

파이썬 기초

# 11 파이썬 표준모듈



홍필두 교수  
(파이썬기초)



## 학습내용

- 01 수학
- 02 시간, 난수 발생
- 03 sys 모듈





## 학습목표

- 수학 모듈의 함수에 대해 이해하고 사용할 수 있다.
- 시간과 난수 모듈의 함수에 대하여 이해하고 사용할 수 있다.
- sys 모듈과 기타 built-in 모듈의 함수에 대해 이해하고 사용할 수 있다.



# 사전 학습

## “표준 모듈 활용”

프로그래밍으로 어떤 업무를 처리하기 위하여  
유용하게 사용되는 많은 함수 기능들이  
**파이썬에는 표준 모듈로 내장**되어 있습니다.

이 표준 모듈을 단순히 불러다 사용함으로써,  
수학, 시간, 시스템제어, 인터넷데이터 수집 등  
다양한 작업을 손쉽게 처리할 수 있습니다.

또한 다양한 기능을 가지고 있는 모듈이  
기존 파이썬 표준 모듈에 없다면 인터넷상에서 쉽게  
해당 기능이 있는 모듈을 불러와 파이썬 내부에  
설치할 수도 있습니다.

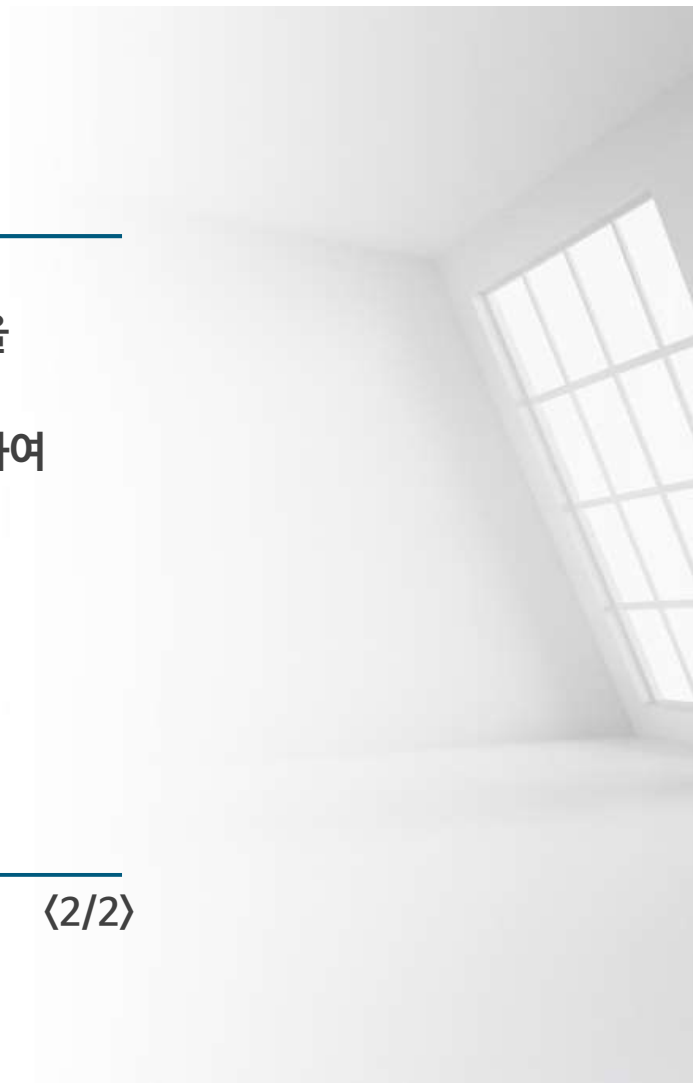
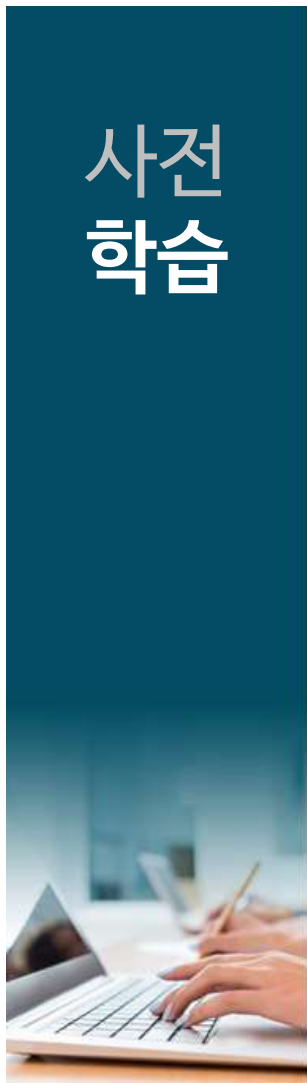
# 사전 학습

## “표준 모듈 활용”

---

이번 강의에서는 이러한 **내 · 외부의 모듈을 사용하는 방법에 관한 특징적인 몇 가지 사항을** 배웁니다.

여러분은 먼저 해당 내용에 대하여 검색을 통하여 미리 알아보고 조사해 본 후, 본 강의를 듣도록 합시다.



