

자료형, 문자열, 리스트



소프트웨어융합대학
교수 진혜진

A photograph of a person's hand typing on a white laptop keyboard. The person is wearing a black smartwatch with a square face. The laptop is on a wooden desk. In the background, there is a small white speaker and a dark object. The image is partially covered by a white overlay on the right side.

목차

1. 자료형
2. 문자열
3. 리스트

- 정수

- 10, 30
- NUMBER = 1

- 실수

- 3.1, 40.67
- GRADE=4.5

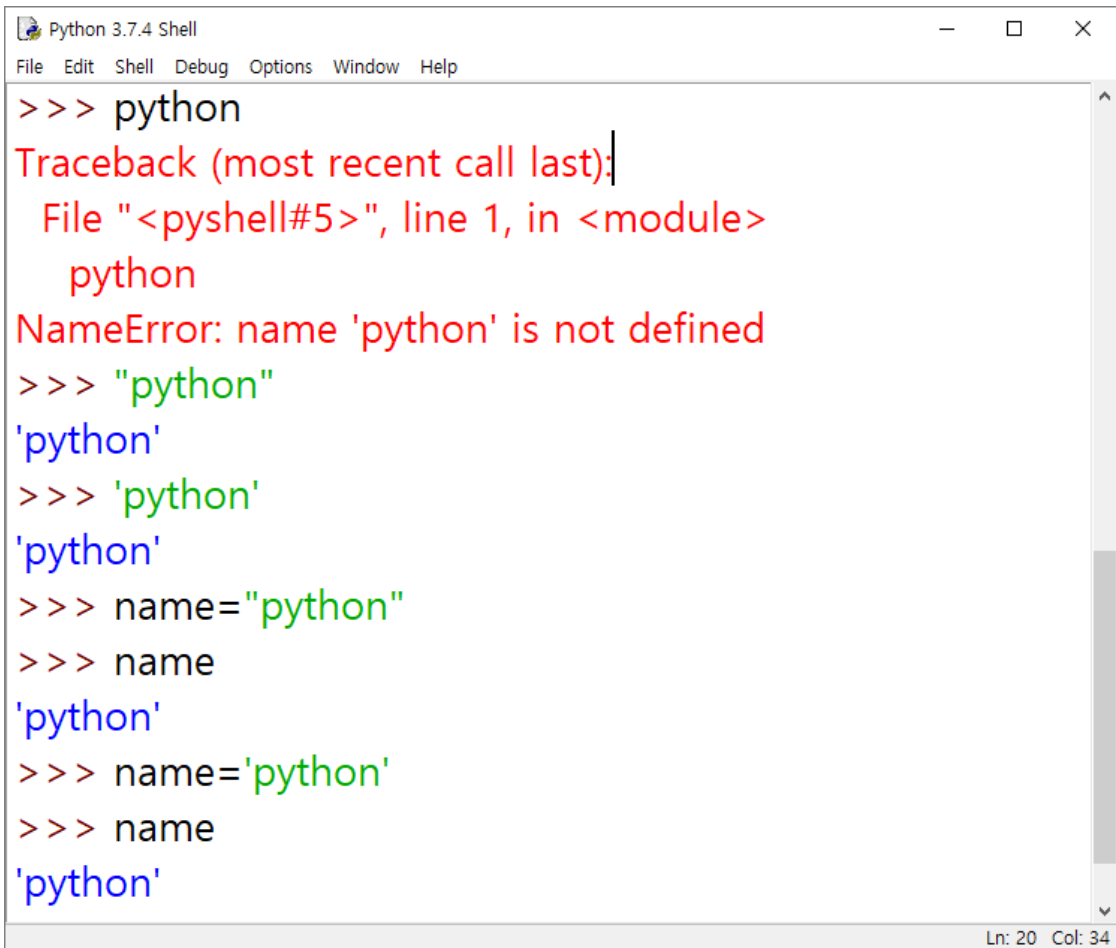
- 문자열

- "AB" , "python"
- BLOOD= "AB"

```
x = 20
print("x =", x)
x = 10.3
print("x =", x)
x = "python"
print("x =", x)
```

