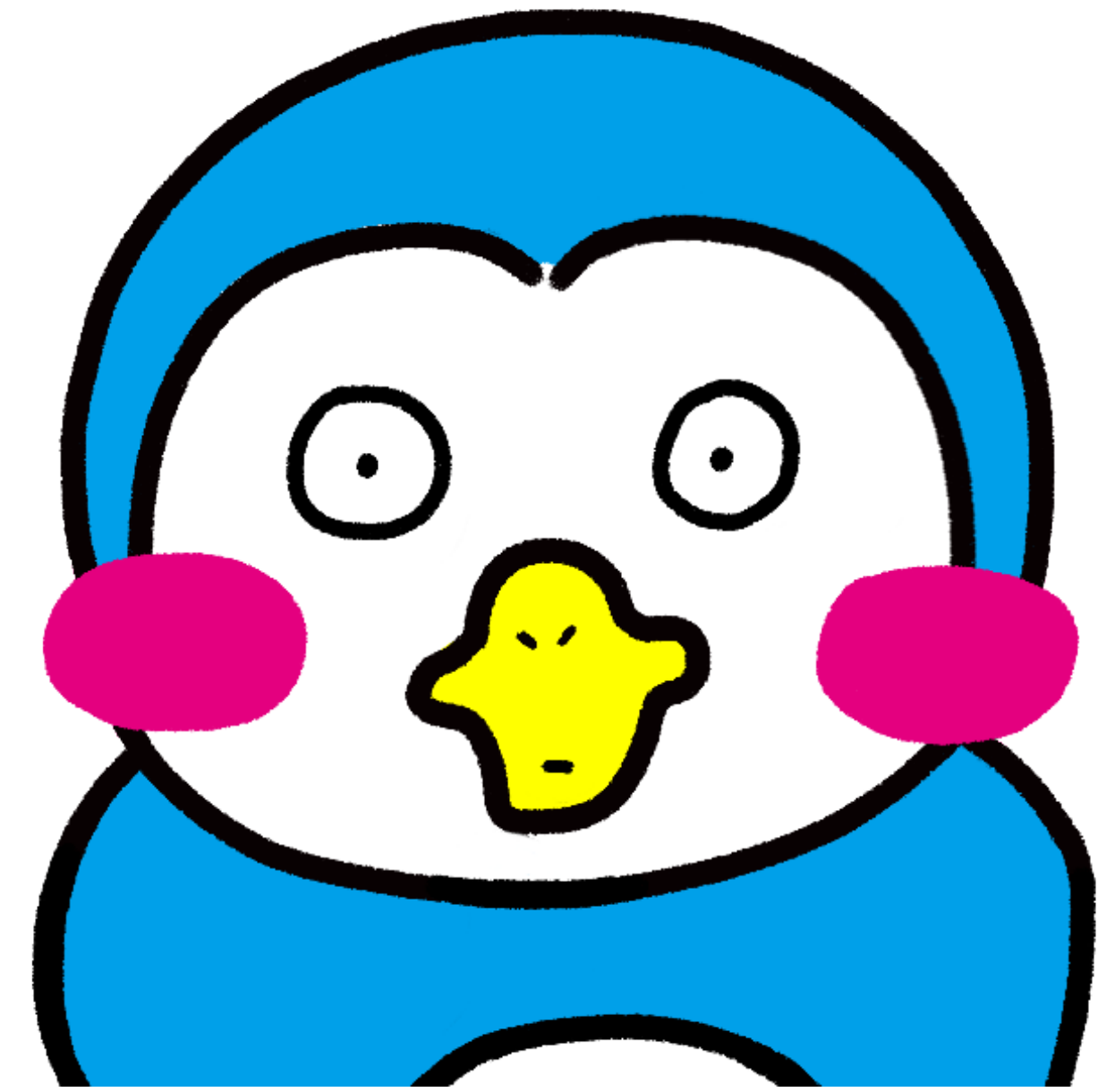


만들고 싶은
데이터 또 있어요



객체 리터럴로 객체 만들기

객체 리터럴

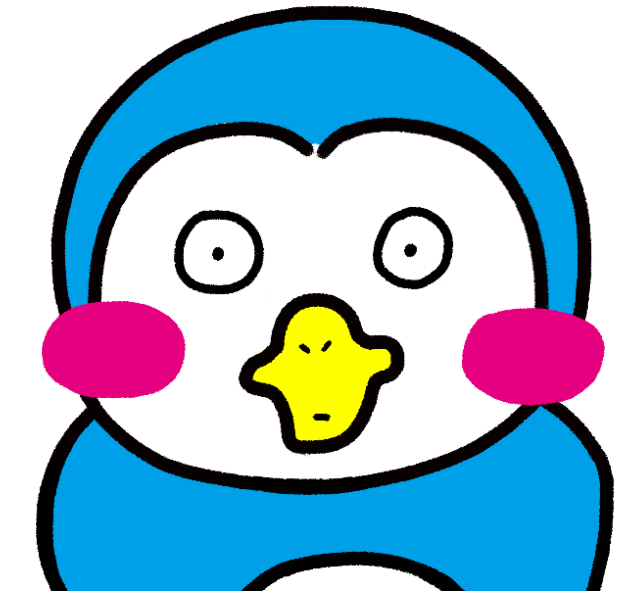


객체를 구성하는 속성(값이나 기능)의 이름과 실제 값을 한 쌍으로 묶어 표현하는 ‘객체 리터럴’을 이용하면 객체를 생성할 수 있다.

