

Type & Interface

객체의 타입의 이름을 지정하는 방법

Type

- 1) 새로운 속성을 추가 하기 위해서 다시 같은 이름으로 선언 할 수 없다.

```
type _Animal = {  
  weight: number;  
};  
  
type _Animal = {//error : 식별자가 중복됨  
  height: string;  
};
```

- 2) 확장하는 방법 -> & 로 확장함.

```
type AnimalType = {  
  species: string,  
  height: number,  
  weight: number  
}  
  
type AfricaAnimal = AnimalInterface & {  
  continent: string  
}
```

- 3) Computed value 사용 가능

```
type names = 'firstName' | 'lastName'  
  
type NameTypes = {  
  [key in names]: string  
}
```

Interface

1) 선언적 확장이 가능하다.

```
interface Animal {  
  weight: number;  
}  
  
interface Animal {  
  height: number;  
}  
  
const tiger: Animal = {  
  weight: 100,  
  height: 200,  
};  
console.log(tiger);  
  
type _Animal = {  
  weight: number;  
};  
  
type _Animal = {  
  //error : 식별자가 중복됨  
  height: string;  
};
```

2) 확장하는 방법 -> extends 로 확장함

```
interface AnimalInterface {  
  species: string  
  height: number  
  weight: number  
}  
  
interface AfricaAnimal extends AnimalInterface {  
  continent: string  
}
```

3) Computed value 사용 불가능

```
const yc: NameTypes = { firstName: 'hi', lastName: 'yc' }  
  
interface NameInterface {  
  // error  
  [key in names]: string  
}
```

Interface가 선언 확장이 가능하므로,

속성이 추가 될 것을 고려하면 interface 사용하는 것이 좋음.