

# Python



---

1강 – 파이썬에 대해, 자료형

# 파이썬 스터디 일정

---



- 1주 (3/16) : 자료형 (int, string, list), 제어문 (if, while, for)
- 2주 (3/23) : 자료형 (tuple, dictionary, set), 함수, 파일 읽기/쓰기
- 3주 (3/30) : GUI
- 4주 (4/6) : url 불러오기, regular expression
- 5주 (4/13) : News Feed Program 만들기 시작
- 6주 (4/20) ~ 이후 : 미정



# 왜 파이썬인가?

---

- 파이썬은 가독성이 좋다
  - Ex) `if 4 in [1,2,3,4]: print ("4가 있습니다")`
  - 직관적으로 무엇을 뜻하는지 알 수 있다.
- 파이썬은 문법이 쉬워 빠르게 배울 수 있다.
- 파이썬은 무료이지만 강력하다.
  - 프로그래머가 만들고자 하는 프로그램의 대부분을 파이썬으로 만들 수 있다. 물론 시스템 프로그래밍이나 하드웨어 제어와 같은 복잡하고 반복 연산이 많은 프로그램은 파이썬에 어울리지 않는다. 하지만 빠른 실행 속도를 필요로 하는 부분은 C로 만들어서 파이썬 프로그램에 포함시킬 수 있다.
- 파이썬은 개발 속도가 빠르다.
  - 웬만한 기능은 모듈이 준비되어 있어 불러오기만 하면 쉽게 구현 가능하다.

# 파이썬이란



- 파이썬은 1989년 귀도 반 로섬이라는 네덜란드 사람이 크리스마스 주에, 연구실이 닫혀 있어서 심심한 김에 만들었다고 한다. 1991년에 발표됐다.
- 인터프리터식, 객체지향적, 동적 타이핑 대화형 언어라고 한다.
  - 인터프리터는 프로그래밍 언어의 소스 코드를 바로 실행하는 컴퓨터 프로그램 또는 환경을 말한다. C, C++, JAVA, C# 등등
  - 객체지향적이란 컴퓨터 프로그래밍의 패러다임의 하나인데 컴퓨터 프로그램을 명령어의 목록으로 보는 시각에서 벗어나 여러 개의 독립된 단위, 즉 “객체”들의 모임으로 파악하고자 하는 것이라고 위키에서 말한다. 자세한 건 수업 때...
  - 동적 타이핑 언어는 자료형 타입 검사를 실행 타임에 하는 언어이다. 반대로, 정적 타이핑은 컴파일 타임에 자료형 타입 검사를 한다. 쉽게 말해, 동적 타이핑은 타입 선언이 필요하지 않고, 정적 타이핑은 자바, C/C++ 처럼 타입 선언이 필요하다.
- C언어로 구현되었다.
- 모든 문자열을 유니코드로 처리하기 때문에, 거의 모든 세계 언어를 처리할 수 있다.



# 파이썬 활용

(나중에 파이썬으로 무엇을 더 해보고 싶은지 생각해 볼 것)

- 시스템 유틸리티 제작
- GUI 프로그래밍
- C/C++와의 결합
- 웹 프로그래밍
  - Django는 파이썬을 이용한 웹 프레임워크다
- 데이터베이스 프로그래밍
  - Sybase, Infomix, Oracle, MySQL, PostgreSQL 등의 데이터베이스에 접근할 수 있게 해주는 도구들을 제공
  - Pickle 모듈
- 데이터 분석, 사물 인터넷
  - Pandas라는 모듈을 이용해 데이터 분석을 할 수 있다.
  - Raspberry Pi라는 리눅스 기반의 작은 기계를 제어할 수 있는 도구로 파이썬이 이용된다. 예를 들어 라즈베리파이에 연결된 모터를 작동시키거나 램프에 불이 들어오게 하는 일들을 파이썬으로 할 수 있다.



## Big Data



```
1 import socket
2
3 serversocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4 serversocket.bind(('localhost', 8089))
5 serversocket.listen(5) #become a server socket, maximum 5 connect
6
7 while True:
8     connection, address = serversocket.accept()
9     buf = connection.recv(64)
10    if len(buf)> 0:
11        print buf
12        break
```

Web Server

# 파이썬 시작하기

---



- [www.python.org](http://www.python.org) 에서 python 3 다운하기.
  - 다운로드시 기본 환경설정? 으로 해주세요.
  - Python 2 와 python 3가 있는데 python 3를 더 많이 쓴다고 함. 둘의 차이는 내장자료형의 내부적인 변화와 표준 라이브러리가 재배치된 점 등이 있다.
- 다음 자료들은 파이썬 기본 문법들이다. 전에 자신이 배운 언어와 어떤 점이 다른지 ‘코드’를 눈여겨봐라.

# 자료형 숫자형 (Numeric)

- 정수형은 int
- 실수형은 float
- 이외에도,
- 복소수는 complex
- 8진수는 숫자가 0o 또는 0O(숫자 0 + 알파벳 소문자 o 또는 대문자 O)로 시작하면 된다.
- Ex) a = 0o177
- 16진수는 0x로 시작하면 된다.
- Ex) a = 0x8ff
- b = 0xABC

```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.3 (default, Apr 11 2012, 07:15:24) [MSC v.1500 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> a=1452
>>> type(a)
<class 'int'>
>>> b=(-4587)
>>> type(b)
<class 'int'>
>>> c=0
>>> type(c)
<class 'int'>
>>> g=1.03
>>> type(g)
<class 'float'>
>>> h=-11.23
>>> type(h)
<class 'float'>
>>> i=.34
>>> type(i)
<class 'float'>
>>> j=2.12e-10
>>> type(j)
<class 'float'>
>>> k=5E220
>>> type(k)
<class 'float'>
>>>
```

Ln: 27 Col: 4

```
>>> a=3
>>> b=3
>>> a**b
27
>>> 7%3
1
>>> 3%7
3
>>> 7/4
1.75
>>> 7//4
1
... █
```

# 자료형 사칙연산

---

- 제곱을 나타내는 \*\*연산자
- 나눗셈 후 나머지를 나타내는 %연산자
- 나눗셈 후 소수점 아랫자리를 버리는 //연산자
- 파이썬 2.7에서  $\frac{7}{4}$ 를 실행할 경우, 값이 0이 나올 것이다. 파이썬 2에서는 정수 형끼리 나오면 정수형 값만 리턴하기 때문이다. 소수점으로 계산을 하고 싶으면  $a/(b*1.0)$  처럼 b를 강제로 실수형으로 변환해야 한다. 파이썬 3는 그럴 필요가 없다.



