

PYTHON BASIC

DATA KUBWA >

1. 프로그래밍 일반

DATA
KUBWA

1. 파이썬 프로그래밍 소개

- “파이썬은 널리 쓰이는 범용, 고급 언어이다. 파이썬의 설계 철학은 코드 가독성에 중점을 두고 있으며 파이썬의 문법은 프로그래머가(C와 같은 언어에서 표현 가능한 것보다도) 더 적은 코드로도 자신의 생각을 표현하도록 한다. 파이썬은 프로그램의 크기에 상관없이 명확하게 프로그램 할 수 있는 구성 요소들을 제공한다.” – 위키피디아
- 1991년 네덜란드 국립 연구소의 Guido Van Rossum에 의해 발표 (현 2019년, 28년의 역사)
- Python이라는 이름은 코미디 프로그램 “Monty Python’s Flying Circus”에서 유래
- Python의 사전적인 의미는 고대 신화의 뱀
- 2000년 Python 2 발표, 2008년 Python 3 발표
- 데이터 처리 업무에 두각
- 데이터 분석 애플리케이션 개발을 위한 최고의 언어

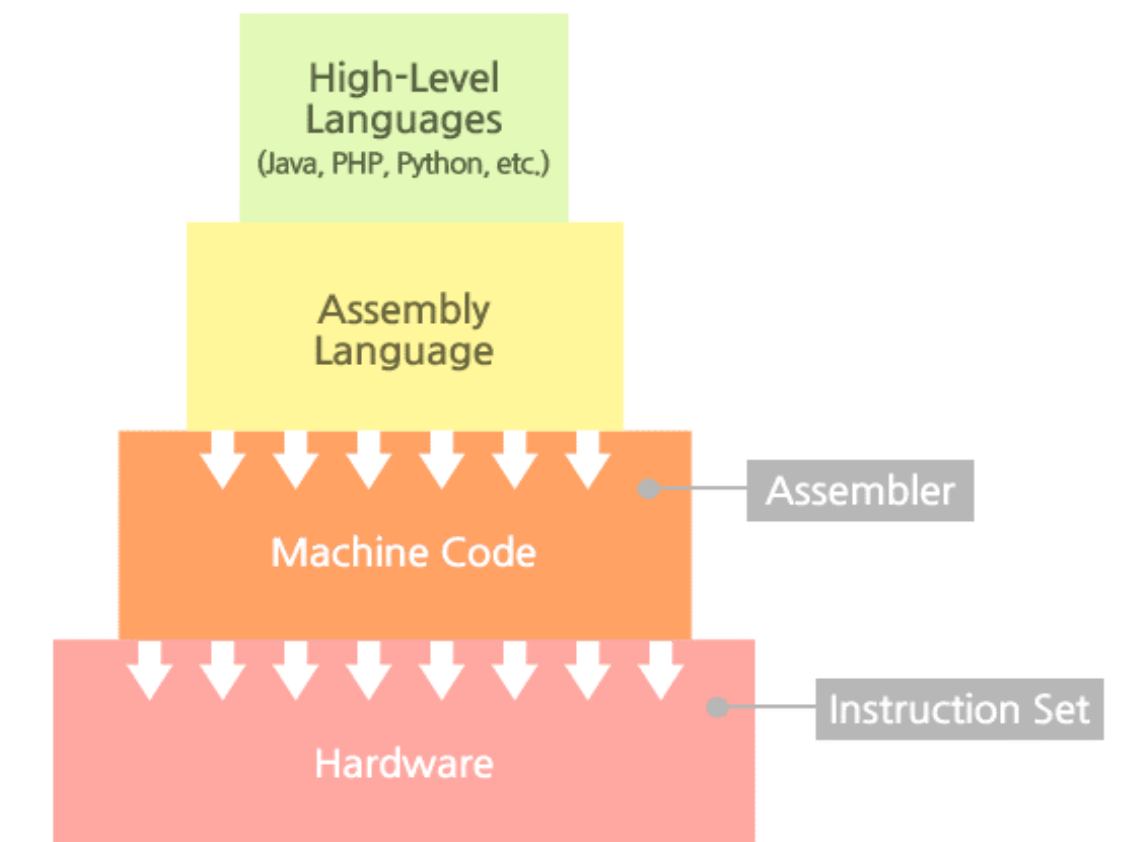


1.2 파이썬 언어의 특징

```
if 7 in [1,3,5,7,9]: print("7이 있습니다")
```

고급 프로그래밍 언어

구분	저급 언어 (Low level language)
장점	컴퓨터가 직접 이해하므로 실행이 빠르고 강력하다 시스템을 세부적으로 조작 할 수 있다
단점	사람이 이해하기 어려우며 사용이 어렵다 사용범위가 제한적이다
예	기계어, 어셈블리어



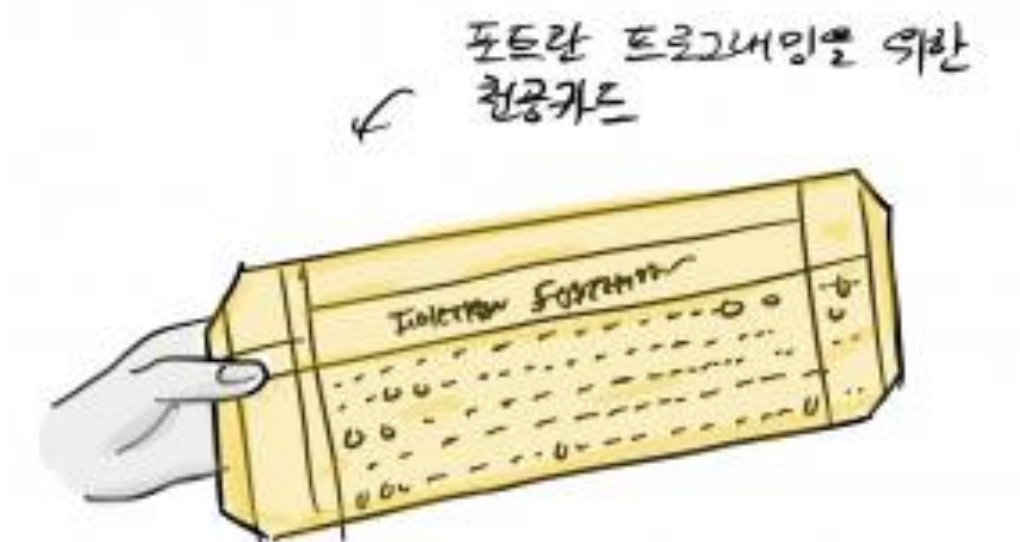
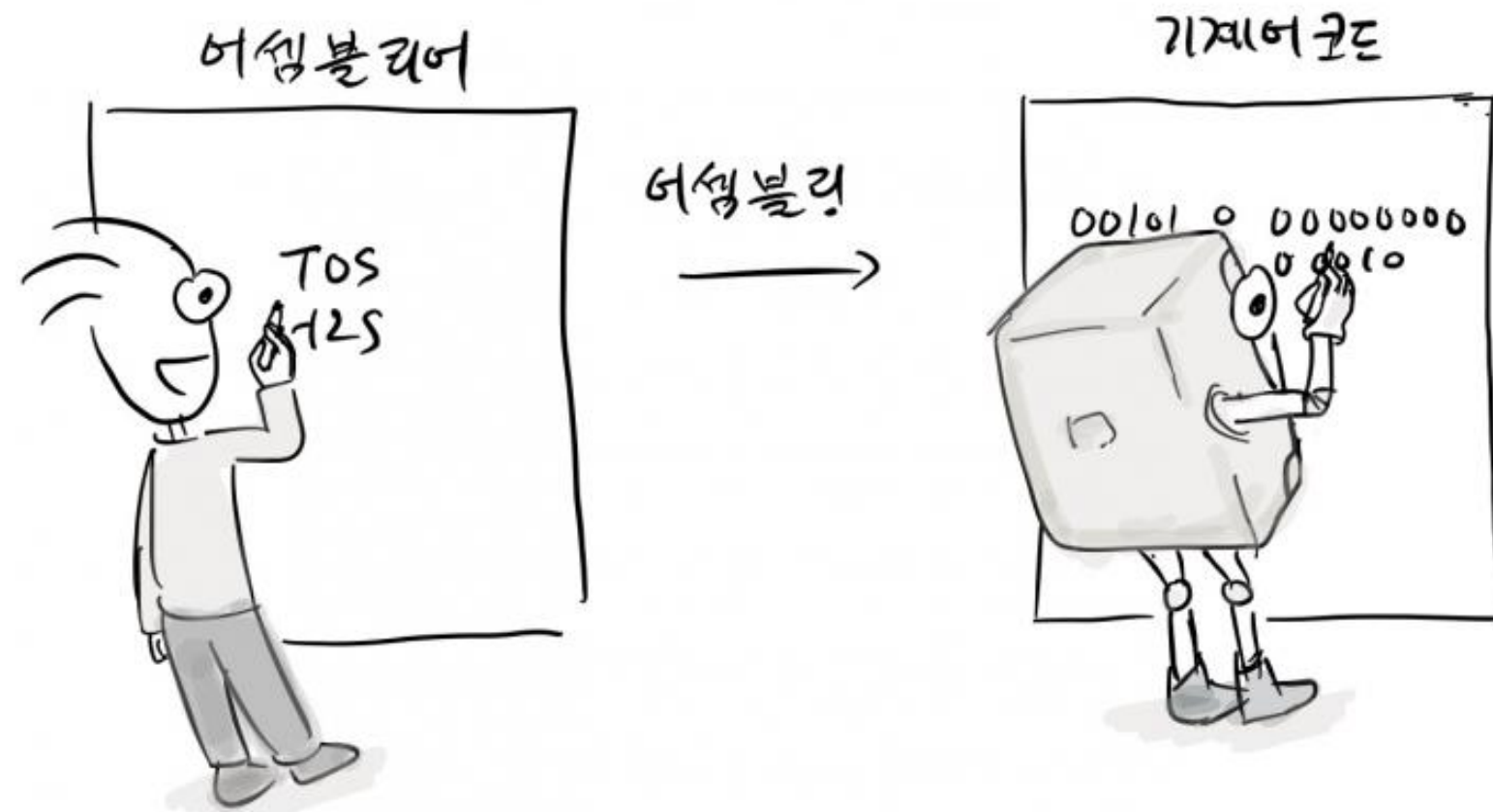
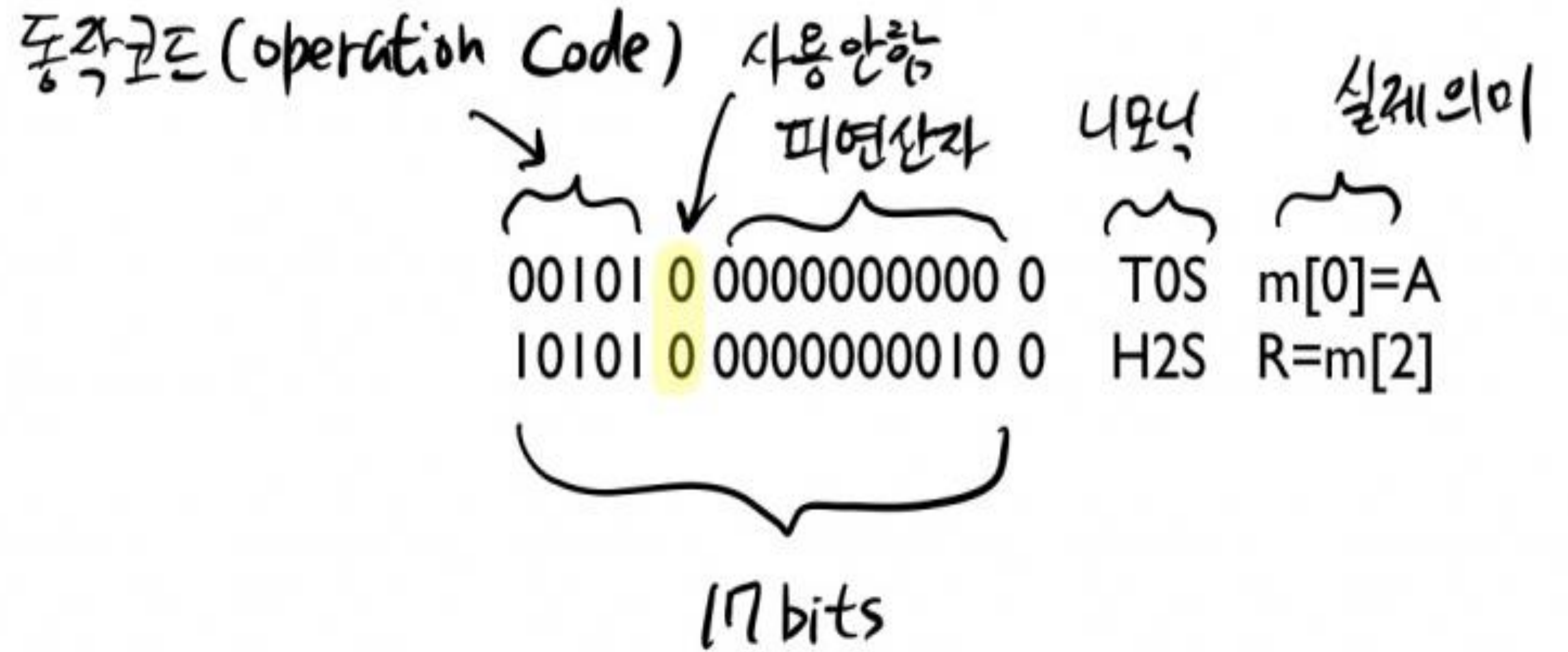
고급 프로그래밍 언어
(High level programming language)

