

# IT 개론

## 2장. 파이썬 설치, 변수, 자료형, 주식

# 목차

---

1. 파이썬 소개하기
2. 파이썬 설치하기
3. hello world 출력하기
4. 변수
5. 자료형
6. 주석 (comment)

# 1. 파이썬 소개하기

## ◆ Python 소개

- 1991년 귀도 반 로섬 (Gudi Van Rossum)이 발표한 프로그래밍 언어
- 인터프리터 언어
- IDLE – 파이썬 인터프리터
- 버전 (현재 버전 2와 버전 3이 공용)
  - 버전 2 : 2020년까지 지원 예정
  - 버전 3 : 새로운 버전으로 버전 2와 완벽하게 호환되지 않음.



# 1. 파이썬 소개하기

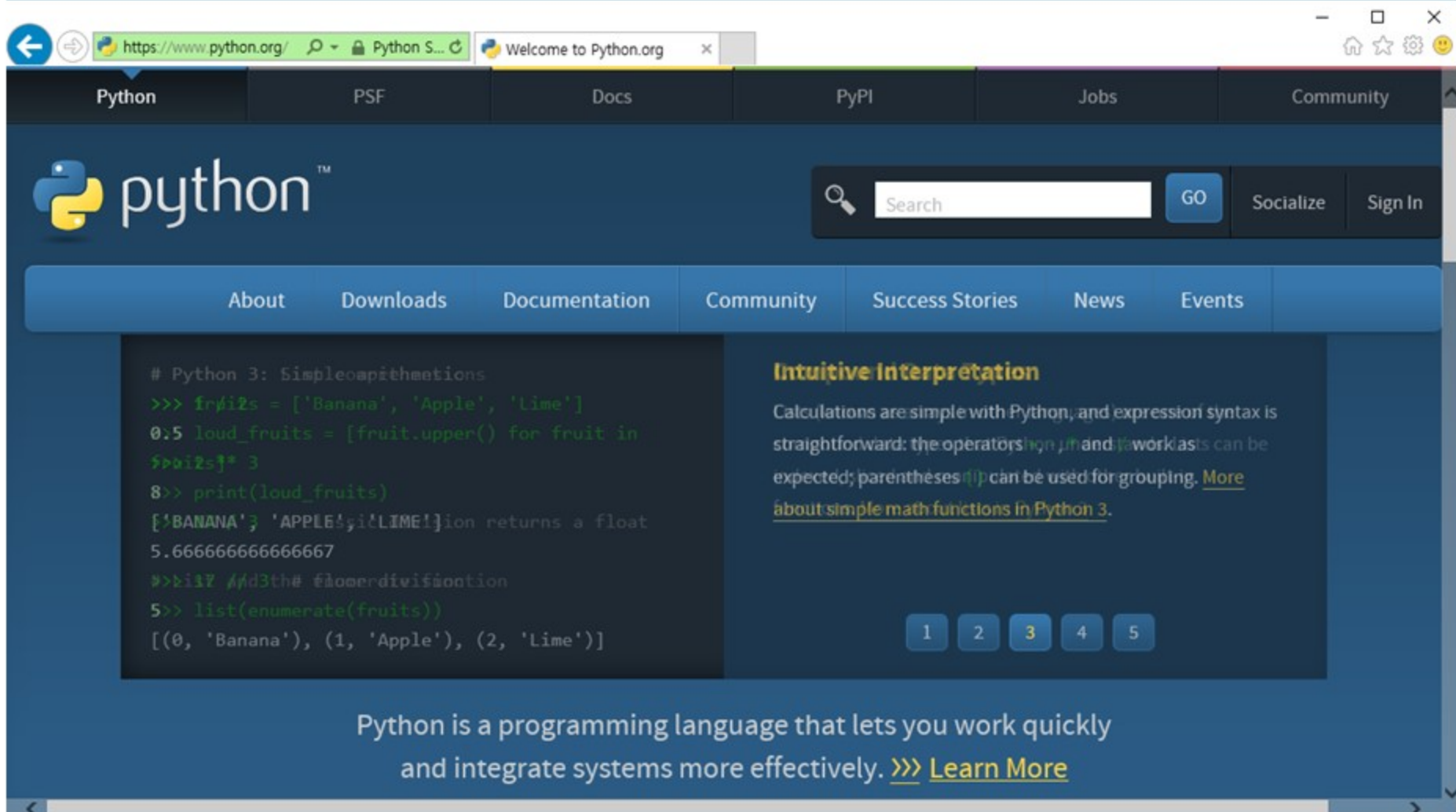
---

## ◆ 왜 파이썬인가?

- Readability (가독성)
- Simplicity (단순함)
- Efficiency (효율성)
- Easy (배우기 쉬움)
- Extensibility (확장성)

## 2. 파이썬 설치하기

### ◆ Python 3.5 설치하기



The screenshot shows the Python.org homepage in a web browser. The browser's address bar displays 'https://www.python.org/'. The page features a dark blue header with navigation links: Python, PSF, Docs, PyPI, Jobs, and Community. Below the header is the Python logo and a search bar. A secondary navigation bar includes links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area is split into two columns. The left column contains a code snippet demonstrating Python 3's simple arithmetic and list operations. The right column, titled 'Intuitive Interpretation', explains that Python's syntax is straightforward and easy to learn, with a link to 'More about simple math functions in Python 3.'. At the bottom, a blue banner states: 'Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)'.

