리눅스 명령어

□ 리눅스 명령어 기초

Accessing Linux System

- Login
  - Remote access using PuTTy (or XRDP: GUI-based)
  - ID: pi
  - Password: raspberry

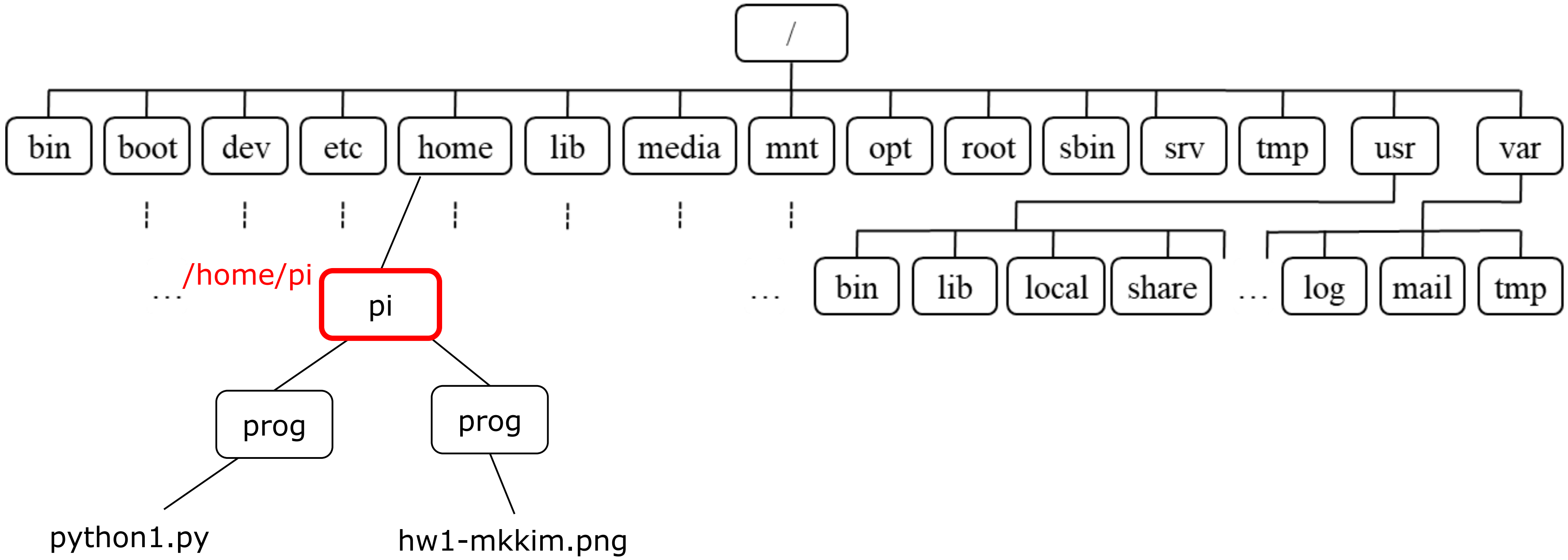
- Shell
  - Command interpreter
  - Shell prompt : % or $
    - pi@raspberrypi ~ $

Linux File System

- Types of files
  - directory: 폴더
    - contains the information about the other files
  - file
    - has data
    - text, ASCII
    - binary file : (e.g.) image, execution files
    - application specific files: word, power point, etc.
  - special file: 장치 파일
    - character device files: /dev/tty1, /dev/stdin, /dev/stdout, etc.
    - block device files: /dev/sda1
    - etc.

Linux File System

- File system
  - Hierarchical tree structure
  - Root: /



Linux Commands

- man
  - help 기능
  - example:
    - man ls
    - man man
- shutdown, restart
  - system shutdown and restart
  - example:
    - sudo shutdown –h now
    - sudo restart

Linux Commands

- ls
  - list files or directories in the current directory
  - examples:
    - ls
    - ls –l
- cd
  - change directory
  - examples:
    - cd /dev
    - cd ../../dev