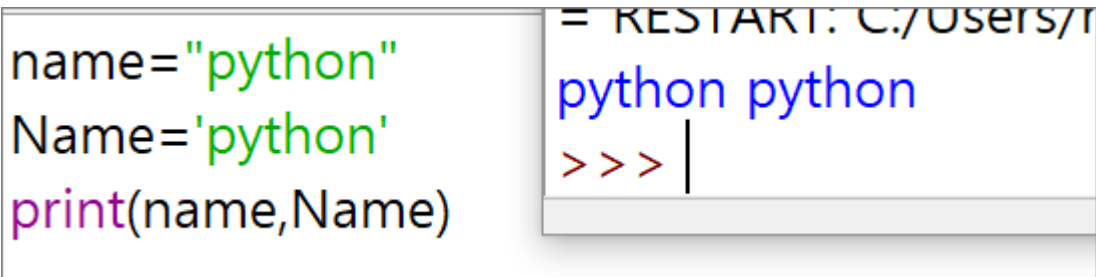
```
= RESTART: C:/Users/hjjin/
x = 20
x = 10.3
x = python
>>> |
```

- 큰따옴표 “ ”
- 작은 따옴표 ‘ ’

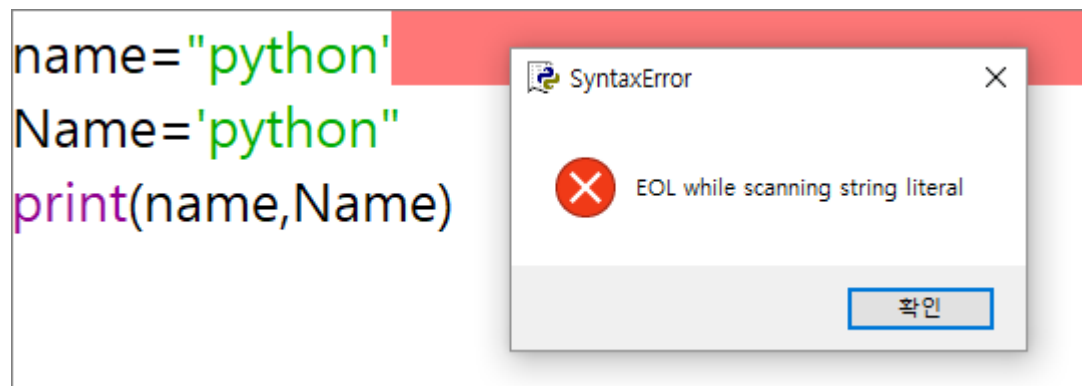


```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
>>> python
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    python
NameError: name 'python' is not defined
>>> "python"
'python'
>>> 'python'
'python'
>>> name="python"
>>> name
'python'
>>> name='python'
>>> name
'python'
```

Ln: 20 Col: 34



```
= RESTART: C:/Users/...
name="python"
Name='python'
python python
>>> |
print(name,Name)
```



```
name="python"
Name='python'
print(name,Name)
```

SyntaxError
EOL while scanning string literal
확인

```
number1=20  
number2=30  
print(number1+number2)
```

```
number1="20"  
number2="30"  
print(number1+number2)
```

```
number1=20  
number2=30  
sum=number1+number2  
print(sum)
```