Python.org homepage screenshot showing navigation links (Python, PSF, Docs, PyPI, Jobs, Community) and the main content area. The main content area includes a code snippet demonstrating Python 3's simple arithmetic and list operations, and a section titled 'Intuitive Interpretation' explaining that Python's syntax is straightforward and easy to learn. The bottom banner states: 'Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)'.

```
# Python 3: Simple arithmetic
>>> fruits = ['Banana', 'Apple', 'Lime']
0:5 loud_fruits = [fruit.upper() for fruit in fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']
5.666666666666667
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```

**Intuitive Interpretation**

Calculations are simple with Python, and expression syntax is straightforward: the operators `+`, `*`, and `/` work as expected; parentheses `()` can be used for grouping. [More about simple math functions in Python 3.](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

## 2. 파이썬 설치하기

### ◆ Python 3.5 설치하기

The screenshot shows the Python.org website with the 'Downloads' menu open. The 'All releases' option is selected, and the 'Python 3.5.1' download link is highlighted. A Windows file explorer window is open at the bottom, showing the download of 'python-3.5.1.exe' and the '실행(R)' (Run) button is highlighted.

Python.org website navigation: Python, PSF, Docs, PyPI, Jobs, Community

python™

Search GO Socialize Sign In

About Downloads Documentation Community Success Stories News Events

All releases ①

Source code

Windows

Mac OS X

Other Platforms

License

Alternative Implementations

Download for Windows

Python 3.5.1 ② Python 2.7.11

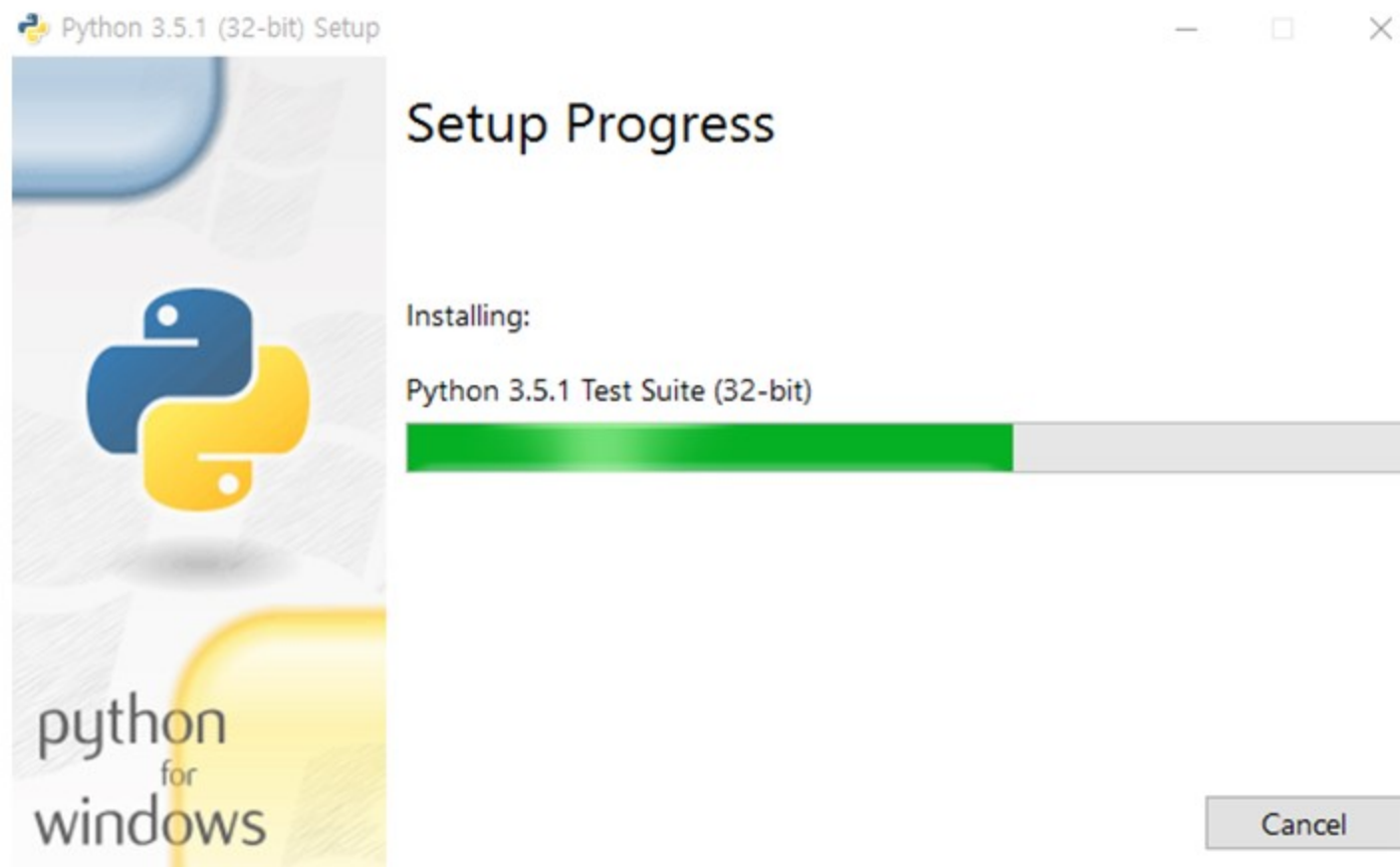
Not the OS you are looking for? Python can be used on 21 different operating systems and environments. [View the full list.](#)

Python is a programming language that lets you work quickly

python.org의 python-3.5.1.exe(27.4MB)를(를) 실행하거나 저장하시겠습니까? ③ 실행(R) 저장(S) 취소(C) x

## 2. 파이썬 설치하기

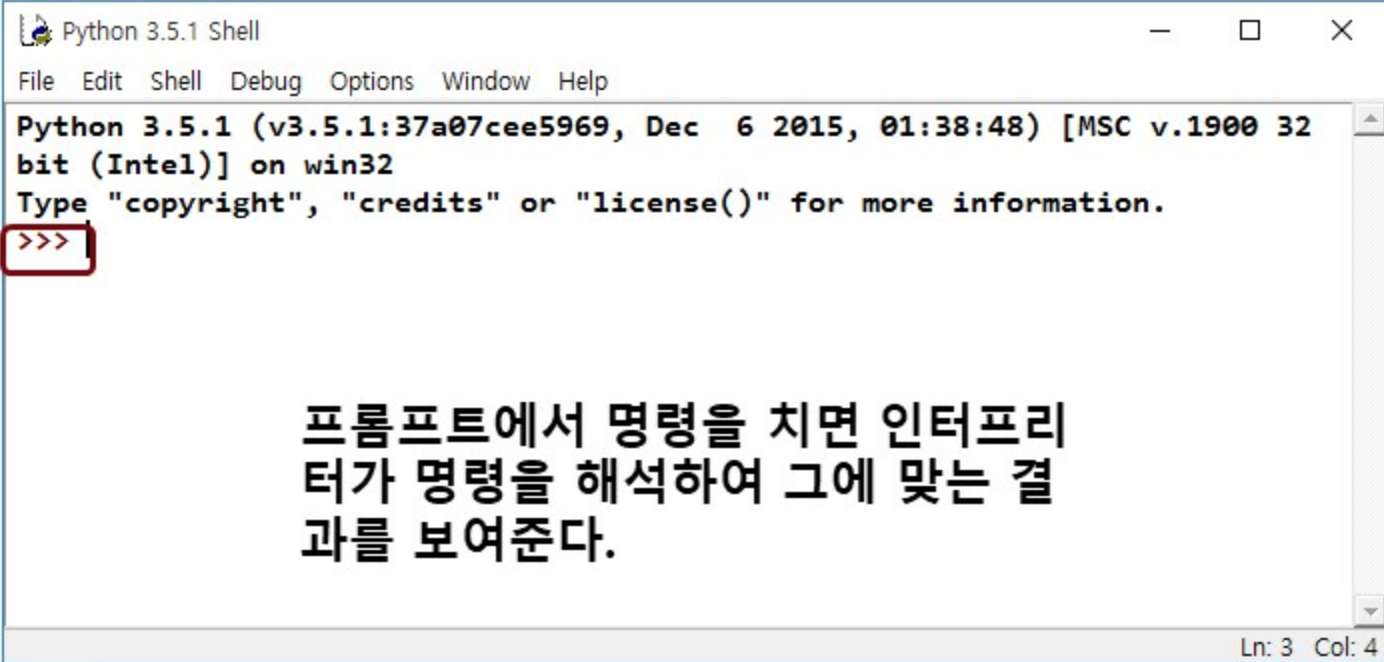
### ◆ Python 3.5 설치하기



## 2. 파이썬 설치하기

### ◆ IDLE 인터프리터 실행하기 (python shell)

프롬프트



