

Day2. 파이썬 기초 문법과 라이브러리

심선영 교수

❖ **강사 소개: 성신여자대학교 경영학부 심선영 교수 (연락처: syshim@sungshin.ac.kr)**
[학력 및 경력]

- 고려대학교 전산과학과 (~1997)
- KOSCOM(~2000) , 키움닷컴 증권 IT개발부서 (~2001)
- 한국과학기술원 경영학 석사 (~2004)
- 한국과학기술원 경영공학박사, MIS (~2008), 텍사스주립대 전자상거래 연구소, 방문연구원(~2009)
- 성신여자대학교 경영학부 교수 (2010~)

[연구분야]

- 경영정보시스템 투자 및 운영전략, 디지털경제 및 기업전략, IT투자분석, IT아웃소싱, IT거버넌스
- 빅데이터, 인공지능, 프로세스 기반 디지털 트랜스포메이션 (RPA)

[K-MOOC 강좌 운영, kmooc.kr]

- 4차 산업혁명과 경영혁신 (2018~), 비즈니스를 위한 인공지능 (2020~), 블록체인 경제와 디지털 자산 혁명 (2022~)

[위원회/자문활동]

- 5G 통신정책협의회 제2소위 자문위원 (과기정통부, 2018)
- 제3기 인터넷상생발전협의회 (방통위, 2020)
- 인공지능을 활용한 디지털 감사 방안 수립 (행안부, 2021)
- 조달청 대형 정보화사업 전문 평가위원 / 지능정보사회진흥원 정보화사업 전문 평가위원 (2019~)
- 범정부 RPA 확산 가이드 작성 (행안부, 2022)



강의 목표

- ❖ 파이썬에 대해 이해하고 분석환경을 이해할 수 있다.
- ❖ 파이썬 언어의 기본 문법을 익힌다.
- ❖ 파이썬 언어에서 사용가능한 기본 구문을 이해한다.

Day2. 강의 스케줄

| 목차 | 활동 |
|------------------------------|---|
| Day2 Overview | 파이썬 개요 및 분석환경 실습 |
| 파이썬 기초 문법 - 기본 데이터 타입과 변수 | 파이썬 실습 파일명 <ul style="list-style-type: none">• Day2_1(1) 변수_기본데이터타입_숫자형.ipynb• Day2_1(2) 기본데이터타입_문자열.ipynb• Day2_1(3) 리스트_튜플.ipynb• Day2_1(4) 딕셔너리.ipynb |
| 파이썬 기초 문법 - 함수 | 파이썬 실습 파일명: Day2_2(1) 함수.ipynb Day2_2(2) 사용자함수.ipynb |
| 파이썬 기초 문법 - 반복문과 조건문 | 파이썬 실습 파일명: Day2_3 if조건문과for반복문.ipynb Day2_3 while반복문.ipynb |
| Wrap-Up | 학습 정리 |

파이썬 개요 및 분석환경 실습

파이썬 개요 및 분석환경 실습

파이썬이란?

파이썬 python

[1991년 네덜란드 출신의
귀도 반 로섬 (Guido Van Rossum)이
개발 하고 2020년대 가장
인기있는 오픈 소스 기반의
인터프리터형 객체지향
프로그래밍 언어]



<출처> <https://www.cwi.nl/news/blogs/interview-guido-van-rossum-201cid-rather-write-code-than-papers.201d/@images/7f6415bf-a2c5-44fd-a10e-99032b6d8c0d.jpeg> (21.08.31)

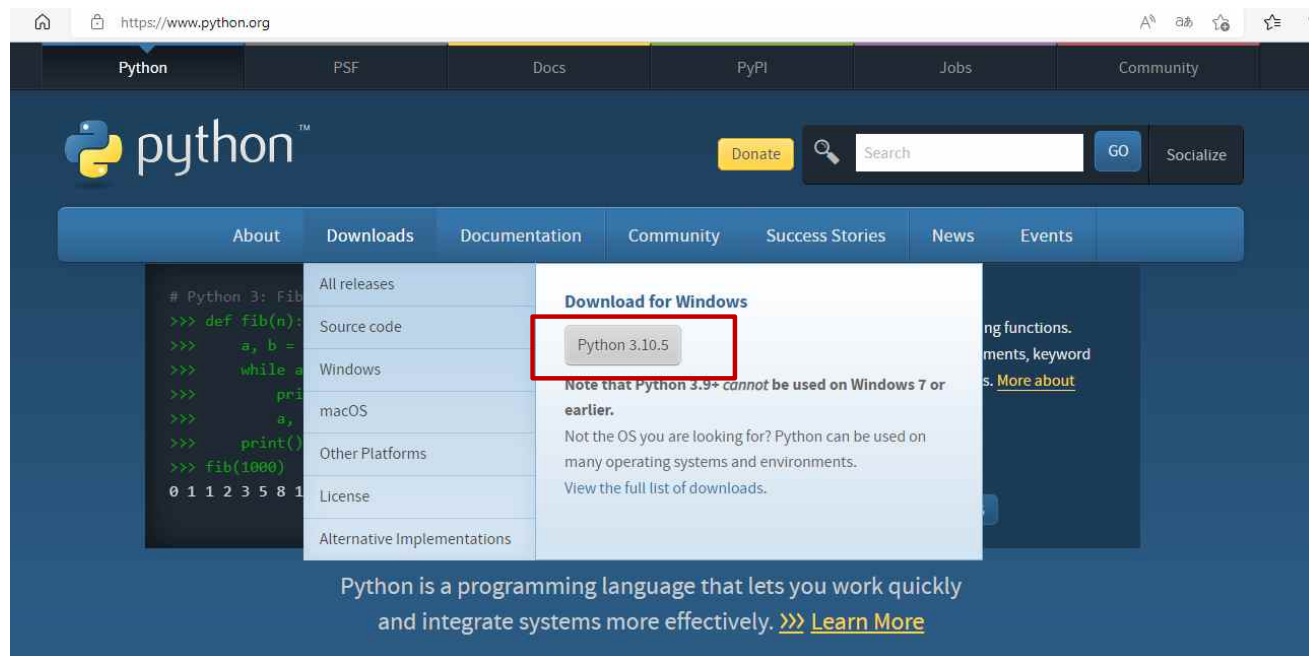
파이썬이란?

