



python

p

쉽게

y t

배우는

파이썬

제 1편. hello, python?

With 정훈희

h o n

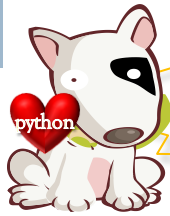


13. 또다른 자료형 리스트, 튜플



이번시간에 학습할 내용은..

앞서 공부한 변수는 하나의 숫자 또는 문자를
저장하는 공간이었습니다. 그런데, 한번에 여러 개
의 숫자나 문자를 하나로 묶어서 저장할 필요도
있습니다. 이런 개념이 리스트 인데요. 함께
학습해보죠^^



리스트(list) 가 뭐냐하면?



연우,영철,길동,아름이는
내 친구야.



그럼, 친구=[연우, 영철,길동,아름]
이렇게 이야기할 수 있어요



내가 좋아하는 색상은
"빨강","노랑","연초록","
연보라"야.



그럼, 색상=[빨강,노랑,연초록,연보라]와
같이 이야기할 수 있어요.

리스트=여러 개의 문자, 숫자를 하나의 그룹으로 묶어서 저장해 놓은 개념

```
친구=['연우','영철','길동','아름']  
색상=['빨강','노랑','연초록','연보라']
```

리스트의 항목

리스트이름

리스트는 대괄호([])로 시작과 끝을 감싸고,
각 항목을 구분할 때 쉼표(,) 이용



리스트엔 별걸 다 집어넣을 수 있다구요^^



내가 좋아하는 과일은 "사과"
내가 좋아하는 친구는 "홍길동"
내가 좋아하는 숫자는 7,9
내가 좋아하는 알파벳은 'z'
이걸 하나로 묶을 수 있을까?



리스트엔 다 넣을 수 있어요
mylist라는 그릇에 이 모든걸 담아 볼게요
`mylist=["사과","홍길동",7,9,'z']`



숫자와 문자열을 그렇게 섞어서
한 곳에 담아도 돼?



리스트의 항목에는 어떤 자료형도 모두
올 수 있어요. 걱정마세요^^ 이런것도
돼요. 아래 보세요.



```
myone=["분홍","노랑","연초록"]  
mylist=[7,9,"사과",["분홍","노랑"]]
```

리스트안에 또다른 리스트를 항목으로 표시할 수 있어요



리스트 만들고 항목추가하기

◆ 리스트 만들기

- 리스트 값들도 결국은 변수에 할당되는 것입니다.
- 따라서, 리스트를 만들 때는 변수와 같은 방법으로 생성하면 됩니다.

```
mylist=[ ]  
이렇게 빈 리스트를 만들 수 있어요^^
```

◆ 리스트에 항목 추가하는 작업

- 리스트에 항목을 추가할 경우 리스트이름.append(항목명) 과 같이 코딩합니다.

```
mylist=[ ]  
mylist.append("사과") => mylist 리스트에 "사과"항목을 추가하라는 의미
```



리스트 항목 가져오고 출력하는 방법

◆ 리스트 항목 가져오기

`mylist=["사과","배","복숭아","포도","오렌지"]` 가 있는 경우

| mylist | <code>mylist[0]</code> | <code>mylist[1]</code> | <code>mylist[2]</code> | <code>mylist[3]</code> | <code>mylist[4]</code> |
|--------|------------------------|------------------------|------------------------|------------------------|------------------------|
| | 사과 | 배 | 복숭아 | 포도 | 오렌지 |



`mylist` 아파트 맨 앞집에 살아요.
나를 `mylist[0]` 이라고 불러주세요



`mylist` 아파트 사과 옆 집에 살아요
`mylist[1]` 이라고 불러주세요



`mylist` 아파트 배 옆 집에 살아요
`mylist[2]` 라고 불러주세요



`mylist` 아파트 복숭아 옆 집에 살아요
`mylist[3]` 이라고 불러주세요

그럼 포도를 화면에 출력하려면
`print(mylist[3])` 이렇게 써야겠네?



```
mylist=["사과","배","복숭아","포도","오렌지"]
print(mylist[3])
```

```
>>>
포도
>>> |
```

그럼 리스트의
배, 복숭아, 포도를
가져오려면?

