



파이썬과 친해지기

조건문, 타입 변환, 리스트, 반복문(for)

2주차

복습

“변하는 수” : 숫자, 문자 등을 담는 주머니

숫자 넣는 법

$x = 5$
 $number = -9.5$

문자 넣는 법

$friend = \text{“다람쥐”}$
 $pet = \text{‘강아지’}$

복습

변수끼리도 계산할 수 있다!

```
>>> 지갑 = 5000
>>> 주인 = "김다솜"
>>> 지갑 = 지갑 * 2
>>> 지갑
10000
```

```
>>> 신발 = "나이키"
>>> 주인 = "이지원"
>>> 주인 + 신발
'이지원나이키'
```

복습

print()

괄호 안의 값을 화면에 출력

```
>>> name = "김다람"
>>> dream = "마법사"
>>> print(name, "님의 장래희망: ", dream)
김다람  님의 장래희망:   마법사
```

복습

```
print("사탕" + 3 + "개")
```



```
>>> %Run test.py
Traceback (most recent call last):
  File "C:\Users\lumer\Desktop\test.py", line 1, in <
module>
    print("사탕" + 3 + "개")
TypeError: can only concatenate str (not
"int") to str
```

오류 발생!
(문자와 숫자를 합칠 수 없음)

```
print("사탕", 3, "개")
```



```
>>> %Run test.py
사탕 3 개
>>>
```

정상 출력

복습

input()

사용자에게 무언가를 질문함

```
name = input("이름이 무엇인가요?")  
print("안녕하세요,", name, "님!")
```

복습

```
>>> 3 > 1
```

```
True (예)
```

```
>>> 10 == 10
```

```
True (예)
```

```
>>> 5 == 6
```

```
False (아니오)
```

복습

if A:

만약에 A가 참이면 if 블록을 실행

```
>>> if 1 + 1 == 2:  
    print("1+1=2 가 맞습니다.")
```

```
1+1=2 가 맞습니다.
```


조건문

if A:, else:

만약에 A가 참이면 if 블록을 실행, 아니면 else 블록을 실행

```
if dinner = "고기":  
    print("오늘 저녁은 고기!")  
else:  
    print("오늘 저녁은 고기가 아니네...")
```

조건문

달면 삼키고 쓰면 뱉는다.

if 달면:
삼킨다.

else:
뱉는다.

조건문

if A, elif B, elif C ..., else:

만약에 A가 참이면 if 블록을 실행,
아니면 B를 보고 참이면 실행,
아니면 C를 보고...
모두 아니면 else 블록을 실행

조건문

간장은 짜고, 설탕은 달고, 나머지는 모르겠어!

if 간장:

 짜다!

elif 설탕:

 달다!

else:

 모르겠다...

※ elif는 몇 개가 있던 상관 없고, else는 없어도 된다.

if 간장:

짜다!

elif 설탕:

달다!

elif 치킨:

맛있어!

elif 물감:

그걸 왜 먹어...

```
a = input("뭘 먹을까?")

if a == "간장":
    print("짜다!")
elif a == "설탕":
    print("달다!")
elif a == "치킨":
    print("맛있어!")
elif a == "물감":
    print("그걸 왜 먹어...")
else:
    print("뭔지 모르겠다.")
```

직접 해보기

1. 사용자에게 이름을 입력 받고 자기 이름이면 “로그인 성공!”, 아니면 “로그인 실패..” 를 출력하는 프로그램을 만들어보자.

```
>>> %Run test.py
```

```
이름을 대라! 강지민  
로그인 성공!
```

```
>>> %Run test.py
```

```
이름을 대라! 한동현  
로그인 실패..
```

```
>>>
```

2. 사용자에게 이름을 입력받고 그 이름의 사람은 몇 살인지 알려주는 프로그램을 만들어보자. 단 모르는 사람의 이름이 입력되면 “모르겠다” 고 출력하자.

```
>>> %Run test.py
```

```
이름을 대라! 심은섭  
21살
```

```
>>> %Run test.py
```

```
이름을 대라! 한지원  
12살
```

```
>>> %Run test.py
```

```
이름을 대라! 우리집 강아지 뽀뽀  
모르겠어..
```


복습

정수	int	0, 123, -5000
실수	float	1.345, -1.0, 999.99
스트링(문자)	str	“치킨”, “먹고싶다”
불리언	bool	True(예), False(아니오)

타입 변환

```
a = input("숫자를 입력해주세요.")  
b = a * 2  
print(a, "의 두배는", b, "입니다.")
```



```
>>> %Run test.py  
숫자를 입력해주세요.5  
5 의 두배는 55 입니다.
```

타입 변환

```
a = input("숫자를 입력해주세요.")  
b = a * 2  
print(a, "의 두배는", b, "입니다.")
```

숫자를 입력해주세요. 5 $a = \text{"5"}$

$b = \text{"5"} * 2$

5 의 두배는 55 입니다.