- ❖ 문법이 간결하여 초보자가 배우기 쉽다.
- ❖ 오픈 소스 라이브러리가 풍부하다
 - 데이터 과학 관련, 무료로 사용할 수 있는 소프트웨어가 많다.
- ❖ 통계 분석가와 IT 개발자가 함께 사용 할 수 있는 언어
 - 통계 패키지 SAS, SPSS, R 와 개발언어 java, c, c++ 처럼 따로 배울 필요 없이 파이썬 하나로 해결된다.

파이썬 설치

❖ 파이썬 (Python)

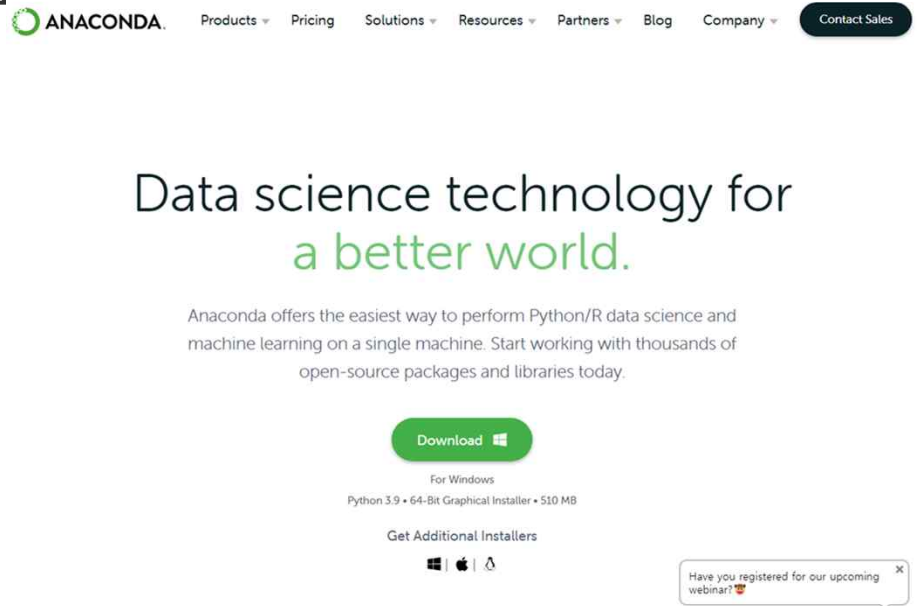
- 비영리로 운영되는 '파이썬 소프트웨어 재단'이 관리하는 개방형, 공동체 기반 개발 모델을 가지고 있으며 무료로 누구나 다운로드 받아 사용 가능
- 공식 홈페이지인 <https://www.python.org> 에서 다운로드



아나콘다 설치

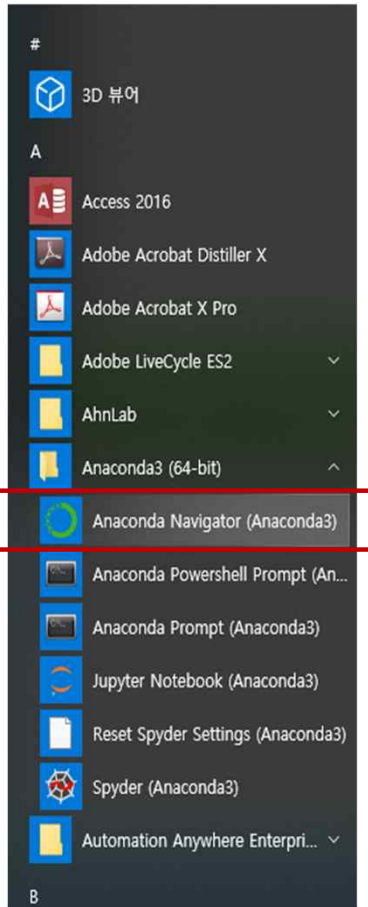
❖ 아나콘다 (Anaconda)

- 파이썬과 함께 인공지능, 머신러닝, 딥러닝 관련 많은 패키지들을 쉽게 설치하여 환경구축을 용이하게 해 주는 플랫폼
- 아나콘다를 설치할 경우 파이썬을 별도로 설치할 필요 없음!
- 공식 홈페이지인 <https://www.anaconda.com/> 에서 다운로드



쥬피터 노트북 실행하기 (1)

Anaconda Navigator



❖ 아나콘다에서 제공하는 각종 서비스들을 한눈에 볼 수 있는 GUI Tool

- 쥬피터 노트북

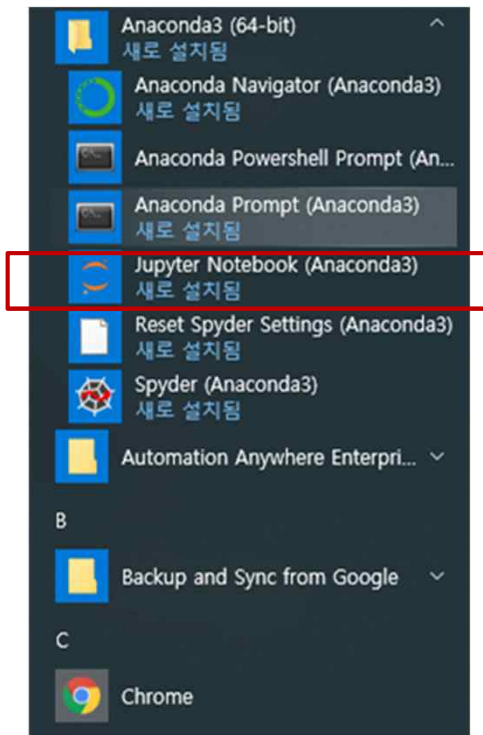
- 파이썬 코드를 웹브라우저 인터페이스를 통해 쉽게 확인하고 관리할 수 있는 개발 환경