사람이 이해하기 쉬우므로 프로그램의 작성이 쉽고
작성된 프로그램이 읽기 쉽다
오류의 수정이 쉽다

저급 언어에 비해 실행 속도가 느리다
번역이라는 추가 작업이 필요하다

C, C++, JAVA, Python, PHP, C# 등

1.2 파이썬 언어의 특징

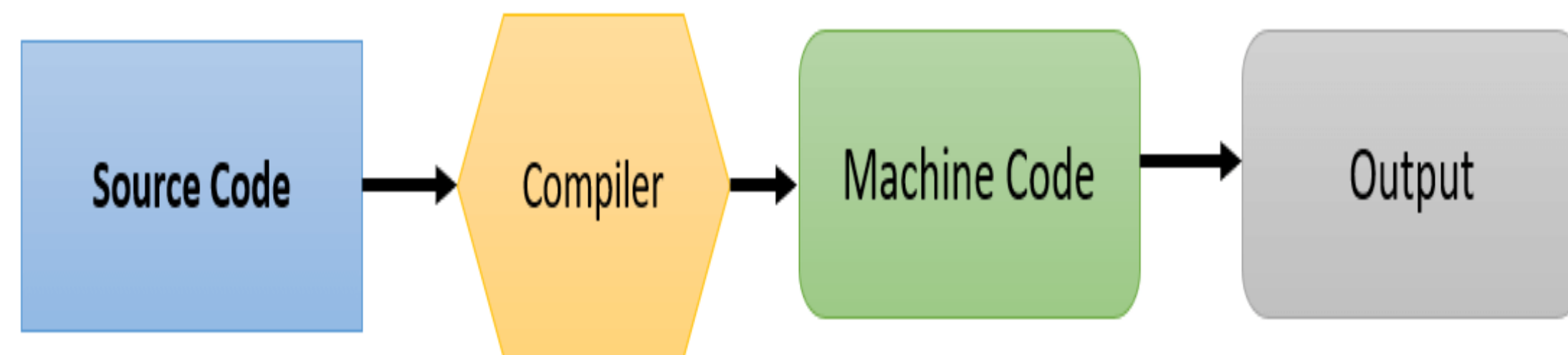


1.2 파이썬 언어의 특징

- 인터프리터 방식의 언어

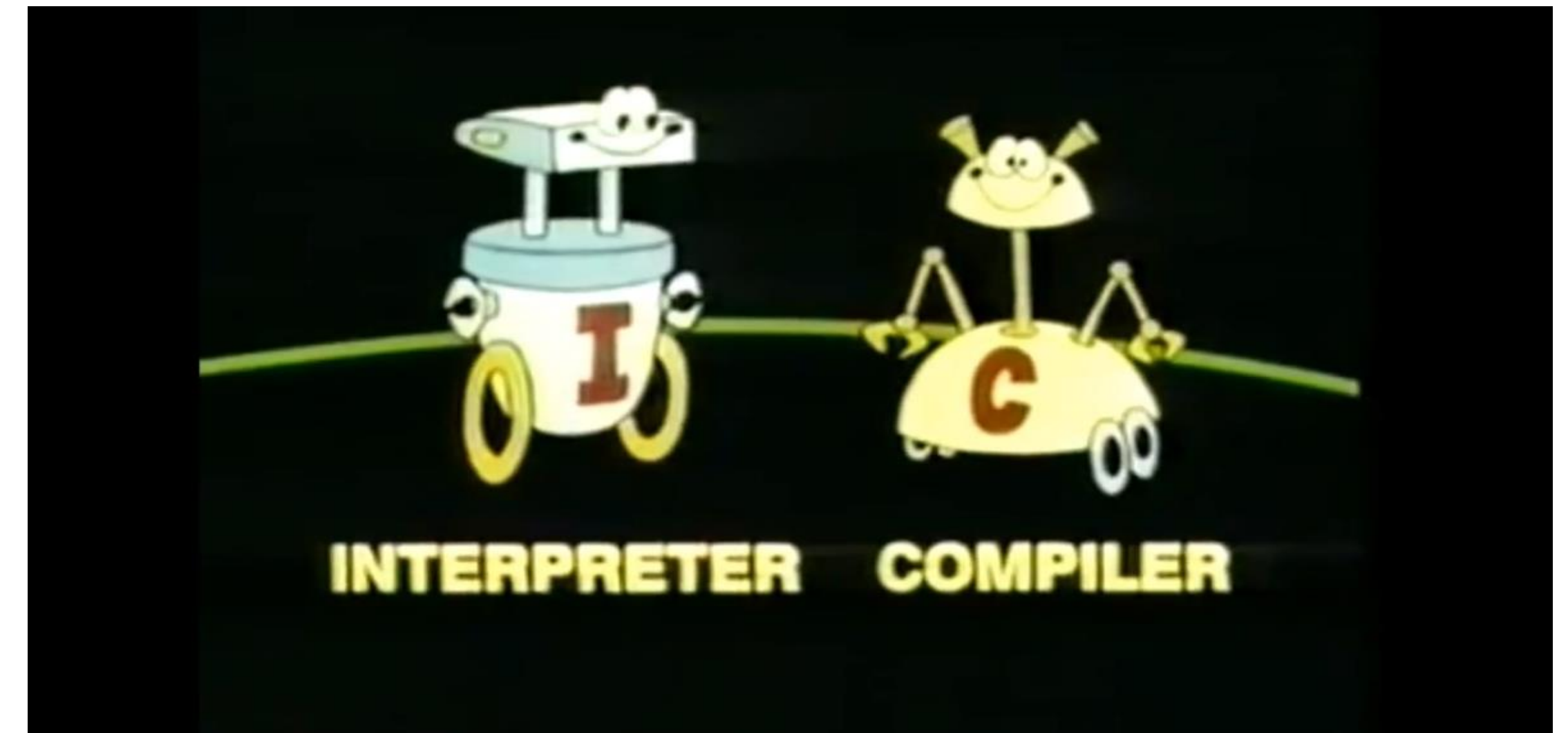
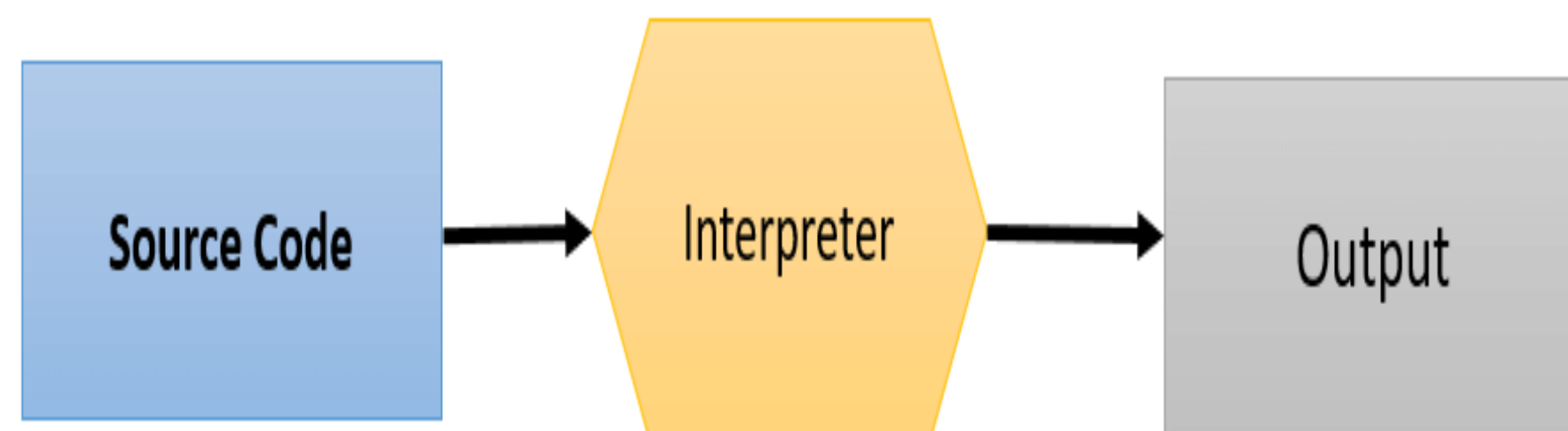
<https://www.youtube.com/watch?v=e4ax90XmUBc>

How Compiler Works



© guru99.com

How Interpreter Works



INTERPRETER	COMPILER
-RUNS SLOWLY	-EXTRA PREPARATION TIME
-STARTS RIGHT AWAY	-BUT THEN PROGRAM RUNS VERY QUICKLY
-LETS YOU SEE RESULTS	

1.2 파이썬 언어의 특징

- 객체 지향 프로그래밍
- (vs. procedure oriented programming)

```
class Box:  
    def __init__(self, length, breadth, height):  
        self.length = length  
        self.breadth = breadth  
        self.height = height  
  
    def getVolume(self):  
        vol = self.length * self.breadth * self.height  
        return vol
```

```
mybox = Box(300, 350, 210)  
volume = mybox.getVolume  
print(volume)
```

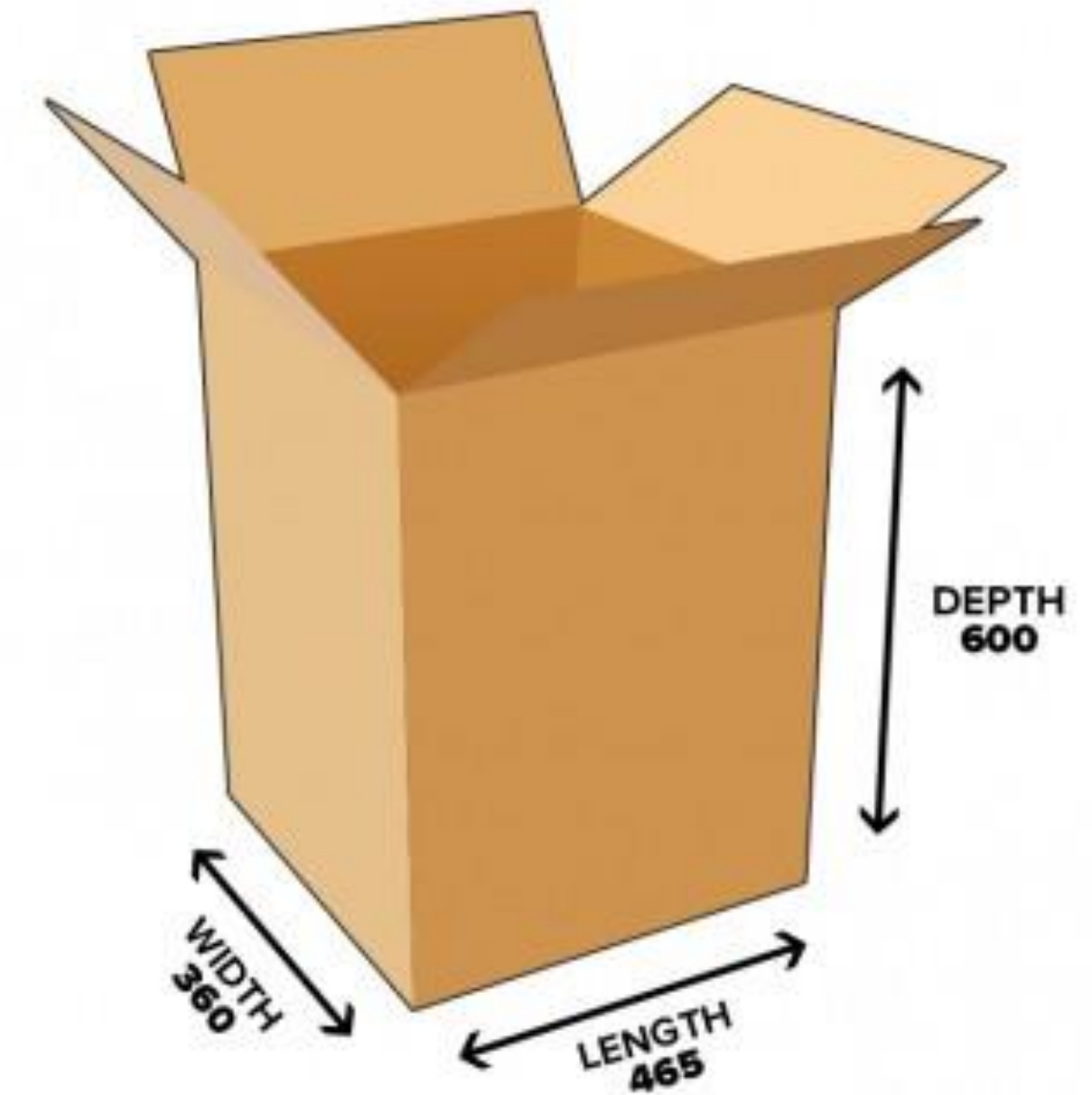


```
length = 300  
breadth = 350  
height = 210
```

```
volume = length * breadth * height  
print(volume)
```

1.2 파이썬 언어의 특징

- 객체 지향 프로그래밍
- (vs. procedure oriented programming)



```
yourbox = Box(465, 360, 600)
volume = yourbox.getVolume

print(volume)
```

```
length = 465
breadth = 360
height = 600

volume = length * breadth * height

print(volume)
```


1.2 파이썬 언어의 특징

- Free



1.3 파이썬 설치

- <https://www.python.org/downloads/>

[About](#) [Downloads](#) [Documentation](#) [Community](#) [Success Stories](#) [News](#) [Events](#)

Download the latest version for Windows

[Download Python 3.7.3](#)

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Pre-releases](#), [Docker images](#)

Looking for Python 2.7? See below for specific releases



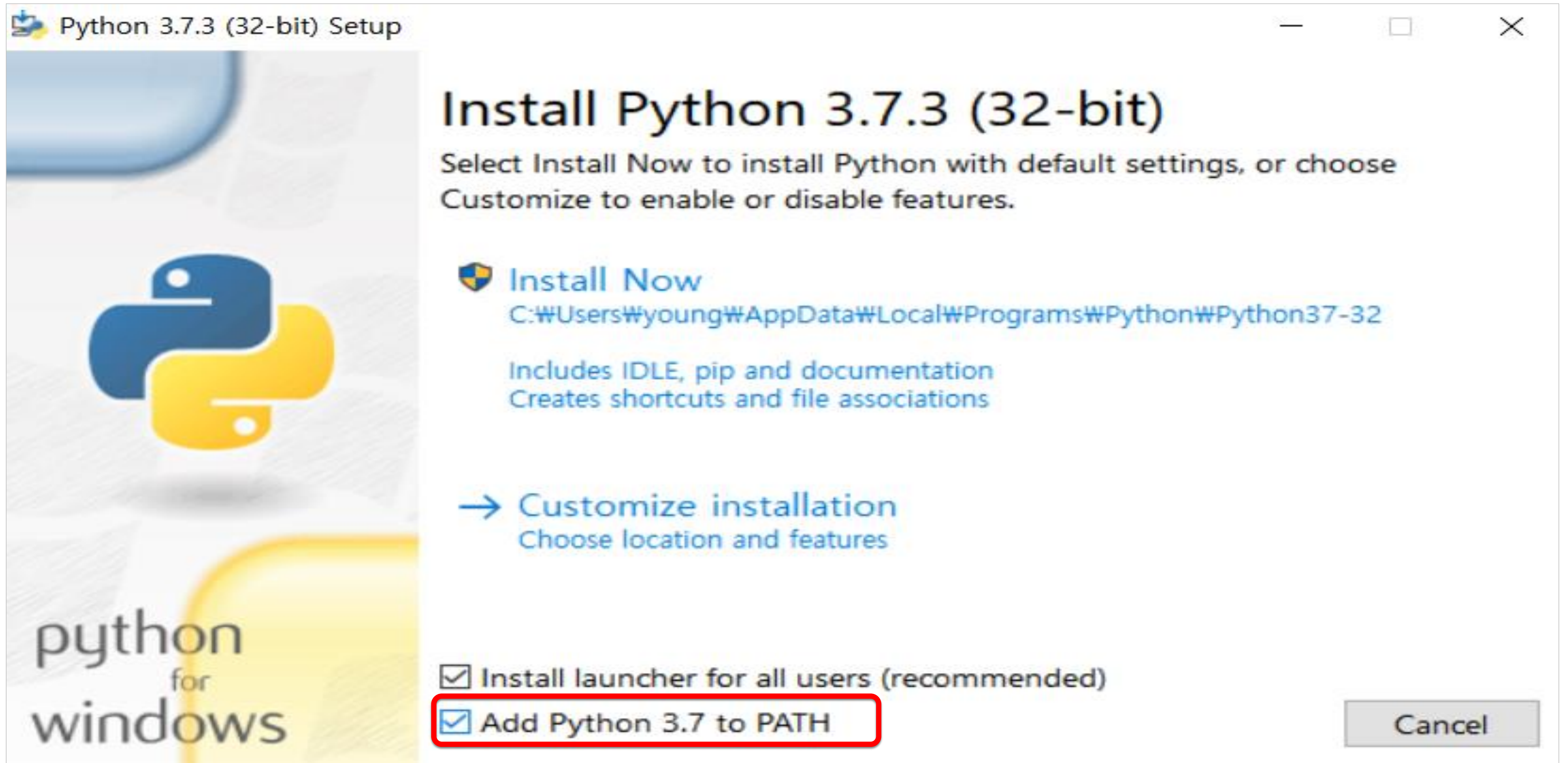
Contribute to the PSF by Purchasing a PyCharm License. All proceeds benefit the PSF. [Donate Now](#)

Looking for a specific release?

