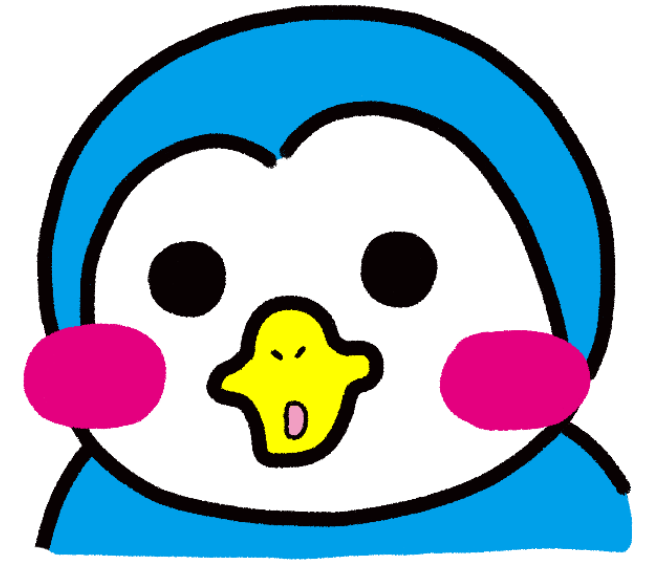


입문자를 위한 파이썬 기초



프로그램을 제어하는 for문

반복문

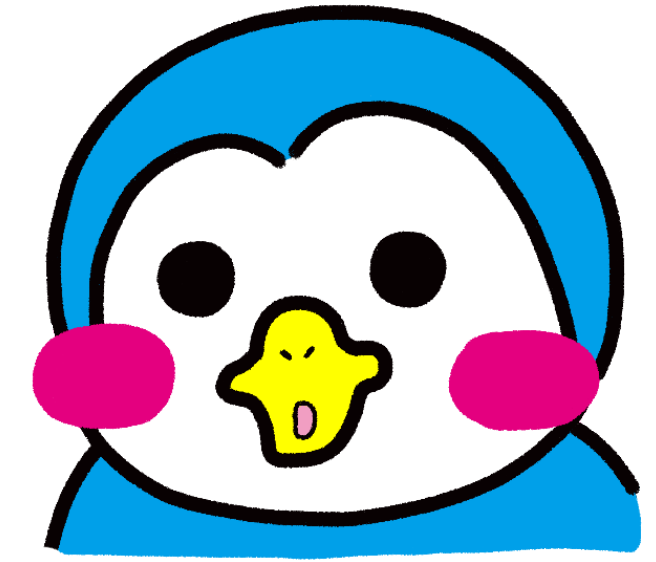


반복문은 비슷하거나 동일한 구문을 반복해서 수행할 수 있는 구문이다.
파이썬의 대표적인 반복문은 다음 두 가지이며, 두 반복문은 구조 및 동작방식에
서 차이를 보인다.

- for문
- while문

=> 반복문은 한 번 작성한 코드가 여러 번 실행되도록 해주는 구문이다!

for문

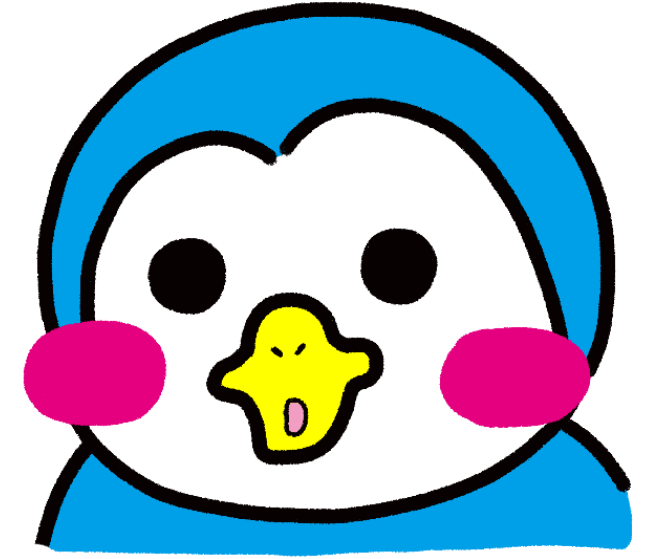


for문은 포함한 코드를 주어진 데이터모음의 멤버 수만큼 반복 실행하는 구문이다. for문의 데이터모음으로 문자열, 리스트, 튜플 등을 사용할 수 있는데, 이러한 자료형을 통틀어 'iterable 객체'라 부르기도 한다.

```
for 변수 in 데이터모음 :  
    # 반복해서 실행하고픈 코드
```