rams/Python/Python37-32/0915.p

50

2030

50

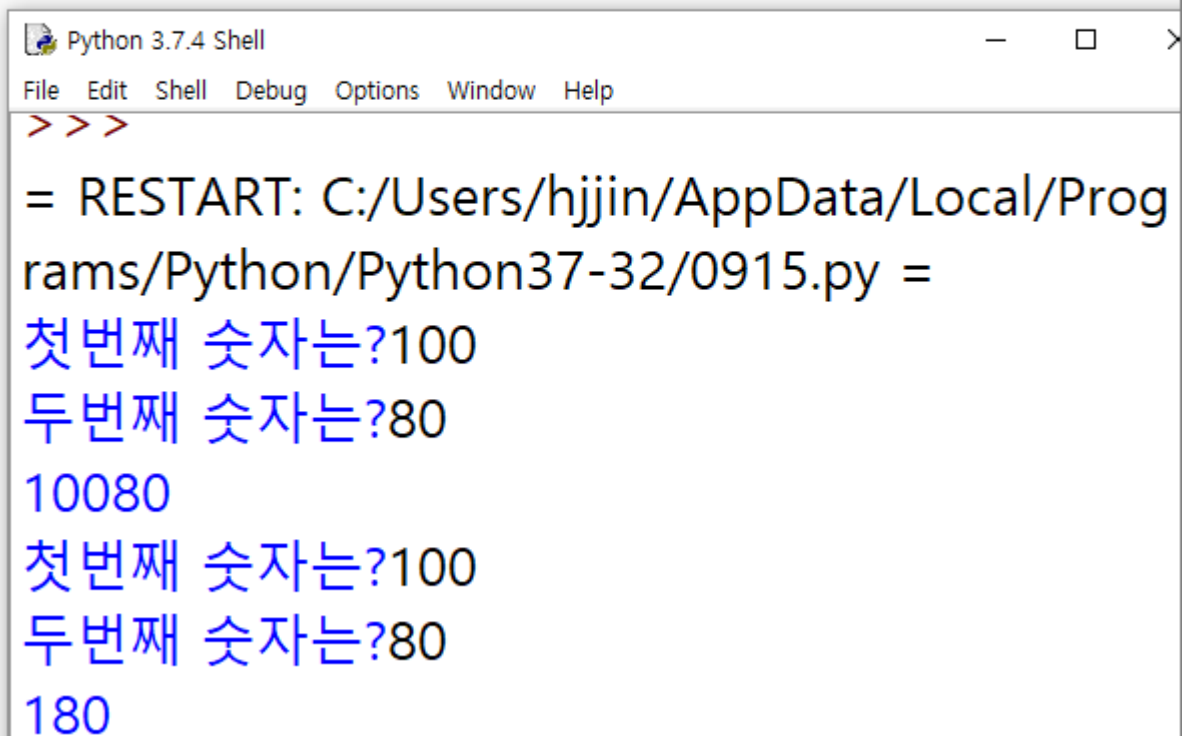
>>> |

#두수의 합

```
number1=input("첫번째 숫자는?") #input():키보드로 부터 값을 입력받아 문자열로 처리
number2=input("두번째 숫자는?")
sum=number1+number2
print(sum)
```

#두수의 합

```
number1=int(input("첫번째 숫자는?"))
number2=int(input("두번째 숫자는?"))
sum=number1+number2
print(sum)
```



```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
>>>
= RESTART: C:/Users/hjjin/AppData/Local/Programs/Python/Python37-32/0915.py =
첫번째 숫자는?100
두번째 숫자는?80
10080
첫번째 숫자는?100
두번째 숫자는?80
180
```

- `int()`: 문자열을 정수로 변환
- `float()`: 문자열을 실수로 변환

#두 실수의 합

```
number1=int(input("첫번째 숫자는?"))
number2=int(input("두번째 숫자는?"))
sum=number1+number2
print(sum)
```

```
Type "help", "copyright", "credits" or "license()" for more
>>>
= RESTART: C:/Users/hjjin/AppData/Local/Programs/Python/Pyth
첫번째 숫자는?30.5
Traceback (most recent call last):
  File "C:/Users/hjjin/AppData/Local/Programs/Python/Py
>
    number1=int(input("첫번째 숫자는?"))
ValueError: invalid literal for int() with base 10: '30.5'
```