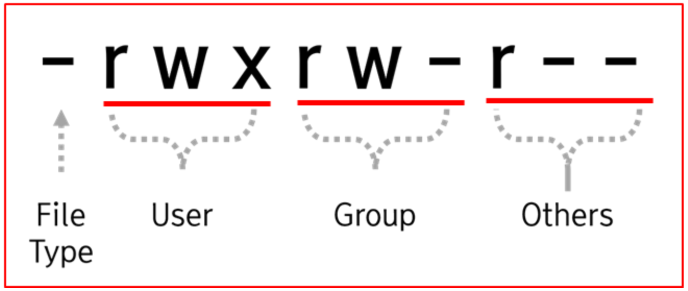
- File commands:
  - file copy: cp
    - cp a.txt b.txt
  - file delete: rm
    - rm b.txt
  - file move: mv
    - mv prog.c ../prog

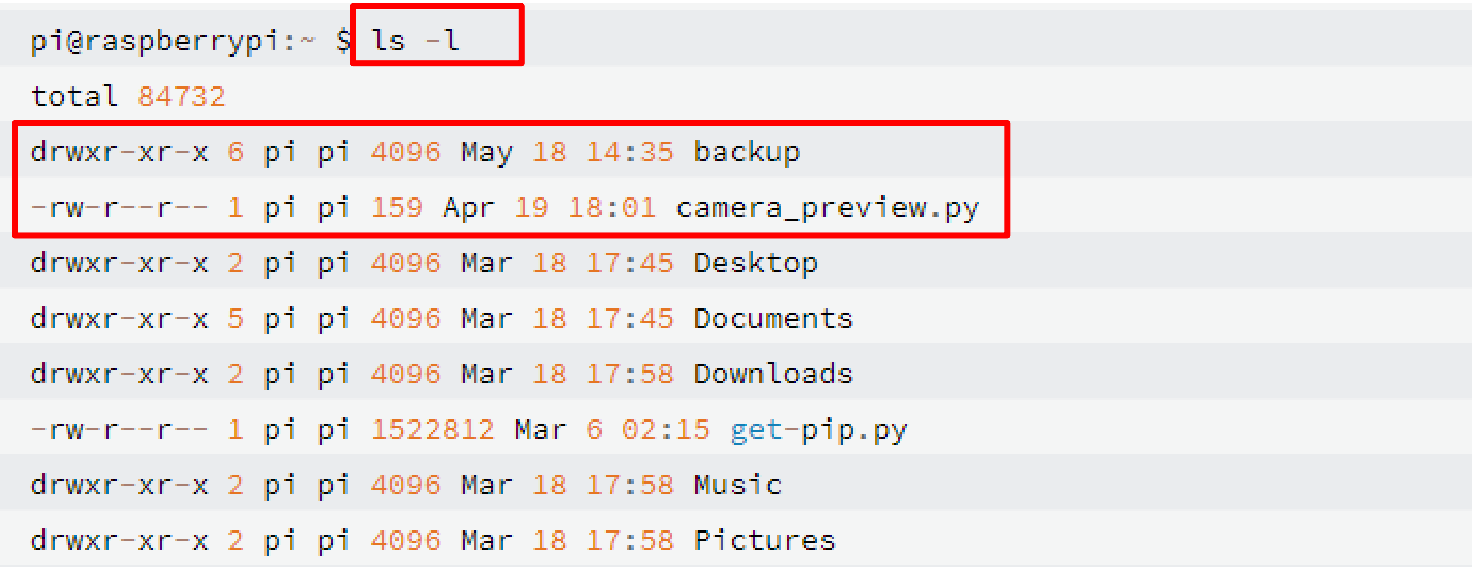
Linux Commands

- directory commands:
  - make directory: mkdir
    - mkdir prog
  - remove directory : rmdir
    - rmdir prog
    - rmdir –r prog