# 자료형 문자열 (string)

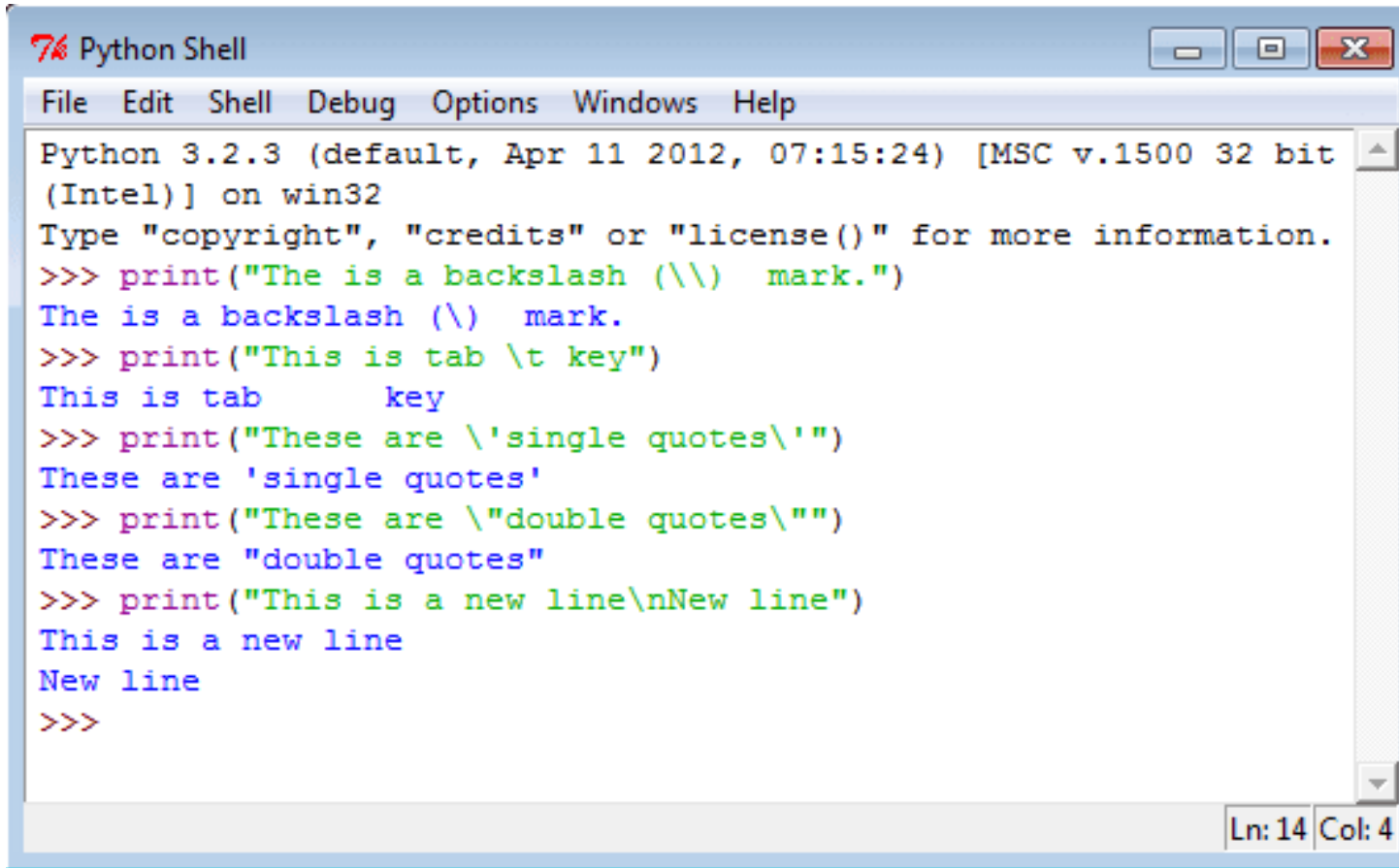
- 문자열은 ‘’, “”로 쓴다

```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.3 (default, Apr 11 2012, 07:15:24) [MSC v.1500 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> str1 = "String" #Strings start and end with double quotes
>>> print(str1)
String
>>> str2 = 'String' #Strings start and end with single quotes
>>> print(str2)
String
>>> str3 = "String' #Strings start with double quote and end wit
h single quote
SyntaxError: EOL while scanning string literal
>>> str1 = 'String" #Strings start with single quote and end wit
h double quote
SyntaxError: EOL while scanning string literal
SyntaxError: EOL while scanning string literal
>>> str2 = "Day's" #Single quote within double quotes
>>> print(str2)
Day's
>>> str2 = 'Day"s' #Double quote within single quotes
>>> print(str2)
Day"s
>>> |
```

Ln: 20 Col: 4

# 문자열 특수 기호

- `\n` 문자열 안에서 줄을 바꿀 때
- `\t` 탭 간격을 줄 때
- `\\` 문자 `\`를 그대로 표현할 때
- `\'` '를 그대로 표현할 때
- `\"` "를 그대로 표현할 때



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.3 (default, Apr 11 2012, 07:15:24) [MSC v.1500 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("The is a backslash (\\) mark.")
The is a backslash (\\) mark.
>>> print("This is tab \t key")
This is tab      key
>>> print("These are \'single quotes\'")
These are 'single quotes'
>>> print("These are \"double quotes\"")
These are "double quotes"
>>> print("This is a new line\nNew line")
This is a new line
New line
>>>
```

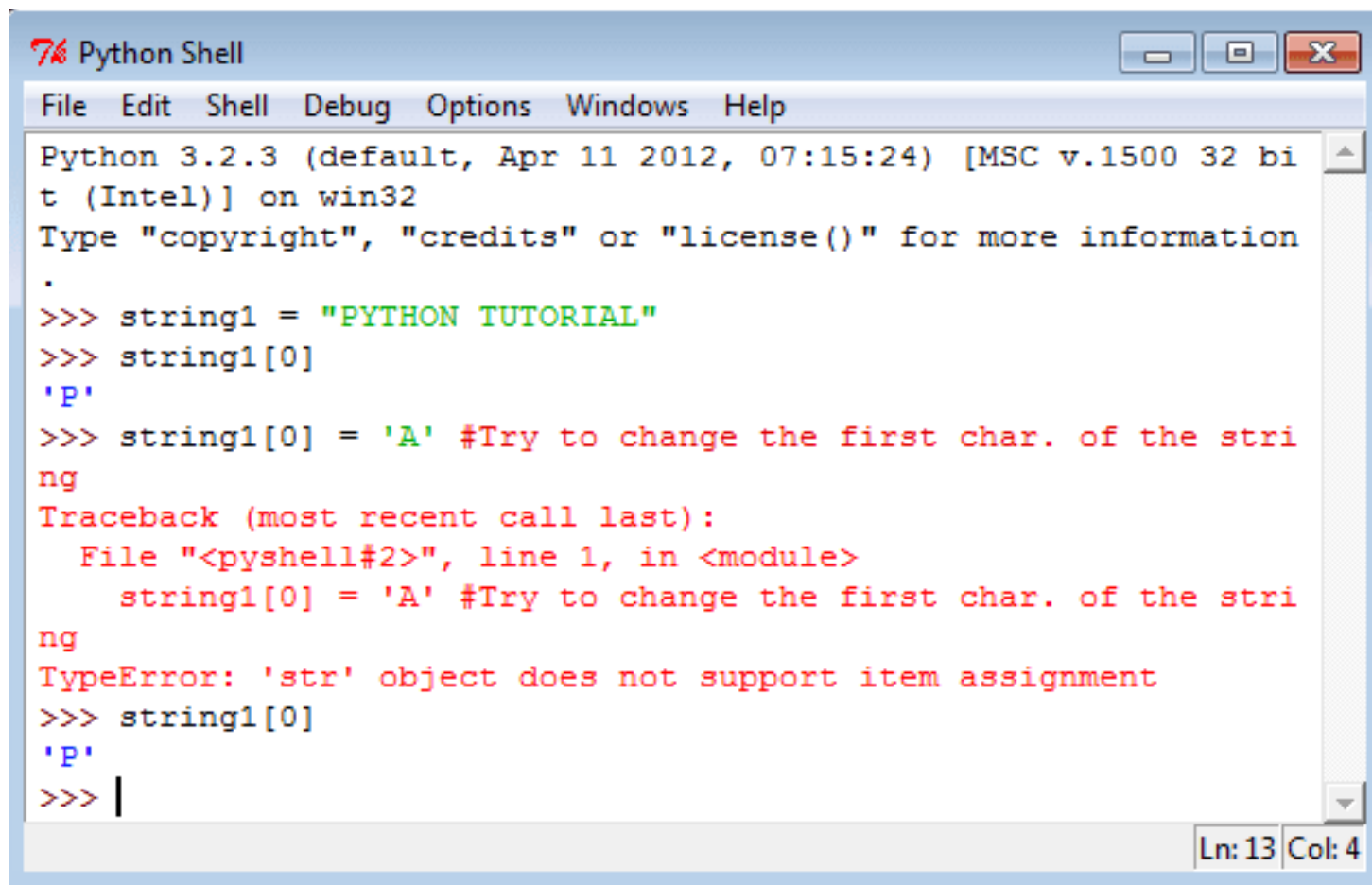
Ln: 14 Col: 4

# 문자열 인덱스

- 문자 '열' 이기에 글자 하나하나 인덱싱이 된다.
- 첫 글자는 0부터 시작.