- Launch

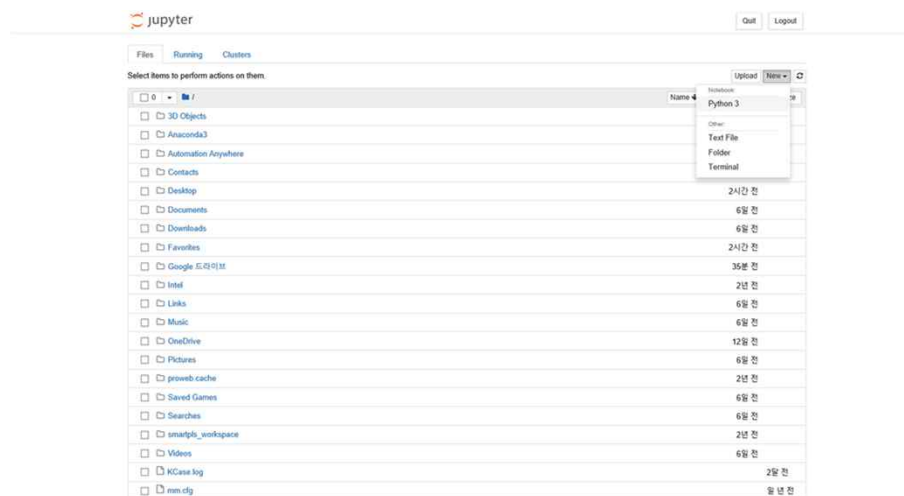
- 백그라운드) 쥬피터 노트북 서버를 수행 → 서버에 자동접속 → 브라우저 창 → 오른쪽 상단 “New” 버튼 → 새로운 파일작성

쥬피터 노트북 실행하기(2)

❖ 프로그램 리스트에서 직접실행



1. 프로그램 리스트에서 'Jupyter Notebook' 실행
2. 브라우저에서 원하는 파일 선택



파이썬 개발 환경설정(Window)

Anaconda Navigator

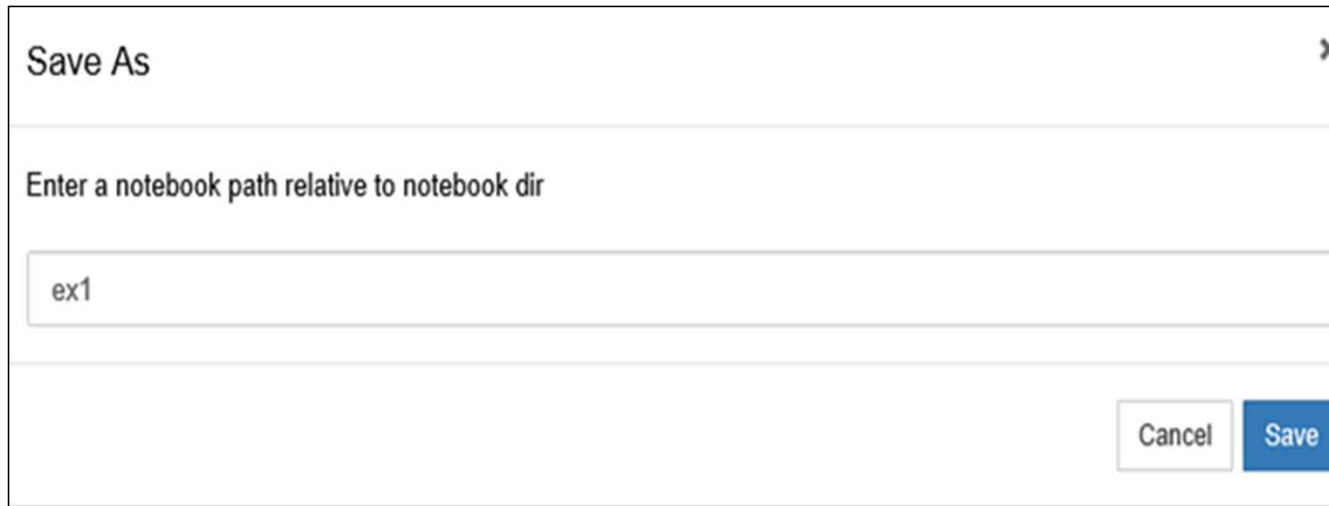


- 오른쪽 상단 “New” 버튼 → 새로운 파일 작성 (Python3)

파이썬 개발 환경설정(Window)

파일 저장

- File > Save as >...
- **ex1.ipynb** 라는 파일이 생성됨!

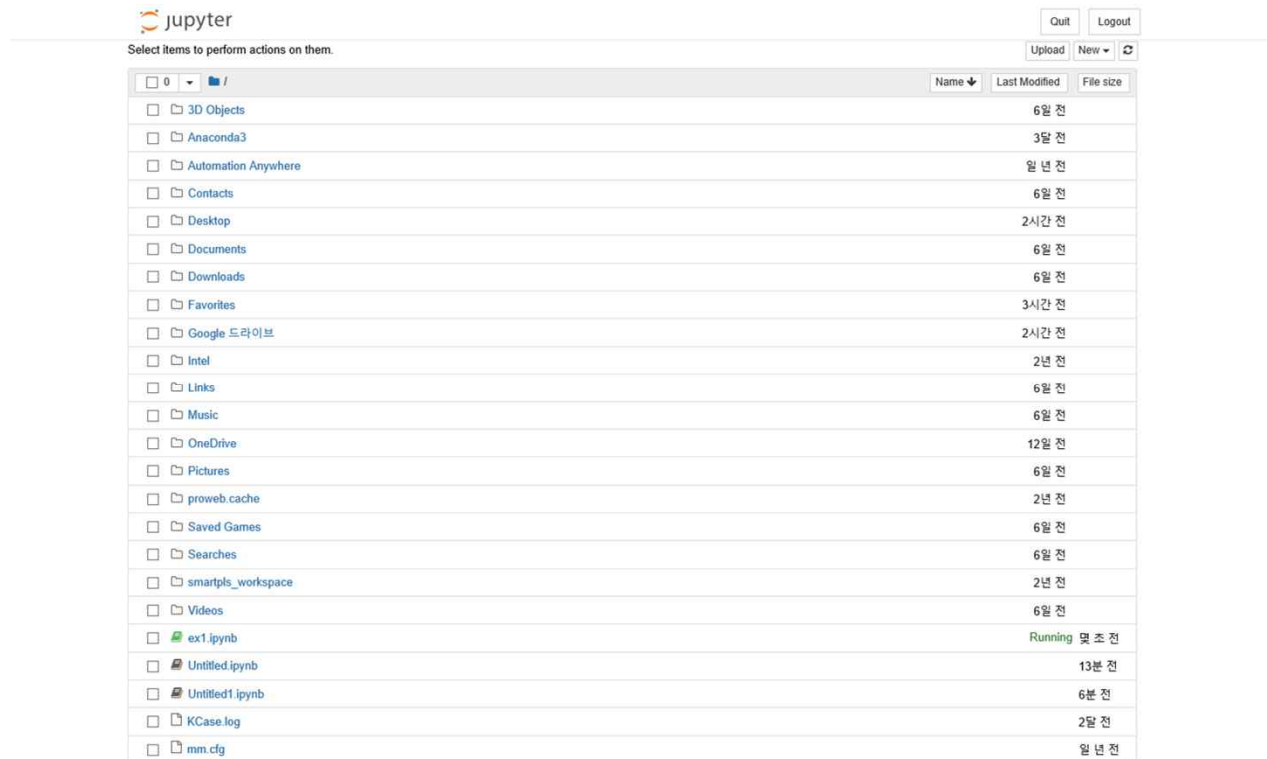


The image shows a 'Save As' dialog box from JupyterLab. The title bar says 'Save As' with a close button (X) on the right. Below the title bar, there is a text prompt: 'Enter a notebook path relative to notebook dir'. A text input field contains the text 'ex1'. At the bottom right, there are two buttons: 'Cancel' and 'Save'.