#두 실수의 합

```
number1=float(input("첫번째 숫자는?"))
number2=float(input("두번째 숫자는?"))
sum=number1+number2
print(sum)
```

```
= RESTART: C:/Users
첫번째 숫자는?3.25
두번째 숫자는?1.35
4.6
>>> |
```

- **str() : 숫자를 문자열로 변환**

```
print("올해는"+2019+" 년입니다.")
```

```
> print("올해는"+2019+" 년입니다.")  
TypeError: can only concatenate str (not "int") to str
```

```
print("올해는"+str(2019)+" 년입니다.")
```

```
= RESTART: C:/Users/hjjin  
올해는2019 년입니다.  
>>> |
```

```
#print("현재온도는 "+22.8+" 도입니다.")
```

```
print("현재온도는 "+str(22.8)+" 도입니다.")
```

```
= RESTART: C:/Users/hjjin/AppDa  
현재온도는 22.8 도입니다.
```


■ 문자열 반복 : * 연산자

```
name = "문자열"
print(name*5)
```

= RESTART: C:/Users/hjjin/AppData/Local/Programs/Python/Python38-64/Python.exe
문자열문자열문자열문자열문자열

■ 문자 추출

■ 인덱스 : 0부터 시작

```
name = "jin hyejin"
print(name[0])
print(name[4])
print(name[9])
print(name[-1])
print(name[-3])
```

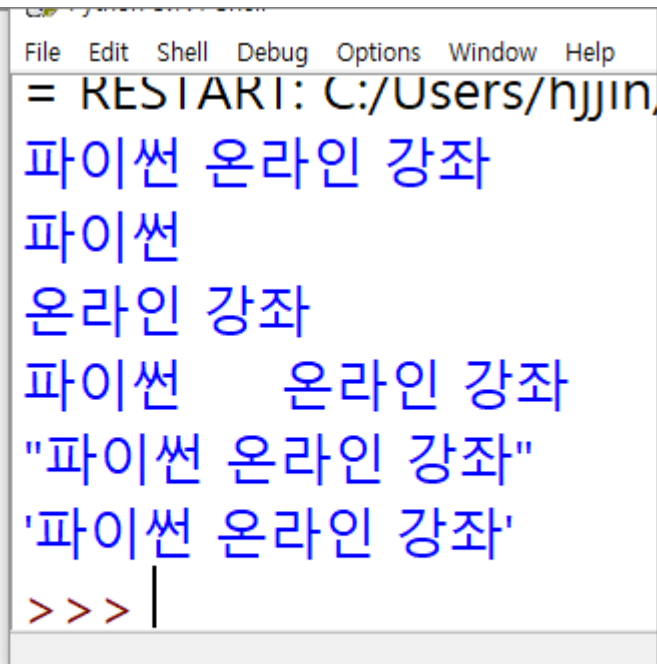
= RESTART: C:/Users/hjjin/AppData/Local/Programs/Python/Python38-64/Python.exe
j
h
n
n
j

```
name = "jin hyejin"
print(name[4:10])
```

= RESTART: C:/Users/hjjin/AppData/Local/Programs/Python/Python38-64/Python.exe
hyejin
>>> |

- `\n` : 줄 바꿈 문자
- `\t` : 탭 문자
- `\\` : 역 슬래시 출력
- `\"` : 큰 따옴표 출력
- `\'` : 작은 따옴표 출력

