1-1-1 절대값 구하기

In [1]:

```
abs(3)
```

Out[1]:

3

In [2]:

```
abs(-3)
```

Out[2]:

3

1-1-2 dir

- 객체가 자체적으로 가지고 있는 변수나 함수를 보여준다.

In [3]:

```
dir([1,2,3])
```

Out[3]:

```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__iadd__',
 '__imul__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__reversed__',
 '__rmul__',
 '__setattr__',
 '__setitem__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'append',
 'clear',
 'copy',
 'count',
 'extend',
 'index',
 'insert',
 'pop',
 'remove',
 'reverse',
 'sort']
```

1-1-3 enumerate

- 자료형을 입력받아 인덱스 값을 포함하는 enumerate객체를 돌려준다.

In [4]:

```
for i, name in enumerate(['body', 'foo', 'bar']):
    print(i, name)
```

```
0 body
1 foo
2 bar
```

실습 1.

- 내가 좋아하는 과일을 리스트로 만들고 이를 i, name으로 받아보자

1-1-4 len 함수

- len(s)는 입력값 s의 길이(요소의 전체 개수)를 돌려준다.

In [5]:

```
len('python')
```

Out[5]:

```
6
```

range는 for문과 함께 자주 사용하는 함수이다.

In [6]:

```
list(range(5))
```

Out[6]:

```
[0, 1, 2, 3, 4]
```

1-2 라이브러리

sys : 파이썬 인터프리터가 제공하는 변수를 직접 제어

In [7]:

```
# argv_test.py
import sys
print(sys.argv)
```

```
['C:₩₩ProgramData₩₩Anaconda3₩₩lib₩₩site-packages₩₩ipykernel_launcher.py', '-f',
'C:₩₩Users₩₩user₩₩AppData₩₩Roaming₩₩jupyter₩₩runtime₩₩kernel-6d40df5c-b1f7-4fc8-9bcc
-018dba352d0a.json']
```

In [8]:

```
# 파이썬 모듈들이 저장되어 있는 위치
sys.path
```

Out[8]:

```
['C:₩₩Users₩₩user₩₩Documents₩₩인공지능기본₩₩01_파이썬기본',
 'C:₩₩ProgramData₩₩Anaconda3₩₩python38.zip',
 'C:₩₩ProgramData₩₩Anaconda3₩₩DLLs',
 'C:₩₩ProgramData₩₩Anaconda3₩₩lib',
 'C:₩₩ProgramData₩₩Anaconda3',
 '',
 'C:₩₩ProgramData₩₩Anaconda3₩₩lib₩₩site-packages',
 'C:₩₩ProgramData₩₩Anaconda3₩₩lib₩₩site-packages₩₩win32',
 'C:₩₩ProgramData₩₩Anaconda3₩₩lib₩₩site-packages₩₩win32₩₩lib',
 'C:₩₩ProgramData₩₩Anaconda3₩₩lib₩₩site-packages₩₩Pythonwin',
 'C:₩₩ProgramData₩₩Anaconda3₩₩lib₩₩site-packages₩₩IPython₩₩extensions',
 'C:₩₩Users₩₩user₩₩.ipython']
```