파이썬 개발 환경설정(Window)

저장된 파일 확인

- 실행 중인 파일 --> 초록색으로 표시됨



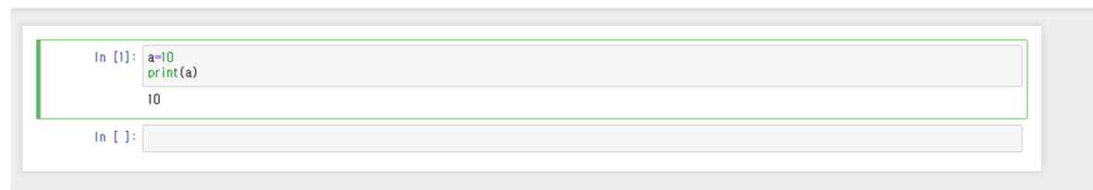
The image shows the JupyterLab file browser interface. At the top, there's a 'jupyter' logo and buttons for 'Quit' and 'Logout'. Below that, a toolbar contains 'Upload', 'New', and a refresh icon. The main area displays a file browser for the root directory '/'. It lists various system folders like '3D Objects', 'Anaconda3', 'Automation Anywhere', 'Contacts', 'Desktop', 'Documents', 'Downloads', 'Favorites', 'Google 드라이브', 'Intel', 'Links', 'Music', 'OneDrive', 'Pictures', 'proweb.cache', 'Saved Games', 'Searches', 'smartpls_workspace', 'Videos', and several '.ipynb' files. The file 'ex1.ipynb' is highlighted in green and has a 'Running' status indicator next to it. Other files include 'Untitled.ipynb', 'Untitled1.ipynb', 'KCase.log', and 'mm.cfg'.

| Name | Last Modified | File size |
|---------------------|---------------|-----------|
| 0 | | |
| 3D Objects | 6일 전 | |
| Anaconda3 | 3달 전 | |
| Automation Anywhere | 일 년 전 | |
| Contacts | 6일 전 | |
| Desktop | 2시간 전 | |
| Documents | 6일 전 | |
| Downloads | 6일 전 | |
| Favorites | 3시간 전 | |
| Google 드라이브 | 2시간 전 | |
| Intel | 2년 전 | |
| Links | 6일 전 | |
| Music | 6일 전 | |
| OneDrive | 12일 전 | |
| Pictures | 6일 전 | |
| proweb.cache | 2년 전 | |
| Saved Games | 6일 전 | |
| Searches | 6일 전 | |
| smartpls_workspace | 2년 전 | |
| Videos | 6일 전 | |
| ex1.ipynb | Running 몇 초 전 | |
| Untitled.ipynb | 13분 전 | |
| Untitled1.ipynb | 6분 전 | |
| KCase.log | 2달 전 | |
| mm.cfg | 일 년 전 | |

쥬피터 노트북 사용

편집모드와 명령모드

- 편집모드 (연두색)

A screenshot of a Jupyter Notebook cell in edit mode. The cell is highlighted with a light green border. The code inside is:

```
In [1]: a=10  
        print(a)  
10
```

The prompt 'In []:' is visible below the code area.

- 명령모드 (파란색) : ESC 키 클릭

A screenshot of a Jupyter Notebook cell in command mode. The cell is highlighted with a light blue border. The code inside is:

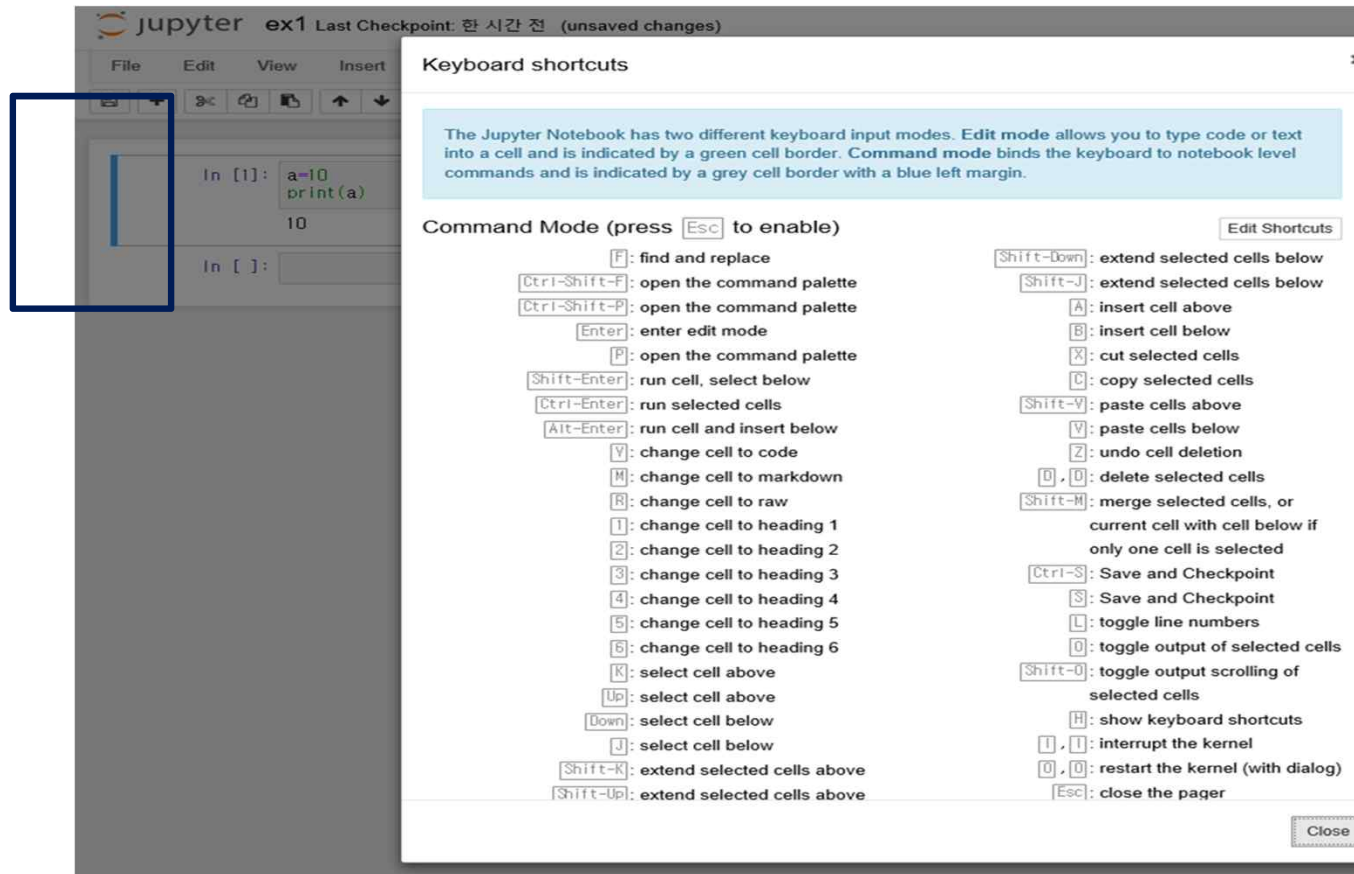
```
In [1]: a=10  
        print(a)  
10
```