Linux Commands

- file access permission
  - separate permission for owner, group, others



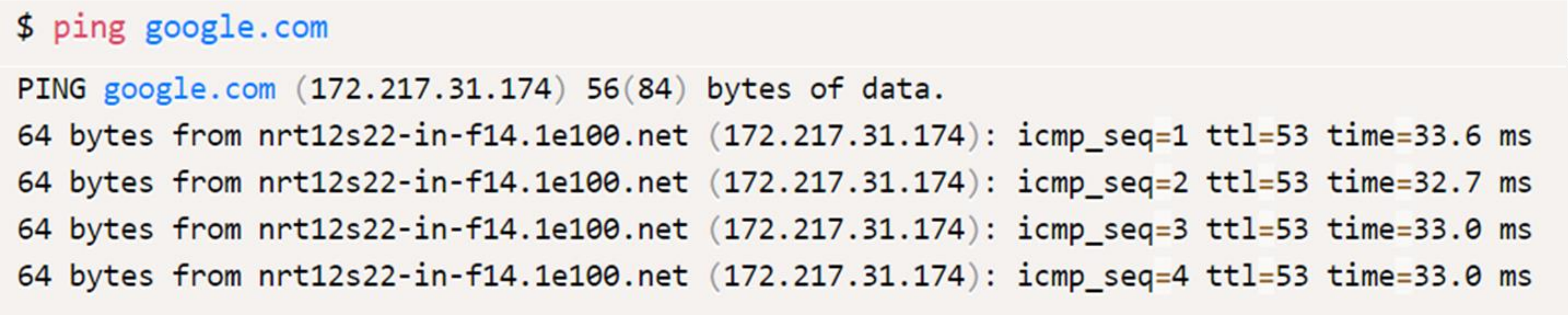


```
pi@raspberrypi:~ $ ls -l
total 84732
drwxr-xr-x 6 pi pi 4096 May 18 14:35 backup
-rw-r--r-- 1 pi pi 159 Apr 19 18:01 camera_preview.py
drwxr-xr-x 2 pi pi 4096 Mar 18 17:45 Desktop
drwxr-xr-x 5 pi pi 4096 Mar 18 17:45 Documents
drwxr-xr-x 2 pi pi 4096 Mar 18 17:58 Downloads
-rw-r--r-- 1 pi pi 1522812 Mar 6 02:15 get-pip.py
drwxr-xr-x 2 pi pi 4096 Mar 18 17:58 Music
drwxr-xr-x 2 pi pi 4096 Mar 18 17:58 Pictures
```

Linux Commands

Network related commands

- remote host accessability: ping

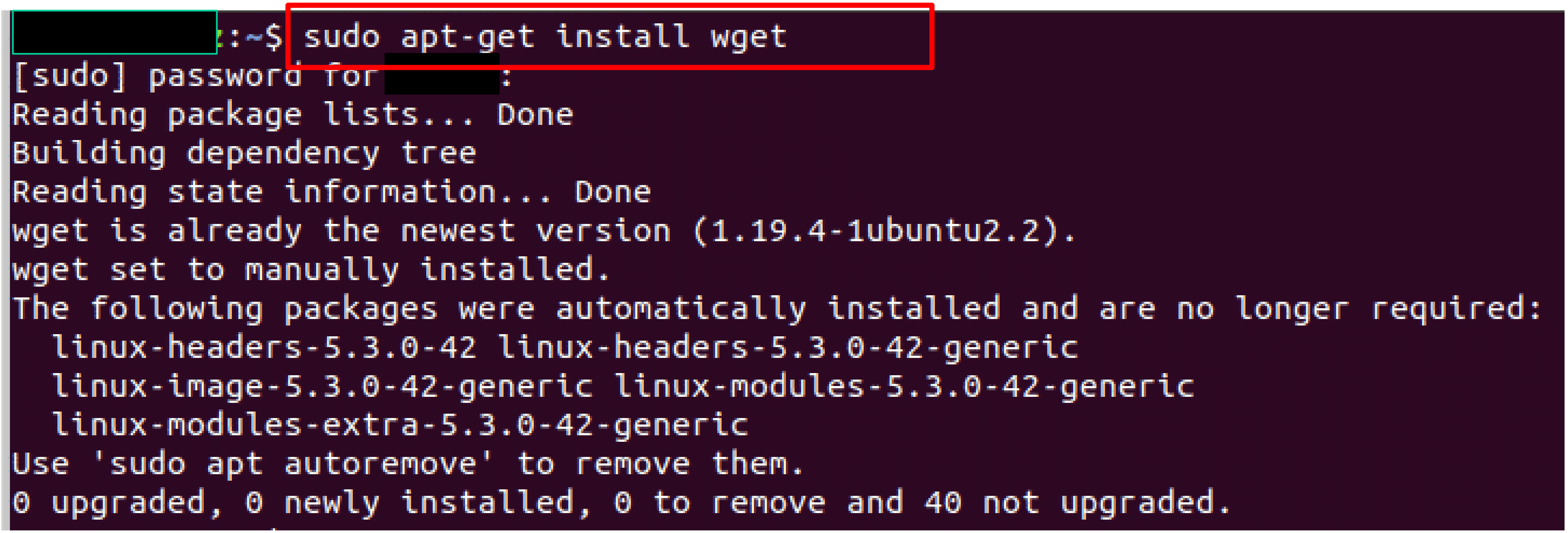


```
$ ping google.com

PING google.com (172.217.31.174) 56(84) bytes of data.
64 bytes from nrt12s22-in-f14.1e100.net (172.217.31.174): icmp_seq=1 ttl=53 time=33.6 ms
64 bytes from nrt12s22-in-f14.1e100.net (172.217.31.174): icmp_seq=2 ttl=53 time=32.7 ms
64 bytes from nrt12s22-in-f14.1e100.net (172.217.31.174): icmp_seq=3 ttl=53 time=33.0 ms
64 bytes from nrt12s22-in-f14.1e100.net (172.217.31.174): icmp_seq=4 ttl=53 time=33.0 ms
```

