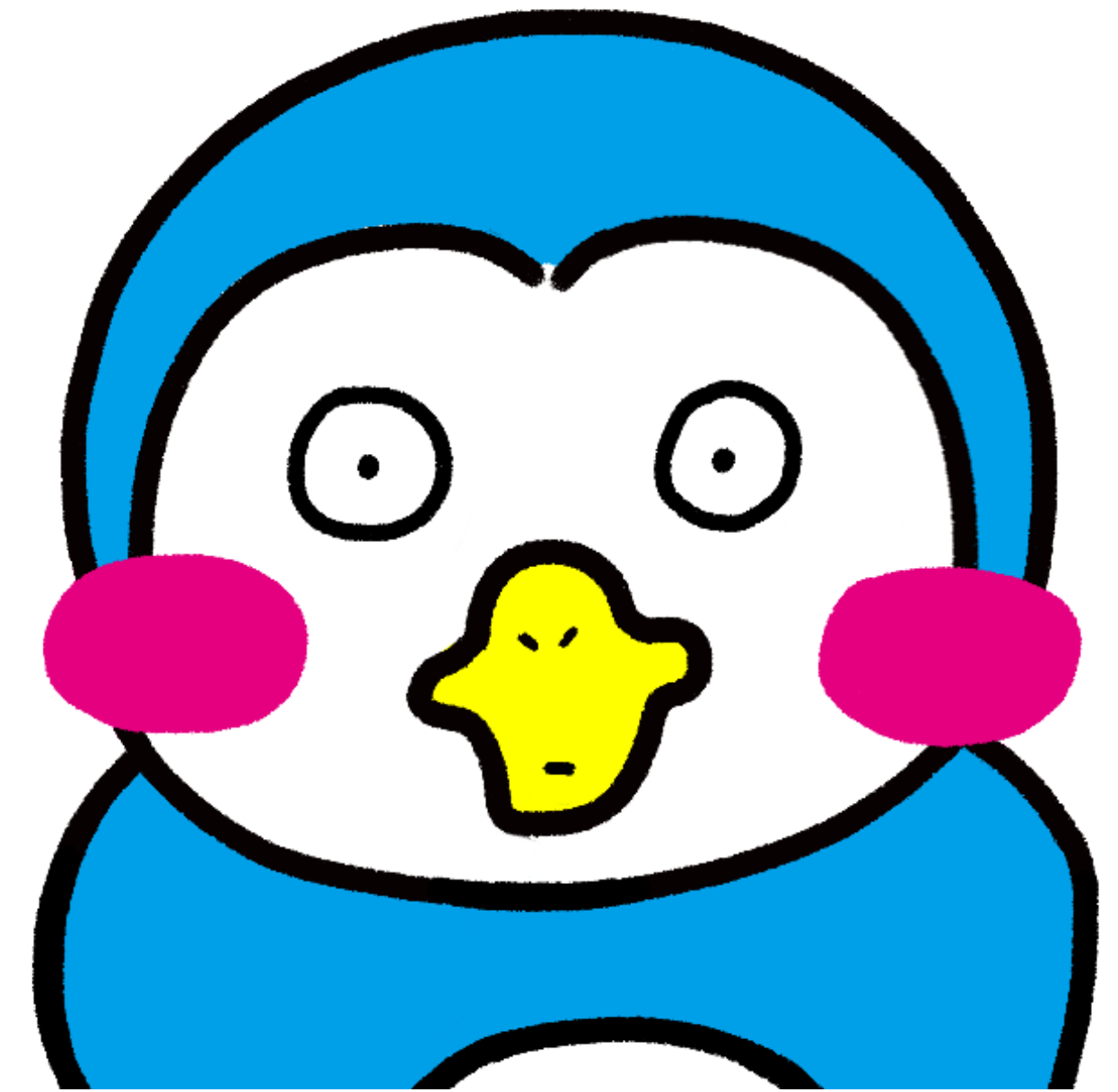
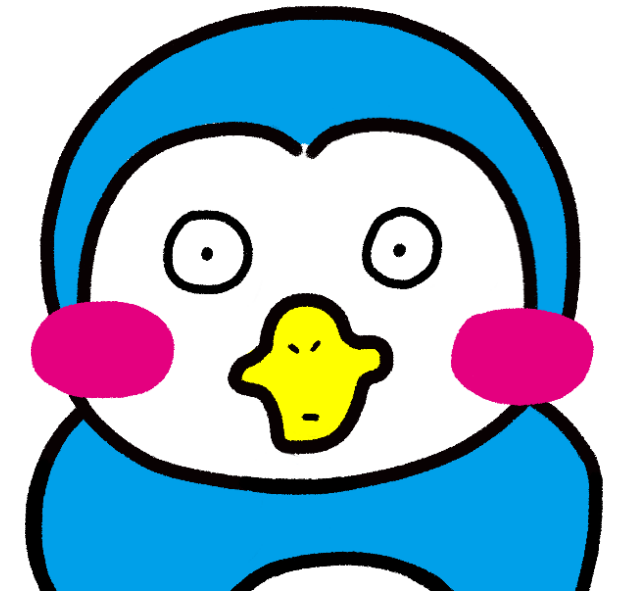