```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.3 (default, Apr 11 2012, 07:15:24) [MSC v.1500 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> string1 = "PYTHON TUTORIAL"
>>> print(string1[0]) #Print first character
P
>>> print(string1[-15]) #Print first character
P
>>> print(string1[14]) #Print last character
L
>>> print(string1[-1]) #Print last character
L
>>> print(string1[4]) #Print 4th character
O
>>> print(string1[-11]) #Print 4th character
O
>>> print(string1[16]) #Out of index range
Traceback (most recent call last):
  File "<pyshell#7>", line 1, in <module>
    print(string1[16]) #Out of index range
IndexError: string index out of range
>>> |
```

Ln: 21 Col: 4



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.3 (default, Apr 11 2012, 07:15:24) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information
>>> string1 = "PYTHON TUTORIAL"
>>> string1[0]
'P'
>>> string1[0] = 'A' #Try to change the first char. of the string
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    string1[0] = 'A' #Try to change the first char. of the string
TypeError: 'str' object does not support item assignment
>>> string1[0]
'P'
>>> |
```

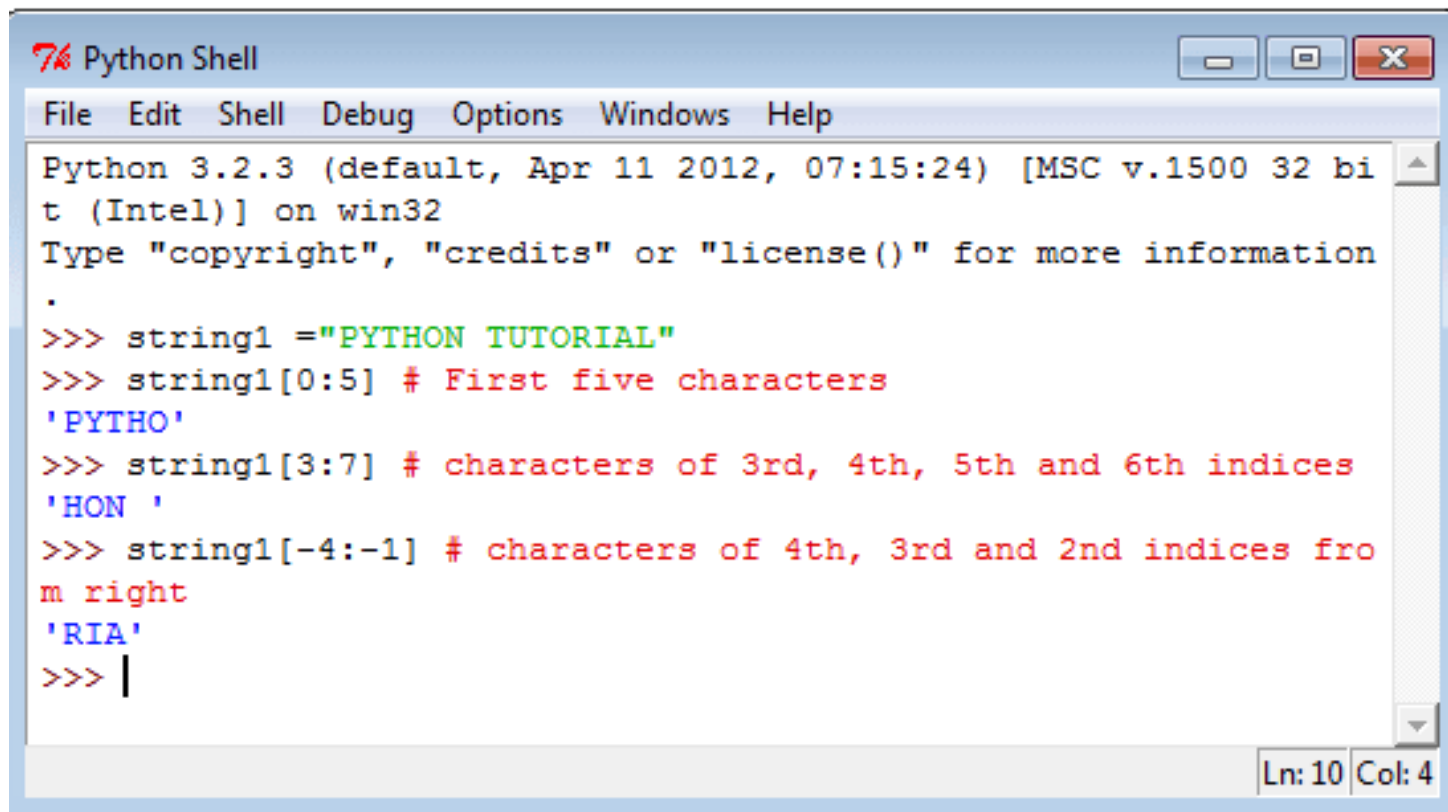
Ln: 13 Col: 4

# 문자열 특성

- 문자열이 생성되면, 요소를 바꿀 수는 없다. 여기서 요소란 글자 하나를 뜻한다.

# 문자열 슬라이싱

- 문자열을 글자 별로 나누어서 골라낼 수 있다.