The prompt 'In []:' is visible below the code area.

쥬피터 노트북 사용

Help 보기

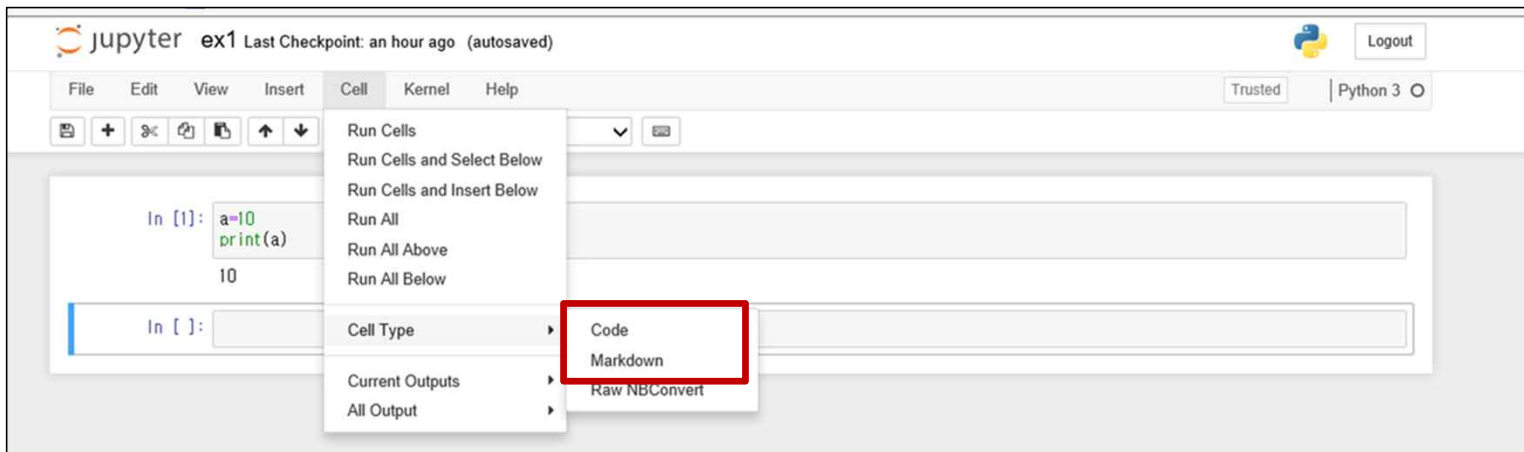
- ESC -> 명령모드 (파란색) -> h (Help)



쥬피터 노트북 사용

코드(code)와 마크다운(markdown)

❖ Cell > Cell Type 에서 선택



코드(code)와 마크다운(markdown)

❖ 코드: 소스코드의 입력

```
In [1]: a=10  
        print(a)  
  
10
```

❖ 마크다운: 소스코드의 제목, 설명 등 (#을 앞에 붙여서 글자크기 조절)

```
# 테스트 프로그램  
- 수식  
- 함수  
  
> 파이썬은 매우 재미있습니다
```



```
테스트 프로그램  
• 수식  
• 함수  
  
파이썬은 매우 재미있습니다
```

[기억하기] 단축키

명령(Command) 단축키

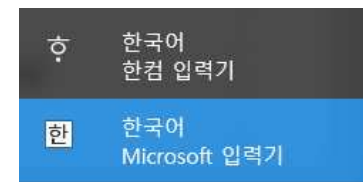
a : above (위에 셀 추가하기)
b : bottom (아래 셀 추가하기)
d+d : delete + delete (셀 삭제하기)
shift + 'l' : line (셀 줄번호 표시하기)
shift + 'm' : merger (셀 병합)

셀선택 + m : → 마크다운 변환
셀선택 + y : → 코드 변환
셀선택 + r : → Raw NBConvert
x : 셀 잘라내기
c : 셀 복사
v : 셀 붙여넣기

편집모드(Edit) 단축키

control + Enter : 셀 실행
shift + Enter : 셀 실행 후 아래 셀로 이동
alt + Enter : 셀 실행 후 아래 셀 삽입하며 이동
shift + Tab : 도움말
Tab : 자동 완성
control + '/' : 코드 주석 처리(해당 코드라인 가장 앞에 #을 넣는 기능)
control + shift + '-' : 셀 나누기 / shift + m 병합하기와 반대 기능

(안될경우)



파이썬 기본 익히기

파이썬 기본 익히기

❖ Jupiter Notebook

- [Step 1] 새 노트북(Notebook) 생성 – Untitled
- [Step 2] 셀 추가
 - 셀의 3가지 종류
 - ✓ Code: 코드 작성
 - ✓ Markdown: 문서 작성 (#, ## 등 사용)
 - ✓ NBConvert* : 작성된 코드가 실행되지 않고 코드 내용이 그대로 출력
- [Step 3] 코드/문서 입력 및 실행

파이썬 기초 문법

기본데이터 타입과 변수 (1)

- 계산식, 숫자타입, 변수

계산식, 숫자타입, 변수

우리는 어렸을 때...