Python releases by version number:

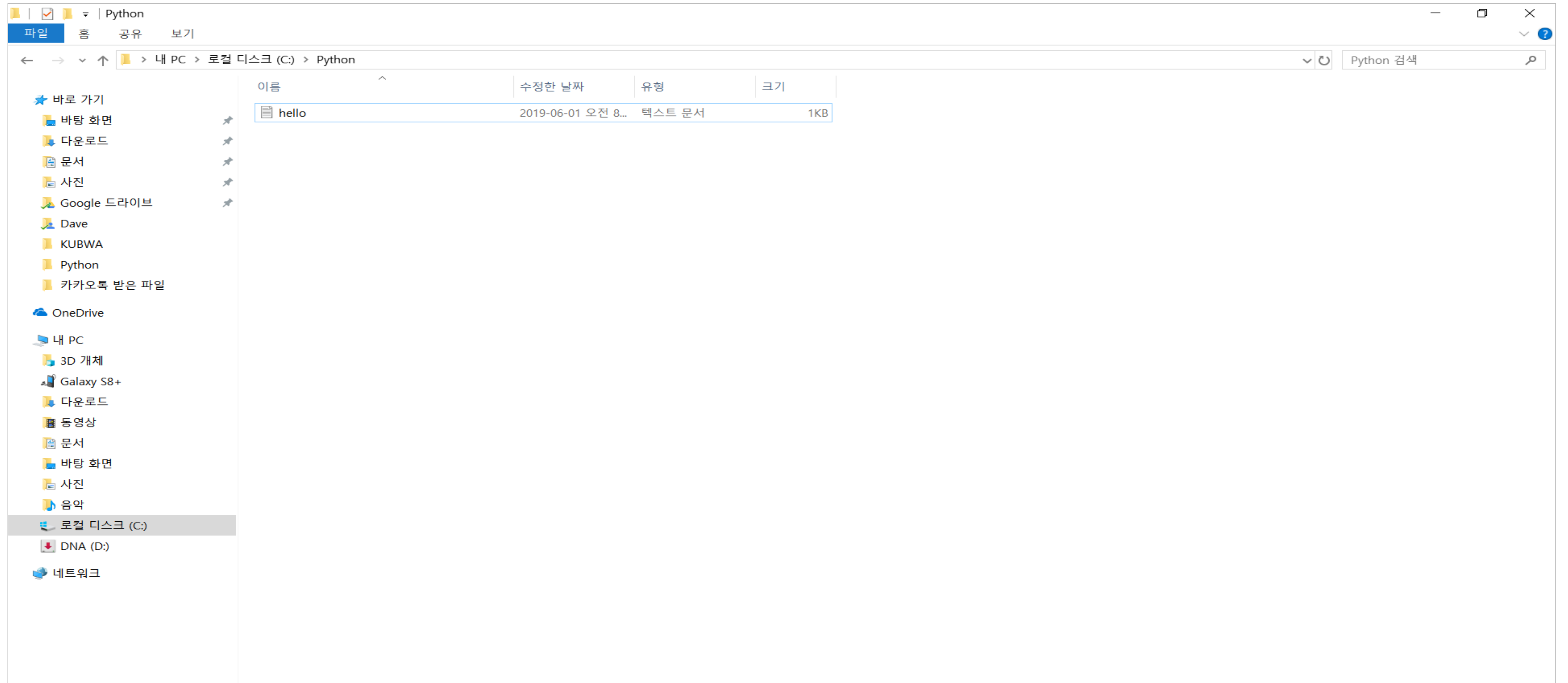
Release version	Release date
Click for more	

1.3 파이썬 설치



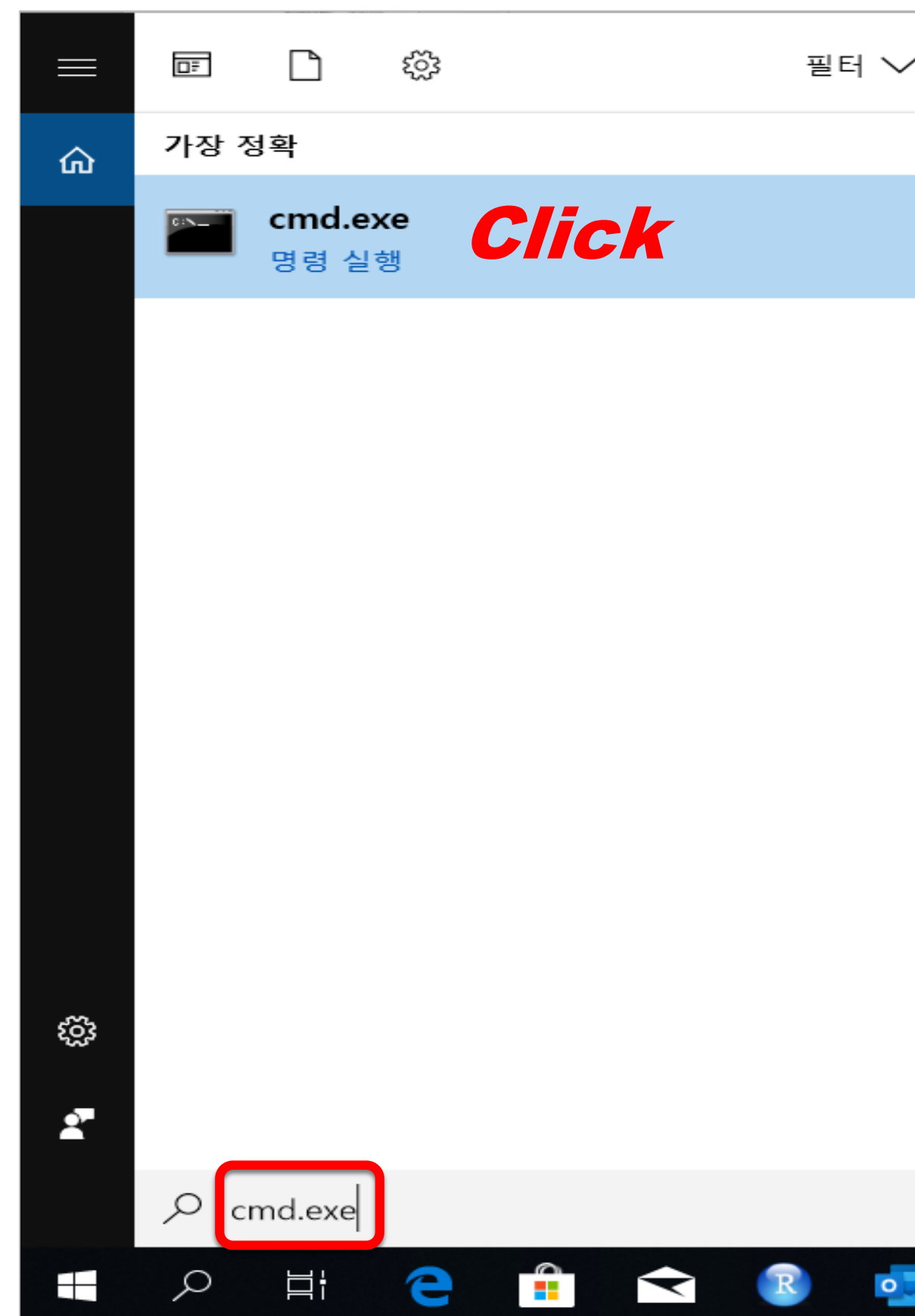
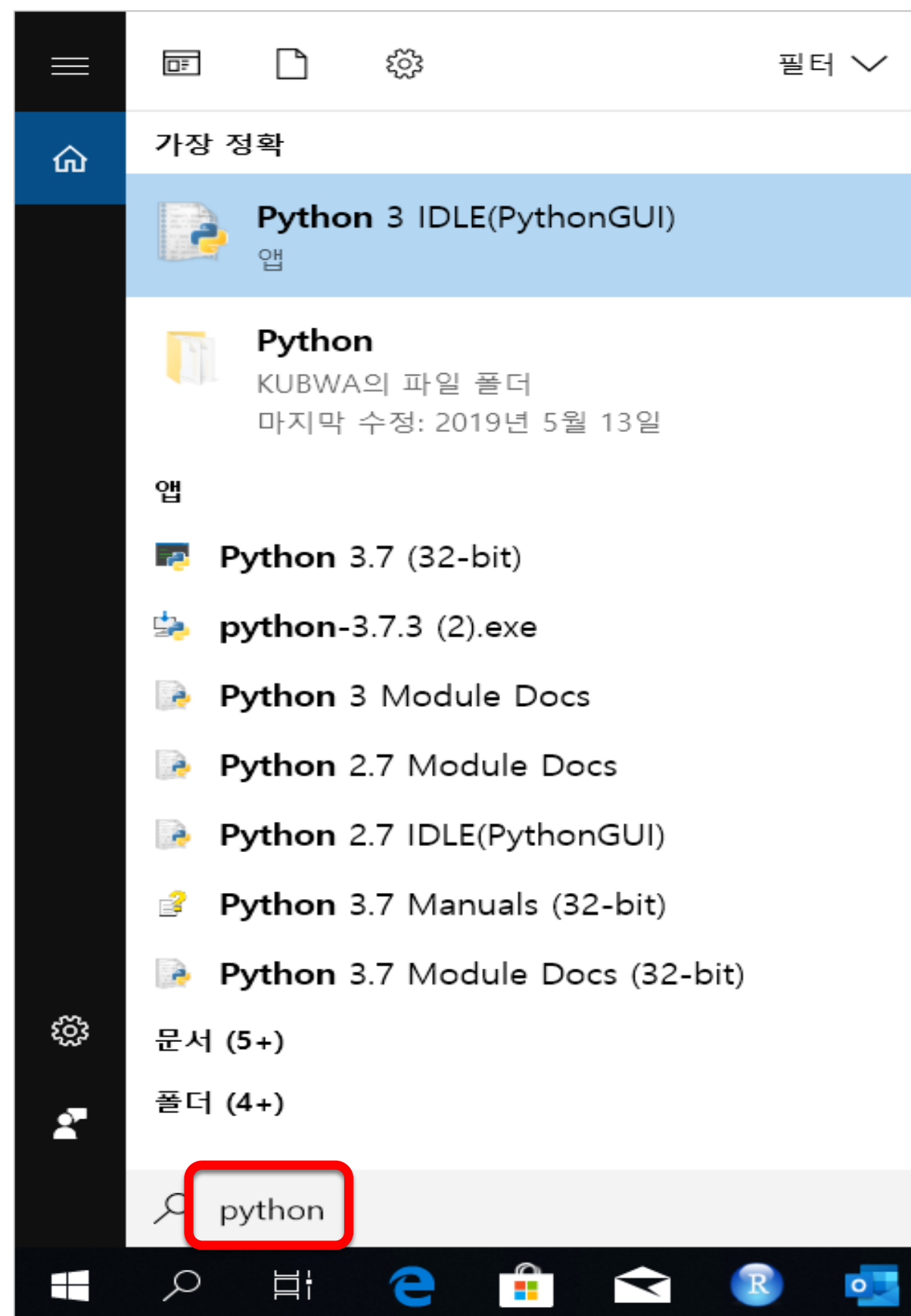
1.3 파이썬 설치