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.3 (default, Apr 11 2012, 07:15:24) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information
.
>>> string1 = "PYTHON TUTORIAL"
>>> string1[0:5] # First five characters
'PYTHO'
>>> string1[3:7] # characters of 3rd, 4th, 5th and 6th indices
'HON '
>>> string1[-4:-1] # characters of 4th, 3rd and 2nd indices from right
'RIA'
>>> |
```

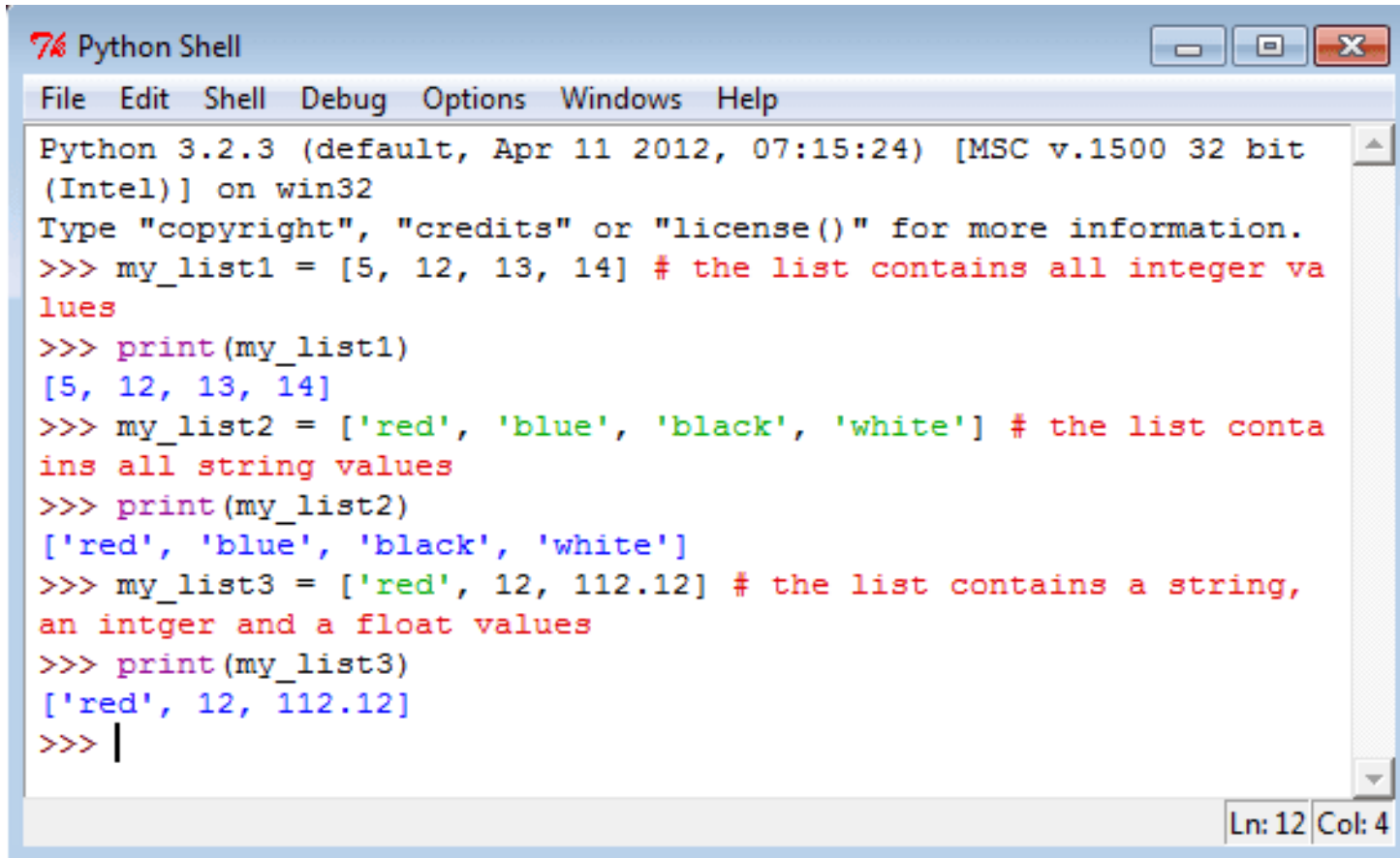
Ln: 10 Col: 4

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:0
1:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more
information.
>>> "I eat %d apples." % 3
'I eat 3 apples.'
>>> "I eat %s apples." % "five"
'I eat five apples.'
>>> number=3
>>> "I eat %d apples." % number
'I eat 3 apples.'
>>> number=10
>>> day="three"
>>>
>>> "I ate %d apples, so I was sick for %s days" % (
number, day)
'I ate 10 apples, so I was sick for three days'
>>> |
```

Ln: 15 Col: 4

# 문자열 포매팅

코드	설명
%s	문자열(string)
%c	문자 1개(character)
%d	정수(integer)
%f	부동 소수(floating-point)
%o	8진수
%x	16진수
%%	문자 %자체

A screenshot of a Python Shell window. The title bar says "Python Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area shows the following code:

```
Python 3.2.3 (default, Apr 11 2012, 07:15:24) [MSC v.1500 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> my_list1 = [5, 12, 13, 14] # the list contains all integer va
lues
>>> print(my_list1)
[5, 12, 13, 14]
>>> my_list2 = ['red', 'blue', 'black', 'white'] # the list conta
ins all string values
>>> print(my_list2)
['red', 'blue', 'black', 'white']
>>> my_list3 = ['red', 12, 112.12] # the list contains a string,
an intger and a float values
>>> print(my_list3)
['red', 12, 112.12]
>>> |
```

