



파이썬과 친해지기

짧은 널기 연산자, for문 활용, 함수 정의,
len(), sum()

6주차

짧은 넣기 연산자

```
>>> a = 5
```

```
>>> a = a + 3
```

```
>>> a
```

8

```
>>> b = 4
```

```
>>> b = b - 1
```

```
>>> b
```

3

어떤 변수에 뭔가를 더하거나 빼서 다시 그 변수에 넣는 계산
자주 쓰이지만 너무 길다.

짧은 늘기 연산자

```
>>> a = 5
```

```
>>> a += 3
```

```
>>> a
```

```
8
```

```
>>> b = 4
```

```
>>> b -= 1
```

```
>>> b
```

```
3
```

훨씬 간단하다.

$+=$ 연산자

$a += b$

a 에 b 를 더해서 다시 a 로 넣는다. $a = a + b$ 와 같다.

```
>>> a = 5
```

```
>>> a += 3
```

```
>>> a
```

```
8
```

짧은 넣기 연산자

+ =

- =

***** =

/ =

+, -, *, /, %, **, //

모두 뒤에 등호를 붙일 수 있다.

복습

리스트는 대괄호, 즉 '[' 와 ']' 를 사용!

```
food = ["고기", "치킨", "삼겹살"]  
money = [1000, 1000, 1000, 500]  
empty_list = []
```

복습

for A in B:

B 안의 모든 것을 한번 씩 A에 넣고 블록을 실행한다.

```
>>> for a in [1, 2, 3, 4, 5]:  
        print("양", a, "마리...")
```

```
양 1 마리...
```

```
양 2 마리...
```

```
양 3 마리...
```

복습

range(a, b)

“a 이상 b 미만인 모든 정수”라는 표현을 뱉는다.

```
>>> for a in range(1, 6):  
    print("양", a, "마리...")
```


복습

```
for a in range(1, 101):  
    print("양", a, "마리...")
```

100이 아니라 101!



```
양 95 마리...  
양 96 마리...  
양 97 마리...  
양 98 마리...  
양 99 마리...  
양 100 마리...
```

같이 해보기

1. 리스트 a 안에 들어있는 모든 값을 출력해보자.
단 for 문을 이용한다.

```
a = [1, 2, 3, 4, 5]
```

```
>>> %Run Asriel.py
```

```
1
```

```
2
```

```
3
```

```
4
```

같이 해보기

1. 리스트 a 안에 들어있는 모든 값을 출력해보자.
단 for 문을 이용한다.

```
a = [1, 2, 3, 4, 5]
```

```
for b in a:
```

같이 해보기

1. 리스트 a 안에 들어있는 모든 값을 출력해보자.
단 for 문을 이용한다.

```
a = [1, 2, 3, 4, 5]
```

```
for b in a:
```

a 안에 있는 모든 값을 한번 씩 b에 집어넣어 블록을 반복한다.
블록 안에서 b를 출력한다면 어떻게 될까?

1. 리스트 a 안에 들어있는 모든 값을 출력해보자.
단 for 문을 이용한다.

```
a = [1, 2, 3, 4, 5]  
for b in a:
```

1. 리스트 a 안에 들어있는 모든 값을 출력해보자.
단 for 문을 이용한다.

```
a = [1, 2, 3, 4, 5]  
for b in a:  
    print(a)
```

같이 해보기

2. 리스트 a 안에 들어있는 값의 갯수를 세서 출력해보자.
단 for 문을 이용한다.

```
a = [1, 3, "안녕", "치킨", True]
```

```
>>> %Run Asriel.py  
5 개입니다.
```

같이 해보기

2. 리스트 a 안에 들어있는 값의 갯수를 세서 출력해보자.
단 for 문을 이용한다.

```
a = [1, 3, "안녕", "치킨", True]
```

```
for b in a:
```


같이 해보기

2. 리스트 a 안에 들어있는 값의 갯수를 세서 출력해보자.
단 for 문을 이용한다.

```
a = [1, 3, "안녕", "치킨", True]
```

```
for b in a:
```

a 안에 있는 모든 값을 한번 씩 b에 집어넣어 블록을 반복한다.
즉 a 안에 값이 5개면, 블록도 다섯 번 실행된다!

