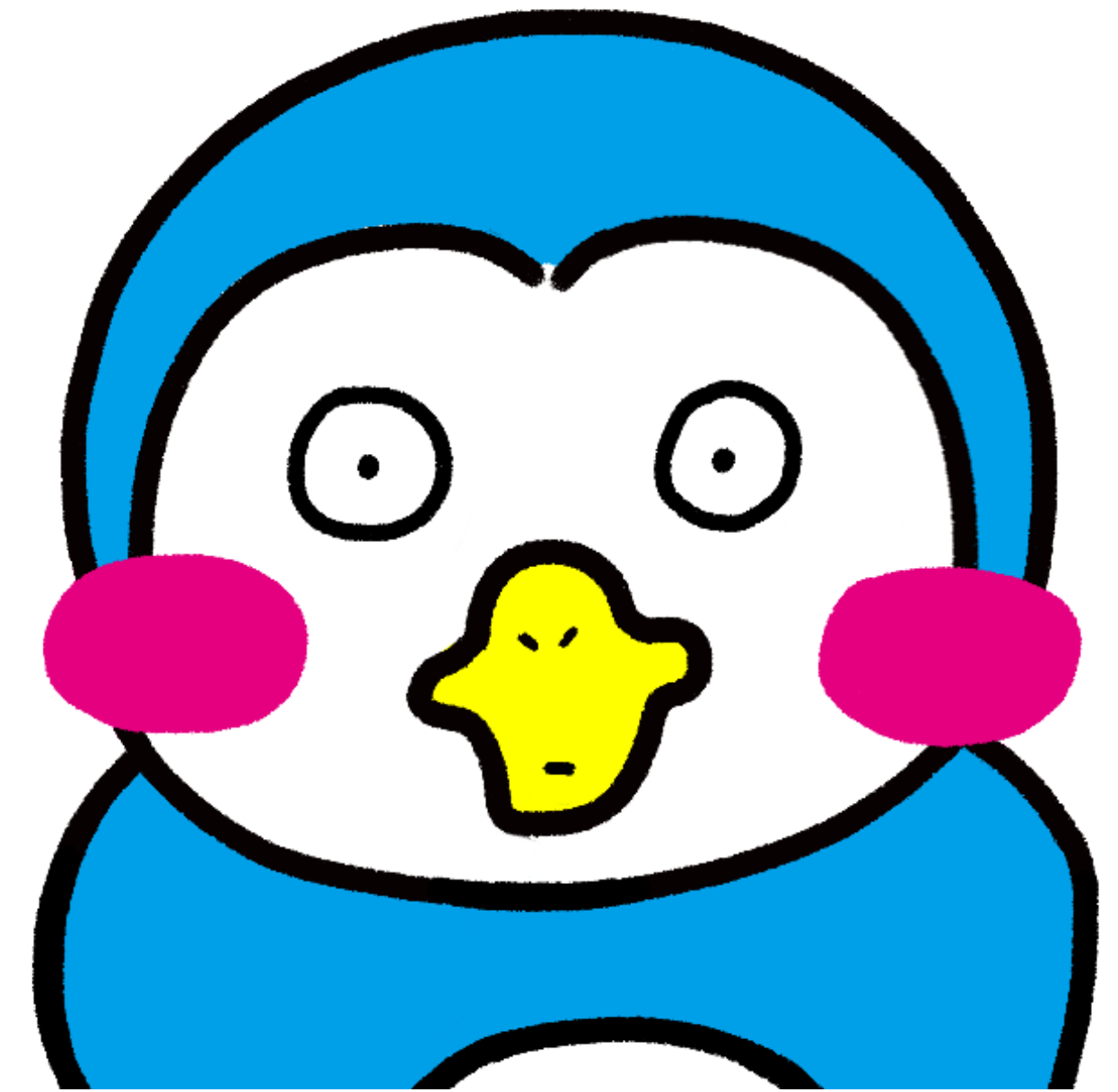


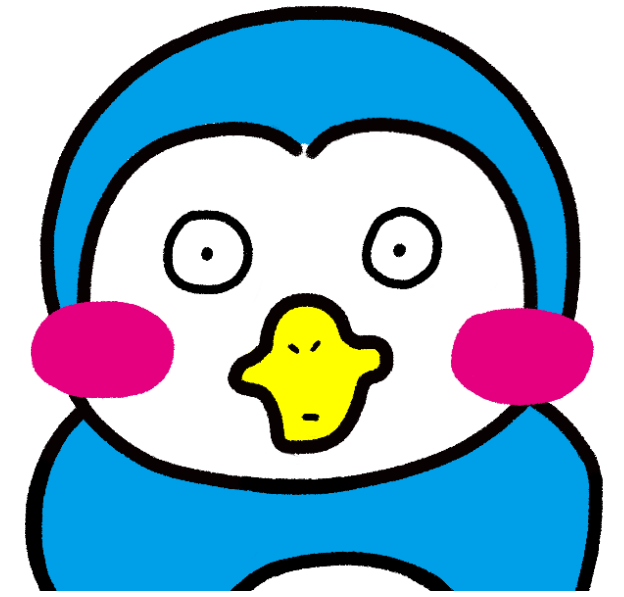
점이 세 개 모이면  
마법같은 일이



---

스프레드 연산자(전개 구문)

# 스프레드 연산자(전개 구문)



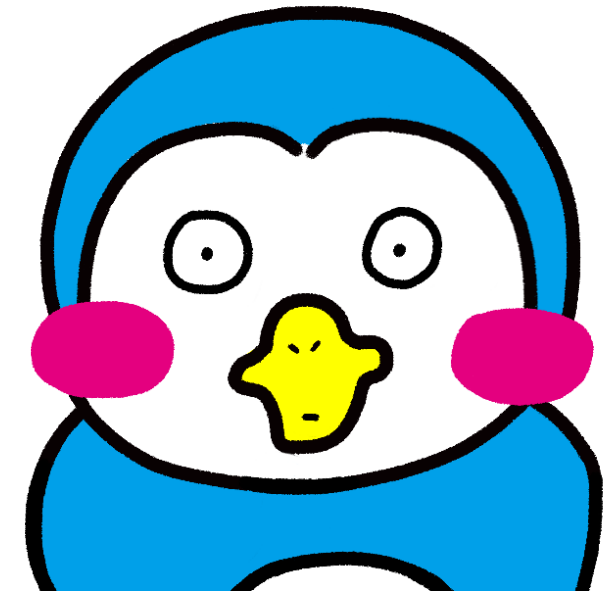
스프레드 연산자를 사용하면 배열이나 객체 리터럴 등의 요소 목록을 펼칠 수 있다. 전개 구문, 전개 연산자 등으로 불린다.

(펼치다 = spread)

스프레드 연산자 :

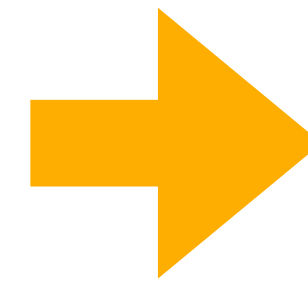