```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 6 2015, 01:38:48) [MSC v.1900 32
bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

프롬프트에서 명령을 치면 인터프리터가 명령을 해석하여 그에 맞는 결과를 보여준다.

Ln: 3 Col: 4



### 3. hello world 출력하기

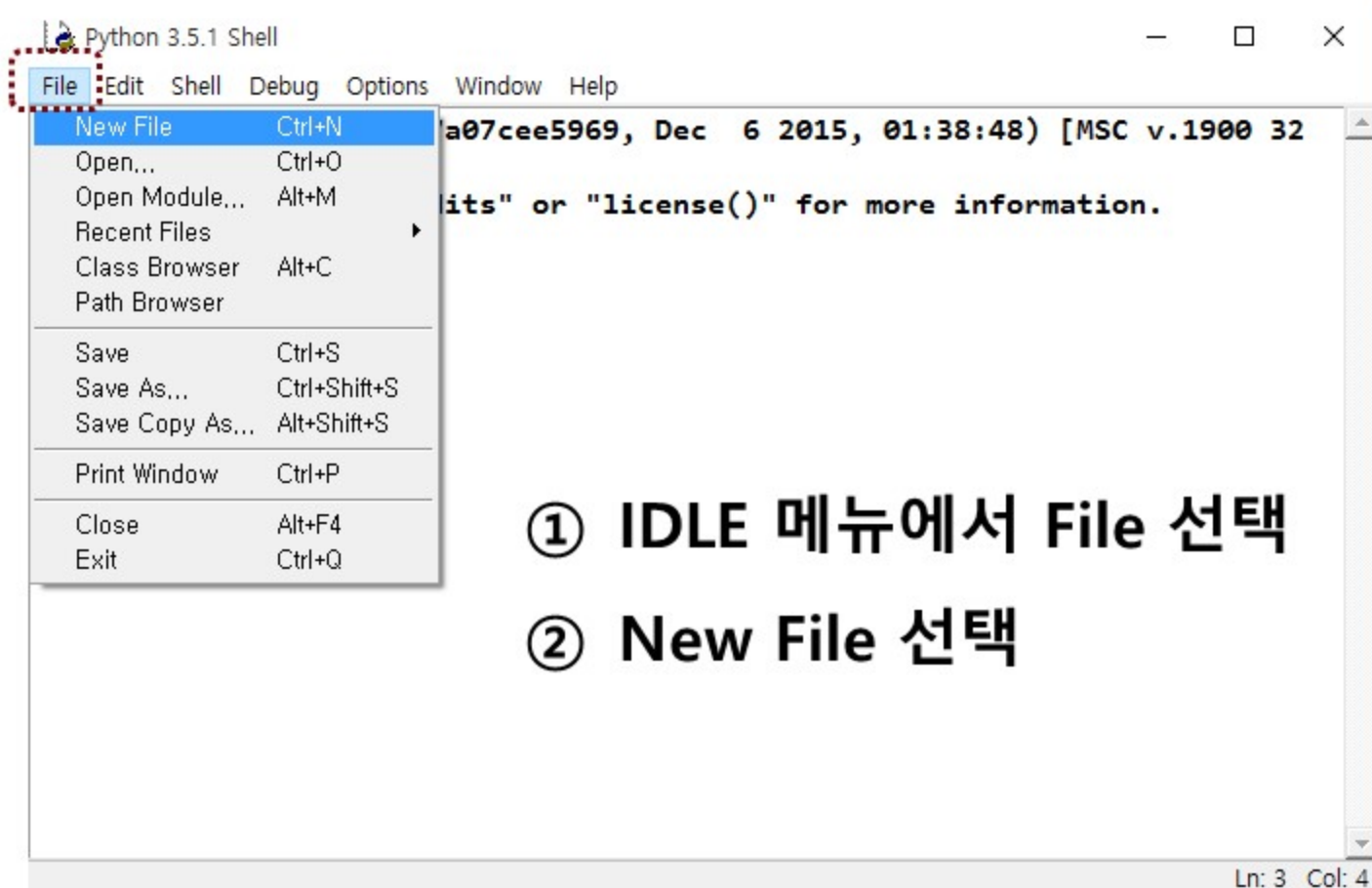
#### ◆ print 내장 함수

- print() 함수는 괄호의 내용을 출력한다.
- 홑따옴표('...') 또는 쌍따옴표("...")를 이용하여 원하는 문자열을 출력할 수 있다.
- 홑따옴표 세 개(''...'') 또는 쌍따옴표 세 개('"'...'")를 사용할 수도 있다.

```
>>> print('hello world')
hello world
>>> print("hello world")
hello world
>>> print(''hello world'')
hello world
>>> print('"'hello world"'')
hello world
```