Linux Commands

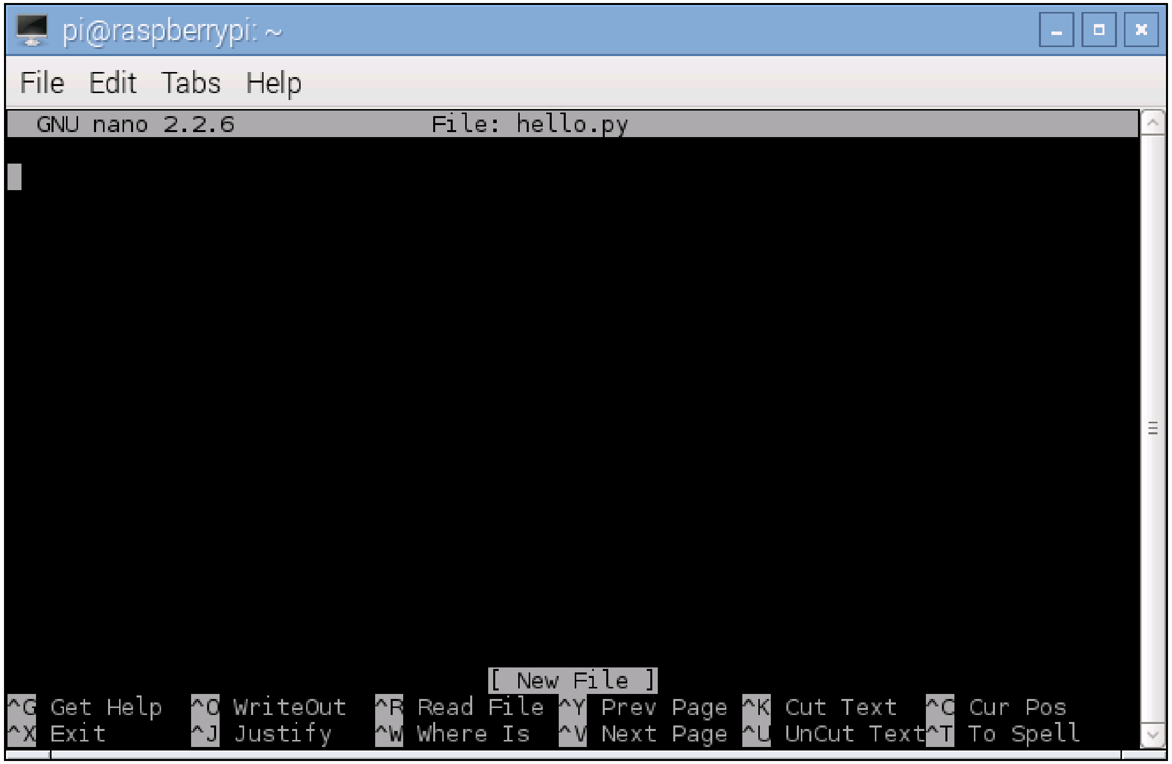
- Package management tool: apt-get
  - Install packet: sudo apt-get install <package>
  - Remove a packet: sudo apt-get remove <package>



```
[redacted]:~$ sudo apt-get install wget
[sudo] password for [redacted]:
Reading package lists... Done
Building dependency tree
Reading state information... Done
wget is already the newest version (1.19.4-1ubuntu2.2).
wget set to manually installed.
The following packages were automatically installed and are no longer required:
  linux-headers-5.3.0-42 linux-headers-5.3.0-42-generic
  linux-image-5.3.0-42-generic linux-modules-5.3.0-42-generic
  linux-modules-extra-5.3.0-42-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 40 not upgraded.
```