• C:\Python



1.3 파이썬 설치

• 파이썬 설치 확인



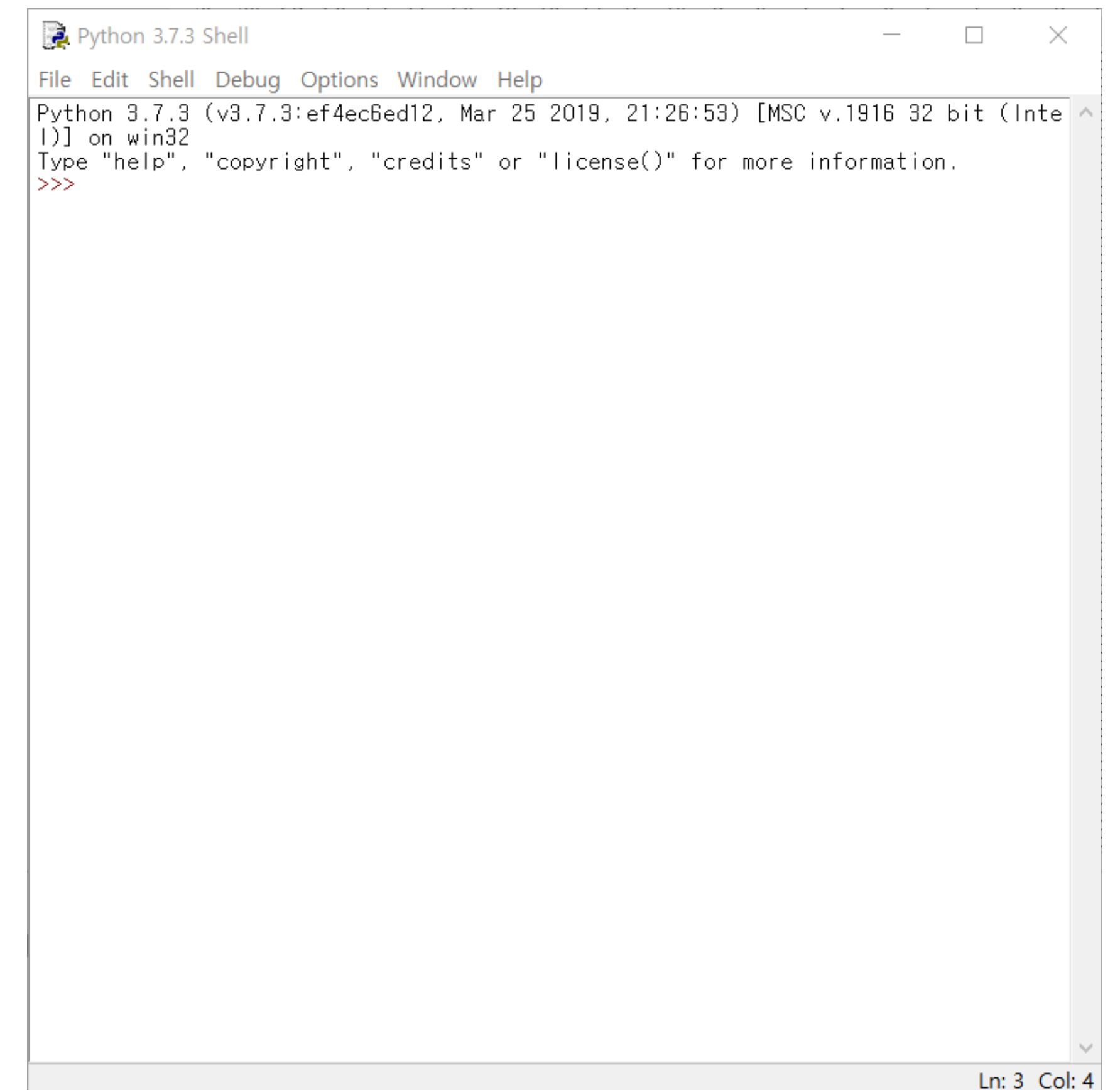
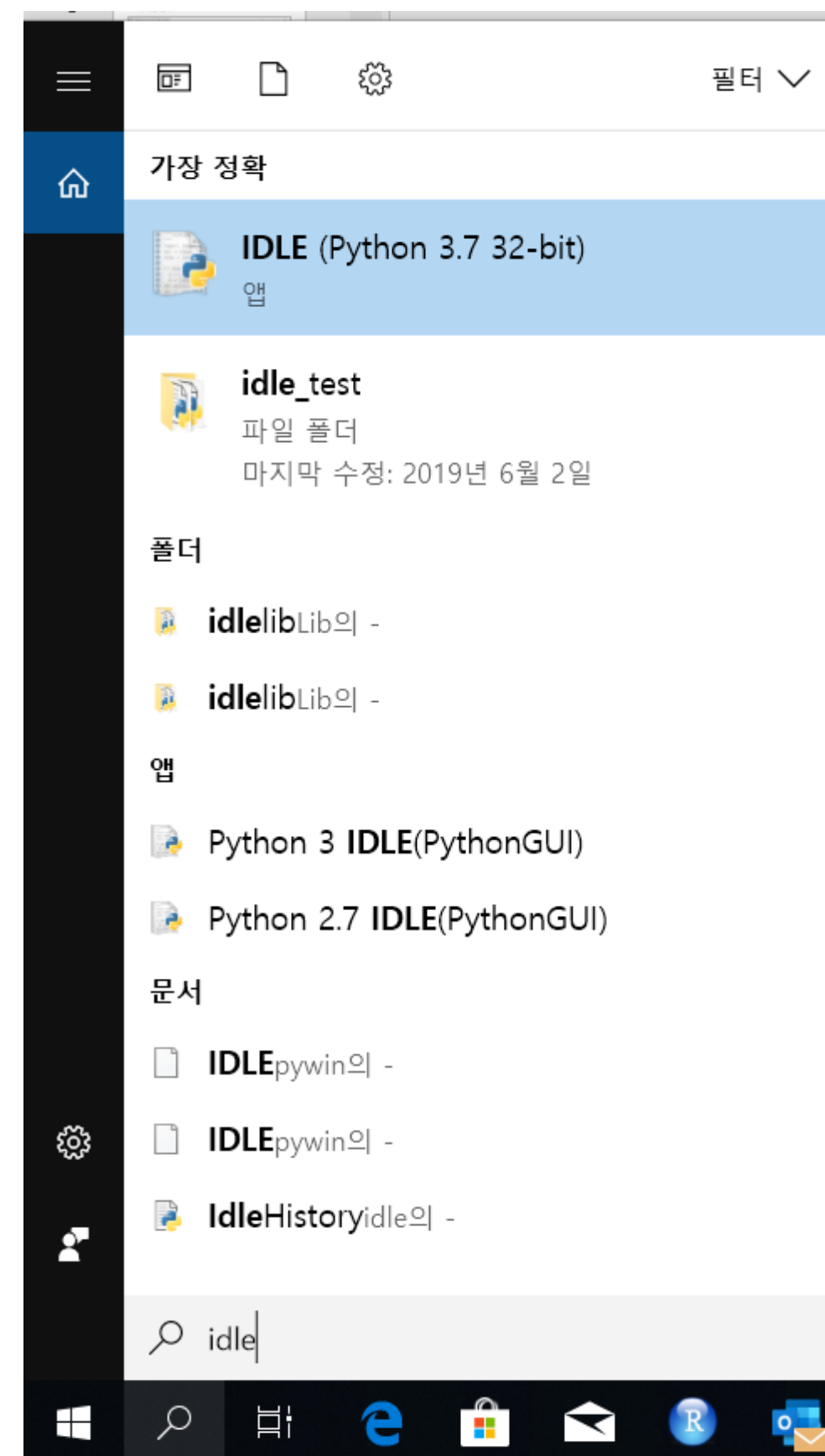
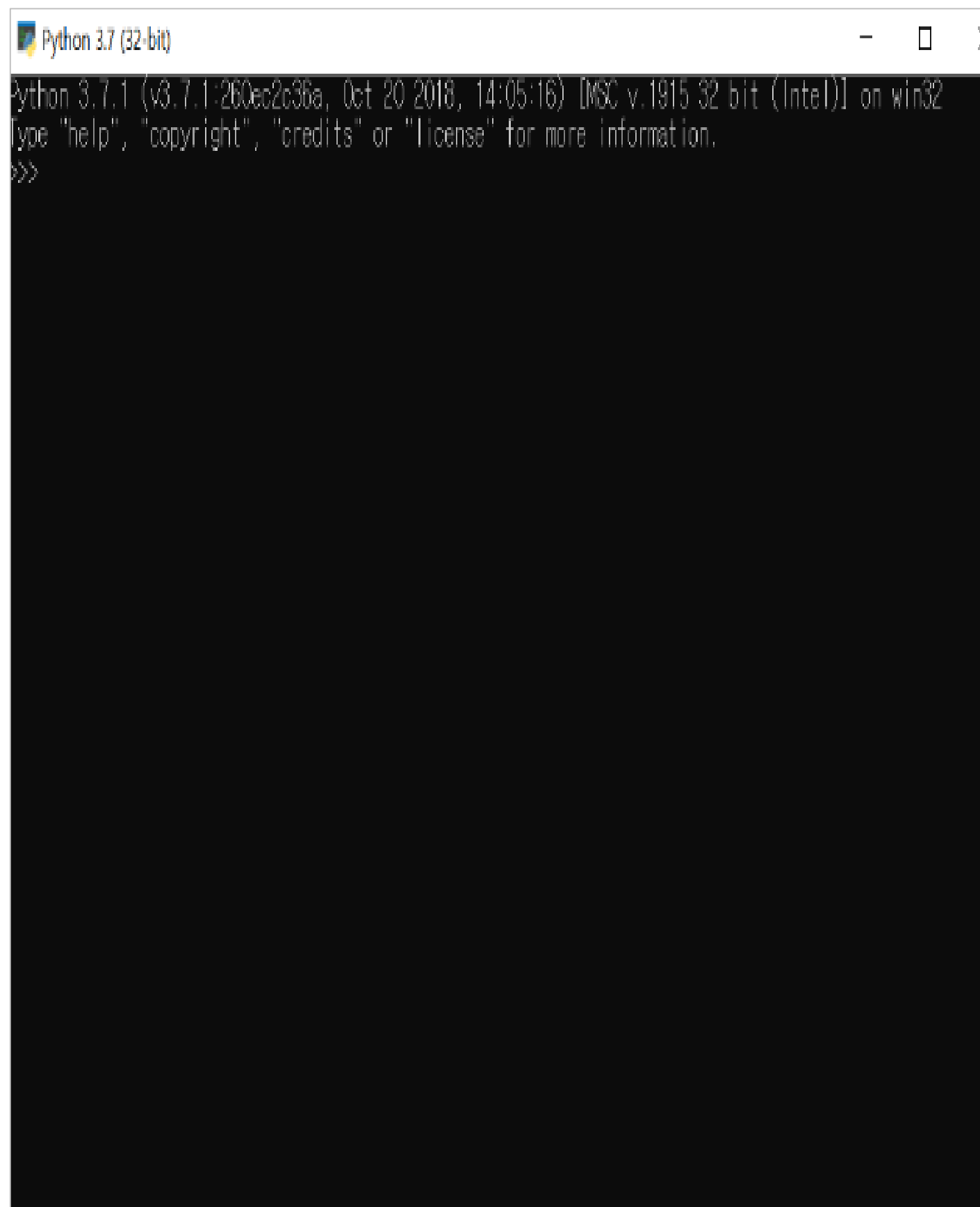
1. 프로그래밍 일반

DATA
KUBWA

II. PYTHON을 위한 JUPYTER NOTEBOOK 활용

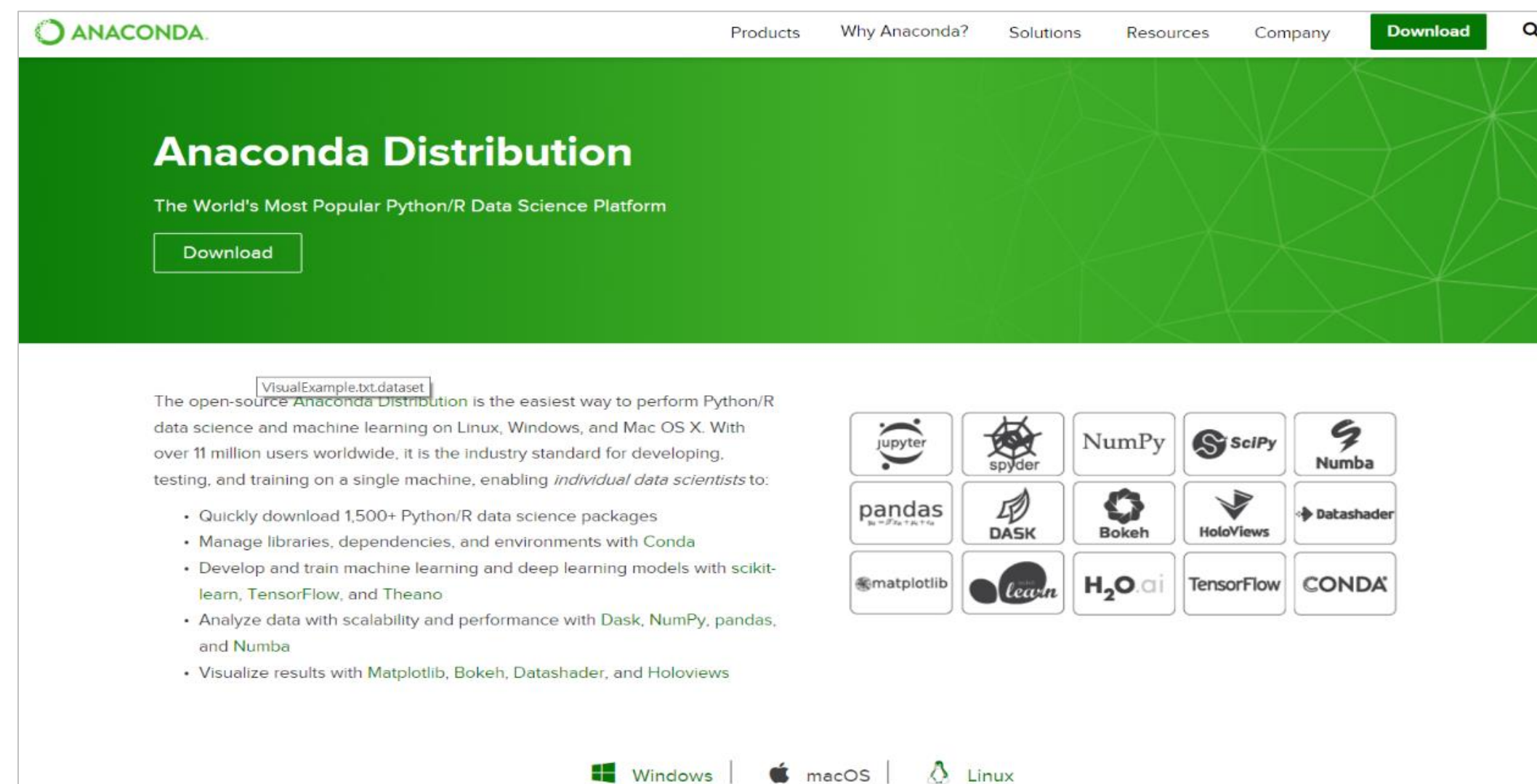
2.1 에디터

- Python IDLE (Integrated Development and Learning Environment)



2.2 ANACONDA

- Anaconda **이**용 Jupyter Notebook 설치
 - 추천: 파이참, 비주얼 스튜디오 코드, 주피터 노트북 등
 - 파이썬/R 활용 데이터 분석 및 시각화
 - <https://www.anaconda.com/distribution/#download-section>
 - Jupyter Notebook 설치



Python 3.7 version

Download

64-Bit Graphical Installer (662 MB)

32-Bit Graphical Installer (546 MB)

2.3 JUPYTER NOTEBOOK 활용

• # 주피터 노트북 실행

- 시작 > Anaconda3 (64-bit) > Jupyter Notebook
- 명령 프롬프트를 실행(윈도우 키+R을 누른 뒤 cmd를 입력)한 뒤 다음 명령을 입력합니다.
- C:\Users\<사용자계정>\Anaconda3\python.exe -m notebook



The screenshot shows the Jupyter Notebook web interface. At the top, there's a header with the Jupyter logo and 'Quit' and 'Logout' buttons. Below the header, there are tabs for 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, showing a file browser. The browser displays a list of files and folders in a table with columns for 'Name', 'Last Modified', and 'File size'. The files listed include '3D Objects', 'Anaconda3', 'Contacts', 'Desktop', 'Documents', 'Downloads', 'Favorites', 'Google 드라이브', 'Links', 'Music', 'OneDrive', 'Pictures', 'PycharmProjects', and 'Saved Games'. Each file has a checkbox on the left and a timestamp on the right.