타입 변환

```
a = input("a를 입력하세요.")  
b = a + 1 ← 오류 발생!!  
print("a + 1 은", b, "입니다.")
```

왜 오류가 날까?

타입 변환

```
a = input("a를 입력하세요.")  
b = a + 1 ← 오류 발생!!  
print("a + 1 은", b, "입니다.")
```

문자(str)와 숫자는 + 할 수 없다!
(문자는 문자끼리, 숫자는 숫자끼리만 가능하다)

정수 변환

int(A)

A를 정수로 변환하여 뱉는다.

```
>>> int("10") 문자를 정수로 변환  
10
```

```
>>> int(3.5) 실수를 정수로 변환  
3
```

정수 변환

```
a = input("숫자를 입력해주세요.")  
b = int(a)  
c = b * 2  
print("2를 곱하면", c, "가 됩니다.")
```



```
>>> %Run test.py  
숫자를 입력해주세요.5  
2를 곱하면 10 가 됩니다.
```

정수 변환

```
a = int(input("숫자를 입력해주세요. "))  
c = a * 2  
print("2를 곱하면", c, "가 됩니다.")
```



```
>>> %Run test.py  
숫자를 입력해주세요. 5  
2를 곱하면 10 가 됩니다.
```


실수 변환

float(A)

A를 실수로 변환하여 뱉는다.

>>> float("3.5") 문자를 실수로 변환

3.5

>>> float(5) 정수를 실수로 변환

5.0

실수 변환

```
a = input("숫자를 입력해주세요. ")  
b = float(a)  
c = b / 2  
print("2로 나누면", c, "입니다.")
```



```
>>> %Run test.py  
숫자를 입력해주세요. 0.5  
2로 나누면 0.25 입니다.
```

문자 변환

str(A)

A를 문자로 변환하여 뱉는다.

```
>>> str(5)  
'5'
```

정수를 문자로 변환

```
>>> str(3.5)  
'3.5'
```

실수를 문자로 변환

문자 변환

```
a = 1 + 1  
b = "1 더하기 1은 " + str(a) + "입니다."  
print(b)
```



```
>>> %Run test.py  
1 더하기 1은 2입니다.
```

직접 해보기

1. 사용자에게 정수 a 와 b 를 입력 받으면, 두 수를 더한 값, 뺀 값, 곱한 값을 출력하자.

```
>>> %Run test.py
정수 a를 입력하세요. 5
정수 b를 입력하세요. 3
더하면 8
빼면 2
곱하면 15
```

2. 치킨의 가격을 입력 받자. 만약 만원보다 싸면 “치킨 시켜야지~”를, 그렇지 않으면 “용돈을 더 모아야겠다...”를 출력해보자.

```
>>> %Run test.py
```

```
치킨이 얼마지?! 8000  
치킨 시켜야지~
```

```
>>> %Run test.py
```

```
치킨이 얼마지?! 50000  
용돈을 더 모아야겠다...
```

리스트(목록)

엄마 심부름

1. 두부
2. 간장
3. 대파
4. 벌꿀
- ...

내 인생 버킷리스트

1. 부자 되기
2. 7일 내내 치킨만 먹어보기
3. 24시간동안 잠 자보기
- ...

리스트

리스트는 대괄호, 즉 '[' 와 ']' 를 사용!

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


리스트 연산

리스트끼리 더할 수 있다. '+'

```
>>> food = ["계란찜", "삼겹살"]  
>>> bob = ["공기밥"]  
>>> food + bob  
['계란찜', '삼겹살', '공기밥']
```

리스트에 넣기 (1)

리스트를 합치는 것을 응용!

```
>>> a = []  
>>> a = a + ["사탕"]  
>>> a = a + ["초콜릿"]  
>>> a  
['사탕', '초콜릿']
```

a에 “사탕” 이 들어간다

a에 “초콜릿” 이 들어간다

리스트에 넣기 (2)

리스트가 가지고 있는 append 함수를 이용

```
>>> a = []  
>>> a.append("사탕")  
>>> a.append("초콜릿")  
>>> a  
['사탕', '초콜릿']
```

a에 “사탕” 이 들어간다

a에 “초콜릿” 이 들어간다

점의 의미

치킨.가격



“치킨의 가격”이라는 뜻

전등.켜기()



“전등”이 가지고 있는
“켜기”라는 함수를 부른다.