File Editing

- Editor : nano
  - similar to Windows 메모장



```
  GNU nano 2.2.6              File: hello.py




                              [ New File ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

File Editing

- Editor: nano

Command	Operation
CTRL+g	도움말 보기
CTRL+o	파일 저장
CTRL+x	편집기 빠져나오기
CTRL+a	현재 행의 처음으로 이동
CTRL+e	현재 행의 마지막으로 이동
CTRL+v	이전 페이지로 이동(page-up)
CTRL+y	다음 페이지로 이동(page-down)
CTRL+w	문자열 찾기
CTRL+d	현재 커서 위치의 한 글자 삭제
CTRL+k	한 줄 삭제
CTRL+u	마지막으로 삭제된 줄 복구 (undo)

Python Basics

- □ Python Programming Basics

Python

Python

- very easy to use and concise
- platform independent and object-oriented interpretive language
- dynamic typed language
- many packages for data analysis: NumPy(배열), SciPy(과학계산), matplotlib, pandas(데이터분석/통계), scikit-learn(기계학습), SymPy(기호계산)
- web site: www.python.org
  - version 2.7 and 3.3

Python Development Environment

□ Python3 install

```
$ sudo apt-get update
$ sudo apt-get install python3
```

□ Run Python

- Python shell

```
$ python3
…
>>> print (“Hello World”)
Hello World
>>> quit()
$
```

Python Development Environment

□ Run Python

- Python batch: “helloworld.py”

```
#!/usr/bin/env python3
print (“Hello World”)
```

스크립트를 실행할 interpreter를 지정

- Run

```
$ python3 helloworld.py
Hello World
$
```

Python Basics

□ Code block

- block structure with indentation

```
a = int(input())
if a > 0:
    print(“a is > 0”)
else:
    print(“a is not > 0”)
```

- comments:

```
# comment line

’’’ multi-lines
   comments
’’’
```

Python Basics

Variables

- no type declaration: type is determined when a value is assigned

```
a = 10
b = 3.14
x, y, z = 5, 10.3, 20
```

- Reserved words: 변수명으로 사용할 수 없음

and	as	assert	break	class	continue	def	del
elif	else	except	exec	finally	for	from	global
if	import	in	is	lambda	nonlocal	not	or
pass	print	raise	return	try	while	with	yield

Python Basics

□ String

- string creation: “ string “ or ‘ string ‘
- string concatenation: +
- string multiplication: *

```
>>> str1 = ‘Raspberry’
>>> len(str1)
9
>>> str2 = ‘ Pi’

>>> str3 = str1 + str2
>>> str3
‘Raspberry Pi’
>>> str3 * 2
‘Raspberry PiRaspberry Pi’
```

Python Basics

- String
  - string extraction by indexing or slicing

```
>>> str3
‘Raspberry Pi’
>>> str3[2]
s
>>> str3[-1]
i
>>> str3[0:4]
‘Rasp’
>>> str3[10:]
‘Pi’
>>> str3[:9]
‘Raspberry’
```

Python Basics

- Conditional statement: if

```
max = 0
if a > b:
    max = a
else:
    max = b
print(max)
```

Python Basics

- □ Repetition:
  - while:

```
while expression:
     statements
```

```
i, sum = 1, 0
while i <= 100:
    sum += i
    i +=1
print (“sum =“, sum)
```

Python Basics

- Repetition:
  - for

```
for var in list (or tuple, string):
    statements
```

```
listA = [‘kim’, ‘lee’, ‘park’]
for i in listA:
    print (i)
```

```
sum = 0
for i in range(1, 11):
    sum += i
print (“sum = “, sum)
```

Python Basics

□ Repetition:

- continue