2. 리스트 a 안에 들어있는 값의 갯수를 세서 출력해보자.
단 for 문을 이용한다.

```
a = [1, 3, "안녕", "치킨", True]  
count = 0
```

2. 리스트 a 안에 들어있는 값의 갯수를 세서 출력해보자.
단 for 문을 이용한다.

```
a = [1, 3, "안녕", "치킨", True]  
count = 0  
for b in a:
```

2. 리스트 a 안에 들어있는 값의 갯수를 세서 출력해보자.
단 for 문을 이용한다.

```
a = [1, 3, "안녕", "치킨", True]
count = 0
for b in a:
    count += 1
```

2. 리스트 a 안에 들어있는 값의 갯수를 세서 출력해보자.
단 for 문을 이용한다.

```
a = [1, 3, "안녕", "치킨", True]
count = 0
for b in a:
    count += 1
print(count, "개가 들어있습니다.")
```

같이 해보기

3. 리스트 a 안에 들어있는 모든 값을 합쳐서 출력해보자.
단 for 문을 이용한다.

```
a = [1, 2, 3, 4, 5]
```

```
>>> %Run Asriel.py  
합은 15 입니다.
```

같이 해보기

3. 리스트 a 안에 들어있는 모든 값을 합쳐서 출력해보자.
단 for 문을 이용한다.

```
a = [1, 2, 3, 4, 5]
```

```
for b in a:
```

같이 해보기

3. 리스트 a 안에 들어있는 모든 값을 합쳐서 출력해보자.
단 for 문을 이용한다.

```
a = [1, 2, 3, 4, 5]
```

```
for b in a:
```

a 안에 있는 모든 값을 한번 씩 b에 집어넣어 블록을 반복한다.
블록에서 b를 어딘가에 합쳐주면 어떻게 될까?

3. 리스트 a 안에 들어있는 모든 값을 합쳐서 출력해보자.
단 for 문을 이용한다.

```
a = [1, 2, 3, 4, 5]  
합 = 0
```

3. 리스트 a 안에 들어있는 모든 값을 합쳐서 출력해보자.
단 for 문을 이용한다.

```
a = [1, 2, 3, 4, 5]
```

```
합 = 0
```

```
for b in a:
```

3. 리스트 a 안에 들어있는 모든 값을 합쳐서 출력해보자.
단 for 문을 이용한다.

```
a = [1, 2, 3, 4, 5]
```

```
합 = 0
```

```
for b in a:
```

```
    합 += b
```

3. 리스트 a 안에 들어있는 모든 값을 합쳐서 출력해보자.
단 for 문을 이용한다.

```
a = [1, 2, 3, 4, 5]
합 = 0
for b in a:
    합 += b
print("합은", 합, "입니다.")
```

직접 해보기

1. a는 [1, 2, 3, 4, 5, 6, 7, 8] 인 리스트라고 하자.
a의 모든 값 중 4 이상인 것 만을 출력해보자.

```
>>> %Run Asriel.py
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

2. a는 [1, 2, 3, 4, 5] 인 리스트라고 하자. a를 출력한 다음, a 안의 모든 값을 곱해 출력해보자.

힌트 : 합을 구할 때 “합” 변수가 0이었지만,
“곱” 변수는 처음에 1이어야 한다.

```
>>> %Run Asriel.py
```

```
a는 [1, 2, 3, 4, 5] 입니다.  
곱은 120 입니다.
```

3. a는 [1, 2, 3, 4, 5] 인 리스트라고 하자. a를 출력한 다음, a 속 값들의 두 배가 들어가는 b를 만들어 출력해보자.

힌트 : b를 우선 빈 리스트 [] 로 두고 b.append() 함수를 이용해 값을 하나씩 추가해보자.

```
>>> %Run Asriel.py
```

```
a는 [1, 2, 3, 4, 5] 입니다.
```

```
b는 [2, 4, 6, 8, 10] 입니다.
```

함수 만들기

```
a = input("안녕?")  
print("hello")  
합 = sum([1,2,3,4])
```

input 함수, print 함수, sum 함수
이런 함수를 직접 만들 수는 없을까?

def

def A():

A 라는 이름의 함수를 만든다.

```
>>> def 집지키기():  
    print("으르르...")  
    print("왈왈!!")
```

```
>>> 집지키기()  
으르르...  
왈왈!!
```

def 예제

```
def 자기소개():  
    print("안녕하세요! 저는 파이썬입니다.")  
    print("파이썬은 어떤 뱀의 이름입니다.")
```

자기소개()



```
>>> %Run Asriel.py
```

```
안녕하세요! 저는 파이썬입니다.  
파이썬은 어떤 뱀의 이름입니다.
```

def - 2

```
def A(a1, a2, ...):
```

여러 인수를 받는 A 라는 이름의 함수를 만든다.

```
>>> def 합출력(a, b):  
    print(a + b)
```

```
>>> 합출력(2, 3)  
5
```

```
def 먹기(음식):  
    print("선생님은", 음식, "을 먹었다.")
```

```
먹기("치킨")  
먹기("음료수")
```



```
>>> %Run Asriel.py  
선생님은 치킨 을 먹었다.  
선생님은 음료수 을 먹었다.
```

```
def 먹기(사람, 음식):  
    print(사람, "은", 음식, "을 먹었다.")
```

```
먹기("선생님", "치킨")  
먹기("동현", "샐러드")
```



```
>>> %Run Asriel.py  
선생님 은 치킨 을 먹었다.  
동현 은 샐러드 을 먹었다.
```

return

return A

함수 안에서 A를 뱉고 함수를 끝낸다.