점의 의미

치킨의 가격
현빈이의 나이

치킨.가격
현빈.나이

닭이 퍼덕대기
동현이가 치킨을 먹기

닭.퍼덕대기()
동현.먹기(치킨)

리스트에 3을 넣기

리스트.append(3)

list.append()

myList.append(A)

A를 myList 리스트 안에 넣는다.

```
>>> a = ["감자", "고구마"]  
>>> a.append("당근")  
>>> a  
['감자', '고구마', '당근']
```

list.append()

myList.clear()

myList 리스트 안의 내용을 없앤다.

```
>>> home = ["먼지", "쓰레기", "파리"]  
>>> home.clear()  
>>> home  
[]
```

직접 해보기

1. “삼겹살”, “계란찜” 이 들어있는 리스트를 만들자.
“비빔냉면”, “아이스크림” 이 들어있는 리스트를 만들자. 두 리스트를 합친 또 다른 리스트를 만들자. 리스트 세 개를 모두 출력해보자.

```
>>> %Run test.py
```

```
['삼겹살', '계란찜']
```

```
['비빔냉면', '아이스크림']
```

```
['삼겹살', '계란찜', '비빔냉면', '아이스크림']
```


직접 해보기

2. 빈 리스트를 하나 만들자. 그 리스트 안에 냄비에 넣을 재료를 사용자에게 입력 받아 총 3가지를 넣자. 마지막으로 그 리스트를 출력해보자.

```
>>> %Run test.py
```

```
물 넣을까? 물
```

```
물 넣을까? 라면스프
```

```
물 넣을까? 마시멜로
```

```
['물', '라면스프', '마시멜로']
```

반복문

```
print("양", 1, "마리...")  
print("양", 2, "마리...")  
print("양", 3, "마리...")  
print("양", 4, "마리...")  
print("양", 5, "마리...")
```

너무 귀찮은데?!

for 반복문

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...
```

for 반복문

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

↓ 같은 뜻이다.

```
print("양", 1, "마리...")  
print("양", 2, "마리...")  
print("양", 3, "마리...")  
print("양", 4, "마리...")  
print("양", 5, "마리...")
```

for 반복문

```
chickens = ["간장", "양념", "뿌링클", "갈릭"]  
for a in chickens:  
    print(a + "치킨이 최고!")
```



```
>>> %Run test.py
```

```
간장치킨이 최고!  
양념치킨이 최고!  
뿌링클치킨이 최고!  
갈릭치킨이 최고!
```

for 반복문

```
name = "Asriel"  
for a in name:  
    print(a * 8)
```



```
>>> %Run test.py  
AAAAAAA  
SSSSSSSS  
rrrrrrrrr  
iiiiiii  
eeeeeeee  
llllllll
```

양을 백마리까지 세려면?

```
print("양", 1, "마리...")
print("양", 2, "마리...")
print("양", 3, "마리...")
print("양", 4, "마리...")
print("양", 5, "마리...")
print("양", 6, "마리...")
print("양", 7, "마리...")
print("양", 8, "마리...")
print("양", 9, "마리...")
print("양", 10, "마리...")
print("양", 11, "마리...")
print("양", 12, "마리...")
print("양", 13, "마리...")
print("양", 14, "마리...")
print("양", 15, "마리...")
print("양", 16, "마리...")
print("양", 17, "마리...")
print("양", 18, "마리...")
print("양", 19, "마리...")
```

너무 길다...

양을 백마리까지 세려면?

```
for a in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]:  
    print("양", a, "마리...")
```

이것도 너무 길다...

직접 다 쓰는 대신,
“1부터 100의 모든 정수”를 뜻하는
간단한 표현이 필요하다!

range()

range(a, b)

“a 이상 b 미만인 모든 정수”라는 표현을 뱉는다.

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

양을 백마리 까지 세려면?

```
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...
```

직접 해보기

1. 좋아하는 음식을 담은 리스트를 만들자.
그 리스트에 있는 모든 음식 a에 대해 for 반복문을 통해 “a는 맛있다!”를 출력해보자.

```
>>> %Run test.py
```

```
치킨은 맛있다!  
과자는 맛있다!  
고기는 맛있다!  
치즈는 맛있다!
```

직접 해보기

2. 사용자에게 정수를 하나 입력 받자. 1부터 그 숫자까지 모두 더한 합을 구하고, 그 합을 출력해보자.

```
>>> %Run test.py
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

숙제

1. 오늘까지 배운 것 복습하기!



파이썬과 친해지기

조건문, 타입 변환, 리스트, 반복문(for)

2주차

References

왕초보를 위한 Python 2.7
뱀 인형 이미지
기타 이미지

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