

typescript에서 enum을 더 우아하게 사용하기

#TIL/dev/language/typescript

Object를 Enum으로 관리하기

1. enum에 object를 대입할 수 있으면 활용도가 매우 높아질 것 같지만 안타깝게도 불가능하다.
| TypeScript **supports numeric or string-based enums** only.
2. 다음은 enum으로 object를 관리하기 위한 다양한 trick들. 근데 막상 써보면 그다지 우아하지는 않다.

```
// 1
enum PizzaSize {
    small,
    medium,
    large
}

interface PizzaSizeSpec {
    key: number,
    value: number
}

function getPizzaSizeSpec(pizzaSize: PizzaSize): PizzaSizeSpec {
    switch (pizzaSize) {
        case PizzaSize.small:
            return {key:0, value: 25};
        case PizzaSize.medium:
            return {key:0, value: 35};
        case PizzaSize.large:
            return {key:0, value: 50};
    }
}

// 2
export class PizzaSize {
    static readonly SMALL = new PizzaSize('SMALL', 'A small pizza');
    static readonly MEDIUM = new PizzaSize('MEDIUM', 'A medium pizza');
    static readonly LARGE = new PizzaSize('LARGE', 'A large pizza');
```

```

// private to disallow creating other instances of this type
private constructor(private readonly key: string, public readonly value: any)
{
}

toString() {
    return this.key;
}
}

```

Enum with Method

namespace를 활용하여 static method가 있는 것처럼 보이게할 수 있다.

```

enum Mode {
    X,
    Y
}

namespace Mode {
    export function toString(mode: Mode): string {
        return Mode[mode];
    }

    export function parse(mode: string): Mode {
        return Mode[mode];
    }
}

const mode = Mode.X;

const str = Mode.toString(mode);
alert(str);

const m = Mode.parse(str);
alert(m);

```

TypeScript: Add functions to an Enum

하지만 lint는 이를 권장하지는 않는 것 같다. 실 프로젝트에서는 사용하지않았음.