

모던 자바스크립트로 작성하기

Item 58

타입스크립트 개발 환경은 모던 자바스크립트도 실행할 수 있으므로
모던 자바스크립트의 최신 기능들을 사용하는 것이 좋다

👍 장점

- 코드 품질 향상
- 타입 추론 향상

1. ECMAScript 모듈 사용하기

ES2015부터는 import와 export를 사용하는 ECMAScript 모듈이 표준이 되었다.

만약 마이그레이션 대상인 자바스크립트 코드가 단일 파일이거나 비표준 모듈 시스템을 사용 중이라면 ES 모듈로 전환하는 것이 좋다.

2. 프로토타입 대신 클래스 사용하기

과거에는 자바스크립트에서 프로토타입 기반의 객체 모델을 사용했지만

애매한 프로토타입 모델보다는 견고하게 설계된 클래스 기반 모델을 사용하는 것을 권장한다.

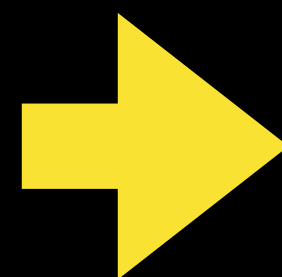
3. var 대신 let/const 사용하기

스코프 문제를 자세히 알지 못하더라도 var 대신 let과 const를 사용하면 스코프 문제를 피할 수 있다.

4. for(;;) 대신 for-of 또는 배열 메서드 사용하기

for-of 루프는 코드가 짧고 인덱스 변수를 사용하지도 않기 때문에 실수를 줄일 수 있다. (인덱스가 필요하면 forEach)

```
for(var i = 0; i < array.length; i++){  
  const el = array[i];  
  // ...  
}
```



```
for(const el of array) {  
  // ...  
}
```

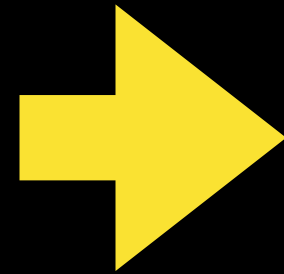
5. 함수 표현식보다 화살표 함수 사용하기

- 상위 스코프의 `this`를 유지 해준다. (`this` 바인딩에 대한 자세한 내용은 Item 49 참조)
- 일반 함수보다 화살표 함수가 더 직관적이며 코드도 간결해진다.

6. 단축 객체 표현과 구조 분해 할당 사용하기

단축 객체 표현

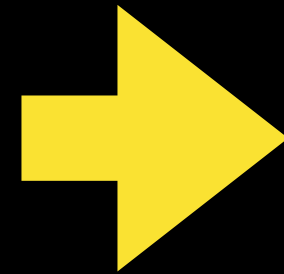
```
const x = 1, y = 2, z = 3;  
const pt = {  
  x: x,  
  y: y,  
  z: z  
};
```



```
const x = 1, y = 2, z = 3;  
const pt = { x, y, z }
```

객체 구조 분해 기본값 지정

```
let {a} = obj.props;  
if (a === undefined) a = 'default';
```



```
const {a = 'default'} = obj.props;
```

7. 저수준 프로미스나 콜백 대신 `async/await` 사용하기

콜백과 프로미스보다 `async`와 `await`가 권장되는 자세한 이유 [Item 25](#)참고

- 코드가 간결해져서 버그나 실수를 방지
- 비동기 코드에 타입 정보가 전달되어 타입 추론을 가능하게 함

8. 타입스크립트에 `use strict` 넣지 않기

타입스크립트에서 수행되는 안정성 검사가 엄격 모드보다 훨씬 더 엄격한 체크를 하기 때문에 타입스크립트 코드에서 `'use strict'`는 무의미함

타입스크립트 컴파일러가 생성하는 자바스크립트 코드에서 `'use strict'`가 추가됨

타입스크립트 코드에서는 `'use strict'`를 쓰지 않고, 대신 `alwaysStrict` 설정을 사용해야 한다.