Name	Last Modified	File size
<input type="checkbox"/> 0		
<input type="checkbox"/> 3D Objects	6달 전	
<input type="checkbox"/> Anaconda3	11시간 전	
<input type="checkbox"/> Contacts	6달 전	
<input type="checkbox"/> Desktop	하루 전	
<input type="checkbox"/> Documents	7일 전	
<input type="checkbox"/> Downloads	11시간 전	
<input type="checkbox"/> Favorites	4달 전	
<input type="checkbox"/> Google 드라이브	18시간 전	
<input type="checkbox"/> Links	24일 전	
<input type="checkbox"/> Music	3달 전	
<input type="checkbox"/> OneDrive	18시간 전	
<input type="checkbox"/> Pictures	16일 전	
<input type="checkbox"/> PycharmProjects	5달 전	
<input type="checkbox"/> Saved Games	6달 전	

2.3 JUPYTER NOTEBOOK 활용

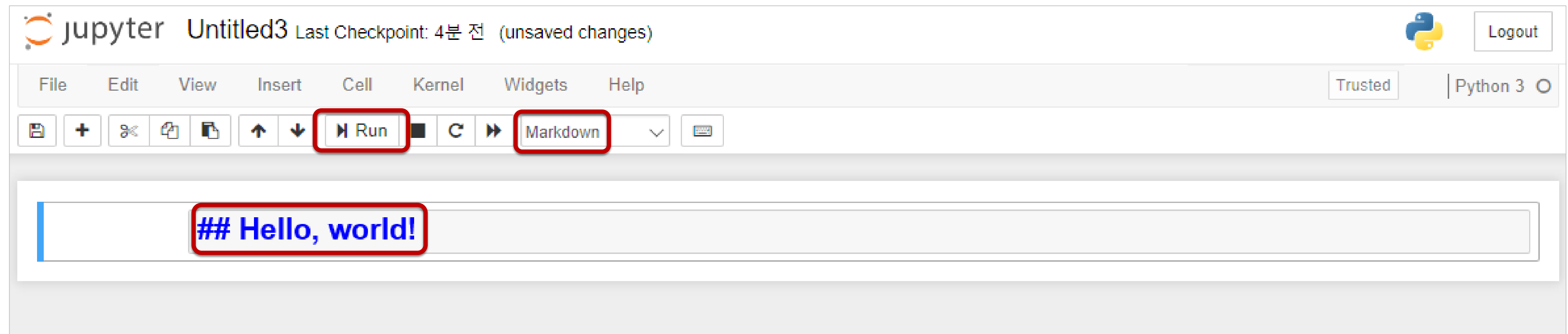
- # 파이썬 노트북 만들기
 - 오른쪽 New 버튼을 클릭
 - Python 3을 클릭
 - 새 노트북 화면



2.3 JUPYTER NOTEBOOK 활용

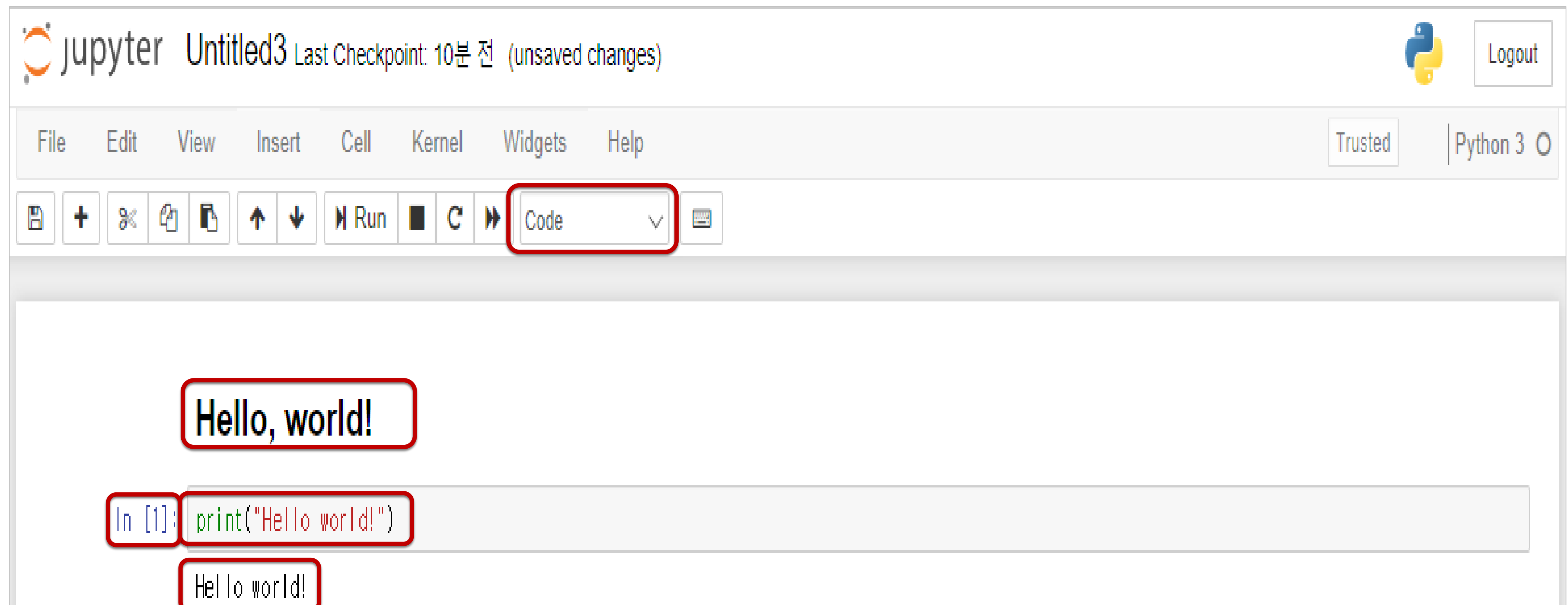
- # 설명 추가

- Markdown 선택
- ## Hello, world! 빈 칸에 입력
- 실행 (Run click or Shift + enter)



2.3 JUPYTER NOTEBOOK 활용

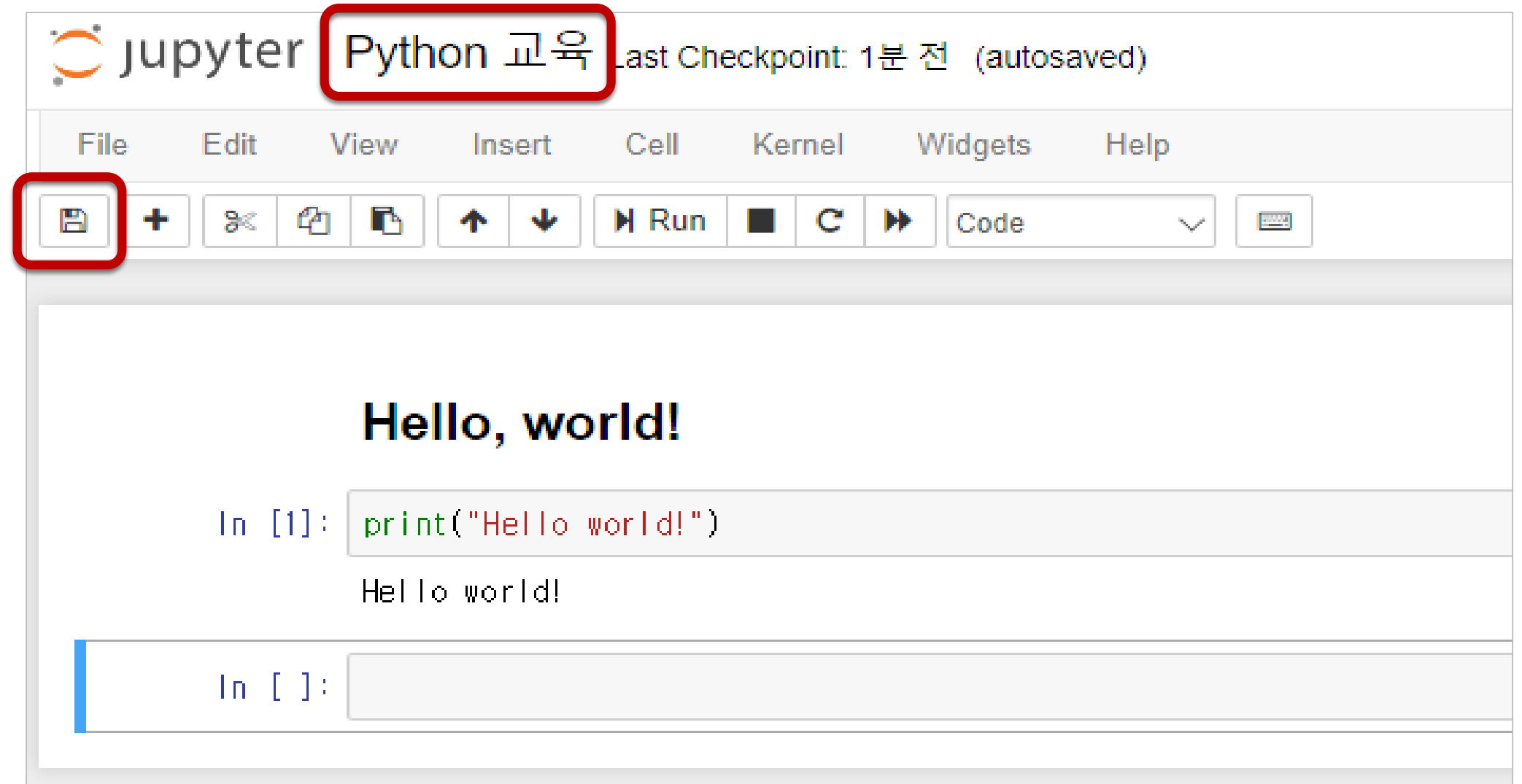
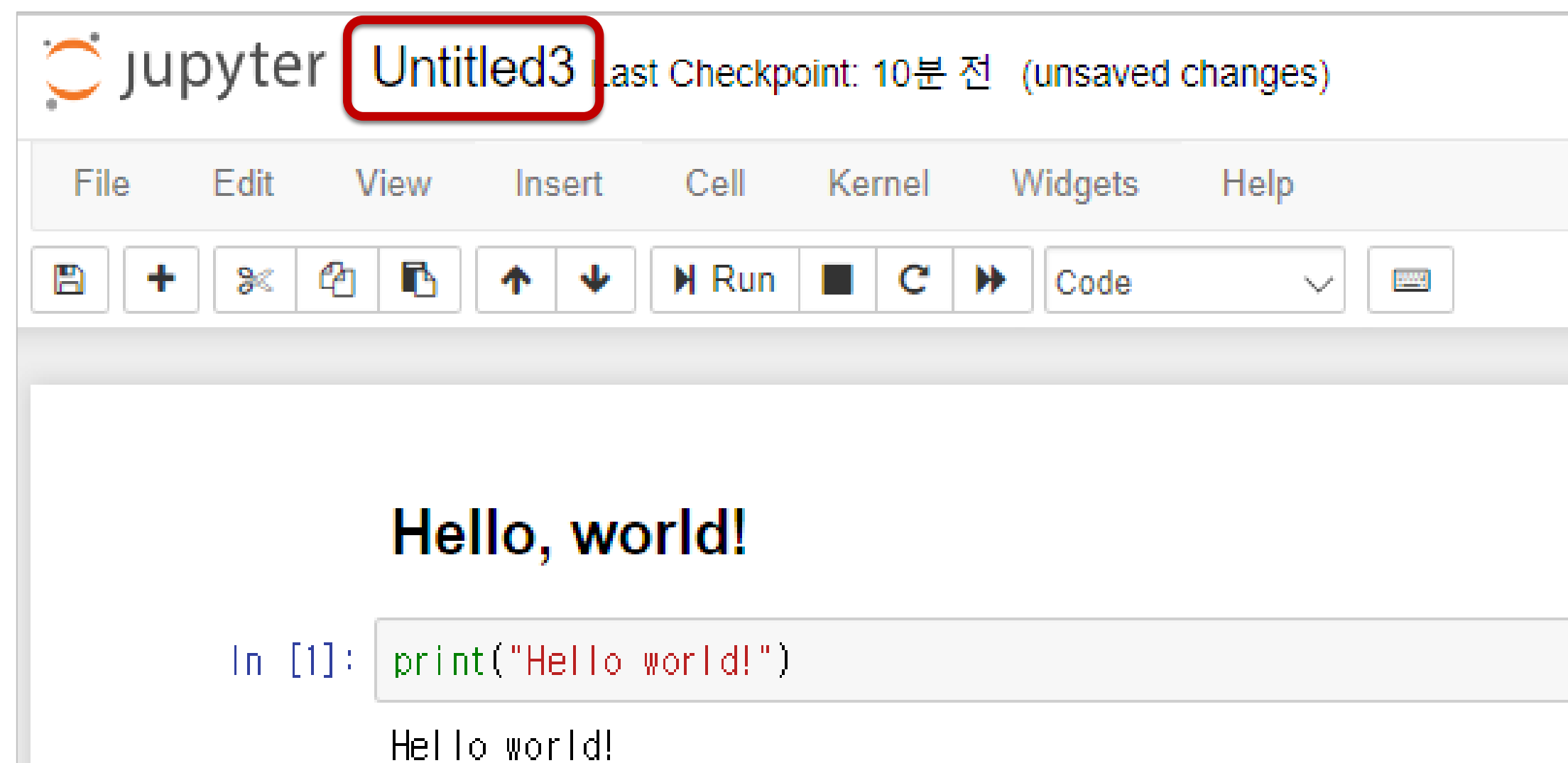
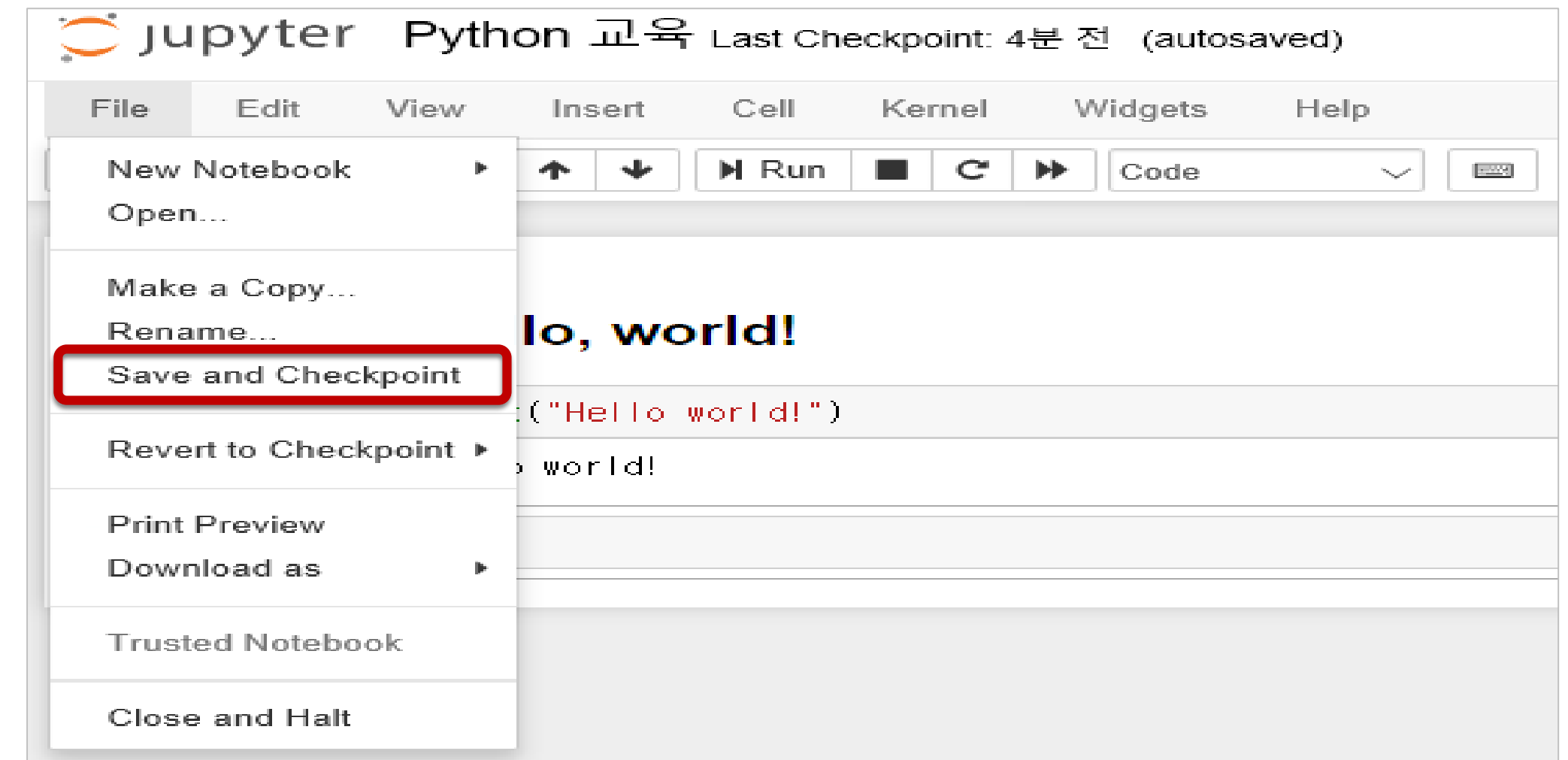
- # 파이썬 코드 입력