# 01

## 수학

- 1) 표준 라이브러리
- 2) math 모듈 사용 예제
- 3) matplotlib 맛보기
- 4) statistics 모듈 사용 예제

## 1) 표준 라이브러리



python을 설치하면 자주 사용되는 다양한 함수(def)가 표준 라이브러리에 포함되어 있음

### 라이브러리

- 함수를 모아 놓은 개념
- 기본적으로 설치되어 있는 라이브러리를 표준 라이브러리라고 부름
- python에서는 이 라이브러리를 module이라고 하며, 표준라이브러리를 built-in module로 명명함 (본 강의에서는 혼용하여 부를 예정임)

## 1) 표준 라이브러리

예

sin함수를 계산하기 위하여 math라는 표준 라이브러리를 프로그램 소스에서 호출하고 해당 math 라이브러리에 있는 sin함수를 가져다 사용하면 됨



표준 라이브러리 및 외부 라이브러리 활용에 대한 몇몇 예제를 본 강의에서 학습함

```
>>> import math
>>> math.sin(0)
0.0
>>> math.pi
3.141592653589793
>>> math.sin(math.pi/2.0)
1.0
```



## 1) 표준 라이브러리

표준 라이브러리를 사용하기 위하여  
다음과 같이 선언하여야 함

- import 라이브러리명
- from 라이브러리명 import 함수명
- import 라이브러리명 as 단축명

## 1) 표준 라이브러리

표준 라이브러리를 사용하기 위하여  
다음과 같이 선언하여야 함

- **import 라이브러리명**
- from 라이브러리명 import 함수명
- import 라이브러리명 as 단축명

```
>>> import math
>>> math.sin(math.pi/2.0)
1.0
```

## 1) 표준 라이브러리

표준 라이브러리를 사용하기 위하여  
다음과 같이 선언하여야 함

- import 라이브러리명
- from 라이브러리명 import 함수명
- import 라이브러리명 as 단축명

```
>>> from math import sin
>>> sin(3.141592/2)
0.99999999999999466
>>>
=RESTART: Shell =====
>>> from math import sin,pi #여러 함수 사용시
>>> sin(pi/2.0)
1.0
```

## 1) 표준 라이브러리

표준 라이브러리를 사용하기 위하여  
다음과 같이 선언하여야 함

- import 라이브러리명
- from 라이브러리명 import 함수명
- **import 라이브러리명 as 단축명**

```
>>> import math as m
>>> m.sin(m.pi/2.0)
1.0
```

# 1) 표준 라이브러리



표준 라이브러리에 대한 내용을 보기 위하여 help()를 활용

```
>>> import math
>>> help(math)
Help on built-in module math:

NAME
    math

DESCRIPTION
    This module is always available. It provides access to the
    mathematical functions defined by the C standard.

FUNCTIONS
    acos(x, /)
        Return the arc cosine (measured in radians) of x.

    acosh(x, /)
        Return the inverse hyperbolic cosine of x.
---이하 생략---
```

## 2) math 모듈 사용 예제



빅데이터, AI, 블록체인, 데이터 분석 등 다양한 분야에서 수학 및 통계학적인 지식이 필요함



일단 삼각함수  $\sin$ 에 대한 사항을 정리해 본 후, 몇 가지 실습을 수행함

- $\sin$  함수 값을 1도에서 360도까지 구해보자
- 각도를 측정하는 단위는 도(degree)와 라디안(radian:  $\pi = 3.141592\dots$ ,  $\pi$  기준으로 표시)가 있다
- $2\pi$ 는 360도이다
- 1도는 몇 라디안인가?  $\pi/180$
- Sin함수는  $-1 \sim 1$ 까지 나오며, 라디안으로 사용  
:  $\sin(0)=0$ ,  $\sin(\pi/2)=1$ ,  $\sin(\pi)=0$ ,  $\sin(\pi*3/4)=-1$

## 2) math 모듈 사용 예제

- 01 1도는  $\pi/180$  라디안 임
- 02 sin함수는 -1~1까지의 값을 가짐
- 03 양수화하기 위하여  $\sin+1$ 을 더하면 0~2까지의 값을 가짐
- 04 이 값에 50배를 곱하면 0~100까지의 값을 가짐
- 05 그 값만큼 띄어 쓴 다음 sin값을 찍어봄

## 2) math 모듈 사용 예제

```
from math import sin, pi

for i in range(360):
    sin_value=sin(i*pi/180)
    space_value=int( (1+sin_value)*50)
    print(" "*space_value, end="")
    print("[%f]"%sin_value)
```



## 2) math 모듈 사용 예제

```
>>>
== RESTART: C:/Users/iamhpd/AppData/Local/Programs/Python/Python37/mysin.py ==
[0.000000]
[0.017452]
[0.034899]
[0.052336]
[0.069756]
[0.087156]
[0.104528]
[0.121869]
[0.139173]
[0.156434]
[0.173648]
[0.190809]
[0.207912]
[0.224951]
[0.241922]
[0.258819]
[0.275637]
[0.292372]
[0.309017]
[0.325568]
[0.342020]
[0.358368]
[0.374607]
[0.390731]
[0.406737]
[0.422618]
[0.438371]
[0.453990]
[0.469472]
```