### 3. hello world 출력하기

#### ◆ 파일에 저장하여 실행하기



① IDLE 메뉴에서 File 선택

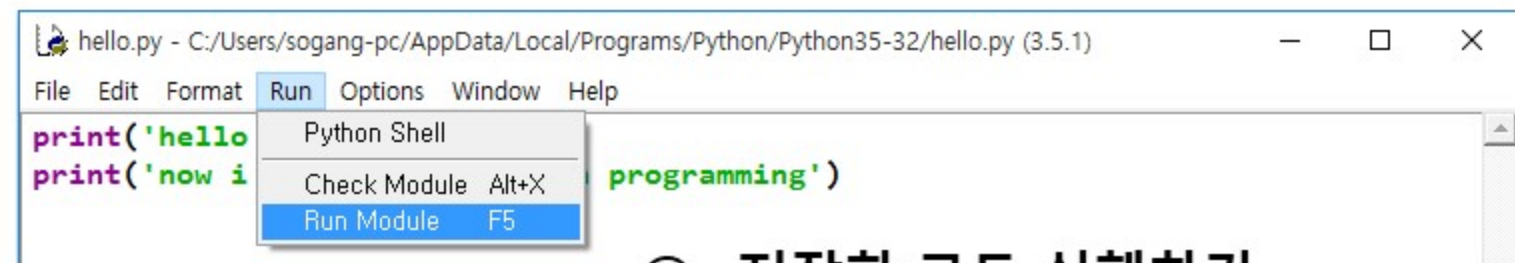
② New File 선택

### 3. hello world 출력하기

#### ◆ 파일에 저장하여 실행하기



- ③ 새로운 화면에서 코드 작성
- ④ File 메뉴에서 Save 선택
- ⑤ 파이썬 파일은 .py가 확장자임.



- ⑥ 저장한 코드 실행하기  
'Run' 메뉴 → 'Run Module'

### 3. hello world 출력하기

**코드** hello.py

```
print('Hello, world!')
```

```
PS C:\Users\User> cd Documents
```

```
PS C:\Users\User\Documents> ls
```

```
PS C:\Users\User\Documents> python hello.py
```

```
Hello, world!
```

현재 위치를 내 문서로 이동

내 문서에 있는 파일 및 폴더 확인

hello.py 실행

Hello, world! 출력됨

#### 결과가 제대로 출력되지 않는 경우

- 올바르게 코드를 작성했는지 확인
- 실행 전 파일을 저장하였는지 확인

### 3. hello world 출력하기

코드 hello.py

`print('Hello, world!)` 작은 따옴표가 없음

```
File "hello.py", line 1
```

```
    print('Hello, world!)
```

```
                ^
```

```
SyntaxError: EOL while scanning string literal
```

hello.py 실행 시  
SyntaxError 발생

### 3. hello world 출력하기

---

```
print ('Hello, world!')
```

함수 이름 하고 싶은 말

### 3. hello world 출력하기

powershell에서 명령어를 입력할 수 없을 때

```
PS C:\Users\User\Documents> python
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 6 2015, 01:38:48) [MSC v.1900
32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> exit()
PS C:\Users\User\Documents>
```

실수로 python만 입력한 경우 발생한다.

`exit()` 명령어로 빠져나올 수 있다.

### 3. hello world 출력하기

---

'Hello World'가 세 번 출력되도록 만들어 보세요..

```
Hello World  
Hello World  
Hello World
```



## 4. 변수

### ◆ 변수 (variables)

- 데이터를 저장하는 공간
- 데이터마다 자료형(data types)을 갖는다.
- 변수는 다음과 같이 '**이름 = 값**'의 형태로 생성한다.

```
>>> a = 10 ← 변수 a를 메모리에 만들어서 정수 10을 저장해라  
>>> b = 3.5 ← 변수 b를 메모리에 만들어서 실수 3.5를 저장해라  
>>> c = 'sogang' ← 변수 c를 메모리에 만들어서 문자열 sogang을 저장해라
```

= 할당 연산자라고 부른다. (assignment)

프로그램은 한줄 한줄이 컴퓨터에게 일을 시키는 것임.

## 4. 변수

### ◆ 변수명 만들기

- 변수명은 영어 소문자, 대문자, 숫자, \_ (underscore)로만 구성한다.
- 변수명은 숫자로 시작할 수 없다.
- 대소문자를 구분한다. 즉, data와 Data는 다른 변수이다.
- 변수명은 한글도 가능하다.
- 키워드(keyword)를 변수명으로 사용하면 안 된다.

