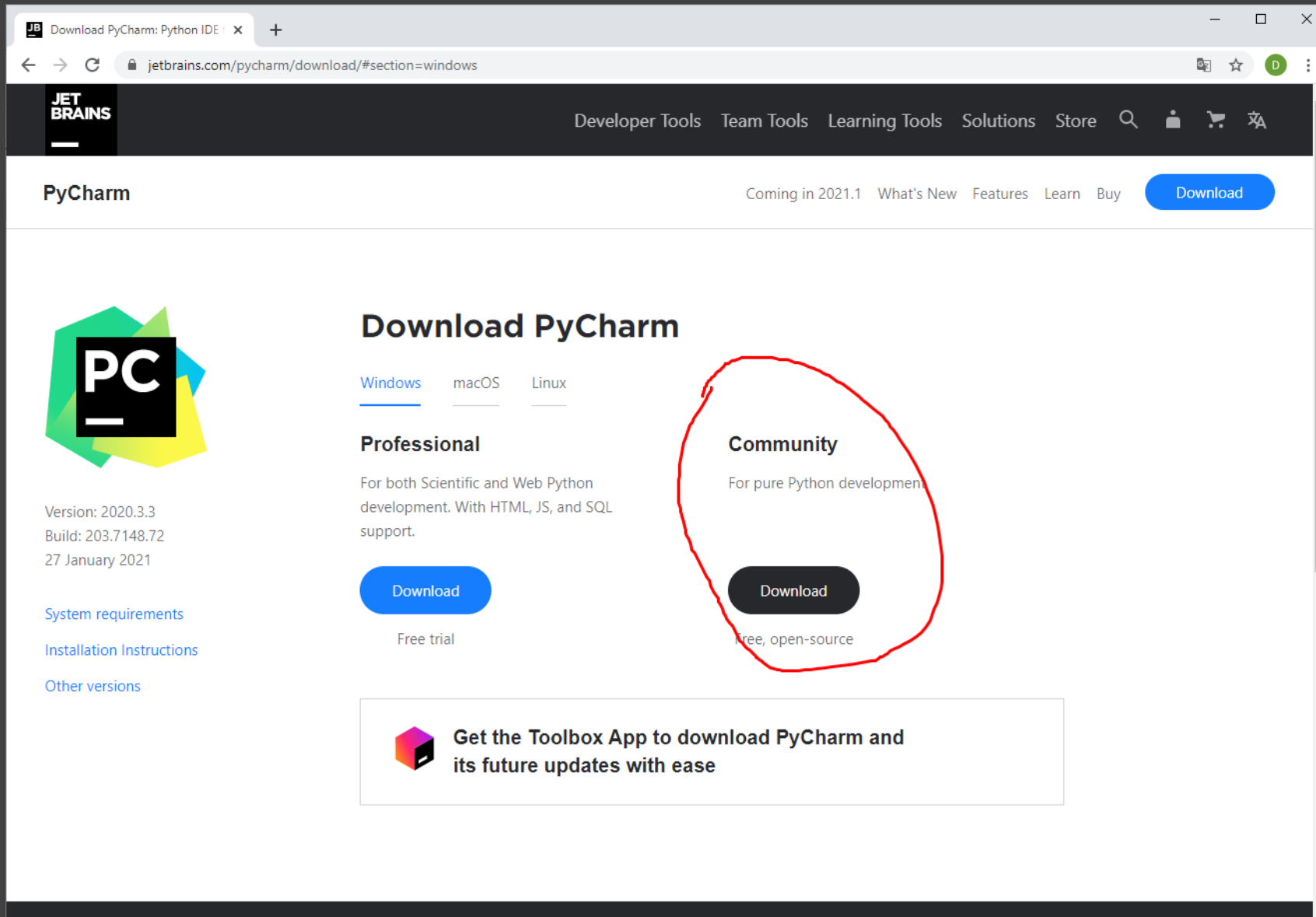


제2강 Control Flow

학습 목차

- PyCharm 설치
- 비교 연산
- Boolean 연산

<https://www.jetbrains.com/pycharm/>



JB Download PyCharm: Python IDE x +


jetbrains.com/pycharm/download/#section=windows

JET BRAINS

Developer Tools Team Tools Learning Tools Solutions Store

PyCharm

Coming in 2021.1 What's New Features Learn Buy [Download](#)



Version: 2020.3.3
Build: 203.7148.72
27 January 2021

[System requirements](#)
[Installation Instructions](#)
[Other versions](#)

Download PyCharm

[Windows](#) [macOS](#) [Linux](#)

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

[Download](#)


Free trial

Community

For pure Python development.