그럼 '배'는 mylist[1]에
'복숭아'는 mylist[2], 포도는
mylist[3]이니까

mylist[1:4]와 같이 써주면 돼요.
(마지막 숫자 바로 이전까지 출력 그래서, 4)

```
mylist=['사과','배','복숭아','포도','오렌지']  
print(mylist[1:4])
```

```
>>>  
['배', '복숭아', '포도']
```

```
mylist=['사과','배','복숭아','포도','오렌지']  
print(mylist[2:])
```

```
>>>  
['복숭아', '포도', '오렌지']  
>>> |
```

mylist[2] 항목부터 끝 항목까지 출력



리스트 항목 삽입, 삭제작업하기

◆ 리스트 항목 삽입하기

`mylist=["사과","배","복숭아","포도","오렌지"]` 에서 '복숭아'앞에 '망고'삽입하는 방법

| | mylist[0] | mylist[1] | mylist[2] | mylist[3] | mylist[4] |
|--------|-----------|-----------|-----------|-----------|-----------|
| mylist | 사과 | 배 | 복숭아 | 포도 | 오렌지 |

| mylist[0] | mylist[1] | mylist[2] | mylist[3] | mylist[4] | mylist[5] |
|-----------|-----------|-----------|-----------|-----------|-----------|
| 사과 | 배 | 망고 | 복숭아 | 포도 | 오렌지 |

==> 기존의 `mylist[2]` 자리에 "망고"를 삽입하라는 의미 입니다.

=> `insert()` 메소드를 이용하여 `mylist.insert(2,"망고")` 와 같이 코딩하면 됩니다.

```
>>> mylist=['사과','배','복숭아','포도','오렌지']
>>> mylist.insert(2,'망고')
>>> print(mylist)
['사과', '배', '망고', '복숭아', '포도', '오렌지']
>>>
```


◆ 리스트 항목 삭제하기

`mylist=["사과","배","복숭아","포도","오렌지"]` 에서 '복숭아'를 제거하는 방법

| | <code>mylist[0]</code> | <code>mylist[1]</code> | <code>mylist[2]</code> | <code>mylist[3]</code> |
|---------------------|------------------------|------------------------|------------------------|------------------------|
| <code>mylist</code> | 사과 | 배 | 포도 | 오렌지 |

==> `remove()` 메소드를 이용하여 삭제하려는 항목이름을 코딩하면 됩니다.

=> 즉, `mylist.remove('복숭아')`

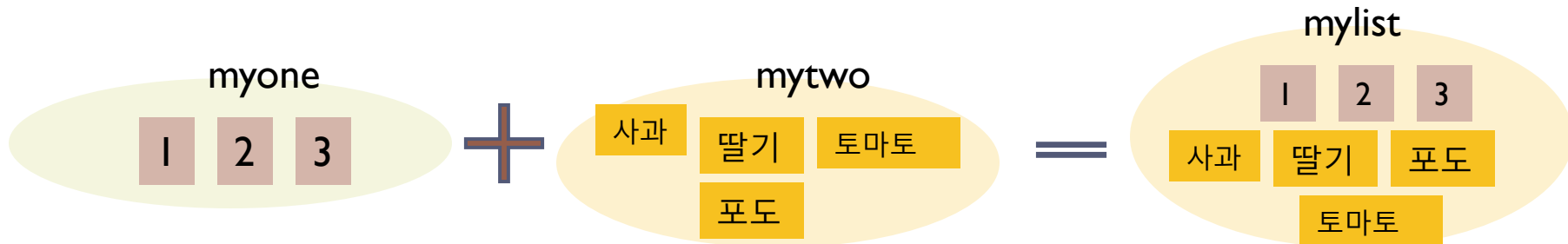
```
>>> mylist=['사과','배','복숭아','포도','오렌지']
>>> mylist.remove('복숭아')
>>> print(mylist)
['사과', '배', '포도', '오렌지']
>>>
```

==> `del` 명령어를 이용하여 삭제해도 됩니다. `del mylist[인덱스번호]`

```
>>> del mylist[1]
>>> print(mylist)
['사과', '포도', '오렌지']
>>>
```