< 파이썬 키워드 (33개) >

False	and	class	elif	for	import	nonlocal	raise	with
None	as	continue	else	from	in	not	return	yield
True	assert	def	except	global	is	or	try	
	break	del	finally	if	lambda	pass	while	18

## 4. 변수

### ◆ 변수 사용 예

```
>>> number = 20162345
>>> score = 90
>>> python_score = 95
>>> math-score = 90 # 에러. 특수 기호는 _ 만 가능하다
SyntaxError: can't assign to operator
>>> math1 = 80
>>> 1math = 80      # 에러. 숫자로 시작할 수 없다
SyntaxError: invalid syntax
>>> 학생수 = 50     # 한글 변수명도 가능함
>>> print(score)
90
>>> print(학생수)
50
>>> a = 10; b = 20; c = 30 # 세미콜론(;) 이용 가능하다
>>> print(a,b,c)
10 20 30
```

## 4. 변수

코드 variable.py

```
print('안녕! 나는 지구인이야. 나는 다리가 4개 있어.')
```

```
PS C:\Users\User\Documents> python variable.py  
안녕! 나는 지구인이야. 나는 다리가 4개 있어.
```



텍스트에서 중요한 의미가 있는 부분을 분리

코드

```
print('안녕! 나는', '지구인', '이야. 나는 다리가', 4, '개 있어.')
```

```
PS C:\Users\User\Documents> python variable.py  
안녕! 나는 지구인 이야. 나는 다리가 4 개 있어.
```

## 4. 변수

```
PS C:\Users\User\Documents> python variable.py  
안녕! 나는 지구인이야. 나는 다리가 4개 있어.
```

전

```
PS C:\Users\User\Documents> python variable.py  
안녕! 나는 지구인 이야. 나는 다리가 4 개 있어.
```

후

코드

```
print('안녕! 나는', '지구인', '이야. 나는 다리가', 4, '개 있어.')
```



※수정된 코드에서 숫자 4에는 따옴표가 없는데, 숫자는 따옴표 없이도 파이썬이 이해할 수 있다.

**중요한 내용** 두 개를 변수로 만든다.

## 4. 변수

### 코드

```
identity = '지구인' ❶  
number_of_legs = 4  
print('안녕! 나는', identity, '이야. 나는 다리가', number_of_legs, '개 있어.') ❷
```



변수

```
PS C:\Users\User\Documents> python variable.py
```

```
안녕! 나는 지구인 이야. 나는 다리가 4 개 있어.
```

## 4. 변수

### 코드

```
identity = '지구인'
number_of_legs = 4
print('안녕! 나는', identity, '이야. 나는 다리가', number_of_legs, '개 있어.')

identity = '한국인'
number_of_legs = 2
print('안녕! 나는', identity, '이야. 나는 다리가', number_of_legs, '개 있어.')
```

안녕! 나는 지구인 이야. 나는 다리가 4 개 있어.

안녕! 나는 한국인 이야. 나는 다리가 2 개 있어.

## 4. 변수

---

identity = '한국인'

변수이름      할당      변수가 할당할 값



## 4. 변수 -Test

---

Season 이라는 변수를 만들고, 변수에 현재 계절이름을 넣어보세요.

“지금은 <계절이름>입니다.” 라고 출력되도록 해보세요.

예)

지금은 봄입니다.

## 5. 자료형

### ◆ 파이썬 자료형 (data types)

#### ■ 기본 자료형

- int (정수)
- float (실수)
- bool (True/False)
- complex (복소수)

#### ■ 군집으로 다루는 자료형

- str (문자열)
  - list (리스트)
  - tuple (튜플)
  - set (집합)
  - dict (사전)
- 4장

```
>>> a = 10
>>> type(a)
<class 'int'>
>>> b = 3.5
>>> type(b)
<class 'float'>
>>> c = True
>>> type(c)
<class 'bool'>
>>> d = 2+5j
>>> type(d)
<class 'complex'>
>>> e = 'sogang'      # 문자열
>>> type(e)
<class 'str'>
>>> f = [1,3,5,7,9]    # 리스트
>>> type(f)
<class 'list'>
>>> g = (2,4,6)         # 튜플
>>> type(g)
<class 'tuple'>
>>> h = {1,2,3,4,5}     # 집합
>>> type(h)
<class 'set'>
>>> i = {3:90, 5:93, 2:85} # 사전
>>> type(i)
<class 'dict'>
```

## 5. 자료형

```
>>> a = 10
>>> type(a)
<class 'int'>
>>> b = 3.5
>>> type(b)
<class 'float'>
>>> c = True
>>> type(c)
<class 'bool'>
>>> d = 2+5j
>>> type(d)
<class 'complex'>
```

```
>>> e = 'sogang'      # 문자열
>>> type(e)
<class 'str'>
>>> f = [1,3,5,7,9]    # 리스트
>>> type(f)
<class 'list'>
>>> g = (2,4,6)        # 튜플
>>> type(g)
<class 'tuple'>
>>> h = {1,2,3,4,5}    # 집합
>>> type(h)
<class 'set'>
>>> i = {3:90, 5:93, 2:85} # 사전
>>> type(i)
<class 'dict'>
```