=> for문에 포함된 코드는 반드시 들여쓰기를 해주어야 합니다!

for 뒤에 있는 변수의 역할은?

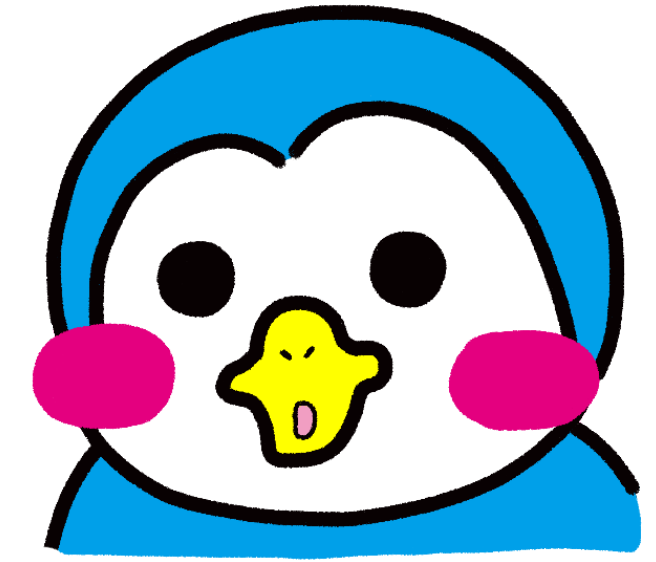


for문의 변수는 데이터모음 안의 멤버를 순서대로 기억하는 역할을 한다. 프로그래머는 이 변수를 통해 반복 코드 안에서 데이터모음의 멤버를 활용할 수 있다.

```
for 변수 in 데이터모음 :  
    # 반복해서 실행하고픈 코드
```

=> for문의 변수 또한 일반 변수와 동일한 이름짓기 규칙을 따릅니다.

for 뒤에 있는 변수의 역할은?



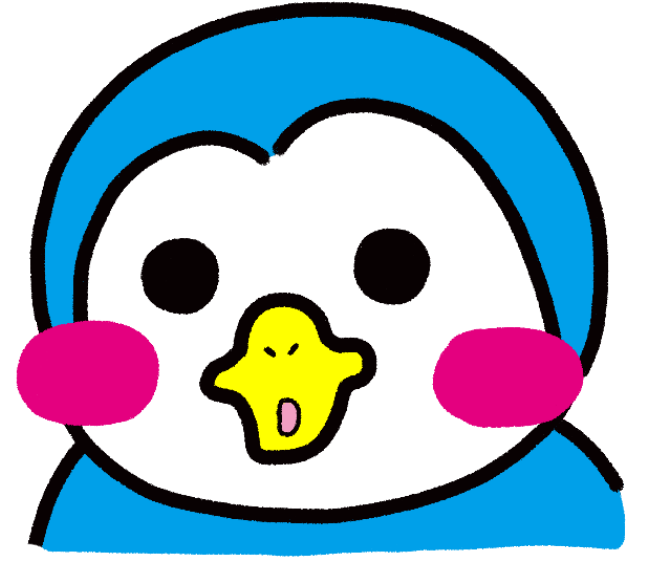
for문의 변수는 데이터모음 안의 멤버를 순서대로 기억하는 역할을 한다. 프로그래머는 이 변수를 통해 반복 코드 안에서 데이터모음의 멤버를 활용할 수 있다.