### 3) matplotlib 맛보기



matplotlib는 외부 모듈로 설치가 필요함



python 3.7이상에서는 pip(설치를 위한 도구)가 설치되어 있고, 그래프를 그리는 콘솔(아나콘다 등)도 설치되어 있기 때문에 편리함



아래 버전은 해당 도구를 다 설치하여야 함

### 3) matplotlib 맛보기

먼저 cmd창에서 pip install matplotlib 이라고 명령하여 matplotlib을 설치함

```

C:\Users\Wiamhpd-myhome>pip install matplotlib
Collecting matplotlib
  Downloading https://files.pythonhosted.org/packages/7e/ce/a4b83c538b48841d4d060569cc0e6afeab5caa95993613f15294ce1b1380/matplotlib-3.0.0-cp37m-win_amd64.whl (8.9MB)
    100% |#####| 8.9MB 802kB/s
Collecting pyparsing!=2.0.4,!>2.1.2,!>2.1.6,>=2.0.1 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/2b/4a/f06b45ab9690d4c37641ec776f7ad691974f4cf6943a73267475b05cbfca/pyparsing-2.2.2-py2.py3-none-any.whl (57kB)
    100% |#####| 61kB 892kB/s
Requirement already satisfied: numpy>=1.10.0 in c:\users\wiamhpd-myhome\appdata\local\programs\python\python37\lib\site-packages (from matplotlib) (1.15.1)
Collecting cycler>=0.10 (from matplotlib)

```

### 3) matplotlib 맛보기

02 이후 `[import matplotlib.pyplot as plt]`로  
matplotlib의 pyplot를 가지고 옴

03 `plt.plot(리스트값)` 함수로 리스트값을 그래프함수에  
전달함

04 `plt.show()`를 통하여 그래프를 그림

※ 더 자세한 사항은 검색을 통하여 심화 학습함

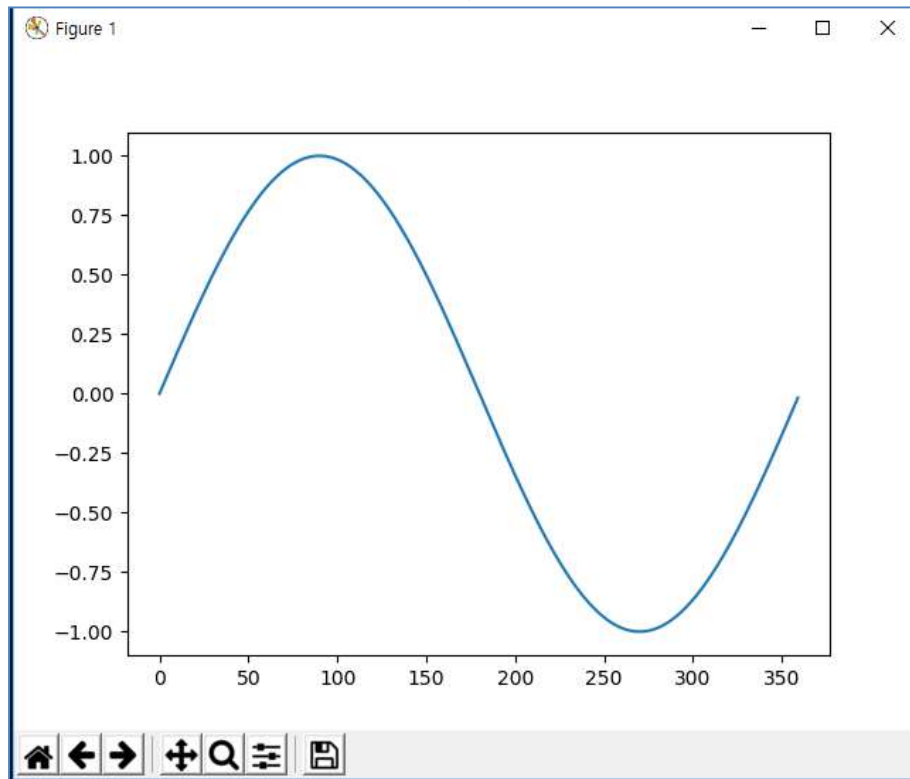
### 3) matplotlib 맛보기

```
from math import sin, pi
import matplotlib.pyplot as plt

graph_value=[] #리스트 선언, 빈 리스트를 만들어야 append가능
for i in range(360):
    graph_value.append(sin(i*pi/180)) #sin값을 하나씩 리스트에 넣음

plt.plot(graph_value) #해당 리스트를 그래프 함수에 보냄
plt.show()
```

### 3) matplotlib 맛보기



## 4) statistics 모듈 사용 예제



통계 관련 모듈



평균(mean), 표준편차(stdev), 분산(variance)등을 구하는데 사용 가능함

```
>>> import statistics as s
>>> a=[100,95,25,33,45,90,100,35,25]
>>> s.mean(a) #평균
60.888888888888886
>>> s.stdev(a) #표준편차
34.18129768032675
>>> s.variance(a) #분산
1168.3611111111111
>>>
```

### TIP

s.까지 입력하고  
몇 초 기다리면  
s 모듈의 함수가  
리스트로  
나옴(실습 참고)

## 4) statistics 모듈 사용 예제

Q

100km의 거리를 갈 때는 시속 120km, 올 때는 80km로 왔을 때 평균 시속 몇 km로 왔는가?

