

# 변수와 연산자



소프트웨어융합대학  
교수 진혜진

# 목차

1. 변수의 이해
2. 기본 자료형
3. 화면 입출력
4. 연산자

## TIOBE Index for August 2019



### August Headline: Silly season in the programming language world

Nothing much has changed during July in the TIOBE index. In the top 10 only Objective-C and SQL have swapped positions. We need a magnifying glass to see some other noteworthy changes: Rust went from #33 to #28, TypeScript from #41 to #35 and Julia from #50 to #39. It is also interesting to note that Kotlin doesn't seem to come closer to the top 20. This month it even lost 2 positions: from #43 to #45.

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the *best* programming language or the language in which *most lines of code* have been written.

The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system. The definition of the TIOBE index can be found [here](#).

Aug 2019	Aug 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.028%	-0.85%
2	2		C	15.154%	+0.19%
3	4	▲	Python	10.020%	+3.03%
4	3	▼	C++	6.057%	-1.41%
5	6	▲	C#	3.842%	+0.30%
6	5	▼	Visual Basic .NET	3.695%	-1.07%
7	8	▲	JavaScript	2.258%	-0.15%

- 프로그래밍에서 변수(variable)는 데이터를 저장하는 임시 저장공간이다.
- 변수는 컴퓨터 메모리 공간에 만들어 진다.
  - 변수에 값이 저장되는 공간을 메모리라고 한다.
- 변수에 값을 넣으라고 선언하는 순간, 물리적으로 메모리 어딘가에 물리적인 공간을 확보할 수 있게 운영체제와 파이썬 인터프리터가 협력하여 메모리 저장 위치를 할당한다.
- 파이썬에서 변수를 생성하려면 다음과 같이 한다.
  - name = "JIN KONG "
  - Age=10

```
test.py - C:/Users/hjjin/AppData/Local/Programs/Python/Python37-32/test.p...
File Edit Format Run Options Window Help
name = "jin kong"
age= 10
print(name, age)
print(name, "님은", age, "살 입니다.")
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:
52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for mo
re information.
>>>
= RESTART: C:/Users/hjjin/AppData/Local/Programs/Python/P
ython37-32/test.py =
jin kong 10
jin kong님은 10 살 입니다.
Ln: 7 Col: 4
```

## ■ 변수명 선언

- 알파벳, 숫자, 밑줄[ \_ ]로 선언할 수 있다.
- 변수명은 의미 있는 단어로 표기하는 것이 좋다.
- 변수명은 대소문자가 구분된다.
- 변수의 이름 중간에 공백이 들어가면 안 된다. 단어를 구분하려면 밑줄[\_]을 사용 한다.
- 특별한 의미가 있는 예약어는 사용할 수 없다.
  - True, False, None, and, or, not, break, continue, return, if, else, elif, for, while, except, finally, global, import, try

## ■ 변수 사용 예

- **Number**
  - 영문 알파벳 문자로 시작
- **\_count**
  - 밑줄 문자로 시작할 수 있다.
- **max\_Number**
  - 중간에 밑줄 문자를 넣을 수 있다.
- **num3**
  - 맨 처음이 아니라면 숫자도 넣을 수 있다.
- **1nd\_base [X]**
  - 숫자로 시작할 수 없다.
- **Score\$ [X]**
  - \$과 같은 기호는 사용할 수 없다.

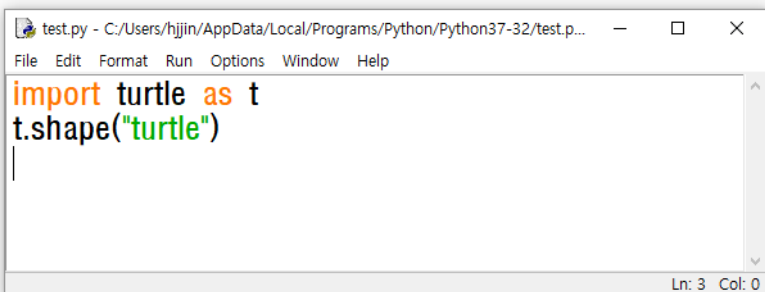
## ■ 변수의 사용

- 변수는 값을 대입하면 사용이 가능하다.
- 변수에 있던 기존 값은 없어지고 새로 입력한 값으로 변경된다.