정해진 횟수만큼
반복해주세요



반복문을 만들어 보자 2편

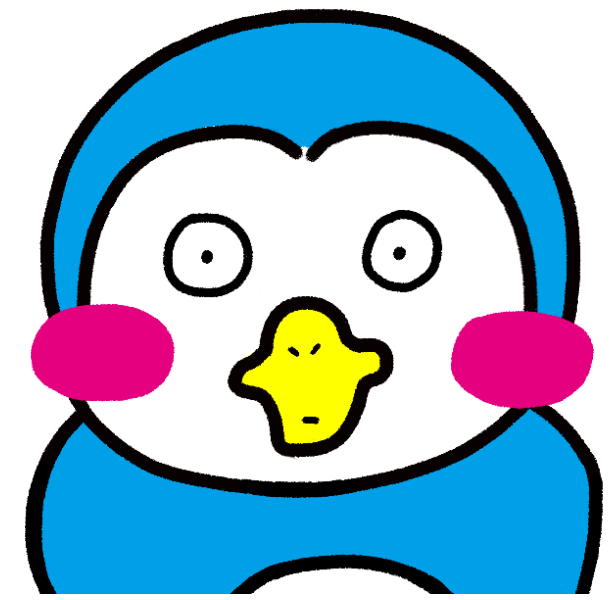
반복문



반복문은 비슷하거나 동일한 구문을 반복해서 수행할 수 있는 구문이다.
대표적인 반복문은 다음 두 가지이며, 두 반복문은 구조 및 동작방식에서 차이를 보인다.

- while문
- for문 (이번 시간에는 이거 함)

for문



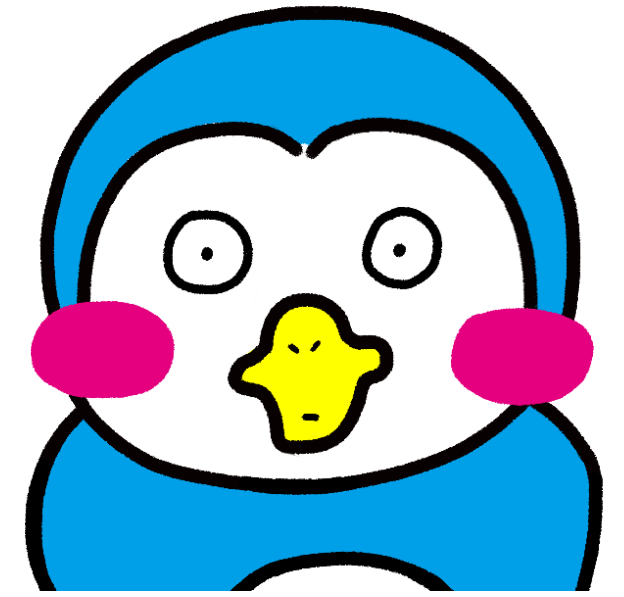
for문은 구문 작성시 반복을 위해 위해 필요한 세 가지 요소를 한 곳에 모아 작성함으로써 보다 명시적으로 반복 횟수를 표현할 수 있는 직관적인 구문이다.

```
for(초기식; 조건식; 반복식){  
    // 조건이 true인 경우 반복 수행할 코드  
}
```