● ● ●

# 배열을 전개하다



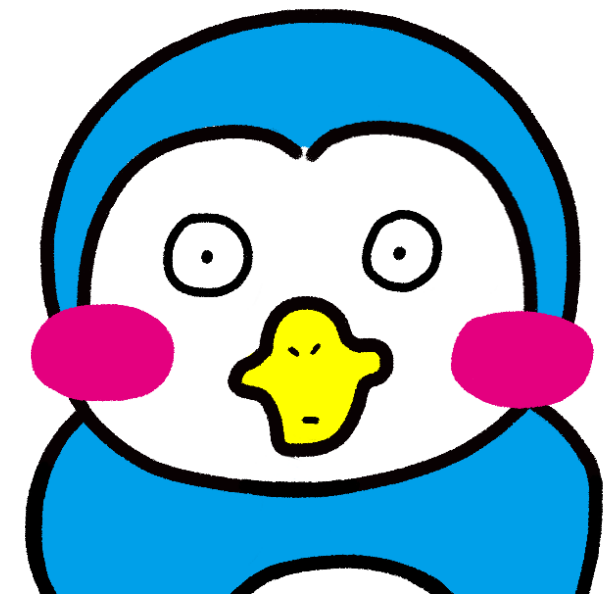
스프레드 연산자는 배열의요소를 복사할 때 효과적으로 사용할 수 있다.

```
const pet = ["강아지", "고양이"]  
console.log(...pet)  
console.log(pet)
```



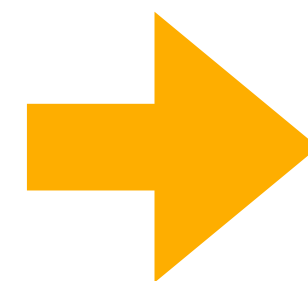
```
강아지 고양이  
[ '강아지', '고양이' ]
```

# 배열을 전개하다



새로운 배열을 선언할 때 스프레드 연산자를 사용하면, 기존 배열의 개별 요소를 멤버로 갖는 배열을 생성할 수 있다.