한번 반복될 때마다
값이 바뀌어요!

```
for 변수 in 데이터모음 :  
    # 반복해서 실행하고픈 코드
```

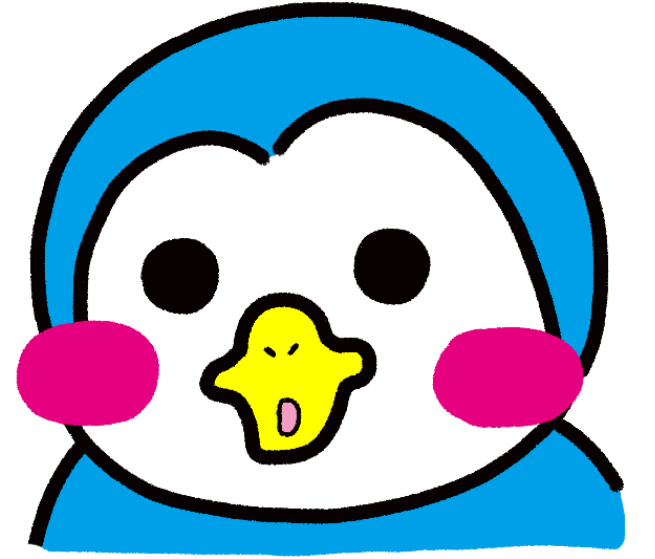
=> for문의 변수 또한 일반 변수와 동일한 이름짓기 규칙을 따릅니다.

동작 흐름을 파악하자



```
for fruit in ["사과", "포도", "자두"] :  
    phrase = "맛있는 과일, %s!" % fruit  
    print(phrase)
```

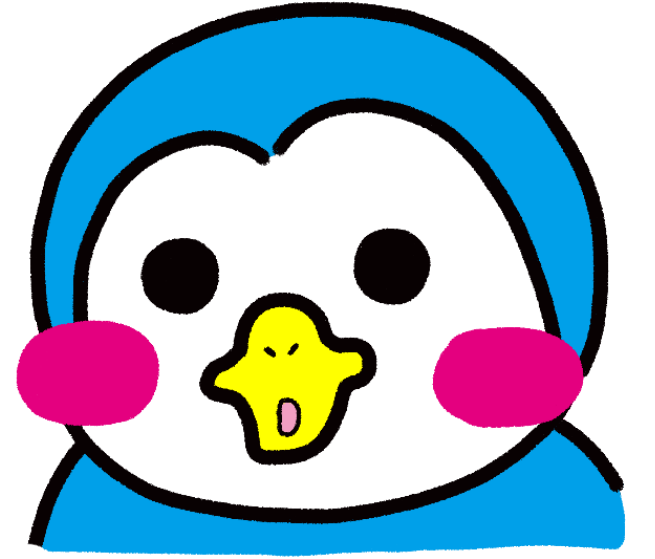

동작 흐름을 파악하자



```
① for fruit in ["사과", "포도", "자두"] :  
    phrase = "맛있는 과일, %s!" % fruit  
    print(phrase)
```

자, for문을 시작하자. 데이터모음으로 멤버가 3개인 리스트가 왔으니,
이 for문은 포함하고 있는 코드를 3회 반복할 것이다.

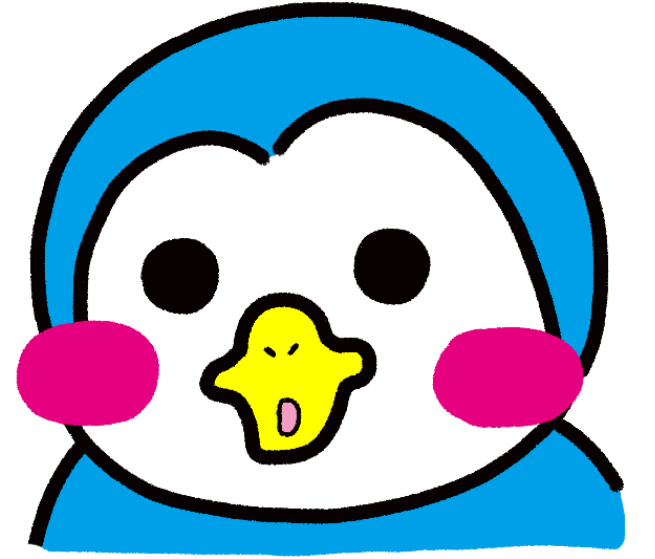
동작 흐름을 파악하자



```
① for fruit in ["사과", "포도", "자두"] :  
    ② phrase = "맛있는 과일, %s!" % fruit  
    print(phrase)
```

for문에 포함된 코드의 첫 번째 실행이다.
이때 변수 fruit의 값은 리스트의 첫 번째 멤버인 “사과”다.