2.3 JUPYTER NOTEBOOK 활용

노트북 제목 변경

- C:\Users\사용자계정 폴더



2. Python 이론

DATA
KUBWA

III. 데이터 타입 종류

3.1 데이터 타입 종류

자료형	타입	설명	예
숫자	int	정수형 데이터	1, 100, -511, 1004
	float	실수형 데이터	10.25, 3.141592, 210e10
	complex	복소수형 데이터	8+4j
불	bool	참/거짓	True
문자열	str	문자열	'LG Electronics'
리스트	list	리스트, 순서가 있는 배열, 수정/추가/삭제가 가능한 자료 구조	[1, 2, 3, 'a', 'b']
튜플	tuple	튜플, 순서가 있는 배열, 수정/추가/삭제가 불가능한 자료 구조	(1, 2, 3, 'a', 'b')
딕셔너리	dict	사전, {key: value}로 구성되어 있는 자료 구조	{'Math': 99, 'English': 88, 'Korean': 78}
집합	set	집합, {key}로 구성되어 있는 자료 구조	{'a', 'b', 'c'}

- # 10진수

- a = 365
- print(a)

- # 2진수

- b = 0b101101101
- print(b)

- # 8진수

- c = 0o555
- print(c)

- # 16진수

- d = 0x16d
- print(d)

3.3 실수(FLOAT) 타입

- fa = 3.14
- print(fa)

- fc = 3.1415e2
- print(fc)

- fb = -3.14
- print(fb)

- fd = 3.1415e-2
- print(fd)
-

3.4 복소수 (COMPLEX) 타입

- $a = 6 + 4j$
- `print(a)`

#변수.real

- `a.real`

- `b = complex(6,4)`
- `print(b)`

#변수.imag

- `a.imag`

3.5 불(BOOL) 타입

- `ba = True`
- `bb = False`

- `ba and bb`

- `6 == 6`

- `6 > 4`

- `6 < 4`

- `bool("ABC")`

- `bool("")`

- `bool([1,2,3])`

- `bool([])`

- `bool()`

- `bool({})`

- `bool(1)`

- `bool(0)`

- `bool(None)`

문자열 생성

- `sa = 'Hello World'`
- `print(sa)`
- `sb = "one way of writing a string"`
- `print(sb)`
- `sc = """another way"""`
- `print(sc)`
- `sd = "This is a longer string that spans multiple lines"`
- `print(sd)`
- `se = ""`
- This is a longer string that
- spans multiple lines
- `""`
- `print(se)`

3.6 문자열(STR) 타입

- a = 'Tom's favorite food is pizza'
- print(a)

- b = "Tom's favorite food is pizza"
- print(b)

- c = "You are right." I said.
- print(c)

- d = `"You are right." I said.`
- print(d)

- e = 'Tom\'s favorite food is pizza'
- print(e)

- # Escape code

- `s = '12\n34'`

- `print(s)`

- `sd = '`

- This is a longer string that

- spans multiple lines

- `'`

- `print(sd)`

- `se = '''`

- This is a longer string that

- spans multiple lines

- `'''`

- `print(se)`

- `sf = 'this\has\no\special\characters'`

- `print(sf)`

- `sg = r'this\has\no\special\characters'`

- `print(sg)`

3.6 문자열(STR) 타입

- # Escape code
- s = '12\n34'
- print(s)

- sd = '
This is a longer string that
spans multiple lines
'
- print(sd)

코드	설명
\n	줄바꿈
\t	수평 탭
\\	문자 “\”
\'	단일인용부호(')
\”	이중인용부호(“)
\r	캐리지 리턴
\f	폼 피드
\a	벨 소리
\b	백 스페이스
\000	널문자

- # 문자열 연산

- a = 'this is the first half '
- b = 'and this is the second half '
- print(a + b)

- a = '='
- print(a * 50)

- # 문자열 길이

- a = 'this is the first half'
- len(a)

- #문자열 인덱싱

- a = 'this is a string'
- print(a[2])
- print(a[0])
- print(a[-1])
- print(a[-0])

- # Slicing

- b = "20010331Rainy"
- print(b[:8]) #print(b[0:8])
- print(b[8:])
- print(b[:])
- print(b[:-5])

3.6 문자열(STR) 타입

- # 문자열 수정
- a = 'this is a string'
- a[10:] = 'rope'
- b = 'rope'
- a[:10] + b
- c = a.replace('string', 'rope')
- c

3.6 문자열(STR) 타입

- # 문자열 format
- “I have %d children.” %3
- “I have %s children.” %”three”
- e = 3
- “I have %d children.” %e
- “I have %d children. I don’t have time %s day.” %(3, “all”)
- # %s, %c, %d, %f, %o, %x,
- “I have %s children. I don’t have time %s day.” %(3, “all”)
- “I have %s children. I don’t have time %s day.” %(3.14, “all”)

3.6 문자열(STR) 타입

- "This is %s% True!" %100
- "This is %s%% True!" %100
- "%10s" %"Good"
- "%-10s" %"Good"
- "%0.5f" %3.141592
- "%20.5f" %3.141592
- "I have {0} children".format(3)
- "I have {0} children".format("three")
- g = 3
- "I have {0} children".format(g)
- "I have {0} children. I don't have time {1} day.".format(3, "all")
- "I have {1} children. I don't have time {0} day.".format(3, "all")
- "I have {a} children. I don't have time {b} day.".format(a=3, b=2)

3.6 문자열(STR) 타입

- "{0:<20}".format("good")
- "{0:>20}".format("good")
- "{0:^20}".format("good")
- "{0:*^20}".format("good")
- "{0:@<20}".format("good")
- '{0:.2f} {1} are worth US\${2}'.format(4.5560, 'Argentine Pesos', 1)
- '{{우리 집}}에 {0} 놀러와요'.format('자주')
-

- #f 문자열 format

- a = 3
- b = 'all'

- f"I have {a} children. I don't have time {b} day."

- f"I have {a+2} children. I don't have time {b} day."

- f{"good":<20}' #{0:<20}'.format("good")

- f{"good":>20}' #{0:>20}'.format("good")

- f{"good":^20}' #{0:^20}'.format("good")

- f{"good":*^20}' #{0:*^20}'.format("good")

- c = 3.141592

- f'{c:0.2f}'

- f'{c:30.2f}'

- f'{{ good }}'{ good }'

- # 문자열 내장함수

- a = "Korean culture"

- a.count('u')

- a.find('u')

- a.find('k')

- a.find('K')

- a.index('u')

- a.index('k')

- ", ".join('Korean culture')

- “,”.join(123456789)

- # 문자열 내장함수

- a.upper()

- a.lower()

- b = ' Korean culture '

- b.strip()

- b.lstrip()

- b.rstrip()

- a.replace("Korean", "American")

- a.split()

- c = "Korean & American"

- c.split("&")

- # 리스트 생성
- a = [] #a = list()
- b = [1, 2, 3, 4, 5]
- c = ['red', 'blue', 'yellow']
- d = [1, 2, 'red', 'blue']
- e = [1, 2, ['red', 'blue']]
- print(a)
- print(b)
- print(c)
- print(d)
- print(e)

- # 인덱싱

- # a = []

- # b = [1, 2, 3, 4, 5]

- b[0]

- b[0] + b[2]

- b[-1]

- a = [1,2,3,4,5,['a','b','c','d'],7,8,9]

- print(a)

- a[0]

- a[-1]

- a[5]

- a[5][2]

- a[5][3]

슬라이싱

- b = [1, 2, 3, 4, 5]

- b[0:4]

- c = "12345"

- c[0:4]

- b[:3]

- b[3:]

- a = [1, 2, 3, 4, 5, ['a', 'b', 'c', 'd'], 7, 8, 9]

- a[0:6]

- a[0:6:1]

- a[0:6:2]

- a[0:6:3]

- a[0:6:4]

- a[0:6:5]

- a[0:6:6]

- a[5]

- a[5][:2]

3.7 리스트(LIST) 타입

- # 연산

- $a = [1, 2, 3, 4]$

- $b = [6, 7, 8, 9]$

- $a + b$

- $a * 2$

- $\text{len}(a)$

- $k = [12345678]$

- $\text{len}(k)$

- $m = '12345678'$

- $\text{len}(m)$

- # 연산
- `c = 'good'` # 문자열
- `b + c` # 리스트 + 문자열

- `c = ['good']`
- `b + c`

- # 수정
- `a[3]`

- `a[3] = 9` # 리스트 수정 가능
- `print(a)`

- `n = '1234'`
- `n[3]`

- `n[3] = '9'` # 문자열 수정 불가

- # function
- a = [1,2,3,4, [a,b,c]]
- b = [6,7,8,9]
- len(a)
- sum(a)
- sum(b)
- max(b)
- min(b)
- del(b[0]) #del b[0]
- print(b)
- b = [6,7,8,9]
- del b[:3]
- print(b)
- str(b[0]) # 숫자형 → 문자형

3.7 리스트(LIST) 타입

- `b = ['6', '7', '8', '9']`
 - `b.append('10')`
 - `print(b)`

 - `b.append(['11', '12'])`
 - `print(b)`
- `c = ['6','9','8','7']`
 - `print(c)`

 - `c.sort()`
 - `print(c)`

 - `d = ['x', 'z', 'y']`
 - `d.sort()`
 - `print(d)`

3.7 리스트(LIST) 타입

- `c = ['6', '9', '8', '7']`
- `c.reverse()`
- `c.index('9')`
- `c.index('5')`
- `c.insert(4, '10')`
- `print(c)`
- `c.insert(0, '5')`
- `print(c)`

3.7 리스트(LIST) 타입

- `e = ['3', '4', '3', '4', '5']`
- `e.remove('4')`
- `print(e)`
- `e.remove('4')`
- `print(e)`
- `e.pop()`
- `print(e)`
- `e.pop(1)`
- `print(e)`
- `e.pop(0)`
- `print(e)`

3.7 리스트(LIST) 타입

- `c = [6, 6, 6, 6, 6, 7, 8, 9]`
- `c.count(6)`
- `c.count(5)`
- `c.extend([4,5])` `#c.append([4,5])`
- `print(c)`

- # 튜플의 생성

- a = tuple() # a = list(), a = []

- a = ()

- type(a)

- b = (1,) # 마지막에 콤마 필요

- type(b)

- ba = (1)

- type(ba)

- c = (1, 2, 3)

- type(c)

- d = 1, 2, 3 # packing & unpacking

- type(d)

- e = (1, 2, ('ab', 'cd'), 3, 4)

- type(e)

3.8 튜플(TUPLE) 타입

- # 튜플은 수정 및 변경 불가
- `e = (1, 2, ('ab', 'cd'), 3, 4)`
- `e[0]`
- `del e[0]`
- `e.remove(1)`
- `e.pop(0)`

3.8 튜플(TUPLE) 타입

```
# c = (1,2,3)
```

```
# e = (1, 2, ('ab', 'cd'), 3, 4)
```

- e[:]

- c + e

- c * 2

- len(c)

- x,y,z = (1,2,3)

- print(x, y, z)

- print(x)

- print(y)

- print(z)

- # 사전의 생성

- a = {'gender':'Female', 'age':30, 'name':'Kristina'}

- print(a)

- type(a)

- a[0]

- a['gender']

- a['age']

- a['name']

- a['Kristina']

- `b = {1: 'good'}`
- `print(b)`

- `c = {a:[1,3,5]}`
- `print(c)`

- `d = {a:(1,3,5)}`
- `print(d)`

- `c = {[1,3,5]:'a'}`

- `d = {(1,3,5):'a'}`
- `print(d)`

- # 사전의 추가

- a = {'gender':'Female'}
- a['age'] = 30
- print(a)

- a['name'] = 'Kristina'
- print(a)

- a['x'] = [1,3,5]
- print(x)

- # 사전의 삭제

- del a['x']
- print(a)

- del a['name']
- print(a)

- del a['age']
- print(a)

- # 사전의 장점

- # {'학생A': 99, '학생B': 91, '학생C': 83, '학생D': 100}

- result = {'학생A': 99, '학생B': 91, '학생C': 83, '학생D': 100}

- result['학생A']

- result['학생B']

- # 중복 key 값

- a = {'fuel': 'gasoline', 'fuel': 'diesel'}

- print(a)

- a = {'fuel': 'gasoline', 'fuel': 'diesel', 'fuel': 'LPG'}

- print(a)

- # 사전 함수

- a = {'gender': 'Female', 'age': 30, 'name': 'Kristina'}
- a.keys()

- list(a.keys())

- a.values()

- a.items()

- a.clear()

- print(a)

- a = {'gender': 'Female', 'age': 30, 'name': 'Kristina'}

- a.get('gender') #a['gender']

- a.get('height', '?')

- 'gender' in a

- 'height' in a

- # 집합의 생성

- a = set([1,2,3,4,5])

- print(a)

- type({1,2,3,4,5})

- type((1,2,3,4,5))

- type([1,2,3,4,5])

- type({1:2, 3:4, 5:6})

- b = set('Kristina')

- b

- b[1] #indexing

- c = list(b)

- c

- c[1]

3.10 집합(SET) 타입

- # 교집합, 합집합, 차집합

- $x = \{1, 2, 3, 4, 5\}$

- $y = \{3, 4, 5, 6, 7\}$

- $x \& y$ # 교집합

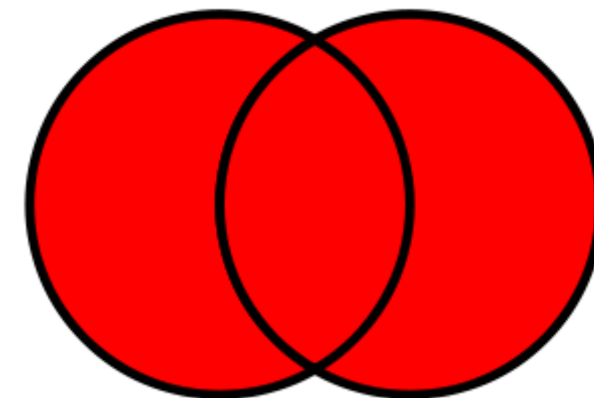
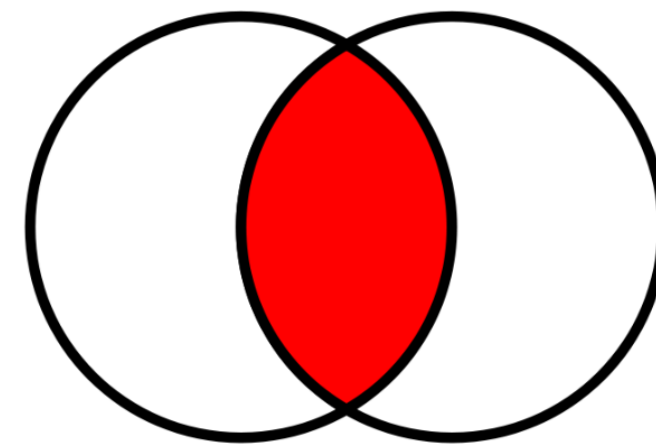
- $x.intersection(y)$