```
>>> def 두배(num):  
    return num * 2
```

```
>>> print(두배(5))  
10
```

return 예제

```
def 사랑스럽게(text):  
    return text + "~!♥"
```

```
a = 사랑스럽게("엄마 치킨사주세요")  
print(a)
```



```
>>> %Run Asriel.py  
엄마 치킨사주세요~!♥
```

return 예제

```
def 입력(질문):  
    return input(질문)
```

```
name = 입력("이름을 대라!")  
print(name, "님, 안녕하세요!")
```



```
>>> %Run Asriel.py  
이름을 대라! 다람쥐  
다람쥐 님, 안녕하세요!
```


return 예제

```
def iinput(질문):  
    a = input(질문)  
    a = int(a)  
    return a
```

```
num = iinput("숫자를 입력하세요.")  
print(num, "의 두배는", num*2, "입니다.")
```

def, return 을 활용해 유용한 함수를 만들 수 있다.

직접 해보기

1. 이름을 인수로 받아 인사를 하는 함수를 만들자.
그 함수로 우리 반 모든 사람들에게 인사를 해보자.

```
Hello("지완")
```



```
>>> %Run Asriel.py  
안녕하세요, 지완 님!
```

2. 두 수를 인수로 받아 곱해서 뱉는 함수를 만들자.
그 함수에 몇가지 숫자를 넣어 부르고, 그 결과를 출력해보자.

힌트 : 함수 안에서 `print()` 대신 `return` 을 사용해야한다.

```
print(곱(2, 4))  
print(곱(6, 5))
```



```
>>> %Run Asriel.py
```

```
8
```

```
30
```

3. 리스트를 인수로 받고, 그 리스트 안의 모든 것을 곱해 출력하는 함수를 만들자. 그 함수에게 **[1, 2, 3, 4, 5]** 등 여러 리스트를 줘서 불러보자.

```
곱([1,2,3,4,5])  
곱([1,2,3])
```



```
>>> %Run Asriel.py  
120  
6
```

길이를 구하는 법

```
a = [1, 3, "안녕", "치킨", True]
count = 0
for b in a:
    count += 1
print(count, "개가 들어있습니다.")
```

너무 길다...



```
>>> %Run Asriel.py
5 개가 들어있습니다.
```

길이를 구하는 법

```
a = [1, 3, "안녕", "치킨", True]
count = len(a)
print(count, "개가 들어있습니다.")
```



len() 함수를 이용!

```
>>> %Run Asriel.py
5 개가 들어있습니다.
```

len()

len(A)

A의 길이를 뱉는다.

```
>>> len("안녕하세요?")  
6
```

```
>>> len([-2, 0, 2])  
3
```

합 구하는 법

```
a = [1, 2, 3, 4, 5]
```

```
합 = 0
```

```
for b in a:
```

```
    합 += b
```

```
print("합은", 합, "입니다.")
```

이것도 너무 길다!



```
>>> %Run Asriel.py
```

```
합은 15 입니다.
```


합 구하는 법

```
a = [1, 2, 3, 4, 5]
합 = sum(a)
print("합은", 합, "입니다.")
```



sum() 함수를 이용!

```
>>> %Run Asriel.py
합은 15 입니다.
```

sum()

sum(A)

A 안의 모든 값의 합을 낸다.

```
>>> sum([1, 3, 5])
```

```
9
```

```
>>> sum([-2, -1, 0, 1, 2])
```

```
0
```

직접 해보기

1. 이름을 입력받자. 이름의 글자 수가 2글자 이상, 4글자 이하이면 “회원가입이 완료되었습니다.”를, 그렇지 않으면 “이름은 2~4글자여야 합니다!”를 출력하자.

```
>>> %Run Asriel.py
이름을 입력해주세요. 뽀
이름은 2~4글자여야 합니다!
```

2. 숫자를 입력받자. 1부터 그 숫자까지의 합을 출력하는 프로그램을 만들자. 예를 들어 “5”를 입력받으면 1 이상 5 미만의 모든 수의 합을 출력한다.

힌트: `int()`, `range()`, `sum()`을 잘 활용해보자.

```
>>> %Run Asriel.py
수를 입력해주세요. 5
15
```

3. 구구단을 출력하는 프로그램을 작성해보자.

힌트: for문을 중첩해서 사용해보자.

```
>>> %Run Asriel.py
```

```
1 x 1 = 1
```

```
1 x 2 = 2
```

```
1 x 3 = 3
```

```
1 x 4 = 4
```

(생략)

```
9 x 6 = 54
```

```
9 x 7 = 63
```

```
9 x 8 = 72
```

```
9 x 9 = 81
```



파이썬과 친해지기

짧은 널기 연산자, for문 활용, 함수 정의,
len(), sum()

6주차

References

왕초보를 위한 Python 2.7
뱀 인형 이미지

<https://wikidocs.net/145>
<https://bit.ly/2WINL65>