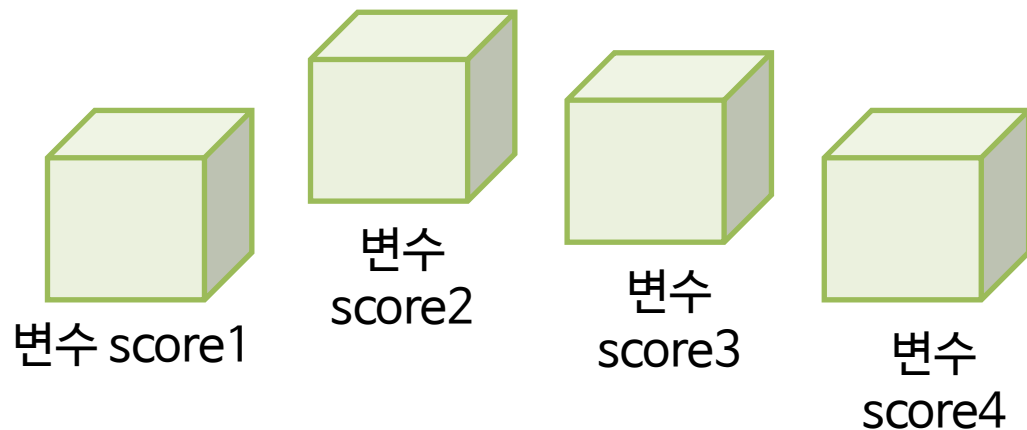
```
message = "파이썬 온라인 강좌"
print(message)
message = "파이썬\n온라인 강좌"
print(message)
message = "파이썬\t온라인 강좌"
print(message)
message = "\n"파이썬 온라인 강좌\n"
print(message)
message = "'파이썬 온라인 강좌'"
print(message)
```



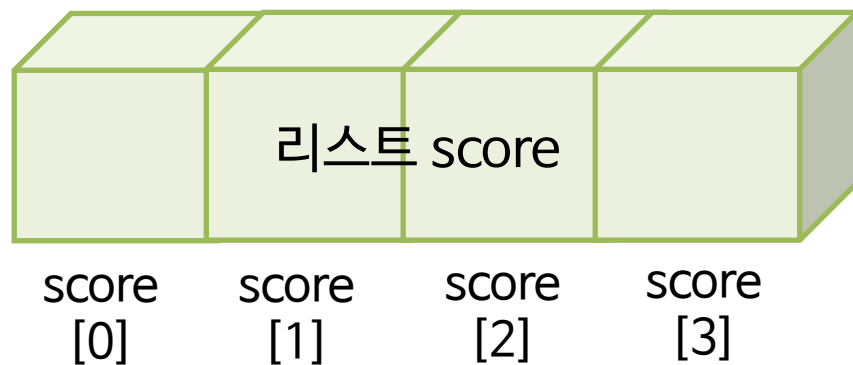
```
File Edit Shell Debug Options Window Help
= RESTART: C:/Users/hjjin/
파이썬 온라인 강좌
파이썬
온라인 강좌
파이썬      온라인 강좌
"파이썬 온라인 강좌"
'파이썬 온라인 강좌'
>>> |
```

- 어떤 경우에는 여러 개의 데이터를 하나로 묶어서 저장하는 것이 필요하다.

- 하나씩 변수로 사용



- 리스트로 정의



- 여러 개의 자료들을 모아서 하나의 묶음으로 저장하는 것
- 리스트 생성 : 항목들을 쉼표로 분리해 대괄호 안에 넣으면 된다.
- ex) 전공을 리스트로 생성
 - major = [“소프트웨어” , “경영” , “금융회계” , “건축”]

```
major = ["소프트웨어", "경영", "금융회계", "건축"]
print(major)
blood = ['A', 'B', 'O', 'AB']
print(blood)
number = [10, 20, 30, 40]
print(number)
```

```
= RESTART: C:/Users/hjjin/AppData/Lo
['소프트웨어', '경영', '금융회계', '건축']
['A', 'B', 'O', 'AB']
[10, 20, 30, 40]
>>> |
```

- 리스트는 정수, 문자열, 실수 등 서로 다른 자료형을 하나로 묶을 수 있다.