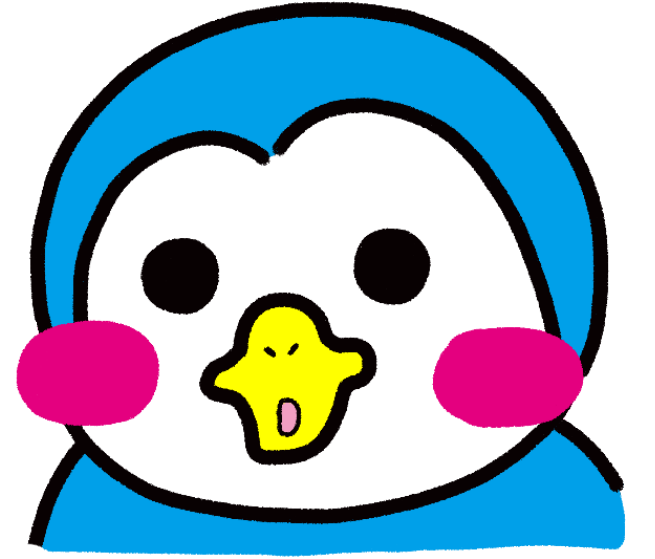
동작 흐름을 파악하자



```
① for fruit in ["사과", "포도", "자두"] :  
    ② phrase = "맛있는 과일, %s!" % fruit  
    ③ print(phrase)
```

for문 아직 안 끝났다! 두 번째 실행이다.
변수 fruit의 값은 리스트의 두 번째 멤버인 “포도”다.

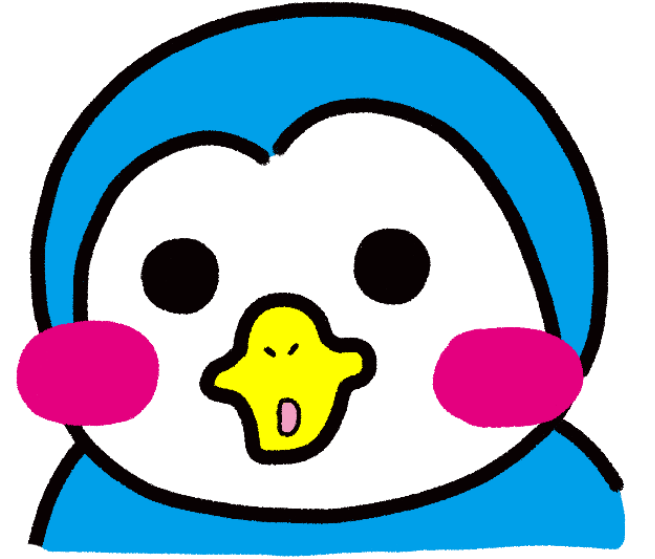
동작 흐름을 파악하자



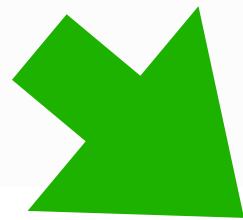
```
① for fruit in ["사과", "포도", "자두"] :  
    ② phrase = "맛있는 과일, %s!" % fruit  
    ③  
    ④ print(phrase)
```

마지막 세 번째 실행! fruit의 값은 리스트의 마지막 멤버 “자두”이고,
이번을 끝으로 for문도 끝이 난다.