os

- 환경변수나 디렉터리, 파일등의 OS자원을 제어할 수 있게 해 주는 모듈

In [22]:

```
import os
os.environ
```

Out[22]:

```
environ{'ALLUSERSPROFILE': 'C:₩₩ProgramData',
        'APPDATA': 'C:₩₩Users₩₩user₩₩AppData₩₩Roaming',
        'COMMONPROGRAMFILES': 'C:₩₩Program Files₩₩Common Files',
        'COMMONPROGRAMFILES(X86)': 'C:₩₩Program Files (x86)₩₩Common Files',
        'COMMONPROGRAMW6432': 'C:₩₩Program Files₩₩Common Files',
        'COMPUTERNAME': 'DESKTOP-CL8VLRA',
        'COMSPEC': 'C:₩₩Windows₩₩system32₩₩cmd.exe',
        'DRIVERDATA': 'C:₩₩Windows₩₩System32₩₩Drivers₩₩DriverData',
        'FPS_BROWSER_APP_PROFILE_STRING': 'Internet Explorer',
        'FPS_BROWSER_USER_PROFILE_STRING': 'Default',
        'HOMEDRIVE': 'C:',
        'HOMEPATH': '₩₩Users₩₩user',
        'LOCALAPPDATA': 'C:₩₩Users₩₩user₩₩AppData₩₩Local',
        'LOGONSERVER': '₩₩₩₩DESKTOP-CL8VLRA',
        'NUMBER_OF_PROCESSORS': '8',
        'ONEDRIVE': 'C:₩₩Users₩₩user₩₩OneDrive',
        'OS': 'Windows_NT',
        'PATH': 'C:₩₩ProgramData₩₩Anaconda3;C:₩₩ProgramData₩₩Anaconda3₩₩Library₩₩min
gw-w64₩₩bin;C:₩₩ProgramData₩₩Anaconda3₩₩Library₩₩usr₩₩bin;C:₩₩ProgramData₩₩Anaconda3
₩₩Library₩₩bin;C:₩₩ProgramData₩₩Anaconda3₩₩Scripts;C:₩₩windows₩₩system32;C:₩₩window
s;C:₩₩windows₩₩System32₩₩₩₩bem;C:₩₩windows₩₩System32₩₩₩₩indowsPowerShell₩₩v1.0₩₩;C:₩₩w
indows₩₩System32₩₩OpenSSH₩₩;C:₩₩Program Files (x86)₩₩NVIDIA Corporation₩₩PhysX₩₩Comm
on;C:₩₩Program Files₩₩NVIDIA Corporation₩₩NVIDIA NvDLISR;C:₩₩Users₩₩user₩₩anaconda3;
C:₩₩Users₩₩user₩₩anaconda3₩₩Library;C:₩₩Users₩₩user₩₩anaconda3₩₩Scripts;C:₩₩Users₩₩u
ser₩₩AppData₩₩Local₩₩Microsoft₩₩WindowsApps;',
        'PATHEXT': '.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC',
        'PROCESSOR_ARCHITECTURE': 'AMD64',
        'PROCESSOR_IDENTIFIER': 'Intel64 Family 6 Model 126 Stepping 5, GenuineInte
l',
        'PROCESSOR_LEVEL': '6',
        'PROCESSOR_REVISION': '7e05',
        'PROGRAMDATA': 'C:₩₩ProgramData',
        'PROGRAMFILES': 'C:₩₩Program Files',
        'PROGRAMFILES(X86)': 'C:₩₩Program Files (x86)',
        'PROGRAMW6432': 'C:₩₩Program Files',
        'PSMODULEPATH': 'C:₩₩Program Files₩₩WindowsPowerShell₩₩Modules;C:₩₩Windows
₩₩system32₩₩WindowsPowerShell₩₩v1.0₩₩Modules',
        'PUBLIC': 'C:₩₩Users₩₩Public',
        'SESSIONNAME': 'Console',
        'SYSTEMDRIVE': 'C:',
        'SYSTEMROOT': 'C:₩₩Windows',
        'TEMP': 'C:₩₩Users₩₩user₩₩AppData₩₩Local₩₩Temp',
        'TMP': 'C:₩₩Users₩₩user₩₩AppData₩₩Local₩₩Temp',
        'USERDOMAIN': 'DESKTOP-CL8VLRA',
        'USERDOMAIN_ROAMINGPROFILE': 'DESKTOP-CL8VLRA',
        'USERNAME': 'user',
        'USERPROFILE': 'C:₩₩Users₩₩user',
        'WINDIR': 'C:₩₩Windows',
        'CONDA_PREFIX': 'C:₩₩ProgramData₩₩Anaconda3',
        'KERNEL_LAUNCH_TIMEOUT': '40',
        'JPY_INTERRUPT_EVENT': '1400',
        'IPY_INTERRUPT_EVENT': '1400',
        'JPY_PARENT_PID': '2344',
```

```
 'TERM': 'xterm-color',
 'CLICOLOR': '1',
 'PAGER': 'cat',
 'GIT_PAGER': 'cat',
 'MPLBACKEND': 'module://ipykernel.pylab.backend_inline'}
```

시스템 PATH 환경변수

In [23]:

```
os.environ['PATH']
```

Out[23]:

```
'C:₩₩ProgramData₩₩Anaconda3;C:₩₩ProgramData₩₩Anaconda3₩₩Library₩₩mingw-w64₩₩bin;
C:₩₩ProgramData₩₩Anaconda3₩₩Library₩₩usr₩₩bin;C:₩₩ProgramData₩₩Anaconda3₩₩Library₩₩b
in;C:₩₩ProgramData₩₩Anaconda3₩₩Scripts;C:₩₩windows₩₩system32;C:₩₩windows;C:₩₩windows
₩₩System32₩₩₩₩bem;C:₩₩windows₩₩System32₩₩₩₩indowsPowerShell₩₩v1.0₩₩;C:₩₩windows₩₩Syste
m32₩₩OpenSSH₩₩;C:₩₩Program Files (x86)₩₩NVIDIA Corporation₩₩PhysX₩₩Common;C:₩₩Progra
m Files₩₩NVIDIA Corporation₩₩NVIDIA NvDLISR;C:₩₩Users₩₩user₩₩anaconda3;C:₩₩Users₩₩us
er₩₩anaconda3₩₩Library;C:₩₩Users₩₩user₩₩anaconda3₩₩Scripts;C:₩₩Users₩₩user₩₩AppData
₩₩Local₩₩Microsoft₩₩WindowsApps;'
```

In [30]:

```
### 디렉터리 위치 변경하기
os.chdir("C:₩₩Users₩₩user₩₩Documents₩₩doit")
```

In [31]:

```
### 디렉터리 위치 변경하기
os.getcwd()
```

Out[31]:

```
'C:₩₩Users₩₩user₩₩Documents₩₩doit'
```

특정 디렉터리의 파일 확인

- 특정 디렉터리 안의 mod로시작하는 파일 확인

In [35]:

```
import glob
glob.glob("C:₩₩Users₩₩user₩₩Documents₩₩doit₩₩mod*")
```

Out[35]:

```
['C:₩₩Users₩₩user₩₩Documents₩₩doit₩₩mod1.py',
 'C:₩₩Users₩₩user₩₩Documents₩₩doit₩₩mod2.py',
 'C:₩₩Users₩₩user₩₩Documents₩₩doit₩₩modtest.py']
```

time

In [37]:

```
import time
time.time()   # UTC를 사용하여 현재 시간을 실수 형태로 1970년 1월 1일 기준
```

Out[37]:

```
1600988612.3222735
```

In [39]:

```
a = time.time()
time.localtime ( a )
```

Out[39]:

```
time.struct_time(tm_year=2020, tm_mon=9, tm_mday=25, tm_hour=8, tm_min=4, tm_sec=1,
tm_wday=4, tm_yday=269, tm_isdst=0)
```

In [40]:

```
### 3초후 출력
time.sleep(3)
print("3초후 출력")
```

```
3초후 출력
```

웹 브라우저 자동 실행

In [41]:

```
import webbrowser
```

In [44]:

```
webbrowser.open("http://google.com")
```

Out[44]:

```
True
```

In [ ]:

```

```