A

- 잘못된 풀이
  - $(120 + 80)/2 = 100$  , 시속 100km
- 올바른 풀이
  - 갈 때는 시속 120km이므로 50분 걸렸고, 올 때는 시속 80km이므로 75분 걸림
  - 총 125분 동안 200km를 간 것이므로  $200 \times 60/125 =$  시속 96km임
  - 이런 평균을 조화평균(Harmonic Mean) 이라고 함



## 4) statistics 모듈 사용 예제

Q

100km의 거리를 갈 때는 시속 120km, 올 때는 80km로 왔을 때 평균 시속 몇 km로 왔는가?

```
>>> import statistics as s
>>> s.harmonic_mean([120,80])
96.0
```

# 02

## 시간, 난수 발생

- 1) time 모듈
- 2) datetime 모듈
- 3) 시간차 계산
- 4) sleep() 함수
- 5) 난수 발생

## 1) time 모듈



time 모듈을 임포트 후 `time()` 함수를 사용하여 현재 시간을 얻음



얻어진 시간은 1970.1.1 0시 0분 0초로 부터 몇 초가 지났는지 표시함



하지만 한국표준시는 +9시간으로 1970.1.1. 오전 9시 정각부터 임



`ctime()` 함수로 시각을 영문 문자형식으로 표시가 가능함

## 1) time 모듈

```
>>> import time as t
>>> a=t.time(); a    #현재시간을 구함
1538473517.0459952
>>> t.ctime(a)    #초단위로 표시된 시간을 문자형식으로 표현
'Tue Oct  2 18:45:17 2018 '
>>> t.ctime(0)    #우리나라 기준으로 time=0의 경우 1970년1월1일 오전9시 정각이다
'Thu Jan  1 09:00:00 1970'
>>>
```

## 1) time 모듈

localtime() 함수

시간 요소를 구조체로 반환함



tm\_year 년도, tm\_mon 월, tm\_mday 일,  
tm\_hour 시, tm\_min 분, tm\_sec 초,  
tm\_wday 요일 (0월요일, 1화요일~7일요일),  
tm\_yday 년간일, tm\_isdst 일광절약일 (썸머타임  
적용시)

└─ 썸머타임적용시



반환되는 구조체를 활용하여 포매팅 작업을 거쳐 시간을  
출력함

## 1) time 모듈

localtime() 함수

시간 요소를 구조체로 반환함

구분	설명	구분	설명
tm_year	년도	tm_mon	월
tm_mday	일	tm_hour	시
tm_min	분	tm_sec	초
tm_wday	요일 (0월요일, 1화요일~7일요일)		
tm_yday	연간 일자	tm_isdst	일광절약일 (썸머타임 적용시)



반환되는 구조체를 활용하여 포매팅 작업을 거쳐 시간을 출력함

## 1) time 모듈

```
>>> import time as t
>>> a=t.localtime()
>>> a
time.struct_time(tm_year=2018, tm_mon=10, tm_mday=2, tm_hour=18,
tm_min=58, tm_sec=8, tm_wday=1, tm_yday=275, tm_isdst=0)
>>>
print("%d년 %d월 %d일 %02d시 %02d분 %02d초"%(a.tm_year,a.tm_mon,a.
tm_mday,a.tm_hour,a.tm_min,a.tm_sec))
2018년 10월 2일 18시 58분 08초
```

## 2) datetime 모듈



datetime 모듈의 now() 함수로 보다 직관적으로 시간을 출력할 수 있음

```
>>> import datetime as t
>>> a=t.datetime.now()
>>> print("%d년 %d월 %d일 %02d시 %02d분
%02d초"%(a.year,a.month,a.day,a.hour,a.minute,a.second))
2018년 10월 2일 19시 23분 48초
```



### 3) 시간차 계산



시작 시간 초와 종료 시간 초의 차이로 실행 시간을 구할 수 있음

```
import time as t

start=t.time()
for i in range(10000):
    pass
end=t.time()

print("%f 초 동안 실행함"%(end-start))
```

```
>>>
== RESTART: ***.py ==
0.000939 초 동안 실행함
```

## 4) sleep() 함수

sleep(n) 함수

n초 동안 실행을 멈춘 후 다시 실행함

```
import time as t

for i in range(10):
    print(t.ctime())
    t.sleep(3)
```

```
>>>
== RESTART: ***.py ==
Tue Oct  2 19:33:38 2018
Tue Oct  2 19:33:41 2018
Tue Oct  2 19:33:44 2018
Tue Oct  2 19:33:47 2018
Tue Oct  2 19:33:50 2018
Tue Oct  2 19:33:53 2018
Tue Oct  2 19:33:56 2018
Tue Oct  2 19:33:59 2018
Tue Oct  2 19:34:02 2018
Tue Oct  2 19:34:05 2018
>>>
```

## 5) 난수 발생



앞서서 9주(8강) 리스트에서 심화학습 부분에 난수를 발생시켜서 프로그램을 처리하였음



random모듈을 import하여 난수 관련 함수를 이용함



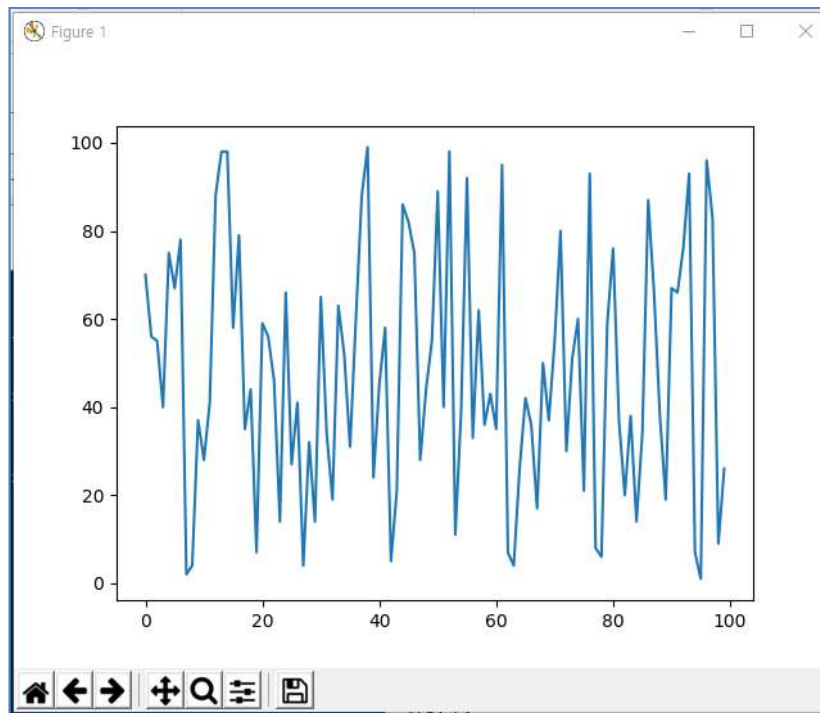
random.randrange(0,100) : 0부터 100까지 임의에 정수를 발생시킴

```
import random as r
import matplotlib.pyplot as plt

score=[]
for i in range(100):
    score.append(r.randrange(0,100))

plt.plot(score)
plt.show()
```