동작 흐름을 파악하자

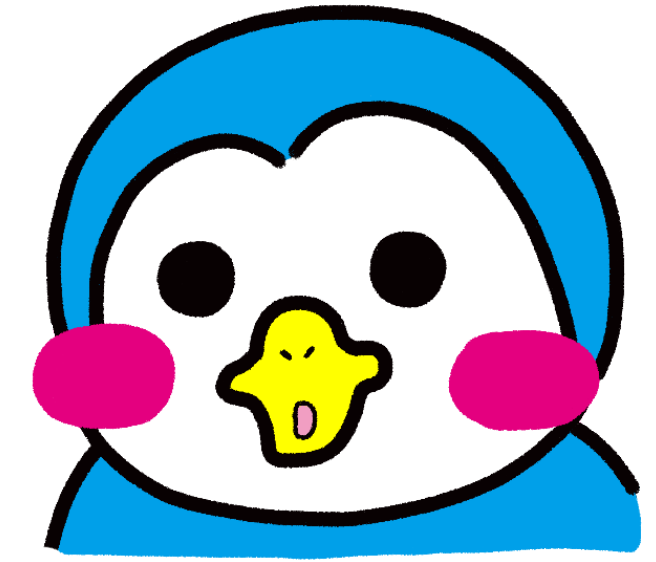


```
for fruit in ["사과", "포도", "자두"] :  
    phrase = "맛있는 과일, %s!" % fruit  
    print(phrase)
```



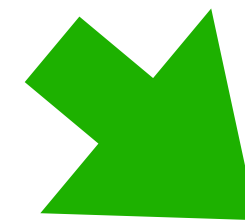
```
맛 있는 과일 , 사과 !  
맛 있는 과일 , 포도 !  
맛 있는 과일 , 자두 !
```

for문의 단짝, range(x)



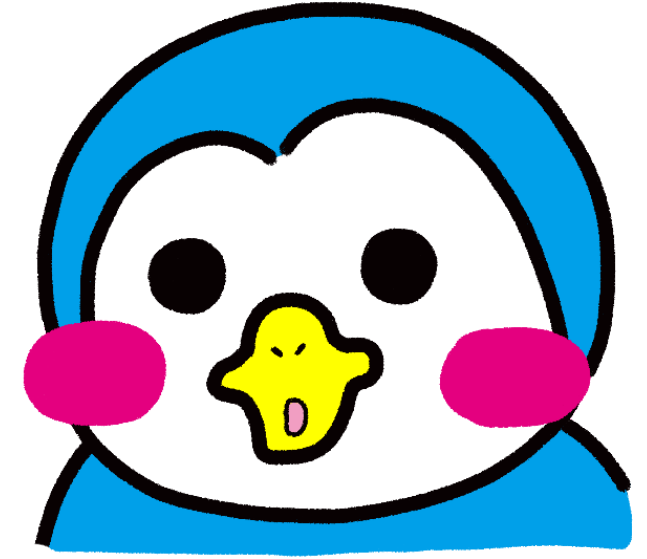
range(x)는 for문의 반복 횟수를 정할 목적으로 곧잘 사용되곤 하는 함수다.
이 함수에 숫자를 입력하면 입력된 크기만큼의 정수를 가진 iterable 객체가 반환된다.

```
for i in range(5) :  
    print( "5회 반복을 위해 range(5) 사용!" )
```



```
5회 반복을 위해 range(5) 사용!  
5회 반복을 위해 range(5) 사용!  
5회 반복을 위해 range(5) 사용!  
5회 반복을 위해 range(5) 사용!  
5회 반복을 위해 range(5) 사용!
```


내용 정리



- 반복문은 비슷하거나 동일한 구문을 반복해서 수행할 수 있는 구문이다.
- for문은 주어진 데이터모음의 멤버 수만큼 코드를 반복 실행하는 구문이다.
- 데이터모음으로는 iterable 객체(문자열, 리스트, 튜플 등)이 올 수 있다.
- for문 안에서 데이터모음의 각 데이터를 활용할 수 있다.
- range(x)는 iterable 객체를 반환하는 함수로, 입력값을 따라 값의 개수 및 범위를 결정하는 특징을 가지고 있다.