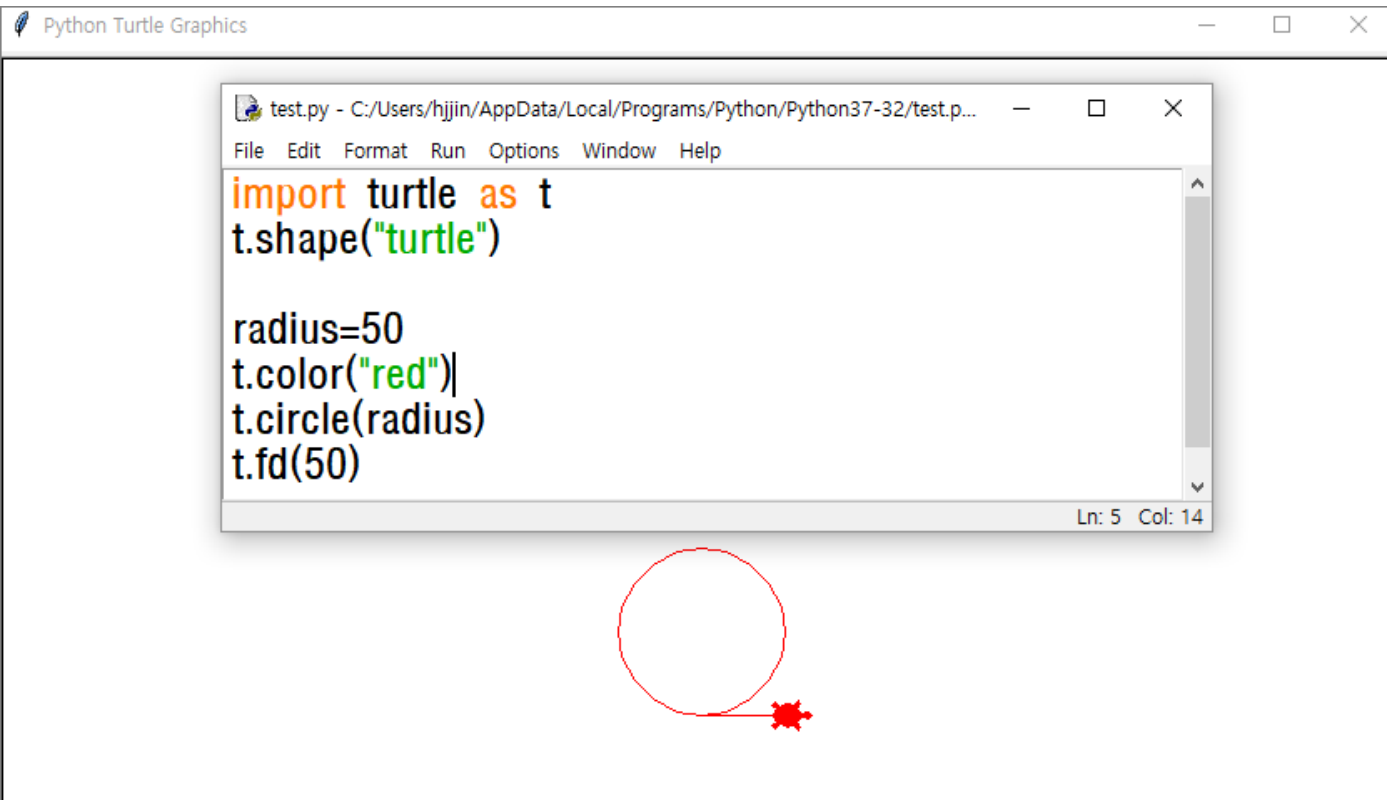
```
>>> boolVar = True
>>> intVar = 0
>>> floatVar = 0.0
>>> strVar = ""
>>> boolVar = False
>>> intVar = 300
>>> floatVar = 213.45
>>> strVar = "진혜진"
>>> boolVar
False
>>> intVar
300
>>> floatVar
213.45
>>> strVar
'진혜진'
```