## 5) 난수 발생





# 03

## sys 모듈

---

- 1) sys 모듈
- 2) built-in 모듈 응용사례

## 1) sys 모듈



시스템의 정보와 시스템을 관리할 수 있는 모듈



기타 응용사례는 help(sys)를 통하여 실습함

```
>>> import sys
>>> sys.version #파이썬 버전 조회

' 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] '
>>> sys.platform #파이썬 설치 플랫폼

'win32'
>>> sys.getwindowsversion() #설치된 윈도우의 버전

sys.getwindowsversion(major=10, minor=0, build=17134, platform=2, service_pack='')
```

## 2) built-in 모듈 응용사례



python IDLE의 도움말(IDLE에서 f1키를 누르거나 메뉴에서 help 를 누르면 나타남) ➡ python docs을 선택함

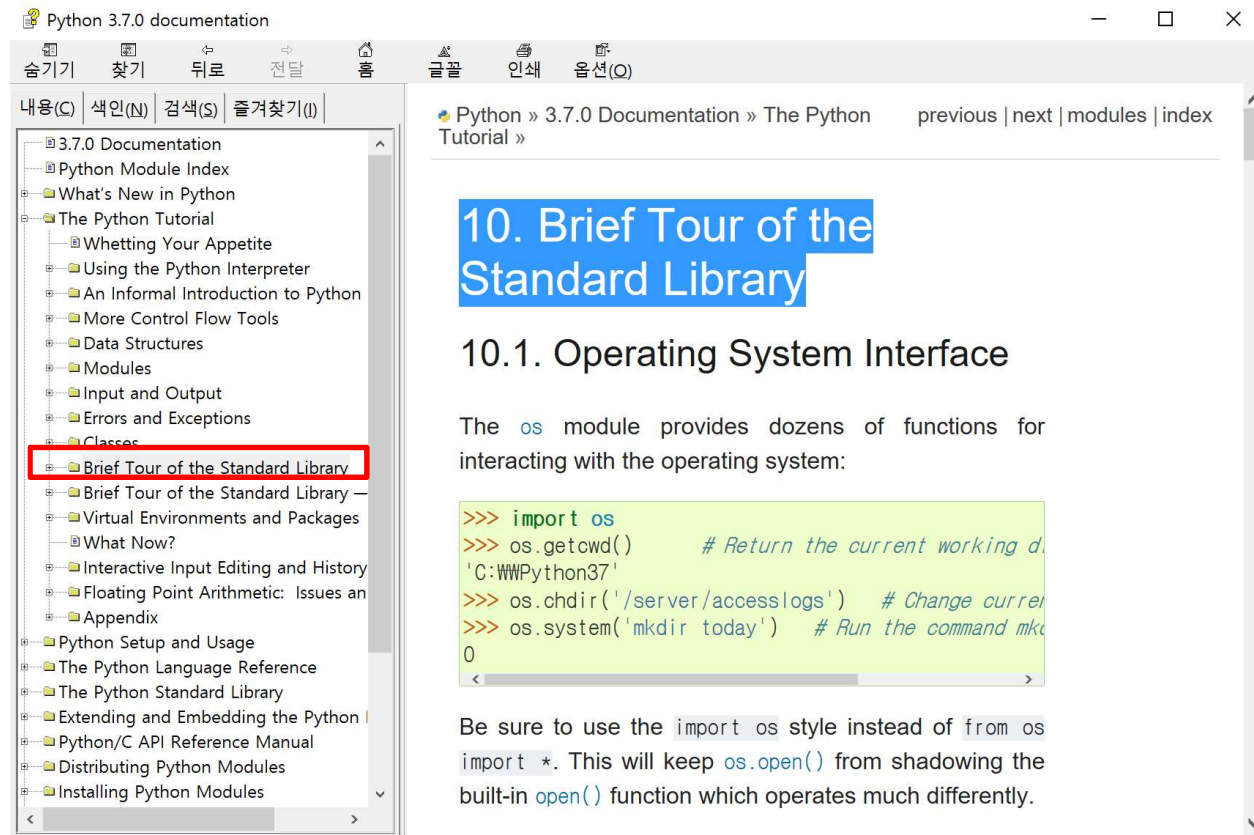


Python 3.7.0 Documentation ➡ The Python Tutorial ➡ 10. Brief Tour of the Standard Library 를 참고할 것



그 중 몇 가지 예제를 같이 해볼 것임

## 2) built-in 모듈 응용사례



The screenshot shows the Python 3.7.0 documentation website. The left sidebar contains a tree view of the documentation structure, with 'Brief Tour of the Standard Library' highlighted in red. The main content area displays the title '10. Brief Tour of the Standard Library' and the sub-section '10.1. Operating System Interface'. Below the sub-section, a paragraph explains that the `os` module provides functions for interacting with the operating system. A code block shows a Python script using `import os` to get the current working directory, change the current directory, and run a system command. The text below the code block advises using `import os` instead of `from os import *` to avoid shadowing the built-in `open()` function.