❖ -2, 27, 35 

❖ 23.1 , 0.3 

❖ $23 + 0.5$

❖ $x = 23 + 0.5$

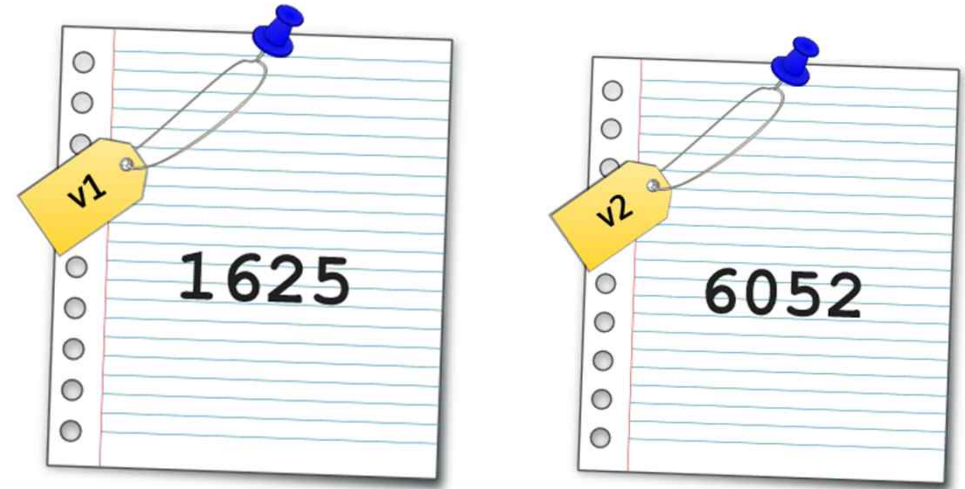
변수

❖ 변수의 개념

- 컴퓨터 메모리에 데이터를 기록
- 기록된 데이터에 붙이는 '태그'

❖ 변수의 필요성

- 중복되는 계산 제거 → 계산속도 향상
- 결과를 다른 곳에 사용



▶ $125 * 13 + 356 * 17$
▶ $125 * 13 - 356 * 17$



▶ $x1 = 125 * 13$
▶ $x2 = 356 * 17$



▶ $x1 + x2$
▶ $x1 - x2$

▶ `print(x1)`
▶ `print(x2)`

계산식, 숫자타입, 변수

변수

❖ 특정 변수의 값 지정하는 코드

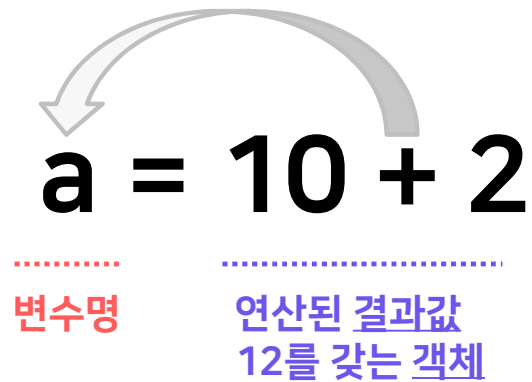
- 변수이름 = 값
- 변수이름 = 계산식

❖ 특정 변수의 값 변경 가능

```
▶ a = 2  
  print(a)  
  a = 8  
  print(a)
```

```
▶ x1 = 125 * 13  
  x2 = 356 * 17  
  print(x1)  
  print(x2)
```

```
▶ v1 = 125 * 13  
  v2 = 356 * 17  
  print(v1 + v2)  
  print(v1 - v2)
```



a = 10 + 2

.....

변수명 연산된 결과값
 12를 갖는 객체

계산식, 숫자타입, 변수

주요 자료형 (Data Type)

파이썬 주요 자료형 Data Type

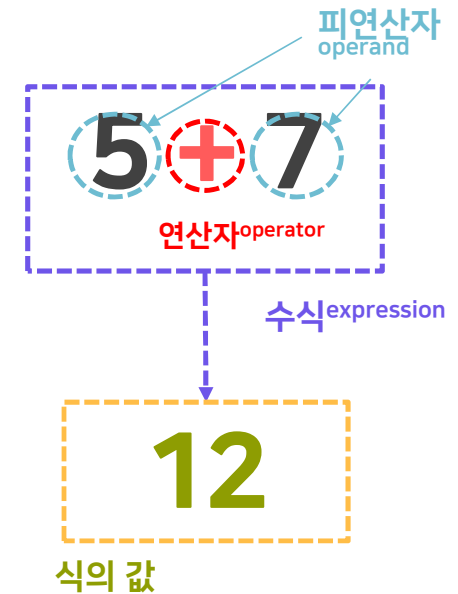
| | | | | | 이름 | 영문명 | 타입 | 설명 | 예시 |
|---|---|-------------|-------------------|--------------|--------------|------------------------------------|----|----|----|
| ① | { | 불리언 | boolean | bool | 논리값 | True, False | | | |
| | | 정수 | integer | int | 정수 | 1,200, -42, 25_000 | | | |
| | | 부동소수점 실수 | floating point | float | 컴퓨터 전용 소수 | 3.14, 2.7e5 | | | |
| ② | { | 문자열 | string | str | 텍스트 | 'oh', "happy" | | | |
| ③ | { | 리스트 | list | list | 목록 자료구조 | [1, 'one', 'ONE'] | | | |
| | | 딕셔너리 | dictionary | dict | 사전 자료구조 | { 'game': 'bingo', 'name': 'Bob' } | | | |

계산식, 숫자타입, 변수

산술 연산자

산술 연산자 Arithmetic operator : + _ * / // % **

| 연산자 | 의미 | 예시 | 결과 |
|-----|-------------------------------------|---------------------|----|
| + | 덧셈 또는 +부호 | 34+42, 'ab'+ 'c' | |
| - | 뺄셈 또는 -부호 | 24.3-50.7 | |
| * | 곱셈 | 32*10, 'a'*3 | |
| / | 나눗셈 division 부동소수점형 나눗셈 | 5/2 | |
| // | 나눗셈의 몫 floor division 정수형 나눗셈, 몫 | 5//2 | |
| % | 나눗셈의 나머지 modulo | 5%2 | |
| ** | 거듭제곱(지수승) | 2**3 | |



실습파일 Go!