```
for i in [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]:
    if i % 2 != 0:
        continue
    print (i, ‘^ 2 =‘, i*i), ‘\n’)
```

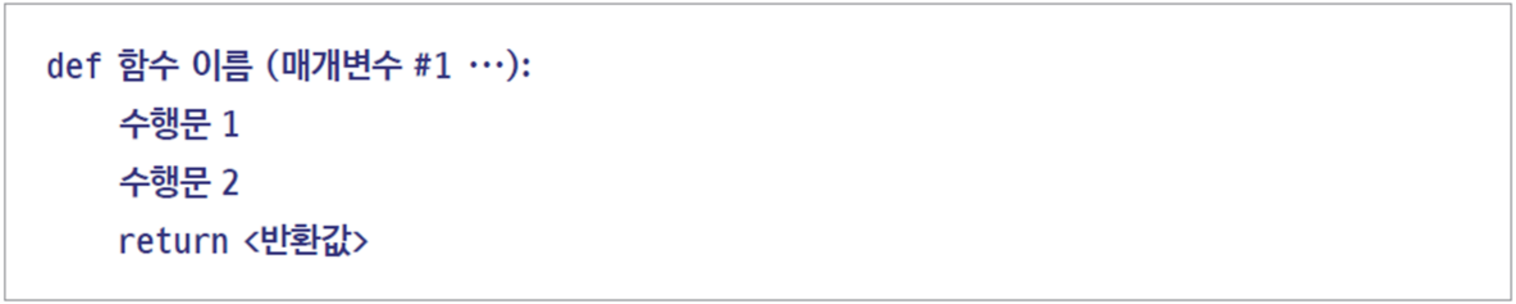
- break

```
i, sum = 1, 0
while True:
    print(i)
    sum += i
    i += 1
    if i == 100:
        break
print(‘sum =‘, sum)
```

Python Basics

☐ Function

- Definition: def
- function block is defined by indentation



```
def 함수 이름 (매개변수 #1 …):
    수행문 1
    수행문 2
    return <반환값>
```

```
def max(a,b):
    if a > b:
        result = a
    else:
        result = b
    return result
```

```
x = max(10, 20)
```

Python Basics

- Recursive function

```
def fact(n):
    if n <= 1:
        return 1
    else:
        return n * fact(n - 1)

result = fact(10)
```

Python Basics

□ Function

- global variable: defined outside of a function
- local variable: defined within a function
- access global variable in a function: global

```
m = 10
n = 20
def func():
    global n
    m = 100
    n = 200
    print (“in-func: m=“, m, “, n=”, n)

func()
print (“out-func: m=“, m, “, n=”, n)
```

Python Basics

□ File I/O

- fp.read(): read entire file and return as a string
- fp.readline(): read file line by line, and return one line
- fp.readlines(): read entire file and return as a list

```
f = open(“test.txt”, “r”)
data = f.readline()
while data:
    print data
    data = f.readline()
f.close()
```

Python Basics

□ File I/O

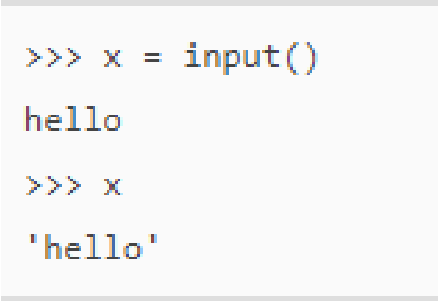
- write file