## ■ 변수 응용

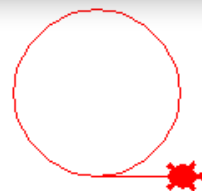


```
test.py - C:/Users/hjjin/AppData/Local/Programs/Python/Python37-32/test.p...
File Edit Format Run Options Window Help
import turtle as t
t.shape("turtle")
|
Ln: 3 Col: 0
```



```
Python Turtle Graphics
test.py - C:/Users/hjjin/AppData/Local/Programs/Python/Python37-32/test.p...
File Edit Format Run Options Window Help
import turtle as t
t.shape("turtle")

radius=50
t.color("red")
t.circle(radius)
t.fd(50)
Ln: 5 Col: 14
```

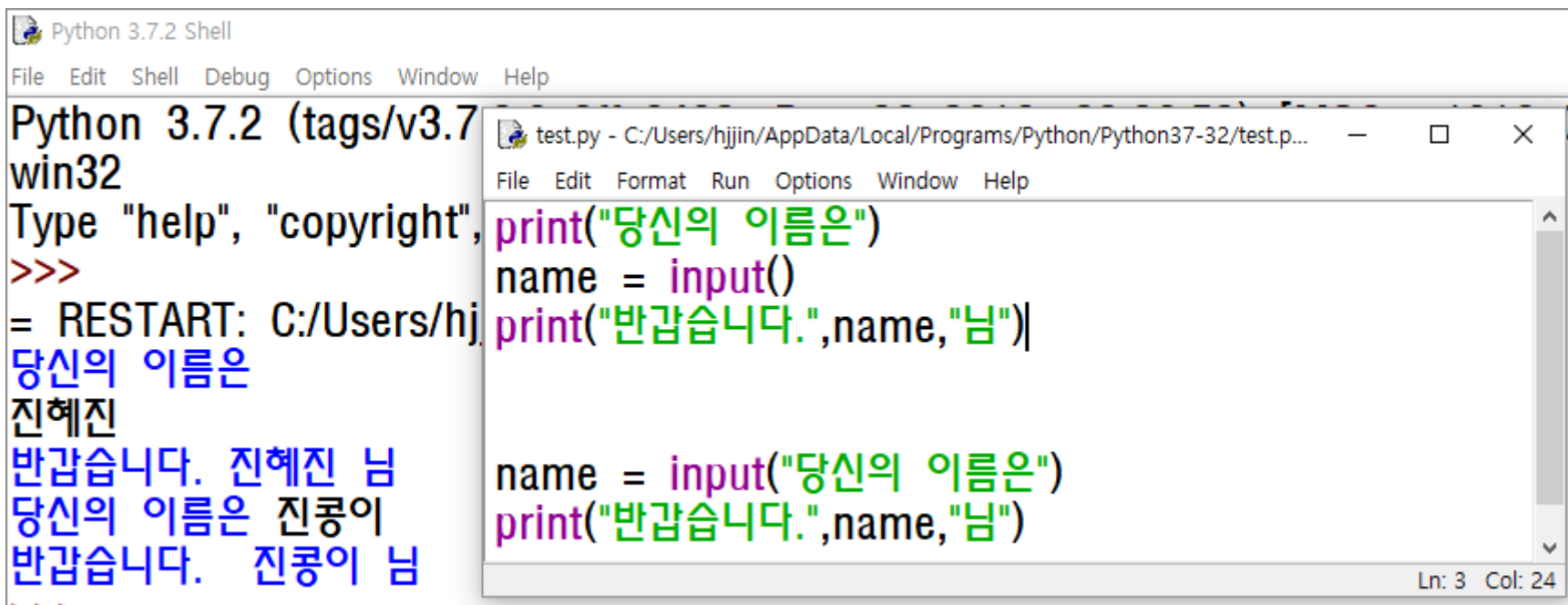




- **정수형(integer type)** : 자연수를 포함해 값의 영역이 정수로 한정된 값
  - EX) 10, 30,
  - NUMBER =1
- **실수형(floating-point type)** : 소수점이 포함된 값
  - 3.1, 40.67
  - GRADE=4.5
- **문자형(string type)** : 값이 문자로 출력되는 값
  - 'A' , "AB"
  - BLOOD= "AB"
- **불린형(boolean type)** : 논리형으로, 참(True) 또는 거짓(False)을 표현할 때 사용
  - True, False
  - RESULT =True

#### ■ 표준 입력 함수: input( ) 함수

- 표준 입력 함수로, 사용자가 문자열을 콘솔 창에 입력할 수 있게 해 준다.