Python 3.7.0 documentation

숨기기 찾기 뒤로 전달 홈 글꼴 인쇄 옵션(O)

내용(C) 색인(N) 검색(S) 즐겨찾기(I)

Python » 3.7.0 Documentation » The Python Tutorial » previous | next | modules | index

### 10. Brief Tour of the Standard Library

#### 10.1. Operating System Interface

The `os` module provides dozens of functions for interacting with the operating system:

```
>>> import os
>>> os.getcwd()          # Return the current working d
'C:\WWPython37'
>>> os.chdir('/server/accesslogs')  # Change curre
>>> os.system('mkdir today')  # Run the command mke
0
```

Be sure to use the `import os` style instead of `from os import *`. This will keep `os.open()` from shadowing the built-in `open()` function which operates much differently.



## 2) built-in 모듈 응용사례

- ✓ import os를 통하여 operating system(운영체제→윈도우)을 제어할 수 있음
- ✓ os.getcwd()로 현재 실행 디렉토리를 얻을 수 있음
- ✓ os.system(명령)으로 윈도우의 명령을 실행할 수 있음

```
>>> import os
>>> os.getcwd()
'C:\\Users\\iamhpd\\AppData\\Local\\Programs\\Python\\Python37'
>>> os.system("calc") #계산기 실행
0
>>> os.system(" IDLE ") #IDLE실행, 실행안됨 path가 다름, 1리턴
1
>>> os.system("python") #파이썬 실행 실행된 상태에서 기다림
-1073741510
>>> os.system("shutdown -f -t 10") #이 명령 수행전 모든 윈도우 프로그램을 정리할 것
```

## 2) built-in 모듈 응용사례



urllib을 이용하여 홈페이지 소스를 분석함



urlopen 함수를 이용하여 학교 홈페이지([www.sju.ac.kr](http://www.sju.ac.kr)) 중 첫 페이지의 소스를 한 줄씩 받아옴



해당 줄에 “세종”이라는 글자가 있으면 해당 줄만 출력함




파이썬 라이브러리를 이용하면 다른 언어에서 수백 줄 작성하는 코드가 몇 줄로 처리될 수 있음

## 2) built-in 모듈 응용사례

```
from urllib.request import urlopen

with urlopen('http://www.kopo.ac.kr/') as response:
    for line in response:
        line = line.decode('utf-8') # 한글 처리
        if "폴리텍" in line:
            print(line)
```



```
>>>
== RESTART: ***.py ==
<meta name="viewport" content="width=device-width, initial-
scale=1.0 ,maximum-scale=1.0, minimum-scale=1.0,user-
scalable=no,target-densitydpi=medium-dpi" />
<meta name="author" content="폴리텍대학" />
<meta name="description" content="폴리텍대학 홈페이지입니다." />
>>>
```

# 04

## 실습하기 I

- 1) 수학 - 표준 라이브러리, math 모듈 사용 예제
- 2) 수학 - matplotlib 맛보기, statistics 모듈 사용 예제
- 3) 시간, 난수 발생 - time 모듈, datetime 모듈

# 실습내용

- 1) 수학 - 표준 라이브러리, math 모듈 사용 예제
- 2) 수학 - matplotlib 맛보기,  
statistics 모듈 사용 예제
- 3) 시간, 난수 발생 - time 모듈, datetime 모듈

# 05

## 실습하기 II

- 1) 시간, 난수 발생 - 시간차 계산, sleep함수, 난수 발생
- 2) sys 모듈 - sys 모듈
- 3) sys 모듈 - built-in 모듈 응용사례

# 실습내용

- 1) 시간, 난수 발생 - 시간차 계산, sleep함수, 난수 발생
- 2) sys 모듈 - sys 모듈
- 3) sys 모듈 - built-in 모듈 응용사례