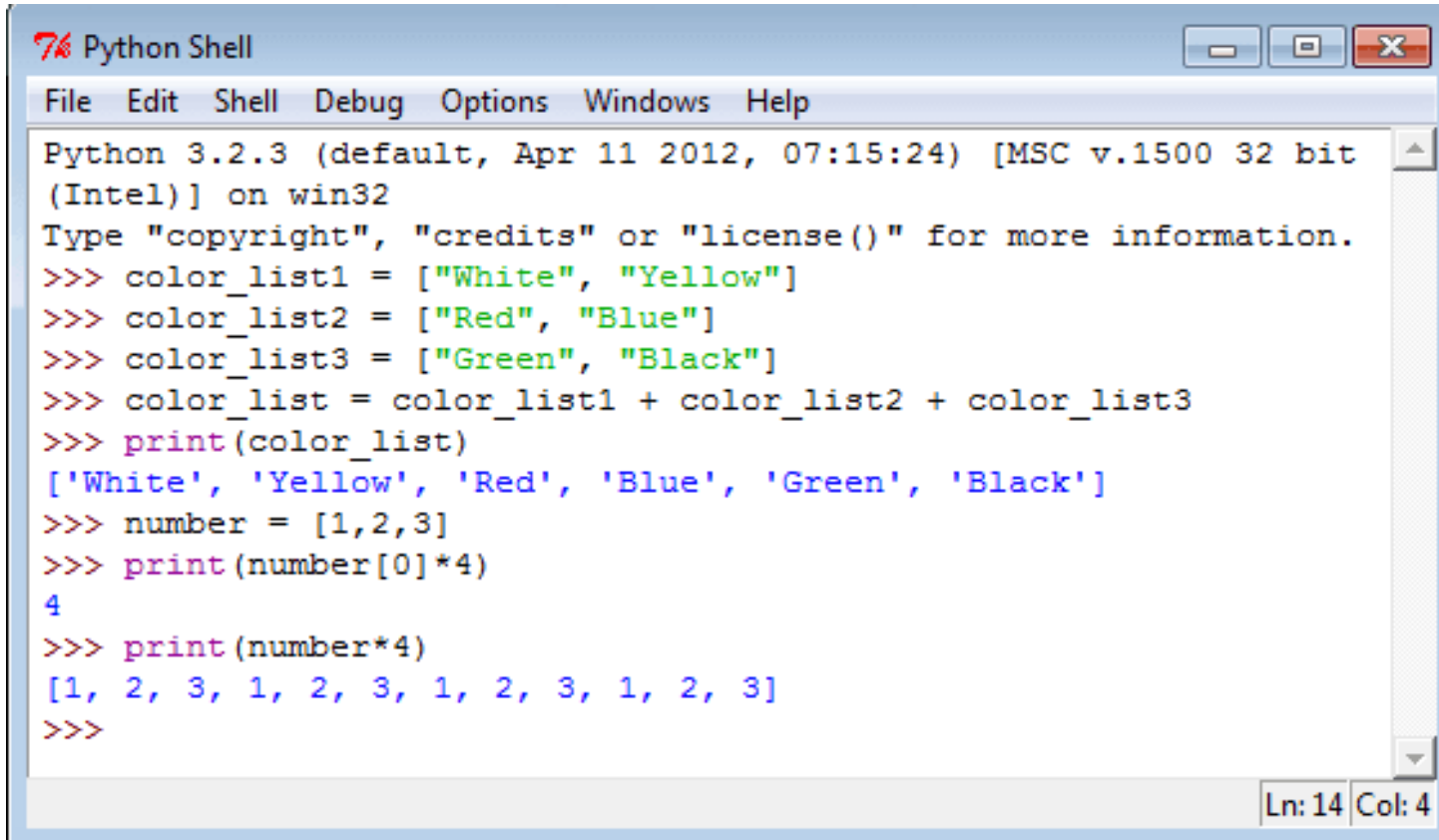
The status bar at the bottom right shows "Ln: 12 Col: 4".

# 리스트

- [] 안에 ,로 구분되는 요소들을 저장하는 타입이라는 게 배열과 비슷하지만 차이점은,
- 리스트엔 다른 타입의 요소가 들어가도 된다.
- 리스트는 크기를 따로 지정해주지 않아도 되고 요소를 넣을수록 크기가 저절로 늘어난다.

# 리스트의 특성

- 리스트는 문자열과 마찬가지로 인덱스와 슬라이싱으로 접근 가능하고 문법도 같다.
- 리스트는 생성 후, 요소들을 바꿀 수 있다. (문자열은 생성 후, 요소들을 바꿀 수 없다. 글자 하나하나 바꿀 수 없다.)
- 리스트는 +, \* 연산자를 사용할 수 있다. (문자열도 똑같은 게 가능하다)

A screenshot of a Python Shell window titled "Python Shell". The window has a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area shows the following code and output:

```
Python 3.2.3 (default, Apr 11 2012, 07:15:24) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> color_list1 = ["White", "Yellow"]
>>> color_list2 = ["Red", "Blue"]
>>> color_list3 = ["Green", "Black"]
>>> color_list = color_list1 + color_list2 + color_list3
>>> print(color_list)
['White', 'Yellow', 'Red', 'Blue', 'Green', 'Black']
>>> number = [1,2,3]
>>> print(number[0]*4)
4
>>> print(number*4)
[1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]
>>>
```

The status bar at the bottom right shows "Ln: 14 Col: 4".



# 리스트 함수들

## Add an item to the end of the list

```
view plain copy to clipboard print ?
01. >>> color_list=["Red", "Blue", "Green", "Black"]
02. >>> print(color_list)
03. ['Red', 'Blue', 'Green', 'Black']
04. >>> color_list.append("Yellow")
05. >>> print(color_list)
06. ['Red', 'Blue', 'Green', 'Black', 'Yellow']
07. >>>
```

## Insert an item at a given position

```
view plain copy to clipboard print ?
01. >>> color_list=["Red", "Blue", "Green", "Black"]
02. >>> print(color_list)
03. ['Red', 'Blue', 'Green', 'Black']
04. >>> color_list.insert
    (2, "White") #Insert an item at third position
05. >>> print(color_list)
06. ['Red', 'Blue', 'White', 'Green', 'Black']
07. >>>
```

## Remove the item at the given position in the list, and return it

```
view plain copy to clipboard print ?
01. >>> color_list=["Red", "Blue", "Green", "Black"]
02. >>> print(color_list)
03. ['Red', 'Blue', 'Green', 'Black']
04. >>> color_list.pop
    (2) # Remove second item and return it
05. 'Green'
06. >>> print(color_list)
07. ['Red', 'Blue', 'Black']
08. >>>
```

\* 차이점, pop은 index로 지우고, remove는 요소로 지움

## Remove an item from the list

```
view plain copy to clipboard print ?
01. >>> color_list=["Red", "Blue", "Green", "Black"]
02. >>> print(color_list)
03. ['Red', 'Blue', 'Green', 'Black']
04. >>> color_list.remove("Black")
05. >>> print(color_list)
06. ['Red', 'Blue', 'Green']
```

## Remove all items from the list.

```
view plain copy to clipboard print ?
01. >>> color_list=["Red", "Blue", "Green", "Black"]
02. >>> print(color_list)
03. ['Red', 'Blue', 'Green', 'Black']
04. >>> color_list.clear()
05. >>> print(color_list)
06. []
07. >>>
```

Sort the items of the list in place.

```
view plain copy to clipboard print ?
01. >>> color_list=["Red", "Blue", "Green", "Black"]
02. >>> print(color_list)
03. ['Red', 'Blue', 'Green', 'Black']
04. >>> color_list.sort(key=None, reverse=False)
05. >>> print(color_list)
06. ['Black', 'Blue', 'Green', 'Red']
07. >>>
```

\* sort() 도 됨!