[Download](#)

Free, open-source

 Get the Toolbox App to download PyCharm and its future updates with ease

비교 연산

기호	뜻
<	작다
<=	작거나 같다
==	같다
>=	크거나 같다
>	크다
!=	다르다
is	동일한 객체인가?
is not	다른 객체인가?

Boolean 연산

- **and**
- **or**
- **not**

Table 2-2: The and Operator's Truth Table

Expression	Evaluates to...
True and True	True
True and False	False
False and True	False
False and False	False

Table 2-3: The or Operator's Truth Table

Expression	Evaluates to...
True or True	True
True or False	True
False or True	True
False or False	False

Table 2-4: The not Operator's Truth Table

Expression	Evaluates to...
not True	False
not False	True

is 과 ==

- is : 같은 객체인가?
- == : 같은 값인가?
- 뭐가 더 빠를까?

```
a = 1000  
b = 1000  
a == b  
a is b
```

```
c = 'abcd'  
d = 'abcd'  
c == d  
c is d
```

```
id(c)  
id(d)
```

```
x = 3  
y = 3  
id(x)  
id(y)  
x is y
```

인터닝

- 자주 사용하는 상수들은 내부적으로 인터닝(Interning)함으로써, 동일한 id 를 갖게 만듦.
- 기본적으로 다음과 같은 값은 기본 인터닝하고 있음.
 - 정수 : -5 ~ 256
 - 문자열: 대소문자와 숫자로 구성된 문자열

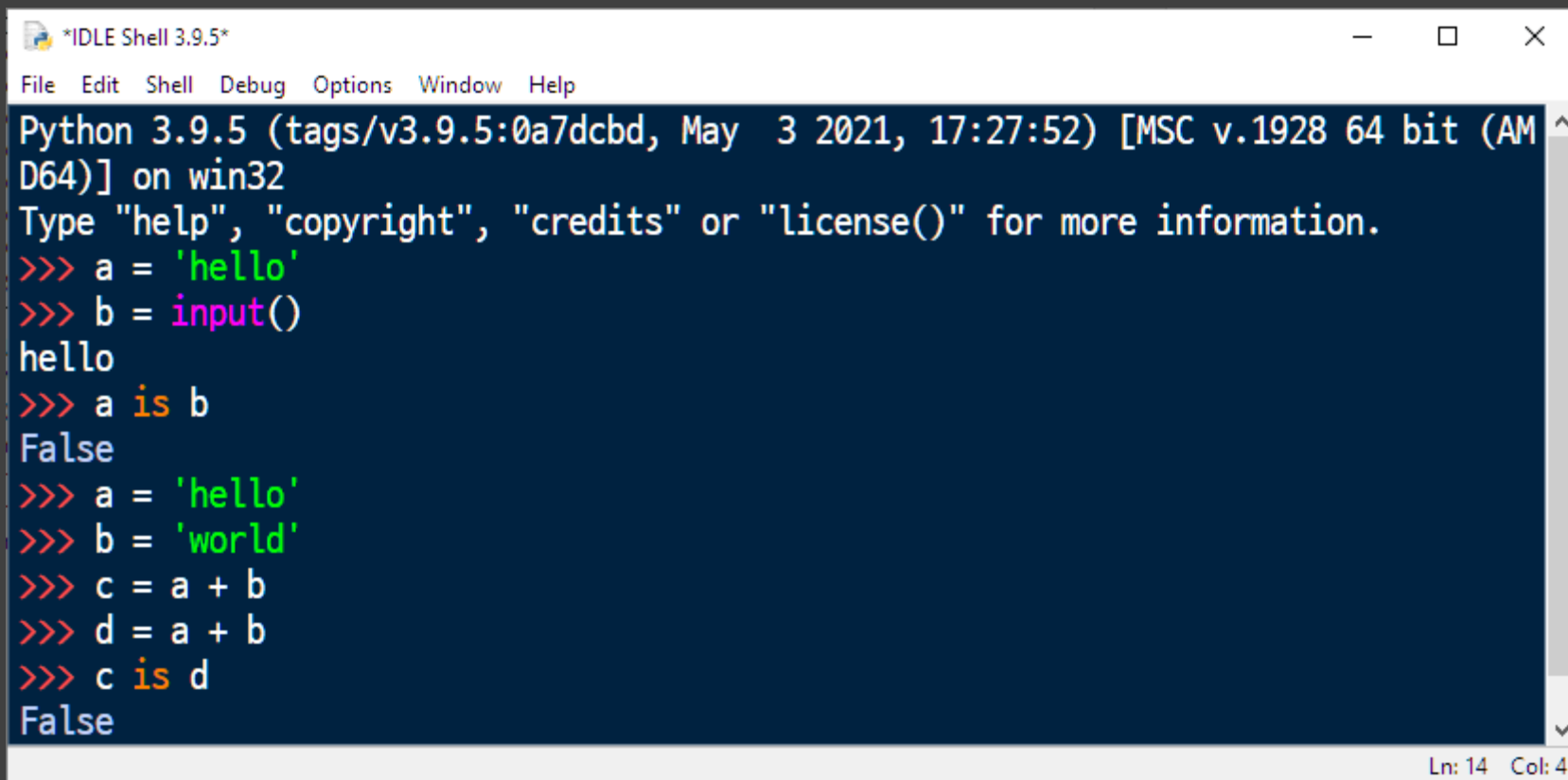
intern() - 상수에 대해서 강제로 내부 캐싱을 통해 속도를 높임.

```
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = 'hello'
>>> b = 'hello'
>>> a is b
True
>>> a = 'hello, world'
>>> b = 'hello, world'
>>> a is b
False
>>> |
```

```
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
>>> a = '이대현'
>>> b = '이대현'
>>> a is b
False
>>> a = 'leedaehyun'
>>> b = 'leedaehyun'
>>> a is b
True
>>> |
Ln: 31 Col: 4
```

```
import sys
a = sys.intern('이대현')
b = sys.intern('이대현')
a is b
```