=> 초기식: 반복 조건의 초기화 작업

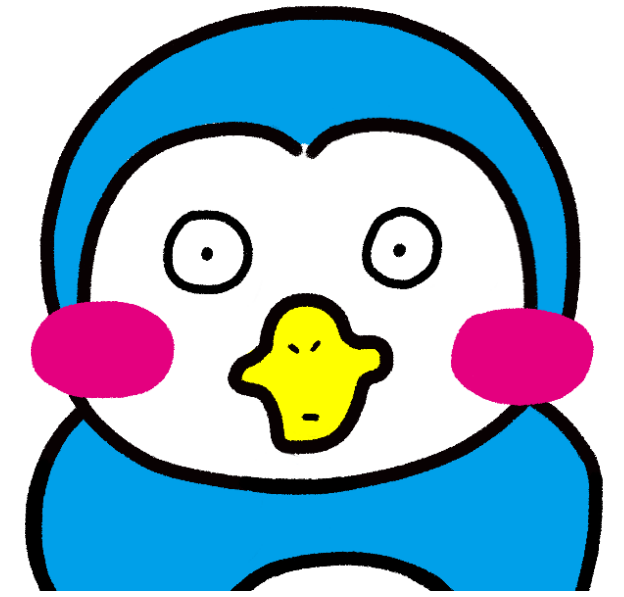
=> 반복식: 반복이 한 번 끝날 때마다 실행될 작업

어떻게 동작하나요?



```
for(let i = 1; i <= 3; i += 1){  
  console.log(i)  
}
```

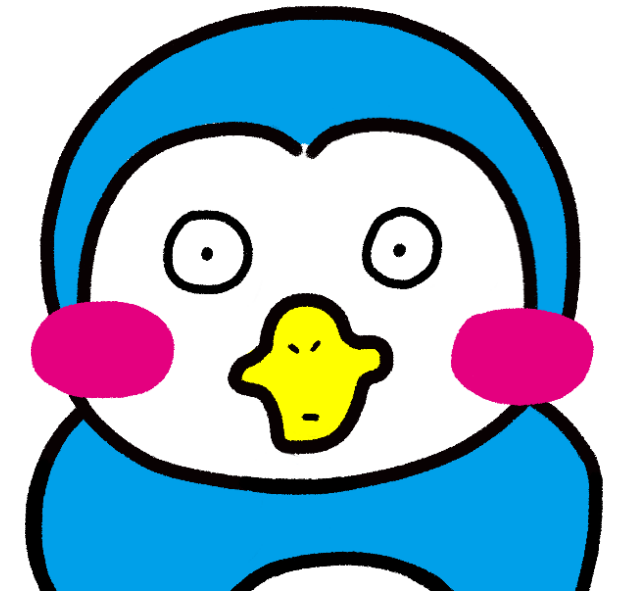
어떻게 동작하나요?



```
for(①let i = 1; i <= 3; i += 1){  
  console.log(i)  
}
```

for문이 시작되면, 가장 먼저 초기식부터!

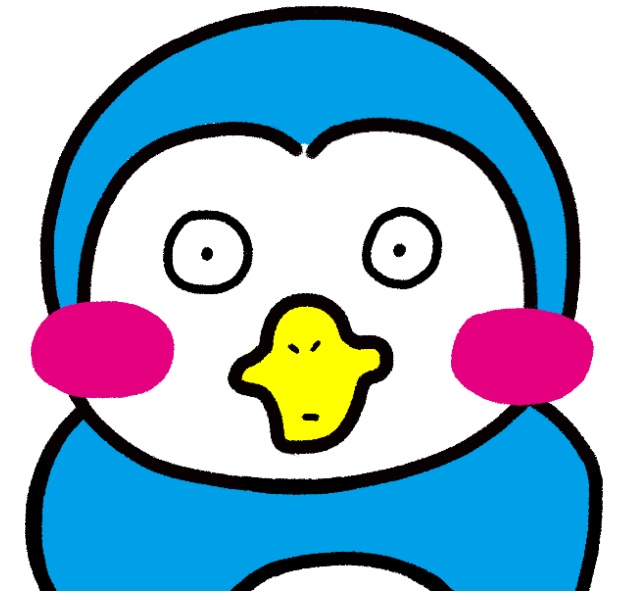
어떻게 동작하나요?



```
for(①let i = 1; ②i <= 3; i += 1){  
  console.log(i)  
}
```

초기식은 처음 한 번만. 초기식 완료 후에는 조건식 확인!
(초기식에서 i를 1로 초기화했으므로, 이 시점에 조건은 true)