# 리스트 함수들

Reverse the elements of the list in place

```
view plain copy to clipboard print ?
01. >>> color_list=["Red", "Blue", "Green", "Black"]
02. >>> print(color_list)
03. ['Red', 'Blue', 'Green', 'Black']
04. >>> color_list.reverse()
05. >>> print(color_list)
06. ['Black', 'Green', 'Blue', 'Red']
07. >>>
```

+ a.count(b) 함수 : a 안에 b가 몇 개지 세줌

Ex) a = [1,2,3,1]

a.count(1)

>>> 2

Return a shallow copy of the list.

```
view plain copy to clipboard print ?
01. >>> color_list=["Red", "Blue", "Green", "Black"]
02. >>> print(color_list)
03. ['Red', 'Blue', 'Green', 'Black']
04. >>> color_list.copy()
05. ['Red', 'Blue', 'Green', 'Black']
06. >>>
```

Search the Lists and find Elements

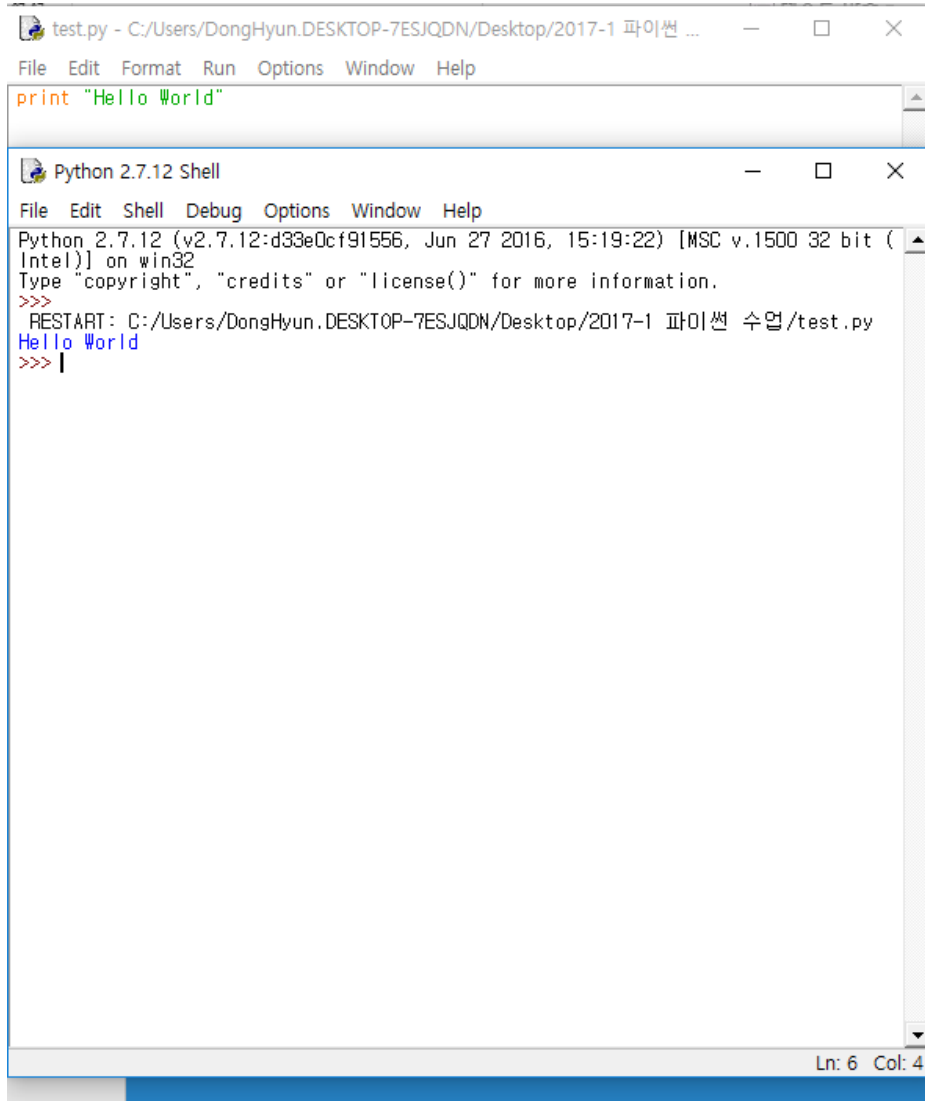
```
view plain copy to clipboard print ?
01. >>> color_list=["Red", "Blue", "Green", "Black"]
02. >>> print(color_list)
03. ['Red', 'Blue', 'Green', 'Black']
04. >>> color_list.index("Green")
05. 2
06. >>>
```

```
>>> a = input("Enter any string: ")
Enter any string: Python
>>> a
'Python'
>>> b = input("Enter any number: ")
Enter any number: 3
>>> type(b)
<class 'str'>
>>> c=int(b)
>>> type(c)
<class 'int'>
>>> |
```

# input(), type()

---

- 파이썬에서 input() 을 하면 str 타입으로 입력된다. 그래서, 입력된 값을 숫자로 바꾸려면, int() 로 캐스트 해주어야 한다.
- type() 은 자료형 타입을 확인해주는 함수이다.



The screenshot shows two windows from the Python IDLE environment. The top window is a file editor titled 'test.py - C:/Users/DongHyun.DESKTOP-7ESJQDN/Desktop/2017-1 파이썬 ...'. It contains a single line of Python code: `print "Hello World"`. The bottom window is the 'Python 2.7.12 Shell'. It displays the Python version and architecture information, followed by a restart command for the file 'test.py'. The output of the program, 'Hello World', is shown in blue text. The shell prompt is currently at the first line.

```
test.py - C:/Users/DongHyun.DESKTOP-7ESJQDN/Desktop/2017-1 파이썬 ...
File Edit Format Run Options Window Help
print "Hello World"

Python 2.7.12 Shell
File Edit Shell Debug Options Window Help
Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:19:22) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/DongHyun.DESKTOP-7ESJQDN/Desktop/2017-1 파이썬 수업/test.py
Hello World
>>> |
```

Ln: 6 Col: 4

# IDLE 사용하기

- IDLE 사용하기.
  - 파이썬을 다운로드 한 후, window 창에 IDLE을 검색해서 실행하면 Python Shell이 나온다.
  - Ctrl+n (File + New File) 을 누르면 에디터 창이 켜진다.
  - 코드를 작성하고 F5(Run + Run Module)을 누르고 Ctrl+n (저장) 하면 shell에 출력 값이 나온다.
- Shell 은 test용, IDLE은 실제 프로그램 작성용으로 사용함.

비교연산자	설명
$x < y$	x가 y보다 작다
$x > y$	x가 y보다 크다
* $x == y$	x와 y가 같다
$x != y$	x와 y가 같지 않다
$x >= y$	x가 y보다 크거나 같다
$x <= y$	x가 y보다 작거나 같다