상수가 아니면 적용되지 않음.




```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = 'hello'
>>> b = input()
hello
>>> a is b
False
>>> a = 'hello'
>>> b = 'world'
>>> c = a + b
>>> d = a + b
>>> c is d
False
Ln: 14 Col: 4
```

if 문법

- 조건을 검사하여, 그 결과에 따라 처리를 하는 문장

```
if age >= 60:  
    print(age)  
    print("you are very old")
```



들여쓰기(indentation)

*** 매우 중요 ***

일반적으로 공백4개씩

조건이 참이면, 들여쓰기된 블록을 실행함.

비교 연산에서 절대 해서는 안될 표현

if success is True:

if success == True:

if success is False:

if success == False:

if 문 확장

```
if age >= 60:  
    print(age)  
    print("you are very old")  
elif age <= 20:  
    print(age)  
    print("you are very young")  
else:  
    print(age)  
    print("you are young")
```

while 반복문 (Iteration Statement)

```
spam = 0
while spam < 5:
    print('Hello, world.')
    spam = spam + 1
```

break

```
while True:
    print('Please type your name.')
    name = input()
    if name == 'your name':
        break
print('Thank you!')
```

continue

```
while True:
    print('Who are you?')
    name = input()
    if name != 'Joe':
        continue
    print('Hello, Joe. What is the password? (It is a fish.)')
    password = input()
    if password == 'swordfish':
        break
print('Access granted.')
```

