError

```
> (function(){  
  'use strict';  
  
  function foo(x, x) {  
    return x + x;  
  }  
  
  console.log(foo(1, 2));  
})();
```

✖ Uncaught SyntaxError: Duplicate parameter name not allowed in this context

```
> (function(){  
  
  function foo(x, x) {  
    return x + x;  
  }  
  
  console.log(foo(1, 2));  
})();
```

4

4. with 문의 사용

- with 문을 사용하면 SyntaxError
(with 문은 전달된 객체를 스코프 체인에 추가)

```
> (function(){  
  'use strict';  
  
  with({x : 1}){  
    console.log(x);  
  }  
})();
```

✖ Uncaught SyntaxError: Strict mode code may not include a with statement

```
> (function(){  
  
  with({x : 1}){  
    console.log(x);  
  }  
})();
```

1

strict mode 적용에 의한 변화

1. 일반 함수의 this

- this에 undefined가 바인딩 된다.
- non-strict mode에서는 전역 객체로 바인딩 됨

```
> (function(){  
  'use strict';  
  
  function foo() {  
    console.log(this);  
  }  
  foo();  
  
  function Foo() {  
    console.log(this);  
  }  
  new Foo();
```

```
}());
```

```
undefined
```

```
▶ Foo {}
```

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

VM2541:4

```
▶ Window {window: Window, self: Window, document: document, name: '', location: Location, ...}
```

```
▶ Foo {}
```

VM2541:9

strict mode 적용에 의한 변화

2. arguments 객체

- 매개변수에 전달된 인수를 재할당하여 변경해도 arguments 객체에 반영되지 않는다

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

VM2586:6

► Arguments [1, callee: (...), Symbol(Symbol.iterator): f]

```
> (function(){  
  
  function foo(x) {  
    x = 2;  
    console.log(arguments);  
  }  
  foo(1);  
  
})();
```

VM2581:5

► Arguments [2, callee: f, Symbol(Symbol.iterator): f]