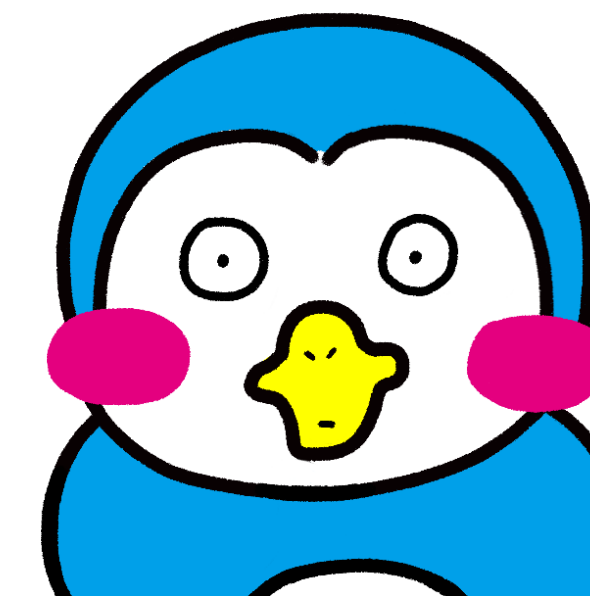
```
const pet1 = ["강아지", "고양이"]  
const pet2 = [...pet1, "햄스터"]  
console.log(pet2)  
  
const car1 = ["아반떼", "소나타"]  
const car2 = ["스파크", ...car1]  
console.log(car2)
```



```
[ '강아지', '고양이', '햄스터' ]  
[ '스파크', '아반떼', '소나타' ]
```

=> 스프레드 연산자는, 사용된 자리에 개별 요소를 '놓고 온다'

# 객체를 전개하다

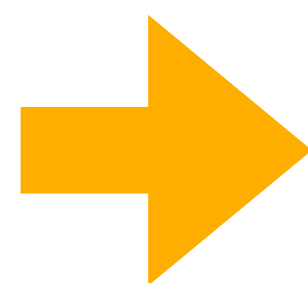


스프레드 연산자는 객체 리터럴에 대해서도 동일한 기능을 수행한다.  
단, 배열과 달리 개별 요소가 키와 밸류로 구성되어 있다.

```
const bird = {  
  parrot: "앵무새",  
  chicken: "닭"  
}
```

```
const animal = {  
  ...bird,  
  dog: "강아지"  
}
```

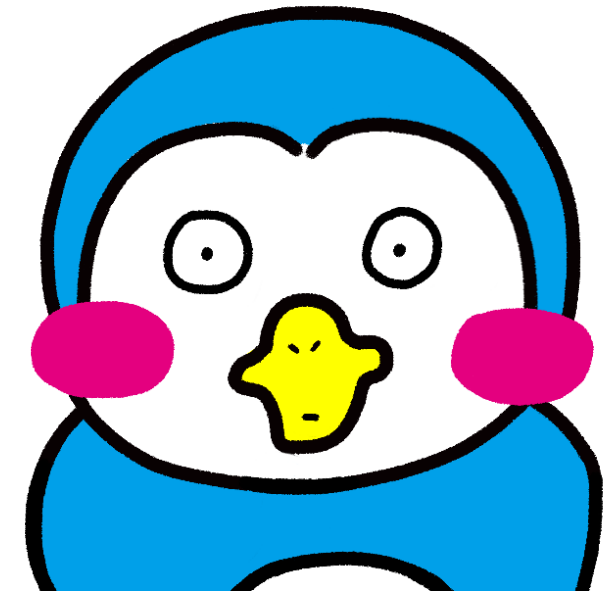
```
console.log(animal)
```



```
{  
  parrot: '앵무새',  
  chicken: '닭',  
  dog: '강아지'  
}
```

# 내용 정리

---



- 스프레드 연산자는 배열, 객체의 요소 목록을 펼친다(spread).
- 스프레드 연산의 결과, 배열과 객체의 개별 요소는 '복사'된다.
- 따라서 배열과 객체에는 별다른 영향을 끼치지 않는다.