## 5. 자료형

### ◆ 정수형 (int)

- 정수형 상수는 10진수, 2진수, 8진수, 16진수가 있다.

```
>>> a = 23
>>> type(a)
<class 'int'>
>>> isinstance(a, int)
True
>>> b = 0o23    # 8진수
>>> c = 0x23    # 16진수
>>> d = 0b1101  # 2진수
>>> print(a,b,c,d)
23 19 35 13
```

**isinstance**(데이터, 자료형)  
데이터가 자료형에 속하는지  
묻는다. True 또는 False로  
결과를 낸다.

## 5. 자료형

### ◆ 정수형 (int)

- 10진수를 2, 8, 또는 16진수로 변환시키는 함수

```
>>> x = bin(10)
>>> y = oct(10)
>>> z = hex(20)
>>> print(x, y, z)
0b1010 0o12 0x14
```

- 다른 자료형으로부터 정수로 변환시키는 함수 - int

```
>>> a = int(5.6)    # 소수점 뒤를 무조건 버림
>>> b = int(7.0)
>>> c = int(-2.9)
>>> d = int('15')
>>> print(a,b,c,d)
5 7 -2 15
```

## 5. 자료형

### ◆ 정수형 (int)

- 정수형을 다른 자료형으로 변환하기

```
>>> a = 123
>>> x = float(a)    # 실수로 변환
>>> y = str(a)       # 문자열로 변환
>>> print(a, x, y)
123 123.0 123
```

- 정수 데이터는 값의 범위 제한이 없다

```
>>> 2 ** 1024      # 2의 1024제곱
179769313486231590772930519078902473361797697894230657273430081157
732675805500963132708477322407536021120113879871393357658789768814
416622492847430639474124377767893424865485276302219601246094119453
082952085005768838150682342462881473913110540827237163350510684586
298239947245938479716304835356329624224137216
```

## 5. 자료형

### ◆ 실수형 (float)

- 소수점을 포함하는 수로 e 또는 E 표현을 이용하기도 한다.

```
>>> a = 1.5
>>> type(a)
<class 'float'>
>>> isinstance(a, float)
True
>>> b = 2e5          #  $2 \times 10^5$ 
>>> c = -2E-4        #  $(-2) \times 10^{-4}$ 
>>> print(a,b,c)
1.5 200000.0 -0.0002
>>> float('inf')     # 무한대의 아주 큰 양수를 표현
inf
>>> float('-inf')     # 무한대의 아주 작은 음수를 표현
-inf
```



## 5. 자료형

코드 number.py

```
my_name = 'Python'      # 문자열(사람이 읽는 텍스트)
my_age = 25              # 숫자
print(my_name, '은 이제', my_age, '살')
```

Python 은 이제 25 살

my\_age는 숫자이다.

코드

```
my_name = 'Python'      # 문자열(사람이 읽는 텍스트)
my_age = '25'            # 숫자(가 든 문자열)
print(my_name, '은 이제', my_age, '살')
```

Python 은 이제 25 살

my\_age는 문자열이다.



## 5. 자료형

코드 number.py

```
my_name = 'Python'      # 문자열(사람이 읽는 텍스트)
my_age = 25              # 숫자
print(my_name, '은 이제', my_age, '살')
```

```
my_next_age = my_age + 1
print('내년에는', my_next_age, '살')
```

Python 은 이제 25 살  
내년에는 26 살

my\_age가 숫자이므로 덧셈 연산을 할 수 있다.

## 5. 자료형

### 코드

```
my_name = 'Python'          # 문자열(사람이 읽는 텍스트)
my_age = '25'                # 숫자(가 든 문자열)
print(my_name, '은 이제', my_age, '살')
```

```
my_next_age = my_age + 1
print('내년에는', my_next_age, '살')
```

Python 은 이제 25 살

Traceback (most recent call last):

File "number.py", line 5, in <module>

```
my_next_age = my_age + 1
```

TypeError: Can't convert 'int' object to str implicitly

`my_age`가 문자열이므로 수를 더하는 연산은 할 수 없다. 오류가 발생한다.

## 5. 자료형

### 코드

```
multiply = 9 * 9    # = 81  
divide = 30 / 5     # = 6  
power = 2 ** 10     # = 1024  
remainder = 15 % 4  # = 3  
  
print(multiply, divide, power, remainder)
```

파이썬은 기본적인 사칙 연산과 제곱이 가능하다.  
24강에서 더 많은 수학 기능을 다룬다.