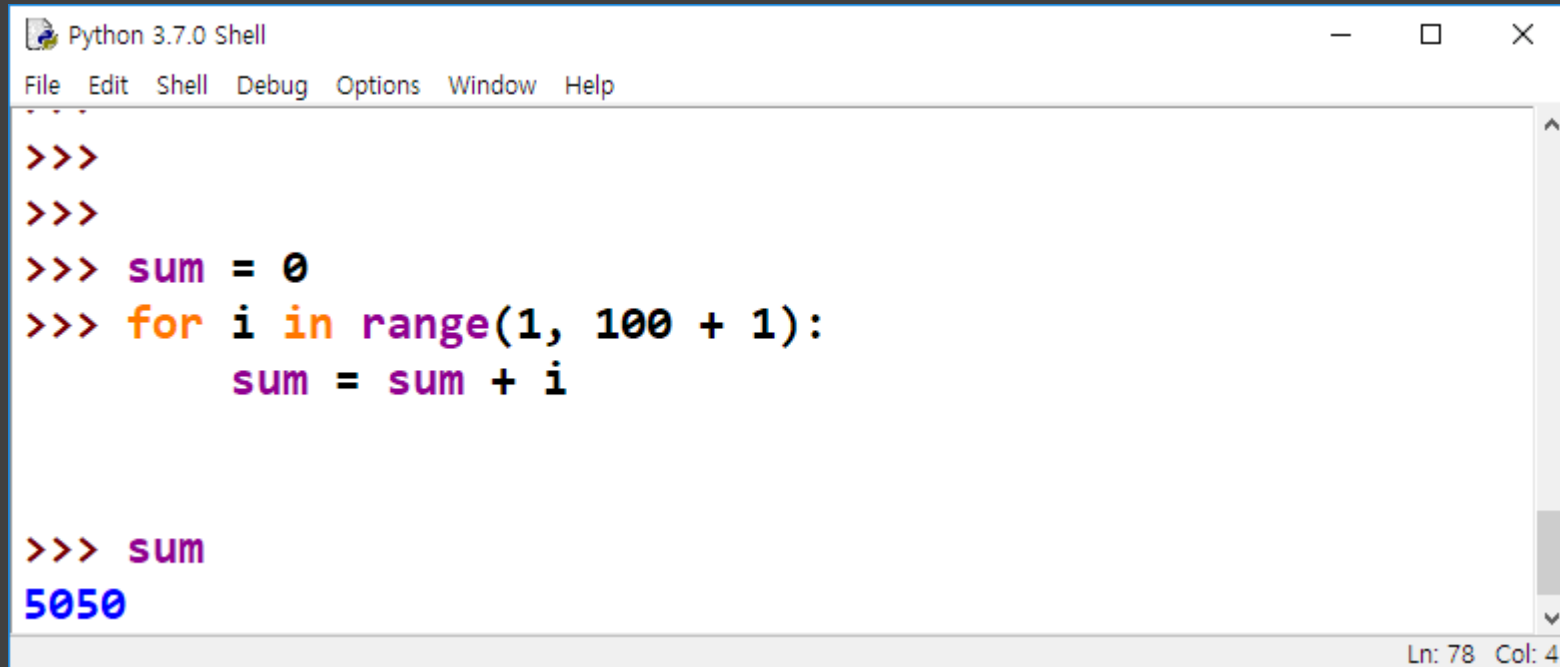
- **while** 과 **if** 안에서 **int 0**, **float 0.0**, **str ""** 은 모두 **False** 로 계산됨.

```
name = ''
while not name:❶
    print('Enter your name:')
    name = input()
print('How many guests will you have?')
numOfGuests = int(input())
if numOfGuests:❷
    print('Be sure to have enough room for all your guests.')
```

```
❸
print('Done')
```


for

- C 의 for 와는 개념적으로 다름.
- 시퀀스 값 (여러 개의 값)에 대해서 하나씩 처리.
- range() 와 결합되면 C의 for 와 비슷하게 사용하는 것이 가능.



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
>>>
>>>
>>> sum = 0
>>> for i in range(1, 100 + 1):
>>>     sum = sum + i
>>>
>>> sum
5050
Ln: 78 Col: 4
```

range()

- 함수처럼 보이지만, 실제로는 클래스임.
- `range(a, b, c)`
 - a부터 b까지, c씩 증가시킨 여러 개의 값들. 단, b는 제외

```
for i in range(12, 16):  
    print(i)
```

```
for i in range(0, 10, 2):  
    print(i)
```

```
for i in range(5, -1, -1):  
    print(i)
```

Module

- 이미 만들어진 함수들의 모음.
- 사용하기 위해서는 import 해야 함.
- 기존 모듈 이름을 파일 이름으로 사용하면 안됨.

```
import random
for i in range(5):
    print(random.randint(1, 10))
```

```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
>>>
>>> import random
>>> random.randint(1,6)
3
>>> random.randint(1,6)
4
>>> random.randint(1,6)
2
>>> random.randint(1,6)
4
>>> random.randint(1,6)
1
>>>
```

Ln: 519 Col: 0

random.randint(시작, 끝)

is 와 ==, 뭐가 더 빠를까?

```
>>> import timeit
>>> timeit.timeit('x == y', setup='x="1234"; y="1234"', number=1000000)
0.02156040002591908
>>> timeit.timeit('x is y', setup='x="1234"; y="1234"', number=1000000)
0.01755580003373325
```

pass

- do nothing
- **엄연한 하나의 문장으로 취급됨.**
- **아직 구체적인 구현 내용이 없을 때, 자리만을 잡아주는 역할.**

정리

- PyCharm
- 파이썬 파일의 실행
- 비교 연산과 불리언 연산
- if
- while
- for
- range
- 모듈
- pass