The image shows two overlapping windows from a Python 3.7.2 environment. The background window is the 'Python 3.7.2 Shell' with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). It displays the prompt 'Python 3.7.2 (tags/v3.7.2:1, Mar 9 2019, win32)' and the instruction 'Type "help", "copyright", "credits()", or "quit()>>>'. Below this, it shows a restart message and the execution of a script. The script's output is displayed in blue text: '당신의 이름은', '진혜진', '반갑습니다. 진혜진 님', '당신의 이름은 진콩이', and '반갑습니다. 진콩이 님'. The foreground window is a text editor titled 'test.py - C:/Users/hjjin/AppData/Local/Programs/Python/Python37-32/test.p...' with a menu bar (File, Edit, Format, Run, Options, Window, Help). It contains the following Python code: 


```
print("당신의 이름은")
name = input()
print("반갑습니다.",name,"님")

name = input("당신의 이름은")
print("반갑습니다.",name,"님")
```

 The status bar at the bottom right of the editor shows 'Ln: 3 Col: 24'.

- **표준 출력 함수: print( ) 함수**

- 표준 출력 함수로, 결과를 화면에 출력하는 함수이다.



The screenshot shows a Python 3.7.4 IDE window titled "0907\_02.py - C:/Users/hjjin/OneDrive/바탕 화면/0907\_02.py (3.7.4)". The code editor contains the line `print("컴퓨터 프로그래밍")`. Below the editor is a "Python 3.7.4 Shell" window. The shell displays the Python version and date information, followed by the prompt `>>>`. The output of the print statement is shown as `컴퓨터 프로그래밍` in blue text.

```
0907_02.py - C:/Users/hjjin/OneDrive/바탕 화면/0907_02.py (3.7.4)
File Edit Format Run Options Window Help

print("컴퓨터 프로그래밍")

Python 3.7.4 Shell
File Edit Shell Debug Options Window Help

Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29)
Type "help", "copyright", "credits" or "license()" for more
>>>
===== RESTART: C:/Users/hjjin/OneDrive/
컴퓨터 프로그래밍
```

#### ■ 형변환

```
test.py - C:/Users/hjjin/AppData/Local/Programs/Python/Python37-32/test.p...
File Edit Format Run Options Window Help

#두 정수의 합을 구하는 프로그램
num1 = input("정수를 입력하세요")
num2 = input("정수를 입력하세요")
sum = num1+num2
print(sum)

Python 3.7.2 Shell
File Edit Shell Debug Options Window Help

정수를 입력하세요20
정수를 입력하세요30
2030
```

```
test.py - C:/Users/hjjin/AppData/Local/Programs/Python/Python37-32/test.p...
File Edit Format Run Options Window Help

#두 정수의 합을 구하는 프로그램
num1 = int(input("정수를 입력하세요"))
num2 = int(input("정수를 입력하세요"))
sum = num1+num2
print(sum)

Python 3.7.2 Shell
File Edit Shell Debug Options Window Help

= RESTART: C:/Users/hjjin/AppData/Local/Programs/Pyt
정수를 입력하세요20
정수를 입력하세요30
50
```