◆ 리스트 연산작업하기 - 더하기

myone=[1,2,3] , mytwo=["사과","딸기","포도","토마토"]




```
>>> myone=[1,2,3]
>>> mytwo=['사과','딸기','포도','토마토']
>>> mylist=myone+mytwo
>>> print(mylist)
[1, 2, 3, '사과', '딸기', '포도', '토마토']
>>> |
```

◆ 리스트 연산작업하기-곱하기




mytwo 리스트 항목들을
5번씩 반복하고 싶어

`mytwo=["사과","딸기"]`




곱하기 작업을 하면 돼요.
`mytwo*5` 이렇게요 ^^

```
>>> mytwo=['사과','딸기']
>>> print(mytwo*5)
['사과', '딸기', '사과', '딸기', '사과', '딸기', '사과', '딸기', '사과', '딸기']
>>> |
```



숫자도 아닌데 곱셈이
되네?



원래 문자열도 연산작업이
된답니다. ^^

```
>>> a="파이썬"
>>> print(a*3)
파이썬 파이썬 파이썬
>>> |
```

도전 !! for 에 list 이용해보기

➤ 앞서 공부한 내용을 이용하여 input을 통해 프로그램언어를 입력받아 리스트에 항목으로 추가한 후 이를 for문 통해 출력하는 프로그램을 작성해보아요. (입력은 5번 반복하여 받도록 합니다.)


1. 먼저 빈 리스트를 선언하고, for문 통해 5번 input 입력받는 작업을 코딩합니다.

```
language=[]  
for i in range(1,6):  
    a=input("언어 입력")
```


2. 입력받은 내용을 language 리스트에 추가하는 작업을 코딩합니다.

```
language=[]  
for i in range(1,6):  
    a=input("언어 입력")  
    language.append(a)
```

3. 5번 입력을 모두 받아 language 리스트에 추가되면 해당 항목을 하나씩 출력하도록 for문을 이용해 코딩합니다.




```
language=[]  
for i in range(1,6):  
    a=input("언어 입력")  
    language.append(a)  
  
for j in language:
```




language 만 썼는데
어떻게 j는 시작값, 마지막
값을 알 수 있어요?

리스트이름만 써줘도
리스트 항목을 모두 나열한
것과 같습니다.

4. print함수 통해 j를 출력하는 작업 코딩하고 저장을 합니다.(python21-2.py)



```
language=[]  
for i in range(1,6):  
    a=input("언어 입력")  
    language.append(a)  
  
for j in language:  
    print(j)
```



```
>>>  
언어 입력 fortran  
언어 입력 perl  
언어 입력 ada  
언어 입력 c  
언어 입력 python  
fortran  
perl  
ada  
c  
python  
>>> |
```



항목을 변경하지 않을때 튜플을 ..

◆ 튜플만들기

튜플명=(항목1, 항목2...)

==>리스트와 만드는 방법은 비슷해요. 단, 대괄호([]) 대신 일반 괄호를 쓴다는 점!!
→차이점) 리스트는 항목을 추가/삭제 할 수 있는데 비해 튜플은 그럴 수 없어요.

```
t1=('사과', '배', '복숭아')  
t1.insert(1, "망고")  
print(t1)
```



```
>>>  
Traceback (most recent call last):  
  File "C:/Python34/ddfdghgj.py", line 2, in  
    t1.insert(1, "망고")  
AttributeError: 'tuple' object has no attribute  
>>>
```



이렇게 에러가 발생해요^^
튜플은 고정된 항목들을
가지고 작업할 때
사용해요



오늘의 핵심내용

1. 리스트는 문자열 뿐만 아니라 숫자도 한꺼번에 저장할 수 있는 큰 그릇!!
2. 리스트에 있는 항목들을 변경할 수 있으며, 서로 다른 리스트를 합칠 수 있습니다.
3. 이에 비해 튜플은 항목들을 수정할 수 없습니다.



잠깐!!! 연습문제라고 볼 수 는 없지만...

1. 여러분의 친구명단을 리스트로 만들어보세요. 리스트명은 friend로 하구요.
(프로그램 소스는 달리 없어요.)