- $\{3, 4, 5\}$

- $x | y$ # 합집합

- $x.union(y)$

- $y.union(x)$

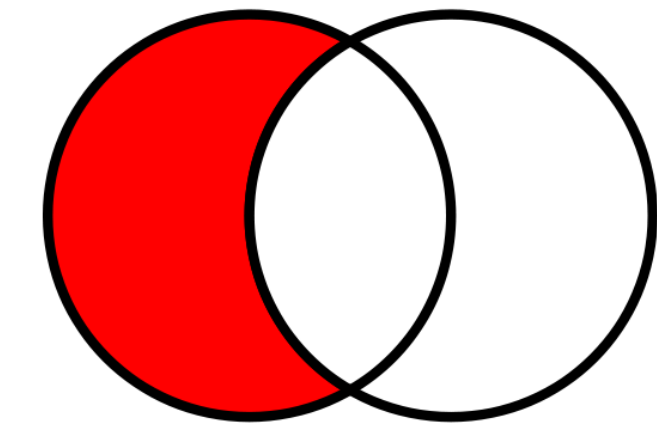


- $x - y$ # 차집합

- $y - x$

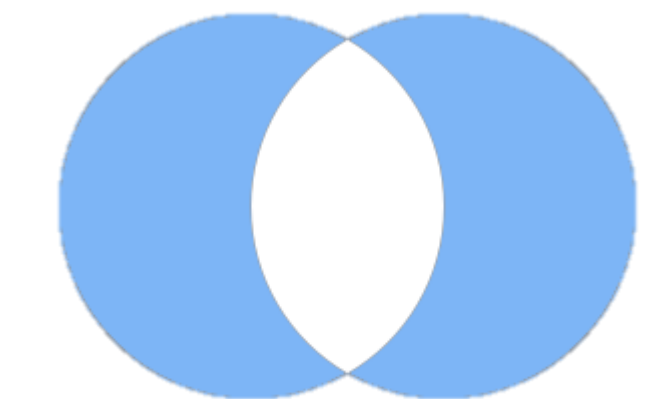
- $x.difference(y)$

- $y.difference(x)$



- $x \wedge y$ # 대칭 차집합

- $(x - y) | (y - x)$



- # 집합 함수
- x.add(10)
- print(x)

- x.update([11,12,13])
- print(x)

- x.remove(13)
- print(x)

2. Python 이론 및 실습

DATA
KUBWA

IV. 연산자

4.1 수치 연산자

- $2 + 3$

- $2 ** 0.5$

- $3 - 2$

- $5 / 2$

- $2 * 3$

- $5 // 2$

- $3.14 ** 2$

- $5 \% 2$

- $a = 6$

- $b = 4$

- $a + b$

- $a - b$

- $a * b$

- $a ** b$ # a의 b승

- a / b

- $a // b$

- $a \% b$

- `a = 20`

- `a += 5`

- `print(a)`

- `a -= 10`

- `print(a)`

- `a *= 2`

- `print(a)`

- `a /= 3`

- `print(a)`

- `a = 3`

- `b = 2`

- `a *= 2 + b`

- `print(a)`

- `a *= b - 5`

- `print(a)`

- $a = 95$
- $a == 90$
- $a != 90$
- $a > 90$
- $a < 90$
- $a \geq 90$
- $a \leq 90$
- $a \geq 90$ and $a \leq 100$
- $90 \leq a \leq 100$
- $100 \leq a \leq 110$
- $70 \leq a \leq 80$

- `a = [1,2,'Hello World']`
- `b = [1,2,'Hello World']`
- `a == b`
- `print(a)`
- `print(b)`
- `a != b`
- `id(a) == id(b)`
- `id(a)`
- `id(b)`
- `a.append('A')`
- `print(a)`
- `id(a)`

- a = True # 참
- b = False # 거짓

- # and 연산

- a and b

- a and a

- b and b

- # or 연산

- a or b

- a or a

- b or b

- # not 연산

- not a

- not b

- not (a and b)

- `a = [1,2,'Hello World']`
- `b = [1,2,'Hello World']`
- `a is b`
- `a is not b`
- `a == b`
- `id(a) == id(b)`

- # in, not in
- a = [1,2,'Hello World']
- 1 in a
- 3 in a
- 3 not in a
- b = {'a':1, 'b':2, 'c':3 }
- 'a' in b
- 'k' in b
- 1 in b

- `a = [`
- `{'num':'1','name':'김철수','kor':90,'eng':80,'math':85,'total':0,'avg':0.0,'order':0 },`
- `{'num':'2','name':'박제동','kor':90,'eng':85,'math':90,'total':0,'avg':0.0,'order':0 },`
- `{'num':'3','name':'홍길동','kor':80,'eng':80,'math':80,'total':0,'avg':0.0,'order':0 }`
- `]`

- `a[0]`

- `a[0]['total']`

- `a[0]['total'] = 1`
- `a[0]`

- `a[0]['total'] = a[0]['kor'] + a[0]['eng'] + a[0]['math']`
- `a[1]['total'] = a[1]['kor'] + a[1]['eng'] + a[1]['math']`
- `a[2]['total'] = a[2]['kor'] + a[2]['eng'] + a[2]['math']`

• # 학생별 합계 구하기

- `a[0]['total']`

- `a[1]['total']`

- `a[2]['total']`

• # 학생별 평균 구하기

- `a[0]['avg'] = a[0]['total'] / 3`

- `a[1]['avg'] = a[1]['total'] / 3`

- `a[2]['avg'] = a[2]['total'] / 3`

- `a[0]['avg']`

- `a[1]['avg']`

- `a[2]['avg']`

- # 학급 평균 구하기

- $b = (a[0]['total'] + a[1]['total'] + a[2]['total']) / 3$
- `print(b)`

- `print(k)`

- `print(e)`

- # 과목별 평균 구하기

- $k = (a[0]['kor'] + a[1]['kor'] + a[2]['kor']) / 3$
- $e = (a[0]['eng'] + a[1]['eng'] + a[2]['eng']) / 3$
- $m = (a[0]['math'] + a[1]['math'] + a[2]['math']) / 3$

- `print(m)`

2. Python 이론 및 실습

DATA
KUBWA

V. 흐름 제어

- # if문의 구조

- if True:
 - print('참입니다')
- else:
 - print('거짓입니다')
- if False:
 - print('참입니다')
- else:
 - print('거짓입니다')

- a = 1000
- if a > 5000:
 - print('택시를 탑니다')
- else:
 - print('버스를 타야 합니다')
- if a <= 5000:
 - print('택시를 탑니다')
- else:
 - print('버스를 타야 합니다')

- # 들여쓰기

- a = 1000
- if a > 5000:
- print('택시를 탑니다')
- else:
- print('버스를 타야 합니다')

- `grade = float(input('총 평점을 입력해 주세요: '))`
- `if grade >= 4.3:`
 - `print('당신은 장학금 수여 대상자입니다.')`
 - `print('축하합니다.')`
- `data = int(input('숫자를 입력하시오: '))`
- `if data % 2 == 0:`
 - `print('입력된 값은 짝수입니다.')`
- `else:`
 - `print('입력된 값은 홀수입니다.')`

- # Print 문

- a = 123

- print(a)

- a = "Python"

- print(a)

- a = [1, 2, 3]

- print(a)

- print("life" "is" "too short")

- print("life"+"is"+"too short")

- print("life", "is", "too short")

- `score = int(input('점수를 입력하시오: '))`
- `if score >= 70:`
 - `print('당신은 시험을 통과했습니다.')`
- `else:`
 - `print('당신은 시험을 통과하지 못했습니다.')`
- `print('공부 열심히 하세요!')`

- #중첩된 if else 구문
- `age = int(input('나이를 입력하시오: '))`
- `height = int(input('키를 입력하시오: '))`
- `if age >= 40:`
 - `if height >= 170:`
 - `print('키가 보통 이상 입니다.')`
 - `else:`
 - `print('키가 보통입니다.')`
- `else:`
 - `if height >= 175:`
 - `print('키가 보통 이상 입니다.')`
 - `else:`
 - `print('키가 보통입니다.')`

- # if elif 구문

- score = int(input('총점을 입력해 주세요: '))
- if score >= 90:
 - print('수')
- else:
 - if 80 <= score < 90:
 - print('우')
 - else:
 - if 70 <= score < 80:
 - print('□|')
 - else:
 - if 60 <= score < 70:
 - print('양')
 - else:
 - print('가')

- score = int(input('총점을 입력해 주세요: '))
- if score >= 90:
 - print('수')
- elif 80 <= score < 90:
 - print('우')
- elif 70 <= score < 80:
 - print('□|')
- elif 60 <= score < 70:
 - print('양')
- else:
 - print('가')

- `score = int(input('총점을 입력해 주세요: '))`

- `if score >= 60:`

- `message = " success "`

- `else:`

- `message = " failure "`

- `print(message)`

- `score = int(input('총점을 입력해 주세요: '))`

- `message = " success " if score >= 60 else
" failure "`

- `print(message)`

for 문

- # for 변수 in 리스트 (튜플, 문자열):
- # 실행 문장 1
- # 실행 문장 2
- for a in [1, 2, 3]:
- print(a)
- for a in [(1,3), {5,7}, 9]:
- print(a)

- for (a,b) in [(1,2), (3,4), (5,6)]:
- print(a+b)
- (a,b) = (1,2)
- print(a)
- print(b)
- print(a,b)

- a = 'Hello!'
- b = ['Hello!', 'World']
- c = (1,2,3)
- for x in a:
 - print(x, end = " ")
- for x in b:
 - print(x)
- for x in c:
 - print(x)

- `d = {'circle': 2, 'rectangle': 3, 'line': 1}`
- `e = {'red', 'green', 'blue', 'red'}`
- `for x in d:`
 - `print(x)`
- `for x in e:`
 - `print(x)`

- # if ~ else 조건문

- Score = [100, 7, 75, 32]
- num = 0
- for a in Score:
 - num = num +1
 - if a >= 70:
 - print('%d번 합격' %num)
 - else:
 - print('%d번 불합격' %num)

- # Continue

- Score = [100, 7, 75, 32]
- num = 0
- for a in Score:
 - num = num +1
 - if a < 70:
 - continue
 - else:
 - print('%d번 합격' %num)

- # range함수

- a = range(10)
- print(a)

- a = 0
- for b in range(10):
- a = a+b
- print(a)

- # Continue, range

- a = [100, 7, 75, 32]
- for b in range(len(a)):
 - if a[b] < 60:
 - continue
 - print("%d번 합격" % (b+1))

- len(a)

- range(4)

- # for ~ for ~ (구구단)

- for a in range(2,10):
 - for b in range(1, 10): [1,2,3,4,5,6,7,8,9]
 - print(a*b, end=" ")
 - print("")

- for a in range(2,10):
 - for b in range(1, 10):
 - print(a*b)

- for a in range(2,10):
 - for b in range(1, 10):
 - print(a*b, end=" ")

- `a = 0`
- `for b in [1,2,3,4,5,6,7,8,9,10]:`
- `a = a + b`
- `print('1부터 10까지 합은 ', a, ' 입니다.')`

- `a = 0`
- `for b in [1,2,3,4,5,6,7,8,9,10]:`
- `a = a + b`
- `print('1부터 10까지 합은 ', a, ' 입니다.')`

- `a = 0`
- `for b in range(1,101):`
- `a = a + b`
- `print('1부터 100까지 합은 ', a, ' 입니다.')`

- `a = 0`
- `for b in range(0,101,2):`
- `a = a + b`
- `print('1부터 100까지 짝수 합은 ', a, ' 입니다.')`

- `a = 0`
- `for b in range(1,101,2):`
- `a = a + b`
- `print('1부터 100까지 홀수 합은 ', a, ' 입니다.')`

- `a = 0`
- `for b in range(1,101):`
- `if (b % 3) == 0 or (b % 7) == 0:`
- `a = a + b`
- `print('1부터 100까지에서 3 혹은 7의 배수의 합은 ', a, ' 입니다.')`

- lst = [1,2,3,4,5]
- result = []
- for num in lst:
- result.append(num*3)
- print(result)

- lst = [1,2,3,4,5]
- result = [num * 3 for num in lst]
- print(result)
-
- lst = [1,2,3,4,5]
- result = [num * 3 for num in lst if num % 2 == 0]
- print(result)

While 문

- # while 조건문:
- 실행문

- a = 1
- while a <= 5:
- print(a)

- `a = 1`
- `while a <= 5:`
- `print(a)`
- `a = a + 1`

- `a = 1`
- `b = 0`
- `while a <= 100:`
- `b = b + a`
- `a = a + 1`
- `print('1부터 100까지의 합은:', b)`

5.4 반복 흐름과 FOR, WHILE 문

	<i>b</i>	<i>a</i>	조건(≤ 100)	While
Start	0	1	참	
1	1	2	참	반복
2	3	3	참	반복
3	6	4	참	반복
4	10	5	참	반복
5	15	6	참	반복
...
97	4753	98	참	반복
98	4851	99	참	반복
99	4950	100	참	반복
100	5050	101	거짓	Print()

- $a = 1$
- $b = 0$
- while $a \leq 100$:
 - $b = b + a$
 - $a = a + 1$
- `print('1부터 100까지의 합은:', b)`

break

- candy = 4
- money = 300
- while money:
- print("지불했으니 사탕을 줍니다.")
- candy = candy - 1
- print("남은 사탕의 양은 %d개입니다." % candy)
- if candy == 0:
- print("사탕이 다 떨어졌습니다. 판매를
중지합니다.")
- break

without break

- candy = 4
- money = 300
- while money:
- print("지불했으니 사탕을 줍니다.")
- candy = candy -1
- print("남은 사탕의 양은 %d개입니다." % candy)
- if candy == 0:
- print("사탕이 다 떨어졌습니다. 판매를
중지합니다.")

for ~ break

- `a = input('중단할 문자를 입력하시오: ')`
- `for b in 'python':`
 - `if b == a:`
 - `break`
 - `print(b, end = " ")`
- `else:`
 - `print('모든 문자 출력 완료!')`

continue

- a = input('건너뛰려 할 문자를 입력하십시오: ')
- for b in 'python':
 - if b == a:
 - **continue**
 - print(b)

pass

- a = input('중단할 문자를 입력하시오: ')
- for b in 'python':
 - if b == a:
 - **pass**
 - print(b)

- coffee = 3
- **while True:**
 - money = int(input("돈을 넣어 주세요: "))
 - if money == 300:
 - print("커피를 줍니다.")
 - coffee = coffee - 1
 - elif money > 300:
 - print("거스름돈 %d를 주고 커피를 줍니다." % (money - 300))
 - coffee = coffee - 1
 - else:
 - print("돈을 다시 돌려주고 커피를 주지 않습니다.")
 - print("남은 커피의 양은 %d개 입니다." % coffee)
 - if coffee == 0:
 - print("커피가 다 떨어졌습니다. 판매를 중지 합니다.")
 - **break**

5.8 예제: 요일 구하기

- #서기 1년 1월 1일은 월요일이다.
- #윤년을 구하는 공식은 다음과 같다.
- #4로 나누어지는 해는 윤년이다.
- #100으로 나누어지는 해는 윤년이 아니다.
- #400으로 나누어지는 해는 윤년이다.
- # 년월일을 입력 받는다.
- year = int(input('년도를 입력하시오: '))
- month = int(input('월을 입력하시오: '))
- day = int(input('일을 입력하시오: '))
- total_days = 0
- # 달별 날수를 리스트로 저장
- year_month_days = [0,31,28,31,30,31,30,31,31,30,31,30,31]
- # 서기 1년부터 year-1 까지의 각 년도 별 날수를 합한다.
- for item in range(1, year):
 - if item % 400 == 0: # 400으로 나뉘는 해: 윤년
 - total_days = total_days + 366
 - elif item % 100 == 0: # 100으로 나뉘는 해: 평년
 - total_days = total_days + 365
 - elif item % 4 == 0: # 4으로 나뉘는 해 : 윤년
 - total_days = total_days + 366
 - else:
 - total_days = total_days + 365
- # 1월 달부터 month-1 까지의 각 달의 날수를 합한다.
- for item in range(1, month):
 - total_days = total_days + year_month_days[item]

- # 입력된 달이 3이상이고 해당년도가 윤년일 경우 1을 추가
- if month >= 3:
 - if year % 400 == 0: # 400으로 나뉘는 해: 윤년
 - total_days = total_days + 1
 - elif year % 100 == 0: # 100으로 나뉘는 해: 평년
 - total_days = total_days + 0
 - elif year % 4 == 0: # 4으로 나뉘는 해 : 윤년
 - total_days = total_days + 1
 - else:
 - total_days = total_days + 0
- total_days += day
- # 총 날수를 7로 나눈 나머지를 구한다.
- remainder = total_days % 7
- if remainder == 0:
- print('일요일 입니다.')
- elif remainder == 1:
- print('월요일 입니다.')
- elif remainder == 2:
- print('화요일 입니다.')
- elif remainder == 3:
- print('수요일 입니다.')
- elif remainder == 4:
- print('목요일 입니다.')
- elif remainder == 5:
- print('금요일 입니다.')
- elif remainder == 6:
- print('토요일 입니다.')