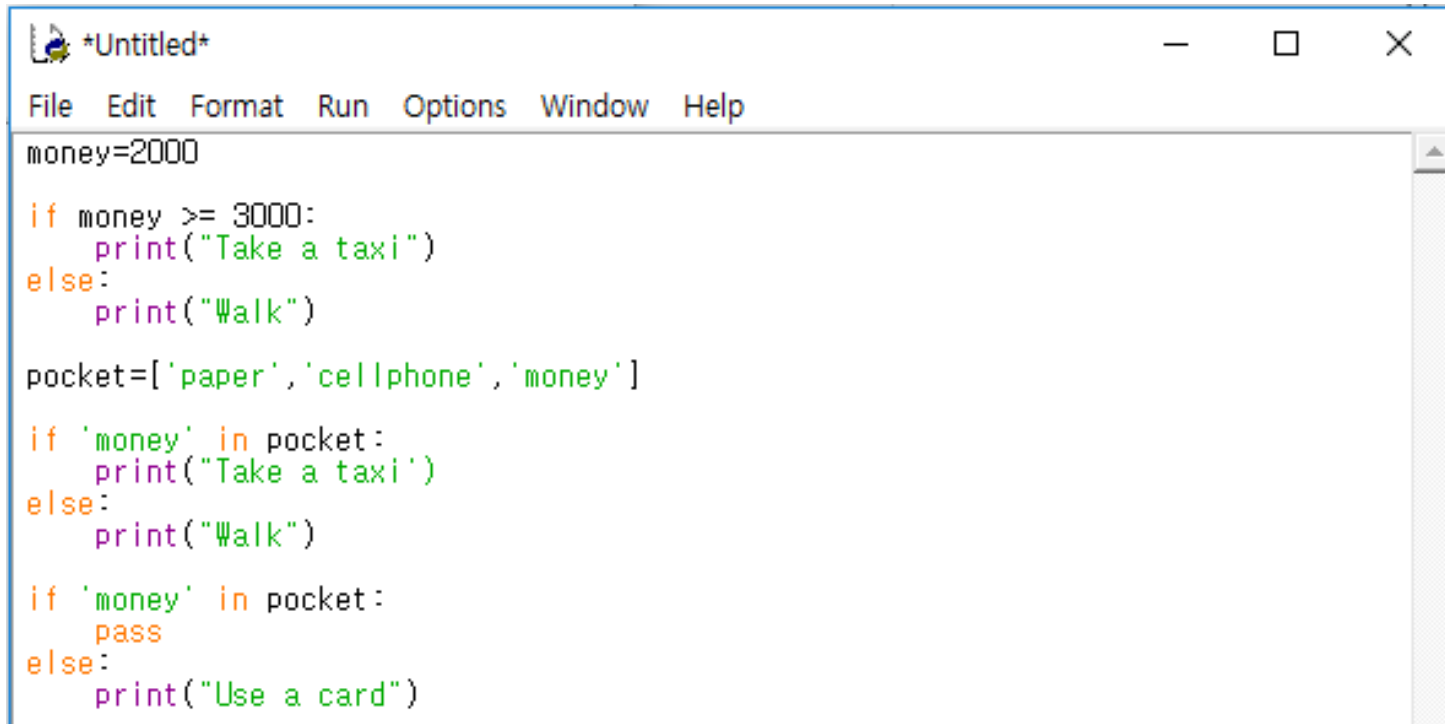
# 비교연산자

---

# 제어문 if

- 조건문 if 입니다.
  - 3번째 줄의 조건문은 '만약 money라는 변수가 3000원 보다 크다면, "Take a taxi"라는 문자열을 출력 하고, 이외의 경우에는 "Walk"라는 문자열을 출력시키라는 뜻이다.
  - 여러 조건문을 사용하고 싶으면 el
  - 파이썬에서 인덴트 (띄어쓰기??) 는 의미가 있다. if 밑 줄에 인덴트 없이 컴파일 해보면 에러가 날 것이다.

```
*Untitled*
File Edit Format Run Options Window Help
money=2000
if money >= 3000:
    print("Take a taxi")
else:
    print("Walk")
pocket=['paper','cellphone','money']
if 'money' in pocket:
    print("Take a taxi")
else:
    print("Walk")
if 'money' in pocket:
    pass
else:
    print("Use a card")
```



```
*Untitled*
File Edit Format Run Options Window Help
money=2000
if money >= 3000:
    print("Take a taxi")
else:
    print("Walk")
pocket=['paper', 'cellphone', 'money']
if 'money' in pocket:
    print("Take a taxi")
else:
    print("Walk")
if 'money' in pocket:
    pass
else:
    print("Use a card")
```

## 제어문 if

- if 는 파이썬에서 이렇게 쓰입니다.
- 단, 파이썬에서 인덴트 (여백??) 는 의미가 있다. if 밑 줄에 인덴트 없이 컴파일 해보면 에러가 날 것이다.

```
>>> treeHit=0
>>> while treeHit<10:
    treeHit=treeHit+1
    print("Hit the tree %d times." %treeHit)
    if treeHit==10:
        print("Tree false down...")
```

```
Hit the tree 1 times.
Hit the tree 2 times.
Hit the tree 3 times.
Hit the tree 4 times.
Hit the tree 5 times.
Hit the tree 6 times.
Hit the tree 7 times.
Hit the tree 8 times.
Hit the tree 9 times.
Hit the tree 10 times.
Tree false down
```

```
>>> coffee=1
>>> coffee=10
>>> money=300
>>> while money:
    print("Give coffee")
    coffee=coffee-1
    print("remaining cofee is %d." % coffee)
    if not coffee:
        print("sold out.")
        break
```

# 제어문 while

- while은 파이썬에서 이렇게 쓰입니다.
- while문의 조건이 참일 경우 반복!



```

1. >>> test_list = ['one', 'two', 'three']
   >>> for i in test_list:
   >>>     print(i)

one
two
three
2. >>> a=[(1,2),(3,4),(5,6)]
   >>> for (first,last) in a:
   >>>     print(first+last)

3
7
11
3. >>> num=0
   >>> for i in range(1,11):
   >>>     num += i

   >>> print(num)
55
4. >>> marks=[90,25,67,45,80]
   >>> for number in range(len(marks)):
   >>>     if marks[number]<60: continue ## 점수가 60점 미만이면 맨 처음으로 돌아간다.
   >>>     print ("Number %d student passed the test. Congratulations!" % (number+1))

Number 1 student passed the test. Congratulations!
Number 3 student passed the test. Congratulations!
Number 5 student passed the test. Congratulations!

```

# 제어문 for

- 1. 전형적인 for 문
- 2. 다양한 for 문의 사용
- 3. range(x,y) 는 x부터 y-1까지의 숫자를 갖는 객체이다.
- 4. len(x)은 x 리스트의 요소 개수를 돌려주는 함수이다.
- 4. continue 는 반복문의 맨 처음으로 돌아가게 하는 제어문이다.

# 과제 – 1주차

---

1. 다음 문자열을 밑의 특정 형식으로 print 하세요. (문자열 프린팅)

```
Twinkle, twinkle, little star,  
    How I wonder what you are!  
        Up above the world so high,  
        Like a diamond in the sky.  
Twinkle, twinkle, little star,  
    How I wonder what you are
```

# 과제 - 1주차

---

2. 사용자의 이름과 성을 입력하고 역순으로 print 하는 프로그램을 작성하세요. (문자열 포매팅, 연산)

Input

Input first name : DongHyun

input last name : Kim

output

Your name is Kim DongHyun

# 과제 – 1주차

---

3. 사용자의 파일 이름을 입력받고 그 확장자를 print 하는 프로그램을 만드세요. (문자열슬라이싱)

*Sample filename* : abc.java

*Output* : java

4. a="101112131415" 에 있는 모든 수의 합을 구하라. (자료형 변환)

5. 다음 리스트에서 첫 번째와 마지막 색을 print 하는 프로그램을 작성하세요

color\_list = ["Red","Green","White" ,"Black"]

# 과제 - 1주차

---

a=[1,2,3,4,5,6,7,8,9]

b=[2,4,8,10]

c=[1,3,5,7,9,10]

6. a 리스트 끝에 10을 추가, b 리스트 4,8 사이에 6 추가, c 리스트에 10 제거 하세요. a,b,c 리스트에 총 2와 3이 몇 개 씩 들어 있는지 세보세요. (리스트 함수)

7. a=[38,5,6,72,1,99], b=[3,9,4,2,70,1] 두 리스트를 합치고 그 리스트를 큰 순서대로 배치하라. 단, sort() 함수를 사용할 것. sort(key=None,reverse=False) 이 함수의 뜻은, key에는 데이터 타입을 쓰고, reverse에는 True면 역순으로, False면 역순을 안한다는 뜻이다. 또한, key값에 int와 str을 넣었을 때 무엇이 다른지 확인해 보세요.