```
f = open(“out.txt”, “w”)
f.write(“This file name is “, “out.txt \n”)
f.close()
```

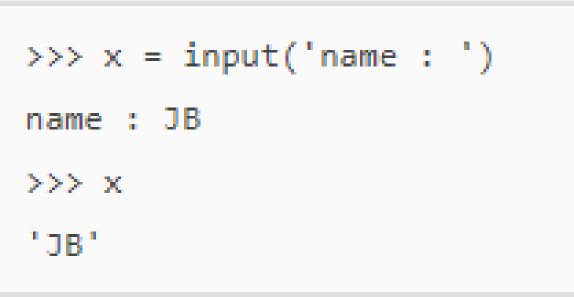
```
f = open("output.txt", "a")
f.write(“This file name is “, “out.txt \n”)
f.close()
```

Python Basics

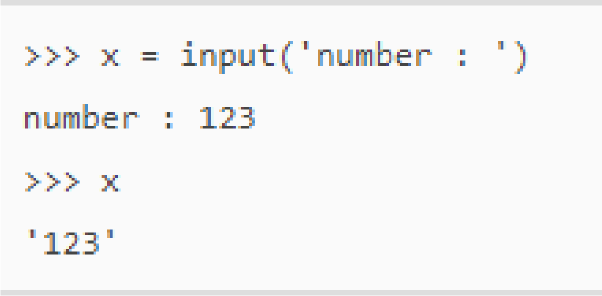
- Standard input / output:
  - input() : 키보드로 부터 입력
  - printf() : 모니터 화면으로 출력



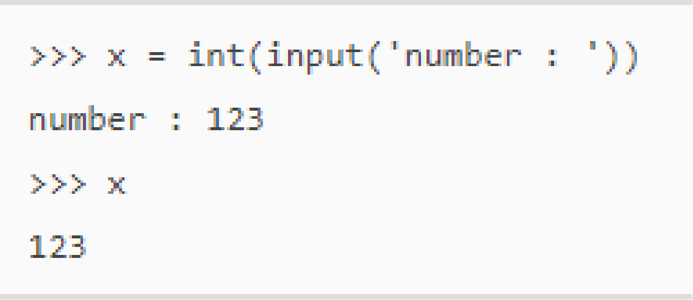
```
>>> x = input()
hello
>>> x
'hello'
```



```
>>> x = input('name : ')
name : JB
>>> x
'JB'
```



```
>>> x = input('number : ')
number : 123
>>> x
'123'
```



```
>>> x = int(input('number : '))
number : 123
>>> x
123
```

Python Basics

Module

- a set of functions, classes, variables: mod_name.py
- divide the program into multiple logical units of modules
- using module: import
- from statement: when accessing a specific variable or function in a module

minimum.py

```
# mininum.py
def min(a,b):
    if a > b:
        return b
    else
        return a
```

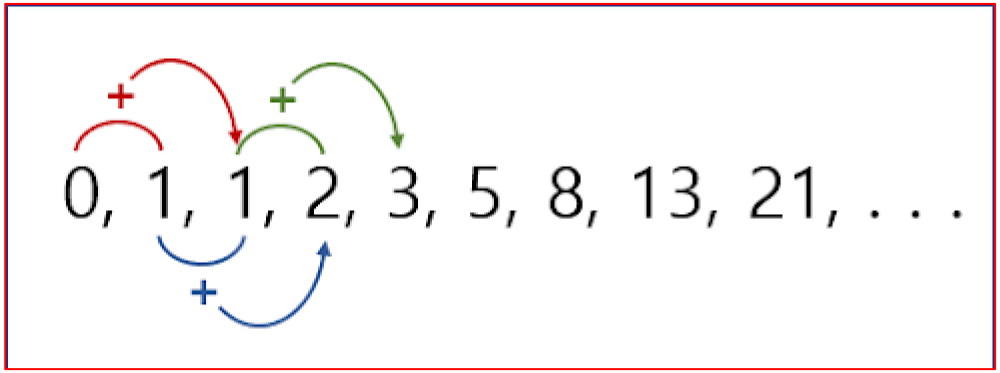
```
import minimum as Min
value = Min.min(10, 20)
```

```
from minimum import min
print(min(10, 20))
```

프로그래밍 실습

□ 피보나치 수열

$$f(n) = \begin{cases} 0 & if \quad n = 0 \\ 1 & if \quad n = 1 \\ f(n-1) + f(n-2) & if \quad n > 1 \end{cases}$$



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

프로그래밍 실습

- 피보나치 숫자를 구하는 파이썬 프로그램 작성
  - fibo1.py: 정수 (k) 를 키보드로부터 입력 받아 k번째 피보나치 숫자를 출력하는 Python 프로그램
  - fibo2.py :
    - 정수 (n)을 인자로 받아 n번째 피보나치 숫자를 return 하는 함수 fibo(n) 을 작성
    - 정수 (k) 를 입력 받아 키보드로부터 k번째 피보나치 숫자를 출력하는 Python 프로그램: fibo(n) 함수 이용
  - 위 각각의 파이썬 프로그램을 작성: nano 에디터 이용