이때 속성의 이름을 키(key), 속성의 실제 값을 밸류(value)라 한다.

```
const cat = {  
  name: "국희",  
  town: "부천"  
}
```

=> name과 town은 키, “국희”와 “부천”은 밸류!



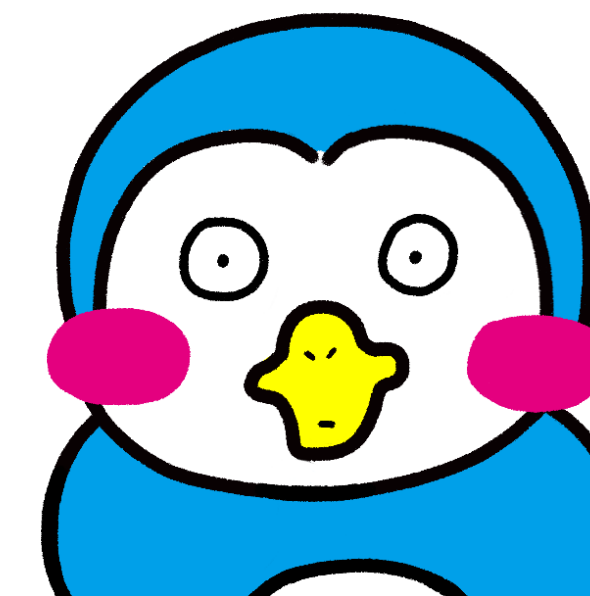
밸류가 함수면 메소드가 된다

속성의 밸류로 함수가 주어지면, 이 속성은 메소드가 된다.

```
const cat = {  
  name: "국희",  
  town: "부천",  
  eat: function(){  
    console.log("밥을 먹는다냥")  
  },  
  scratch: function(){  
    console.log("건드리면 할퀴다냥")  
  }  
}
```

=> 중요! 키는 중복될 수 없다. 그러나 밸류는 같은 것이 있어도 상관 없다.

속성 참조하기

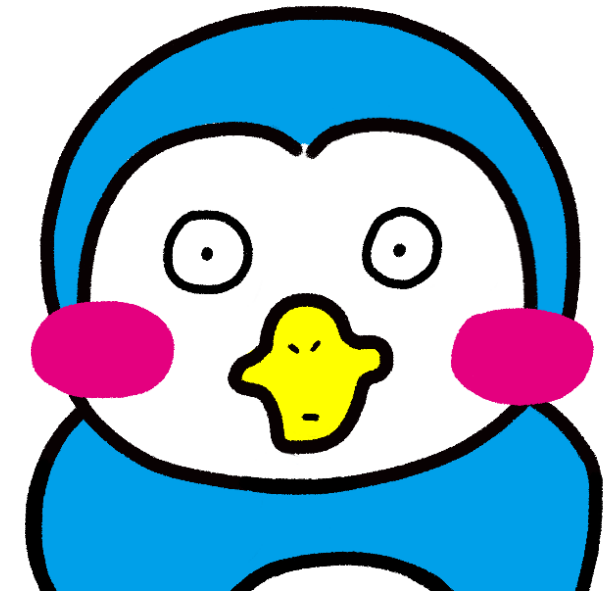


객체 안의 속성 값을 참조할 때는(읽거나 쓸 때는)
도트(.) 연산자 또는 대괄호([]) 연산자를 사용할 수 있다.

```
// 도트 연산자를 사용할 때  
console.log(cat.town) // 부천  
cat.scratch() // 건드리면 할퀴다냥
```

```
// 대괄호 연산자를 사용할 때  
console.log(cat['name']) // 국회  
cat['eat']() // 밥을 먹는다냥
```

내용 정리



- 객체 리터럴은 객체를 구성하는 속성의 이름과 실제 값을 한 쌍으로 묶는 표현 방식이며, 이를 이용하면 객체를 생성할 수 있다.
- 속성의 이름을 키(key), 속성의 실제 값을 밸류(value)라 한다.
- 속성의 밸류로 함수가 주어지면, 이 속성은 메소드가 된다.
- 도트 연산자나 대괄호 연산자로 객체의 속성을 참조할 수 있다.
- 키는 중복될 수 없다. 그러나 밸류는 중복될 수 있다.