## ■ 산술 연산자

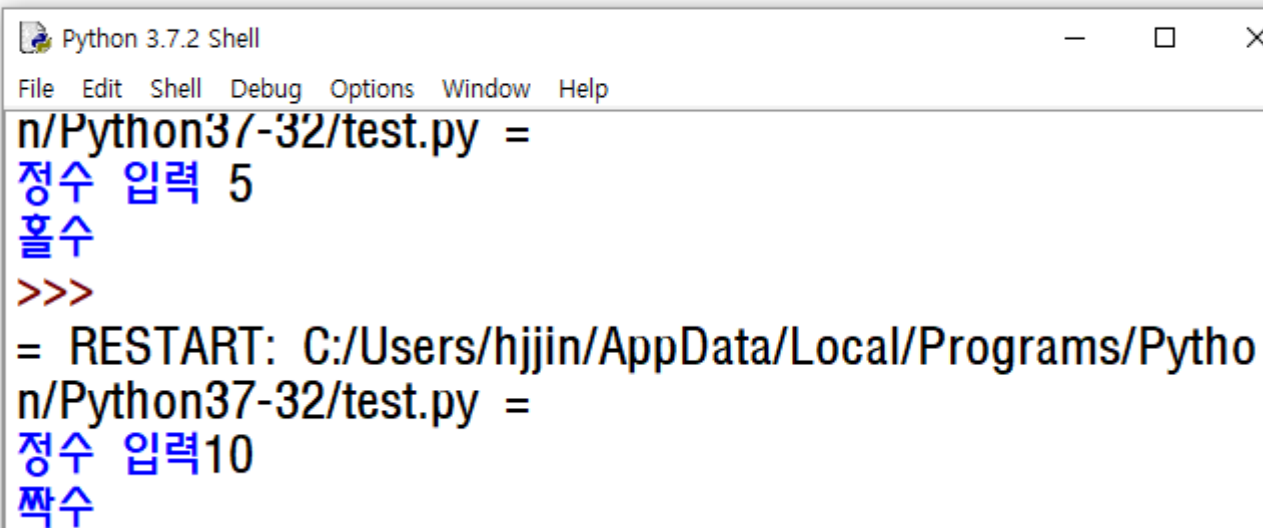
연산자	기호	사용예	결과
덧셈	+	5+3	8
뺄셈	-	5-3	2
곱셈	*	5*3	15
나눗셈	/	5/3	1.6666666666666667
나눗셈	//	5//3	1
나머지	%	5%3	2

```

Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
>>> 5+3
8
>>> 5-3
2
>>> 5*3
15
>>> 5**3
125
>>> 5/3
1.6666666666666667
>>> 5//3
1
>>> 5%3
2
>>> |
Ln: 37 Col: 4
    
```

#입력 받은 정수가 짝수인지 홀수 인지 판단하는 프로그램

```
number = int(input("정수 입력"))  
if number%2 ==0:  
    print("짝수")  
else:  
    print("홀수")
```



```
Python 3.7.2 Shell  
File Edit Shell Debug Options Window Help  
n/Python37-32/test.py =  
정수 입력 5  
홀수  
>>>  
= RESTART: C:/Users/hjjin/AppData/Local/Programs/Python/Python37-32/test.py =  
정수 입력10  
짝수
```

## ■ 관계 연산자

- 어떤 것이 크거나 작거나 같은지 비교하는 것(참은 True로, 거짓은 False로 표시)
- 주로 조건문(if)이나 반복문(while)에서 사용, 단독으로는 거의 사용하지 않는다.

$x < y$	~보다 작음	x가 y보다 작은지 검사
$x > y$	~보다 큼	x가 y보다 큰지 검사
$x == y$	같음	x와 y의 값이 같은지 검사
$x \text{ is } y$	같음(메모리 주소)	x와 y의 메모리 주소가 같은지 검사
$x != y$	같지 않음	x와 y의 값이 같지 않은지 검사
$x \text{ is not } y$	같지 않음(메모리 주소)	x와 y의 메모리 주소가 같지 않은지 검사
$x \geq y$	크거나 같음	x가 y보다 크거나 같은지 검사
$x \leq y$	작거나 같음	x가 y보다 작거나 같은지 검사

```

>>> x=30; y=50
>>> x<y
True
>>> x>y
False
>>> x==y
False
>>> x is y
False
>>> x !=y
True
>>> x is not y
True
>>> x>=y
False
>>> x<=y
True
    
```

### ■ 논리 연산자

and	두 값이 모두 참일 경우 True, 그렇지 않을 경우 False
or	두 값 중 하나만 참일 경우 True, 두 값 모두 거짓일 경우 False
not	값을 역으로 반환하여 판단

```
>>> a=5; b=7
>>> a==8 and b==7
False
>>> a==5 and b==7
True
>>> a==3 or b==7
True
>>> a==3 or a==2
False
>>> not a==5
False
>>> not b!=7
True
```