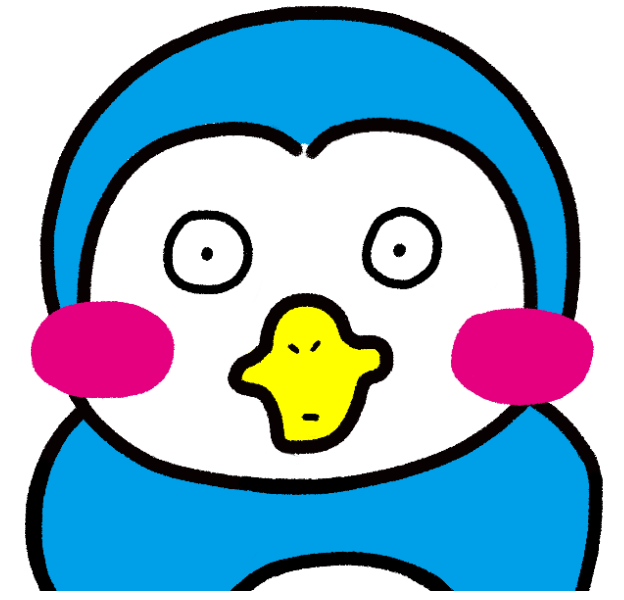
어떻게 동작하나요?



```
for(① let i = 1; ② i <= 3; i += 1){  
  ③ console.log(i)  
}
```

조건식의 결과가 참이면, 중괄호 안의 구문을 실행한다!

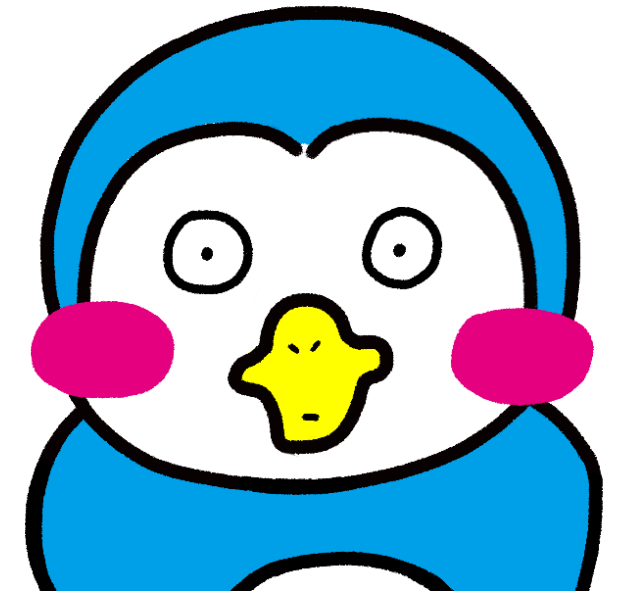
어떻게 동작하나요?



```
for(① let i = 1; ② i <= 3; ④ i += 1){  
  ③ console.log(i)  
}
```

중괄호 안의 구문이 한 번 끝날 때마다 반복식이 실행된다!
(이 시점에 i가 2로 변한다)

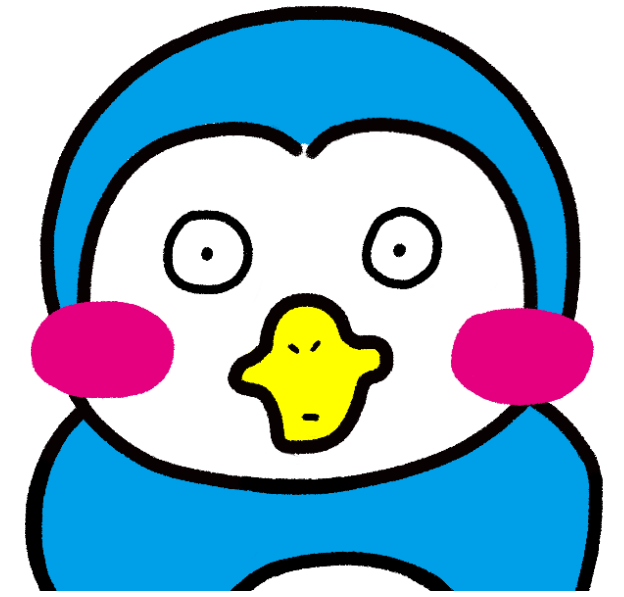
어떻게 동작하나요?



```
for(① let i = 1; ② i <= 3; ④ i += 1){  
  ③ console.log(i)  
}
```

다시 조건식으로 돌아와 반복을 할지 말지 결정한다!
(i가 2이므로 true, 구문은 다시 반복된다)