# 과제 – 1주차

---

8. 쇼핑을 갈 때 지불해야 할 금액을 계산 해야합니다. 20달러 짜리 지폐를 가지고 다음을 사 십시오.

2 우유 상자 @ \$2.50

5 아이스크림 @ \$ 1.20

1 박카스 @ \$ 3.50

다음 목록의 음식을 사고 20달러를 내면 받을 거스름돈을 구하시오.

# 과제 - 1주차

\* Turtle 그래픽은 유아용 프로그래밍 교육 툴로 유명하다고 합니다. 정말 간단한 그래픽 툴이고 파이썬에 내장되었기 때문에 별다른 다운로드 없이 import turtle 을 통해 불러와서 사용할 수 있습니다. 다음 페이지에 있는 함수들을 이용해서 문제를 풀어보시기 바랍니다.

9. turtle 을 사용해서 별, 하트, 꽃을 그려보세요.

참고) 삼각형을 그리세요.

```
import turtle
```

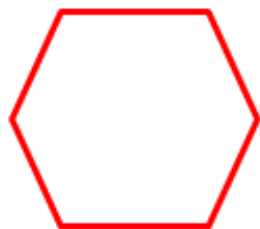
```
turtle.forward(100)
```

```
turtle.left(120)
```

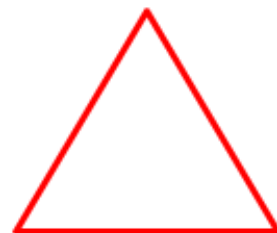
```
turtle.forward(100)
```

```
turtle.left(120)
```

```
turtle.forward(100)
```



e) An hexagon



f) An equilateral triangle



g) 5 squares of different sizes



h) 3 squares

## Some commonly-used turtle functions

<code>forward(<i>n</i>)</code>	move forward <i>n</i> pixels
<code>pendown()</code> , <code>penup()</code>	start/stop drawing
<code>goto(<i>x</i>, <i>y</i>)</code>	go to absolute coordinate ( <i>x</i> , <i>y</i> )
<code>left(<i>d</i>)</code> , <code>right(<i>d</i>)</code>	turn left/right by <i>d</i> degrees
<code>setheading(<i>a</i>)</code>	turn to face angle <i>a</i> (default: 0° is east)
<code>circle(<i>r</i>)</code>	walk in a circle of radius <i>r</i> pixels (by veering left)
<code>dot(<i>d</i>)</code>	draw a dot of diameter <i>d</i> pixels at the current position
<code>begin_fill()</code> ... <code>end_fill()</code>	fill the shape drawn between these two statements with the current colour
<code>color(<i>c</i>)</code>	set the pen colour to <i>c</i> ( <i>c</i> can be a string)
<code>hideturtle()</code> , <code>showturtle()</code>	make the turtle invisible/visible
<code>done()</code>	release the drawing window when finished

## 자주 쓰이는 turtle 함수들

- `pendown()` 을 하면 펜을 종이위에 놓고, `penup()`을 하면 종이에서 펜을 떼다고 생각하시면 됩니다.
- `goto`를 사용하면, 스크린의 절대값으로 펜을 이동합니다.
- `circle`은 선으로 그리는 원이고, `dot`은 속이 채워진 원입니다.
- `begin_fill()` 과 `end_fill()`을 이용해서 색깔을 넣어보세요.
- 인터넷에 찾아보면 사용법을 볼 수 있습니다.

<https://docs.python.org/2/library/turtle.html>



# 과제 - 1주차

---

10. turtle로 9번에서 그린 도형들을 추가해서 자유롭게 그림을 그려보세요. 그리고 수업 시간에 친구들에게 보여줍니다.

# 과제 – 2주차

---

1. 두 수를 입력 받아 짝수인지 홀수인지 알려주는 프로그램을 만드시오.

2.  $a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]$

$b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]$

두 리스트를 비교하고, 같은 수의 요소를 새로운 리스트에 넣어 출력하세요.

3. 몇 년에 100살이 되는지 알려주는 프로그램을 만드시오. 입력 값 : 이름, 나이

# 과제

---

4. sum\_list에 있는 모든 수를 더하는 프로그램을 만드시오.

```
sum_list= [1,2,-20]
```

5. a 리스트에 2자 이상이고 첫 글자와 마지막 글자가 같은 문자열이 몇 개인지 세는 프로그램을 만드시오. a = ['abc', 'xyz', 'aba', '1221']

6. dup\_list중복되는 수를 제거하는 프로그램을 만드시오. [어렵]

```
dup_list = [10,20,30,20,10,50,60,40,80,50,40]
```

# 과제

---

7. 1500과 2700 (둘 다 포함) 사이의 숫자 중 7로 나뉘지고, 5의 배수인 수를 찾는 프로그램을 만드시오.
8. 랜덤으로 생성되는 1과 9 사이의 숫자를 유저가 맞추면 이기고 틀리면 계속 시도할 수 있는 프로그램을 만드시오. [어렵?]  
다음 코드로 시작하시오.

```
import random  
target_num = random.randint(1,10)
```

9. 다음 과 같은 패턴의 \*를 출력하시오.

```
*  
* *  
* * *  
* * * *  
* * *  
* *  
*  
*
```

10. 온도를 Celsius 를 Fahrenheit 으로 또는 반대로 전환시키는 프로그램을 만드시오

(공식:  $c/5 = f-32/9$ )

(입력 방법 예 : 60F, 108C)

11. 0에서 50 사이의 피보나치 수열을 출력하시오 [어렵]

Ex) 0,1,1,2,3,5,8,13,21...

12. 입력 된 알파벳이 모음인지 자음인지 체크하는 프로그램을 만드시오. ( a, e, i, o, u 는 모음)

# 과제

---

13. 10000원 이상을 버는 순간 장사를 마치는 자판기를 만드세요. 메뉴: coke 1500원 juice 1200 원, energy drink 2000원

이 프로그램은 그날 무슨 음료를 몇 개 팔았는지 기록을 마지막에 출력합니다.

14. 10 ~ 15 까지의 각 숫자의 개수를 구해보자

10 = 1, 0

11 = 1, 1

12 = 1, 2

13 = 1, 3

14 = 1, 4

15 = 1, 5

그러므로 이 경우의 답은 0:1개, 1:7개, 2:1개, 3:1개, 4:1개, 5:1개

# 출처

---

- <https://www.wikipedia.org/>
- <http://www.w3resource.com/python/python-tutorial.php>
- 책 '점프 투 파이썬!' 이지스 퍼블리싱 출판사