## 학습활동

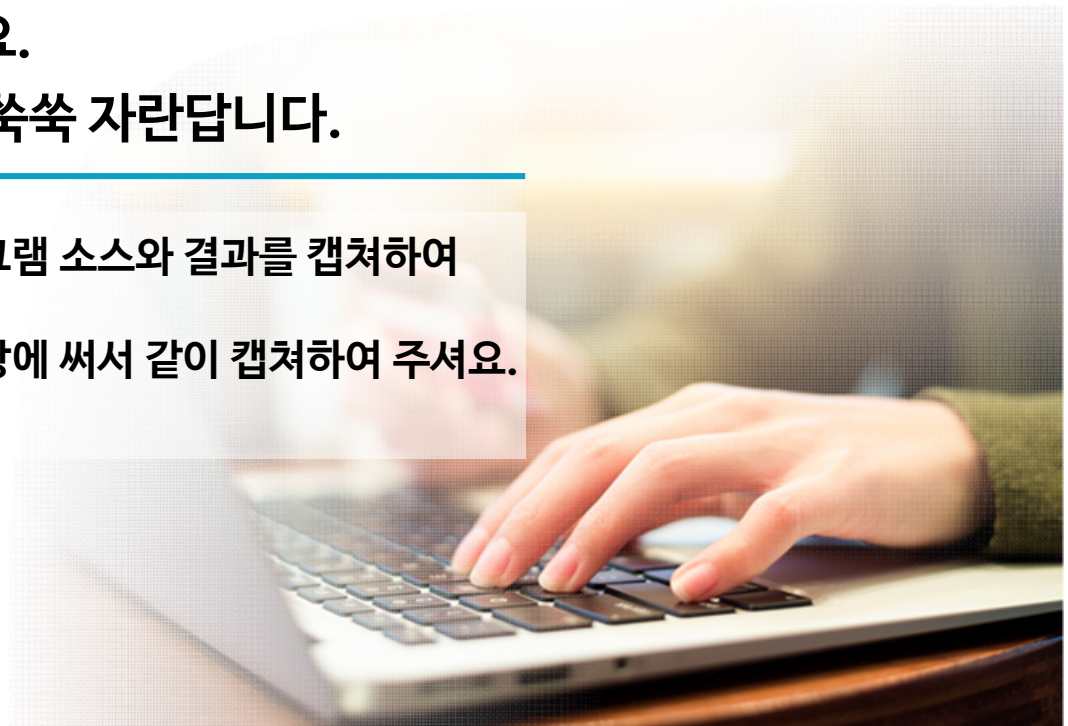
일시정지 버튼을 누른 후, 아래의 학습활동에 참여하세요.

Q

오늘 배운 내용을 스스로 실습하여  
자유게시판에 올려 주세요.

이렇게 정리하면 실력이 쑥쑥 자란답니다.

- ① 본인이 실습한 내용을 프로그램 소스와 결과를 캡처하여 올려주세요.
- ② 본인의 학번과 이름을 메모장에 써서 같이 캡처하여 주세요.
- ③ 그리고 설명도 달아 주세요.







## 학습활동에 대한 교수님 의견

Q

오늘 배운 내용을 스스로 실습하여 자유게시판에 올려 주세요.  
이렇게 정리하면 실력이 쑥쑥 자란답니다.

A

[ 오늘 학습한 내용의 실습 사항 ]

- ① 수학 - 표준 라이브러리, math 모듈 사용 예제
- ② 수학 - matplotlib 맛보기, statistics 모듈 사용 예제
- ③ 시간, 난수 발생 - time 모듈, datetime 모듈
- ④ 시간, 난수 발생 - 시간차 계산, sleep함수, 난수 발생
- ⑤ sys 모듈 - sys 모듈
- ⑥ sys 모듈 - built-in 모듈 응용사례

# 학습 평가

Q1

Q2

Q3

Q4

Q5

Q1

다음 표준라이브러리를 사용하기 위한 선언 중  
**잘못된** 내용은?

- 1 import 라이브러리명
- 2 from 라이브러리명 import 함수명
- 3 import 라이브러리명 as 단축명
- 4 import 라이브러리명 with 함수명

# 학습 평가

Q1

Q2

Q3

Q4

Q5

Q1

다음 표준라이브러리를 사용하기 위한 선언 중  
**잘못된** 내용은?

- 1 import 라이브러리명
- 2 from 라이브러리명 import 함수명
- 3 import 라이브러리명 as 단축명
- ☒ import 라이브러리명 with 함수명

정답

4번

해설

표준 라이브러리를 사용하기 위하여  
보기 ①, ②, ③과 같이 선언하여야 합니다.

# 학습 평가

## Q2

Q1

Q2

Q3

Q4

Q5

다음은 코딩하였을 때 해당되는 내용은?

```
for i in range(360): sin_value=sin(i*pi/180)
```

- 1 sin함수의 0부터 360라디안에 대한 값을 구하고 있다.
- 2 sin함수의 0부터 180라디안에 대한 값을 구하고 있다.
- 3 sin함수의 0부터 360도 (degree)에 대한 값을 구하고 있다.
- 4 sin함수의 0부터 180도 (degree)에 대한 값을 구하고 있다.

# 학습 평가

Q1

Q2

Q3

Q4

Q5

## Q2

다음은 코딩하였을 때 해당되는 내용은?

```
for i in range(360): sin_value=sin(i*pi/180)
```

- 1 sin함수의 0부터 360라디안에 대한 값을 구하고 있다.
- 2 sin함수의 0부터 180라디안에 대한 값을 구하고 있다.
- ☒ 3 sin함수의 0부터 360도 (degree)에 대한 값을 구하고 있다.
- 4 sin함수의 0부터 180도 (degree)에 대한 값을 구하고 있다.

정답

3번

해설

sin함수는 라디안의 값으로 입력 받기 때문에  
0부터 360도의 값을 해당 표현과 같이 변환합니다.

# 학습 평가

Q1

Q2

Q3

Q4

Q5

## Q3

다음 time 모듈의 time() 함수의 값이 0이 반환되었을 경우 나타낸 국내 기준 시간은?

- 1 1900년 1월 1일 0시 0분 0초
- 2 1900년 1월 1일 9시 0분 0초
- 3 1970년 1월 1일 0시 0분 0초
- 4 1970년 1월 1일 9시 0분 0초



## 학습 평가

Q1

Q2

Q3

Q4

Q5

### Q3

다음 time 모듈의 time() 함수의 값이 0이 반환되었을 경우 나타낸 국내 기준 시간은?