파이썬 기초 문법

기본데이터 타입과 변수 (2)

- 문자열

문자열

Indexing & slicing string(문자열 인덱스 및 추출)

❖ a 문자열 길이: 11

Index: 0 1 2 3 4 5 6 7 8 9 10

a = 'Hello World'

| | | | | |
|------|------|------|-----|-------|
| a[0] | a[4] | a[5] | ... | a[10] |
| H | o | _ | | d |

문자열

Indexing & slicing string(문자열 인덱스 및 추출)

❖ a 문자열 길이: 11

‘-’ (마이너스): 뒤에서 몇 번째

a = ‘Hello World’

a[-11] a[-2] a[-1]

실습파일 Go!

파이썬 기초 문법

기본데이터 타입과 변수 (3)

- 데이터 묶음의 데이터 타입

데이터 묶음의 데이터 타입

데이터 묶음의 데이터 타입

❖ 데이터의 묶음을 효과적으로 다루기 위한 데이터 타입

- 리스트
- 튜플
- 딕셔너리
- (셋)

리스트

❖ 리스트란?

- 목록을 관리하는 자료구조

```
score = [90, 85, 93, 77, 78 ]
```

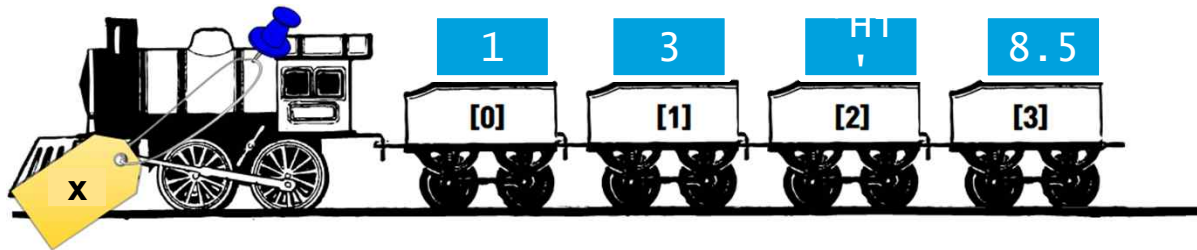
❖ 리스트 특징

- 대괄호 [] 를 사용하여 리스트를 표기
- 각 항목은 요소element(원소)라 하고, 콤마로 구분
- 요소를 선택하는 인덱싱과 잘라내는 슬라이싱이 가능
- 요소 값으로 숫자, 문자열 등 어떠한 자료형의 데이터도 올 수 있으며, 하나의 리스트에 섞어서 구성할 수 있다.

데이터 묶음의 데이터 타입

리스트

- ❖ 여러 개의 데이터 공간이 이어져 있는 데이터 타입
- ❖ 하나의 이름으로 전체 리스트 관리
- ❖ 인덱스: 리스트의 번호
 - 대괄호를 사용하여 중간에 있는 데이터도 직접적으로 관리(읽기, 수정) 가능



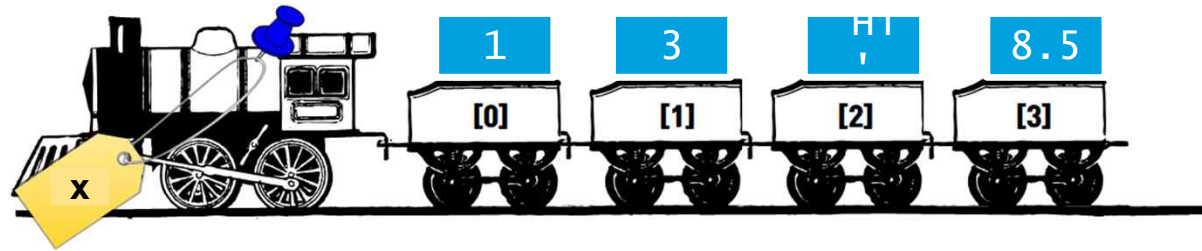
```
x = [1, 3, 'Hi' , 8.5]  
Print(x)  
print(x[2])
```

데이터 묶음의 데이터 타입

리스트

❖ 수정

- 대괄호를 사용하여 위치를 지정하고 대입 연산자(=)를 사용하여 값 수정



```
x = [1, 3, 'Hi' , 8.5]
print(x)
print(x[2])
x[2] = 100
print(x[2])
```

데이터 묶음의 데이터 타입

리스트

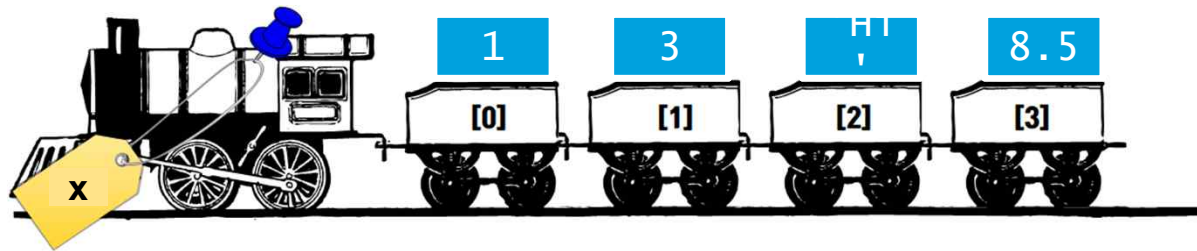
❖ 리스트의 멤버함수

- `append()`
- `extend()`
- `insert()`
- `remove()`
- `index()`
- `sort()`
- `sorted()`

리스트

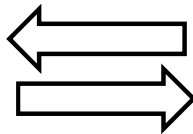
❖ 데이터 추가

- append함수를 사용하여 리스트 가장 뒤에 데이터 추가 가능
- 변수에 []를 지정하면 빈 리스트 생성 가능



```
vs = []  
vs.append(1)  
vs.append(3)  
vs.append('Hi')  
vs.append(8.5)
```