File Edit Format Run Options Window Help	File Edit Shell Debug Options Window Help
<pre>list=[] score=[90,89,60,70] major=["소프트웨어", "정보공학", "정보융합"] print(list) print(score) print(major)</pre>	<pre>= RESTART: C:/Users/hjjin/AppData/ [] [90, 89, 60, 70] ['소프트웨어', '정보공학', '정보융합'] >>> </pre>

File Edit Format Run Options Window	>>>
<pre>list=["진콩이",10,3.1] print(list)</pre>	<pre>= RESTART: C:/User ['진콩이', 10, 3.1] >>> </pre>

- 공백 리스트를 생성한 후에 코드로 리스트에 값을 추가할 수도 있다.

- ex) name=[]

- **append()함수**

- 리스트에 값을 추가

```
name=[]  
print(name)
```

```
name.append("진혜진")  
name.append("진선미")  
print(name)
```

```
>>>  
= RESTART: C:/Users/1  
[]  
['진혜진', '진선미']  
>>> |
```

```
File Edit Format Run Optio >>>  
list=[]  
list.append(7)  
list.append(17)  
list.append(27)  
list.append(37)  
print(list)  
= RESTART: C:/U  
[7, 17, 27, 37]  
>>> |
```

- 리스트 항목 접근하기

score=[90,89,60,70]	= RESTAP
print(score[0])	90
print(score[1])	89
print(score[2])	60
print(score[3])	70
print(score[-1])	70
print(score[-2])	60
print(score[-3])	89

■ 슬라이싱

- 리스트에서 한 번에 여러 개의 항목을 추출하는 기법
- 리스트에 접근할 때 콜론(:)을 사용해 범위를 지정

```
score=[90,89,60,70] = RESTART: |
print(score[0:2])    [90, 89]
print(score[2:4])    [60, 70]
>>> |
```

■ 콜론의 앞이나 뒤 숫자 생략

```
File Edit Format Run Options Win = RESTART: C:/Us
score=[90,89,60,70] [60, 70]
print(score[2:])    [90, 89, 60, 70]
print(score[:4])    >>> |
```


■ 리스트끼리 덧셈, 곱셈 연산

```
File Edit Format Run Options Window Help
a=[1,2,3]
b=[4,5,6]
print(a+b)
print(a*4)
name=["진혜진","진선미","진콩이"]
city=["서울","부산","대구"]
print(name+city)
print(name*2)
```

```
>>>
= RESTART: C:/Users/hjjin/AppData/Local/Programs/Python/Python38-64/Python.exe
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]
['진혜진', '진선미', '진콩이', '서울', '부산', '대구']
['진혜진', '진선미', '진콩이', '진혜진', '진선미', '진콩이']
>>>
```

■ 값을 변경하는 방법

```
a=[1,2,3]
b=[4,5,6]
name=["진혜진","진선미","진콩이"]
city=["서울","부산","대구"]

a[2]=30
name[1]="진달래"

print(a)
print(name)
```

```
>>>
>>>
= RESTART: C:/Users/hjjin/AppData/Local/Programs/Python/Python38-64/Python.exe
[1, 2, 30]
['진혜진', '진달래', '진콩이']
>>>
>>>
>>>
```

- 두 번째 값 인 20을 100과 200이라는 값 2개로 변경하는 방법

<pre>a=[10,20,30] print(a) a[1:2]=[100,200] print(a)</pre>	<pre>= RESTART: C:/Users/hjji [10, 20, 30] [10, 100, 200, 30] >>> >>> >>> </pre>
---	--

- a[1:2] 대신 그냥 a[1] 사용

<pre>a=[10,20,30] print(a) a[1]=[100,200] print(a)</pre>	<pre>= RESTART: C:/Users/hjji [10, 20, 30] [10, [100, 200], 30] >>> >>></pre>
---	---

- 두 번째인 a[1]의 항목 삭제
- 두 번째인 a[1]에서 네 번째인 a[3]까지 삭제

```
a=[10,20,30,40,50] [10]  
del(a[1])          >>>  
print(a)           = RESTART: C:/Use  
a[1:4]=[]          [10, 30, 40, 50]  
print(a)           [10]  
                   >>> |
```