- 1 1900년 1월 1일 0시 0분 0초
- 2 1900년 1월 1일 9시 0분 0초
- 3 1970년 1월 1일 0시 0분 0초
- ☒ 1970년 1월 1일 9시 0분 0초

정답

4번

해설

time() 함수는 1970년 1월 1일 9시 0분 0초(국내 시 기준 +9)부터 몇 초가 경과하였는지 알려줍니다.

# 학습 평가

Q1

Q2

Q3

Q4

Q5

## Q4

다음 time 모듈의 localtime() 함수로 부터 반환되는 값 중 요일 값이 반환되는 구조체 값은?

- 1 tm\_week
- 2 tm\_mday
- 3 tm\_wday
- 4 tm\_yday





# 학습 평가

Q1

Q2

Q3

Q4

Q5

## Q4

다음 time 모듈의 localtime() 함수로 부터 반환되는 값 중 요일 값이 반환되는 구조체 값은?

1 tm\_week

2 tm\_mday

☒ 3 tm\_wday

4 tm\_yday

정답

3번

해설

tm\_year 년도, tm\_mon 월, tm\_mday 일, tm\_hour 시, tm\_min 분, tm\_sec 초, tm\_wday 요일 (0월요일, 1화요일~7일요일), tm\_yday 연간일, tm\_isdst 일광절약일 (썸머타임적용시)을 의미합니다.

# 학습 평가

Q1

Q2

Q3

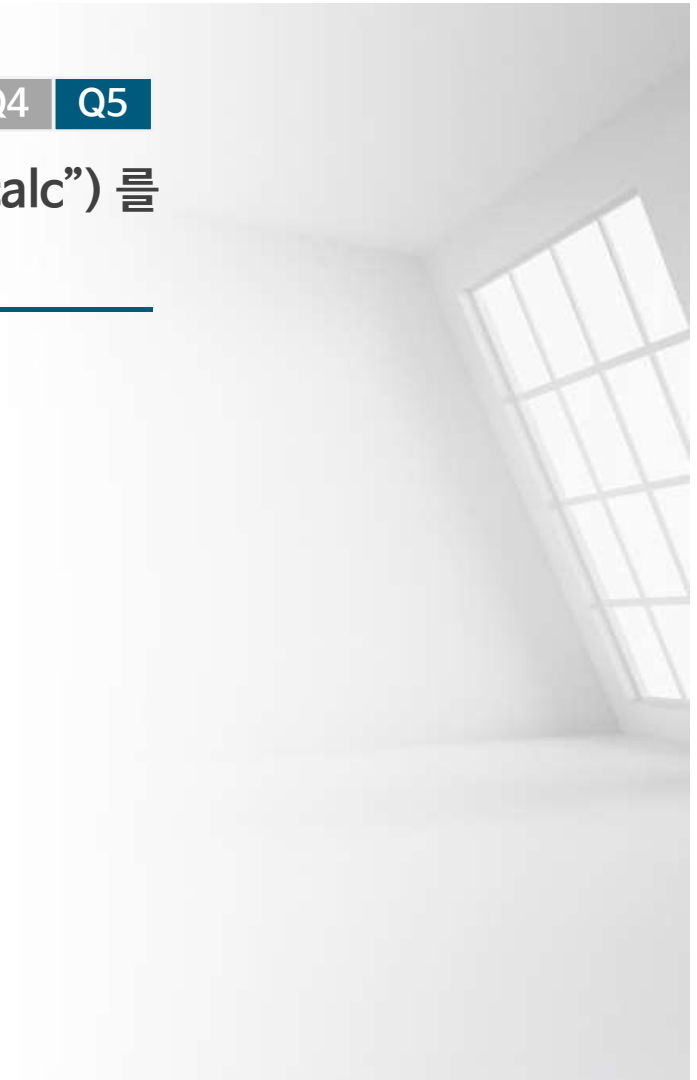
Q4

Q5

## Q5

다음 `import os` 이후 `os.system("calc")` 를 실행하였을 경우 올바른 설명은?

- 1 윈도우 버전 계산
- 2 윈도우 계산기 실행
- 3 저장된 변수 계산
- 4 저장된 상수 계산



# 학습 평가

Q1

Q2

Q3

Q4

Q5

## Q5

다음 `import os` 이후 `os.system("calc")` 를 실행하였을 경우 올바른 설명은?

- 1 윈도우 버전 계산
- ☒ 2 윈도우 계산기 실행
- 3 저장된 변수 계산
- 4 저장된 상수 계산

정답

2번

해설

`os.system(명령)`으로 윈도우의 명령을 실행할 수 있습니다.



## 정리하기

### 수학

- ✓ python을 설치하면 자주 사용되는 다양한 함수(def)가 표준 라이브러리에 포함되어 있음
- ✓ [import 라이브러리명] 등으로 가지고 올 표준 모듈을 선언하여야 함
- ✓ math 모듈은 수학관련, statistics 모듈은 통계 관련 유용한 함수를 제공함





## 정리하기

### 시간 난수 발생

- ✓ time 모듈은 현재시간, 실행시간 등을 구하는데 유용함
- ✓ datetime 모듈로 현재 시간을 구할 수 있음
- ✓ sleep 함수로 실행을 지연시킬 수 있으며,  
random 모듈로 특정 난수를 발생시킬 수 있음





## 정리하기

### sys 모듈

- ✓ sys 모듈은 시스템의 정보와 시스템을 관리할 수 있는 모듈
- ✓ built-in 모듈 응용사례로 IDLE에 포함되어 있는 The Python Tutorial을 활용할 것