- # 학생 데이터 초기화

- students = [
 - {'num':'1','name':'김철수','kor':90,'eng':80,'math':85,'total':0,'avg':0.0,'order':0 },
 - {'num':'2','name':'박제동','kor':90,'eng':85,'math':90,'total':0,'avg':0.0,'order':0 },
 - {'num':'3','name':'홍길동','kor':80,'eng':80,'math':80,'total':0,'avg':0.0,'order':0 }
-]

- # 반 총점, 평균 및 각 과목별 총점과 평균 초기화

- class_total = 0
- class_avg = 0.0
- kor_total = 0
- kor_avg = 0.0
- eng_total = 0
- eng_avg = 0.0
- math_total = 0
- math_avg = 0.0

- # 학생들의 성적 총점과 평균 및 반 총점과 과목별 총점을 구한다.

- for student in students:
 - student['total'] = student['kor'] + student['eng'] + student['math']
 - student['avg'] = student['total'] / 3
- class_total = class_total + student['total']
- kor_total = kor_total + student['kor']
- eng_total = eng_total + student['eng']
- math_total = math_total + student['math']
- class_avg = class_total / len(students)
- kor_avg = kor_total / len(students)
- eng_avg = eng_total / len(students)
- math_avg = math_total / len(students)

5.8 예제: 성적 처리 시스템

- # 학생별 성적 처리 결과를 출력한다.
- for student in students:
- print('번호: {:s}, 이름: {:s}, 국어: {:d}, 영어: {:d}, 수학: {:d}, 총점: {:d}, 평균: {:.2f}'.format(
• student['num'], student['name'], student['kor'], student['eng'], student['math'],
• student['total'], student['avg']))
-)
- print()
- # 반 평균 및 각 과목별 평균을 출력한다.
- print('반 평균 : {:.2f}'.format(class_avg))
- print('국어 평균: {:.2f}'.format(kor_avg))
- print('영어 평균: {:.2f}'.format(eng_avg))
- print('수학 평균: {:.2f}'.format(math_avg))

- # 모스부호

- dic = {
 - '.-': 'A', '-...': 'B', '-.-.': 'C', '-..': 'D', '.': 'E', '...': 'F',
 - '---': 'G', '....': 'H', '..': 'I', '.---': 'J', '-.-': 'K', '-...': 'L',
 - '--': 'M', '-.': 'N', '---': 'O', '-.-.': 'P', '---.': 'Q', '-.': 'R',
 - '...': 'S', '-': 'T', '...': 'U', '...-': 'V', '---': 'W', '-...': 'X',
 - '-.-.': 'Y', '---': 'Z'
- }

- # 풀어야할 암호

- code = '.... -... .-.-. -.-.'

- # Write your code below.

- for word in code.split(" "):
 - for char in word.split():
 - print(dic.get(char), end="")
 - print(" ", end="")

- word = code.split(" ")

- char = word[0].split()

- print(dic.get(char[0]))

- # 게시물 제목

- `title = ""On top of the world! Life is so fantastic if you just let it. I have never been happier. #nyc #newyork #vacation #traveling""`

- `tag_list = []`
- `for tag in title.split(" "):`
 - `if tag.startswith("#"):`
 - `tag_list.append(tag[1:])`
- `print(tag_list)`

- `tag = title.split(" ")`
- `['On', 'top', 'of', 'the', 'world!', 'Life', 'is', 'so', 'fantastic', 'if', 'you', 'just', 'let', 'it.\n', 'have', 'never', 'been', 'happier.', '#nyc', '#newyork', '#vacation', '#traveling']`

- # enumerate()
- # 반복문 사용 시 몇 번째 반복문인지 확인이 필요할 때 사용.
- # 인덱스 번호와 요소값을 tuple형태로 반환

- some_list = ['foo', 'bar', 'baz']
- for i, v in enumerate(some_list):
- print('i: {}, v: {}'.format(i,v))

- # zip()
- # 동일한 자리끼리 묶어줌
- seq1 = ['foo', 'bar', 'baz']
- seq2 = ['one', 'two', 'three']
- for a, b in zip(seq1, seq2):
- print('a: {}, b: {}'.format(a, b))

2. Python 이론 및 실습

DATA
KUBWA

VI. 함수

- # 사용자 정의 함수의 구조

- def 함수명: # def는 함수를 만들 때 사용하는 예약어

- 실행문1
- 실행문2
-

- def say():

- print('Hello World')
- print('Hello World')
- print('Hello World')

- say()

- # 매개변수(parameter), 인수(arguments)

- def add(a, b): # a, b는 매개변수

- return a+b

- print(add(3, 4)) # 3, 4는 인수

- # 입력값과 결과값에 따른 함수의 형태 (입력값 → 함수 → 리턴값)

- # ① 입력값 有, 리턴값 有

- `def add(a, b):`
- `result = a + b`
- `return result`

- `a = add(9, 10)`
- `print(a)`

- # ② 입력값 無, 리턴값 有

- `def abcd():`
- `return 'efg'`
- `a = abcd()`
- `print(a)`

- # ③ 입력값 有, 리턴값 無

- `def abcd(a, b):`
- `print("%s, %s의 곱은 %s입니다." % (a, b, a*b))`
- `print(abcd(9, 10))` # 리턴값이 없음

- # ④ 입력값 無, 리턴값 無

- `def abcd():`
- `print('efg')`
- `print(abcd())`

- # 매개변수 지정하여 호출하기

- def add(a, b):
 - return a+b
- result = add(a=9, b=10)
- print(result)

- result = add(b=10, a=9)
- print(result)

- # 입력값이 몇 개가 될지 모를 때

- def abcd(*args):
 - result = 0
 - for i in args:
 - result = result + i
 - return result

- result = abcd(1,2,3,4,5)
- print(result)

- `def abc(a, *args):`
- `if a == "cute":`
- `x = 0`
- `for i in args:`
- `x = x + i`
- `elif a == "pretty":`
- `x = 1`
- `for i in args:`
- `x = x * i`
- `return x`
- `x = abc('cute', 1,2,3,4,5)`
- `print(x)`

- `x = abc('pretty', 1,2,3,4,5)`
- `print(x)`
- 120

- # 키워드 파라미터 **

- `def abc(**kwargs):`
- `print(kwargs)`

- `abc(a=10)`

- `abc(num = 1, name='Kristina', age=30)`

- # 함수의 결과값은 언제나 하나

- def abcd(a, b):
 - return a+b, a*b

- x = abcd(9, 10)
- print(x)

- x1, x2 = abcd(9, 10)
- print(x1)
- print(x2)

- def abcd(a,b):
 - return a+b
 - return a*b # 생략됨
- x = abcd(9, 10)
- print(x)

- def abcd(a,b): # 앞의 함수와 완전 동일
 - return a+b

- # return

- def abcd(a):
 - if a == "dog":
 - return
 - print("I am %s " % a)

- abcd('human')

- abcd('dog')

- # 초기값 설정

- def abcd(name, age, woman=True):
 - print("Name:%s" % name)
 - print("Age:%s" % age)
 - if woman:
 - print("Female")
 - else:
 - print("Male")
- abcd("Kristina", 27)
- abcd("Kristina", 27, True)

- abcd("Kristina", 27, False)

- def abcd(name, **woman=True**, age):
 - print("Name: %s" % name)
 - print("Age: %s" % age)
 - if woman:
 - print("남자입니다.")
 - else:
 - print("여자입니다.")

- `k = 1`
- `def abcd(k):`
 - `k = k + 1`
- `abcd(k)`
- `print(k)`

- `def abcd(k):`
 - `k = k + 1`
- `abcd(9)`
- `print(k)`

- `def abcd(k)`
 - `k = k+1`
 - **`return k`**
- `k = abcd(9)`
- `print(k)`

- `k = 1`
- `def abcd(k):`
 - `k = k + 1`
 - **`return k`**
- `k = abcd(k)`
- `print(k)`

- `k = 1`
- `def abcd():`
 - **`global k`**
 - `k = k+1`

- `abcd()`
- `print(k)`

lambda

- abcd = lambda a, b: a+b
 - result = abcd(9, 10)
 - print(result)
- def abcd(a, b):
 - return a+b
 - result = abcd(9, 10)
 - print(result)

print

- a = 123
- print(a)
- a = "Python"
- print(a)
- a = [1, 2, 3]
- print(a)
- print("You" "are" "my everything")
- print("You"+"are"+"my everything")
- print("You", "are", "my everything")

파일 생성

- `f = open("새파일.txt", 'w')`
- `f.close()`
- 파일 객체 = `open(파일 이름, 파일 열기 모드)`
- `f = open("C:/python/hello.txt", 'w')`
- `f.close()`

파일 열기 모드	설명
'r'	읽기 모드로 연다 Default
'w'	쓰기 모드로 연다 기존 내용 삭제
'a'	쓰기 모드로 연다 기존 내용 보존
'b'	이진 모드로 연다
't'	텍스트 모드로 연다 Default

Write

- `f = open("C:/python/hello.txt", 'w')`
 - `for i in range(1, 4):`
 - `data = "%d번 항목.\n" % i`
 - `f.write(data)`
 - `f.close()`
- `for i in range(1, 4):`
 - `data = "%d번째 줄입니다.\n" % i`
 - `print(data)`

- # readline 파일의 첫번째 줄을 읽어 출력

- a = open("C:/python/hello.txt", 'r')
- b = a.readline()
- print(b)
- a.close()

- a = open("C:/python/hello.txt", 'r')
- while True:
- b = a.readline()
- if not b: break
- print(b)
- a.close()

readlines()

- `a = open("C:/python/hello.txt", 'r')`
- `b = a.readlines()`
- `for i in b:`
 - `print(i)`
- `a.close()`

- `print(b)`

read()

- `a = open("C:/python/hello.txt", 'r')`
- `b = a.read()`
- `print(b)`
- `a.close()`

write 'a'

- `a = open("C:/python/hello.txt", 'a')`
- `for i in range(4, 7):`
 - `b = "%d번 항목.\n" % i`
 - `a.write(b)`
- `a.close()`

with

- a = open("abcd.txt", 'w')
- a.write("efghijklmn")
- a.close()

- **with** open("abcd.txt", "w") as a: # close 포함
- a.write("efghijklmn")

2. Python 이론 및 실습

DATA
KUBWA

VII. 객체 지향 프로그래밍 (CLASS)

Class

- result = 0
 - def abcd(num):
 - global result
 - result += num
 - return result
 - print(abcd(5))
 - print(abcd(6))
- b1 = 0
 - b2 = 0
 - def add1(num):
 - global b1
 - b1 += num
 - return b1
 - def add2(num):
 - global b2
 - b2 += num
 - return b2
 - print(add1(5))
 - print(add1(6))
 - print(add2(5))
 - print(add2(7))

- **class Calculator:**
 - `def __init__(self):`
 - `self.result = 0`
 - `def add(self, num):`
 - `self.result += num`
 - `return self.result`
- `cal1 = Calculator()`
- `cal2 = Calculator()`
- `print(cal1.add(3))`
- `print(cal1.add(4))`
- `print(cal2.add(3))`
- `print(cal2.add(7))`

틀: 클래스 (class)

제품: 객체 (object)

- class Chocolate: # 초콜릿 클래스 생성
- pass
- a = Chocolate() # a 객체 생성
- b = Chocolate() # b 객체 생성

- class F:
- pass
- a = F()
- b = F()
- c = F()
- type(a)



- class F:
- def abcd(self, x, y):
- self.x = x
- self.y = y
- a = F()
- b = F()
- a.abcd(5, 8)
- b.abcd(6, 7)
- print(a.x)
- print(a.y)
- print(b.x)
- print(b.y)

- print(a.x)

id함수, 객체 주소 확인

- id(a.x)
- id(b.x)

- class F:
- def add(self, x, y):
- self.x = x
- self.y = y
- def add(self):
- result = self.x + self.y
- return result
- def mul(self):
- result = self.x * self.y
- return result
- def sub(self):
- result = self.x - self.y
- return result
- def div(self):
- result = self.x / self.y
- return result
- a = F()
- b = F()
- a.add(5, 8)
- b.add(6, 9)
- a.add()
- a.mul()
- a.sub()
- a.div()
- b.add()
- b.mul()
- b.sub()
- b.div()

생성자(Constructor)

- class F:
- def **__init__**(self, x, y): # def add(self, x, y)
- self.x = x
- self.y = y
- def setdata(self, x, y):
- self.x = x
- self.y = y
- def add(self):
- result = self.x + self.y
- return result
- def mul(self):
- result = self.x * self.y
- return result
- def sub(self):
- result = self.x - self.y
- return result
- def div(self):
- result = self.x / self.y
- return result
- **a = F(5, 9)**

- print(a.x)
- a = F()
- a.add(5, 8)

클래스의 상속

- class M(F):
- pass
- a = M(5, 9)
- a.add()
- a.mul()
- a.sub()
- a.div()
- class M(F):
- def squ(self):
- result = self.x ** self.y
- return result
- a = M(5, 9)
- a.squ()

메서드 오버라이딩

- a = F(4, 0)
- a.div()
- class S(F):
- def div(self):
- if self.y == 0:
- return 0
- else:
- return self.x / self.y
- a = S(4, 0)
- a.div()
- id(Z.name)
- id(a.name)
- id(b.name)

• # 클래스 변수

- class Z:
- name = "Kristina"
- print(Z.name) #(클래스명.클래스 변수)
- a = Z()
- b = Z()
- print(a.name)
- print(b.name)
- Z.name = "Olha"
- print(a.name)
- print(b.name)