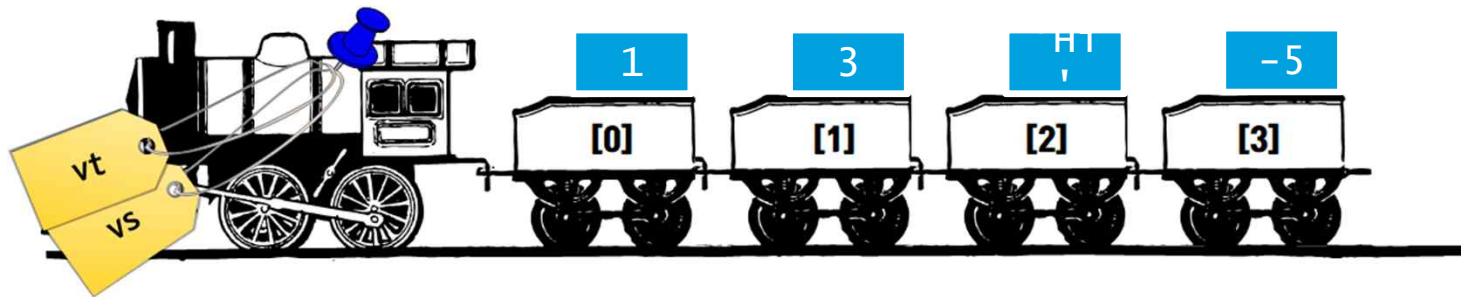
동일한 결과



```
vs = [1, 3, 'Hi', 8.5]
```

리스트

- ❖ 리스트 객체: 메모리 상에서 존재하는 리스트의 실재(實在)
- ❖ 리스트 변수: 리스트 객체에 붙어있는 태그



```
vs = [1, 3, 'Hi' ,  
      8.5]  
vt = vs  
vt[3] = -5  
print(vs)
```

- vt[3]의 값을 -5로 변경하면, vs[3]의 값 역시 -5로 변화
- 2개의 값이 바뀌는 것이 아니라, 두 변수가 가리키는 객체가 동일하기 때문에 발생하는 현상

Tuple(튜플)

❖ 튜플

- 리스트와 유사

❖ 차이점

- 대괄호 없이 생성
- 소괄호 () 를 사용하여 요소를 순서대로 나열
- 한번 만들면 수정할 수 없음



```
tup = 3,4,5  
print(tup[0])
```



```
tup = (3,4,5)  
print(tup[0])
```

Dictionary, dict(딕셔너리)

❖ 딕셔너리

- 리스트와 유사
- 중괄호 { } 를 이용하여 생성
- 키(key) : 값(value) 형태의 구조

❖ 차이점

- 항목에 순서가 없음, 즉, 인덱스가 없음
- 각 항목에 이름이 붙어 있음

❖ 예) 나라이름 : 수도이름 (실습)

파이썬 기초 문법

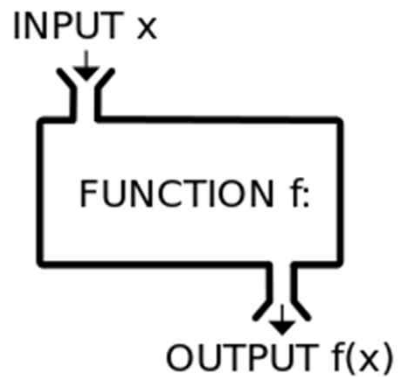
함수의 이해(1)

- 함수 개요, 내장형 함수

함수 개요, 내장형 함수

함수 function

- ❖ 프로그램에서의 함수는 일련의 명령문을 묶어서 실행하여 같은 코딩을 반복하지 않도록!
- ❖ 수학의 함수처럼 인수를 전달 받아 결과를 반환.



함수의 정의

$$f(x) = 3x + 7$$

함수의 호출

$$f(2) \Rightarrow 13$$

$$f(3) \Rightarrow 16$$

<출처> <https://ko.wikipedia.org/wiki/%ED%95%A8%EC%88%98> (21.08.30)

함수 function

❖ 파이썬 함수의 종류

① 내장(Built in) 함수 : 파이썬에서 기본으로 제공

- 기본 탑재된 함수를 바로 사용 (print(), len() 등)

② 라이브러리 함수 : 제 3자^{3rd party}가 제공

- 라이브러리를 import 문으로 포함한 후에 사용 (pandas 등)

③ 사용자 정의(User-defined) 함수

- 사용자가 직접 함수를 정의 (def 예약어를 이용)

실습파일 Go!

파이썬 내장 함수 파이썬 3.9.4 기준 69개

L

| | | | | |
|---------------|-------------|--------------|--------------|----------------|
| abs() | delattr() | hash() | memoryview() | set() |
| all() | dict() | help() | min() | setattr() |
| any() | dir() | hex() | next() | slice() |
| ascii() | divmod() | id() | object() | sorted() |
| bin() | enumerate() | input() | oct() | staticmethod() |
| bool() | eval() | int() | open() | str() |
| breakpoint() | exec() | isinstance() | ord() | sum() |
| bytearray() | filter() | issubclass() | pow() | super() |
| bytes() | float() | iter() | print() | tuple() |
| callable() | format() | len() | property() | type() |
| chr() | frozenset() | list() | range() | vars() |
| classmethod() | getattr() | locals() | repr() | zip() |
| compile() | globals() | map() | reversed() | __import__() |
| complex() | hasattr() | max() | round() | |

<출처> <https://docs.python.org/ko/3/library/functions.html#bin> (21.08.30)

수치연산 내장 함수

내장함수 built in function

함수

abs round divmod pow sum max min

`abs(x)`: 절대값 반환

`round(number[,ndigits])`: 반올림, *ndigits*는 유효자리수

`divmod(a,b)`: 2개의 숫자를 나누어 몫과 나머지를 반환

`pow(base,exp[,mod])`: 거듭제곱을 반환한다. *mod* 인수가 전달되면 나머지를 반환

`sum(iterable[,start=0])`: *start* 및 *iterable*를 왼쪽부터 합한 결과를 반환

`max(iterable, *[,key, default])`

`max(arg1, arg2, *args[,key])`: *iterable*에서 가장 큰 항목이나 두 개 이상의 인수 중 가장 큰 것을 반환. *key*는 순서 함수. *default*는 *iterable*이 비었을 때의 반환값

`min(iterable, *[,key, default])`

`min(arg1, arg2, *args[,key])`: *iterable*에서 가장 작은 항목이나 두 개 이상의 인수 중 가장 작은 것을 반환. *key*는 순서 함수. *default*는 *iterable*이 비었을 때의 반환값

Wrap-Up

❖ Day2. Wrap-up 설문

■ <https://forms.gle/AxtSLQah425r8sWG7>