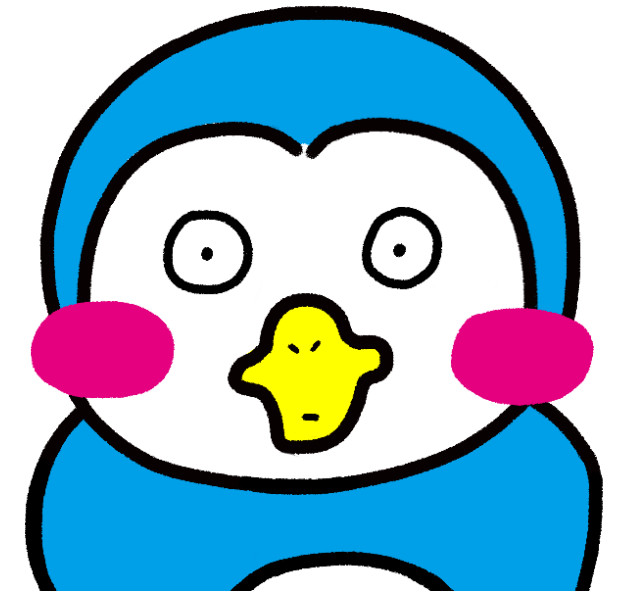
어떻게 동작하나요?



```
for( ①let i = 1; ②⑤i <= 3; ④i += 1){  
  ③⑥console.log(i)  
}
```

다시 조건식으로 돌아와 반복을 할지 말지 결정한다!
(i가 2이므로 true, 구문은 다시 반복된다)

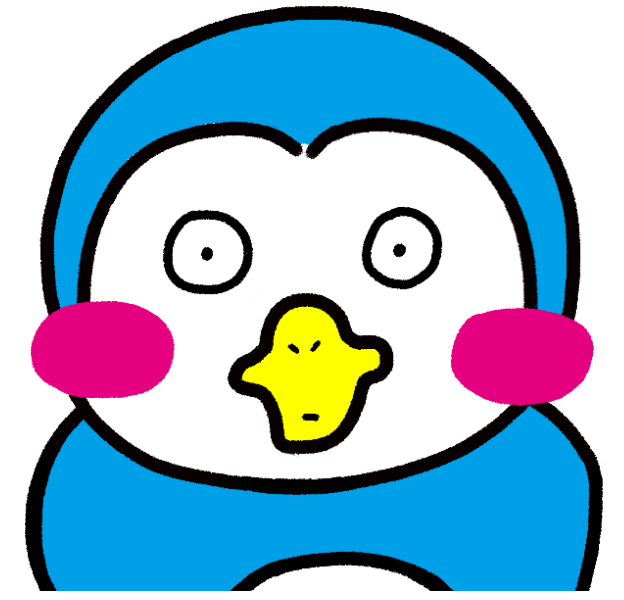
어떻게 동작하나요?



```
for( ①let i = 1; ②⑤i <= 3; ④⑦i += 1){  
  ③⑥console.log(i)  
}
```

다시 조건식으로 돌아와 반복을 할지 말지 결정한다!
(i가 2이므로 true, 구문은 다시 반복된다)

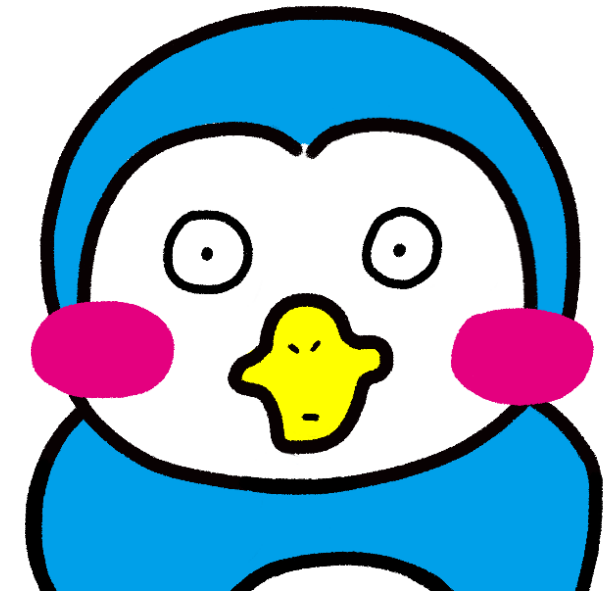
어떻게 동작하나요?



```
for(①let i = 1;②⑤⑧ i <= 3;④⑦ i += 1){  
  ③⑥console.log(i)  
}
```

이 과정은 조건식이 false를 반환할 때까지 반복된다!

내용 정리



- 반복문은 비슷하거나 동일한 구문을 반복해서 수행할 수 있는 구문이다.
- for문은 초기식, 조건식, 반복식으로 반복 횟수를 명시적으로 표현할 수 있다.
- for문의 초기식에서 let 키워드를 사용해 선언한 변수는 for문의 실행이 끝나면 사용할 수 없다.