## 5. 자료형

연산자	기능
+	덧셈
-	뺄셈
*	곱셈
/	나눗셈
**	거듭제곱
%	나머지

## 5. 자료형

### 코드

```
text = '2015' + '1991'  
number = 2015 + 1991  
  
print(text)  
print(number)
```

20151991

← 문자열 두 개가 연결되어 출력됨

4006

← 두 숫자를 더한 값이 출력됨

+ 연산자가 문자열에 쓰일 경우 문자열 두개를 연결시키는 기능을 하고,  
수에 쓰일 경우 일반 사칙연산 기능을 한다.

## 5. 자료형

---

파이썬의 '수'에는 두 가지 종류가 있는 것으로 이해한다.

**정수**로만 표현 가능한 수  $\rightarrow$  정수

**소수점** 이하도 함께 표현 가능한 수  $\rightarrow$  실수

## 5. 자료형

### 코드

```
five4 = 5 * 1    # 정수만 다룰 때  
five5 = 5 * 1.0  # 실수도 함께 다룰 때  
  
print(five4)  
print(five5)
```

5

5.0

+, -, × : 정수만 사용하면 결과값도 정수, 실수가 함께 사용되면 결과값은 실

### 스 코드

```
print(6 / 5)  
print(10 / 5)
```

1.2

2.0

나눗셈은 예외로 정수만 사용해도 실수가 나올 수 있다.

## 5. 자료형

### 코드

```
print(6 % 5)
print(10 % 5)
```

1

0

### 코드

```
div1 = 6 / 5      ❶ / : 나눗셈
div2 = 6 // 5     ❷ // : 몫을 구한다
```

```
print(div1)
print(div2)
```

```
a = 6
b = 5
```

```
print(a == b * (a // b) + (a % b))    # 나눗셈의 정의
```

1.2

1

True



## 5. 자료형

### 코드

```
print(int(5.0)) # 정수로 바꾸기  
print(float(5)) # 부동소수점으로 바꾸기  
print(5 * 1.0) # 부동소수점으로 바꾸기
```

5

5.0

5.0

정수 integer

부동소수점 floating point

## 5. 자료형\_Test

---

정수변수 a와 b를 선언하고, 연산값을 출력해보세요.

A = 21

B = 5

Sum =

Multiply =

Divide =

Remainder =

Power =

## 5. 자료형\_Test

---

**문자열 변수 a와 b, c를 선언하고, 연결해보세요.**

**Year = 1978**

**Month = 07**

**Date = 07**

## 5. 자료형

### ◆ 실수형 (float)

- 실수를 정수로 변환하기 – 반올림(round), 올림(ceil), 내림(floor)

```
>>> round(1.2)
```

```
1
```

```
>>> round(2.5)
```

```
2
```

```
>>> round(2.6)
```

```
3
```

```
>>> round(2.51)
```

```
3
```

```
>>> round(2.50001)
```

```
3
```

```
>>> import math
```

```
>>> math.ceil(1.2)    # 1.2보다 크거나 같은 정수
```

```
2
```

```
>>> math.floor(1.9)   # 1.9보다 작거나 같은 정수
```

```
1
```

## 5. 자료형

### ◆부울형 (bool)

파이썬의 부울형은 참 또는 거짓을 나타내는 True와 False 두 만을 갖는다.

```
>>> a = 1
```

```
>>> a < 0
```

```
False
```

```
>>> a > 0
```

```
True
```

```
>>> b = True
```

```
>>> c = False
```

```
>>> print(b,c)
```

```
True False
```

## 5. 자료형

### ◆복소수형 (complex)

복소수형은 실수부와 허수부로 표현한다. 허수부 뒤에는 j 또는 J를 붙인다.

```
>>> a = 3 + 5j
>>> b = 10 - 2j
>>> c = a + b
>>> print(c)
(13+3j)
```

```
>>> a.real      # 복소수 a의 실수부
3.0
```

```
>>> a.imag      # 복소수 a의 허수부
5.0
```

```
>>> b.conjugate()  # 복소수 b의 켤레복소수
(10+2j)
```

## 6. 주석

---

- 주석은 코드를 설명하는 내용이나 참고 사항 등을 코드 내에 기술하기 위해 사용한다.
- 코드 실행에는 영향을 미치지 않는다.
- 주석의 종류
  - # - 이 기호부터 줄 끝까지 주석 처리한다.
  - 여러 줄 주석 - 홀따옴표 세 개(''...'') 또는 쌍따옴표 세 개(''''...''')로 주석을 표현한다.

## 6. 주석

```
name = 'sogang' # 학교명을 저장한다.
```

```
''' 여기부터는 주석입니다.  
프로그램 실행과는 무관한 줄입니다.  
프로그램 관련해서 설명할 내용이 있으면  
이렇게 주석으로 기록해 두면 좋습니다'''
```

docstring 이  
라고도 한다.

```
a = 10  
b = 20  
sum = a + b  
print(sum) # 변수 sum 값을 출력한다.
```