

# Python 프로그래밍

# 파이썬 (Python) 이란?



Guido Van Rossum

- 1991년 귀도 반 로섬(Guido Van Rossum)이 발표
- 플랫폼 독립적인 인터프리터 언어며,
- 객체 지향적, 동적 타이핑 언어
- 처음 C언어로 구현되었음

## 알아두면 좋은 이야기

- 귀도 반 로섬은 1989년 크리스마스에 할 일이 딱히없어 파이썬을 개발하였다고 함
- '파이썬'이라는 이름은 코메디 프로그램 'Monty Python's Flying Circus'에서 유래
- 파이썬의 원래 의미는 그리스 신화에 나오는 거대한 뱀
- 귀도는 구글에 근무했고 현재 DropBox에서 근무 중
- 귀도는 파이썬 개발자를 찾는 헤드헌터로 부터 취업 제안 메일을 받은 적이 있음

# 파이썬 (Python)

---



파이썬의 로고

두 마리의 뱀이 겹쳐 있는 모습

**플랫폼(?)**

**독립적인(?) 인**

**터프리터(?) 언어**

# 파이썬의 특징

---

## 플랫폼 = OS

: 윈도우, 리눅스, 안드로이드, 맥OS, iOS 등 프로그램이 실행되는 운영 체제를 플랫폼이라고 함

## 독립적인 = 관계없는, 상관없는

: OS에 상관없이 한번 프로그램을 작성하면 사용가능

## 인터프리터 = 통역기를 사용하는 언어...

: 소스코드를 바로 실행할 수 있게 지원하는 프로그램 실행 방법

원말인지 모르면 넘어가자. 괜찮다.

# 파이썬의 특징 2

---

- **가독성**
  - 간결하고 가독성이 좋습니다.
  - 코드블럭을 들여쓰기(indentation)로 구분.
- **풍부한 라이브러리**
  - 매우 광범위한 라이브러리가 기본으로 포함되어 있음.
  - 외부 라이브러리가 풍부하며 확장이 쉬움.
- **접착성**
  - 쉽게 라이브러리를 추가할 수가 있습니다.
  - 파이썬에서 C로 구현된 부분을 사용할 수도 있으며, C에서 파이썬을 사용할 수도 있습니다.
- **무료**
  - 파이썬은 파이썬 소프트웨어 재단(Python Software Foundation)에서 관리하고 있음.
  - 무료와 다름없는 Python Software Foundation License.
- **유니코드**
  - 파이썬에서의 문자열들은 모두 유니코드입니다.
  - 우리가 한글, 한자 등을 표현하려고 특별한 노력을 할 필요가 없습니다.
- **동적타이핑**
  - 런타임 시에 타입 체크를 하는 동적타이핑을 지원하며,
  - 메모리 관리 자동으로 합니다.

**객체 지향적(?)**  
**동적(?) 타이핑 언어**

# 파이썬의 특징 3

---

## 객체 지향적 언어

: 실행 순서가 아닌 단위 모듈(객체) 중심으로 프로그램을 작성 하나의 객체는 어떤 목적을 달성하기 위한 행동(method)와 데이터(attribute)를 가지고 있음

## 동적 타이핑 언어

: 프로그램이 실행하는 시점 프로그램이 가지고 있는 데이터에 대한 타입을 검사함

원말인지 모르면 넘어가자. 괜찮다.



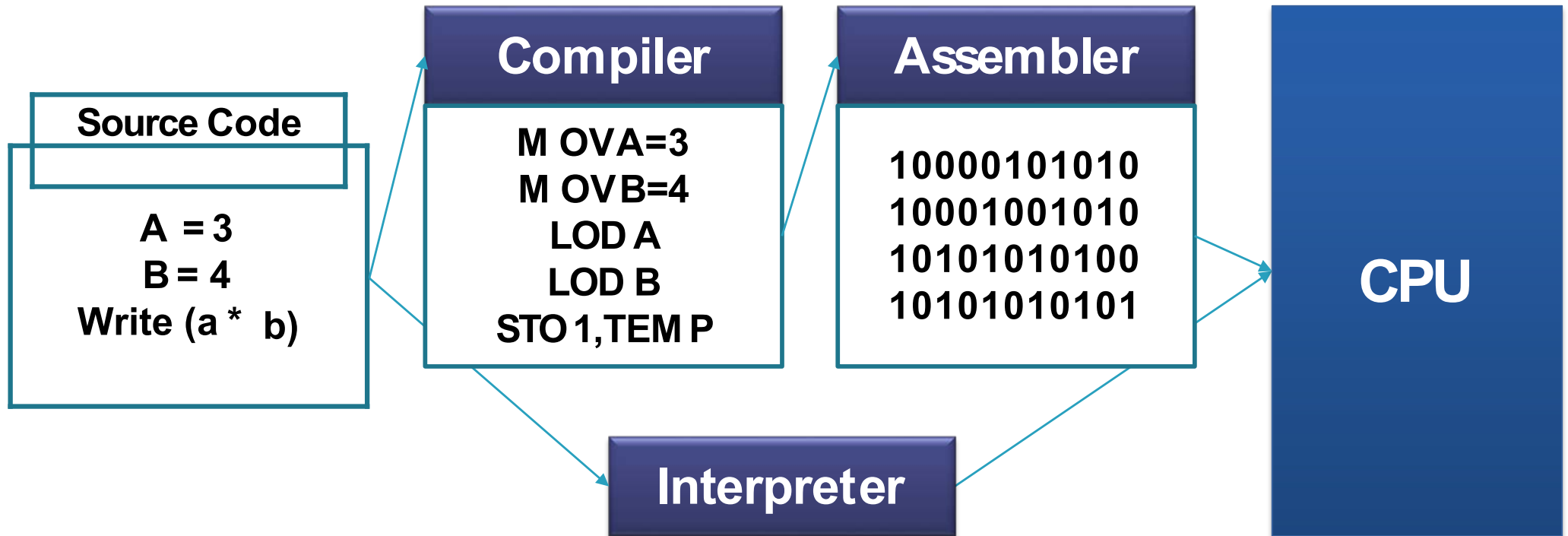
# 컴파일러언어 vs 인터프리터 언어

컴파일러	인터프리터
소스코드를 기계어로 먼저번역하고 해당 플랫폼에 최적화되어 프로그램을 실행	별도의 번역과정 없이 소스코드를 한줄 한줄분석하여 컴퓨터가 처리할 수 있도록 함
실행속도가 빠름 한번의 많은기억장소 필요	간단히 작성, 메모리가 적게필요 실행속도가 느림
C, 자바, C++, C#	파이썬, 스칼라

최근 인터프리터언어도 실행속도를올리기 위해사용자가 모르게컴파일한 다음 번역없이 실행

# 소프트웨어(프로그램)의 동작과정

사람이 알 수 있는 고급언어를 기계가 알수 있는 저급언어로 변환



파이썬은 처음에 컴파일러 언어인 C로 작성되었다.

당연히 실행시 **Assembler**와 같은 기계어 변환 과정을 거친다

# **Why Python?**

# 왜 파이썬인가 ?

## 인간 지향적인 간단한 문법

## JAVA vs Python

JAVA 변수 선언 여부

```
int myCounter = 0;  
String myString = String.valueOf(myCounter);  
if (myString.equals("0")) ...
```

PYTHON

```
myCounter = 0  
myString = str(myCounter)  
if myString == "0": ...
```

```
// print the integers from 1 to 9
```

```
for (int i = 1; i < 10; i++)
```

```
{  
    System.out.println(i);  
}
```

Loop 선언 형태

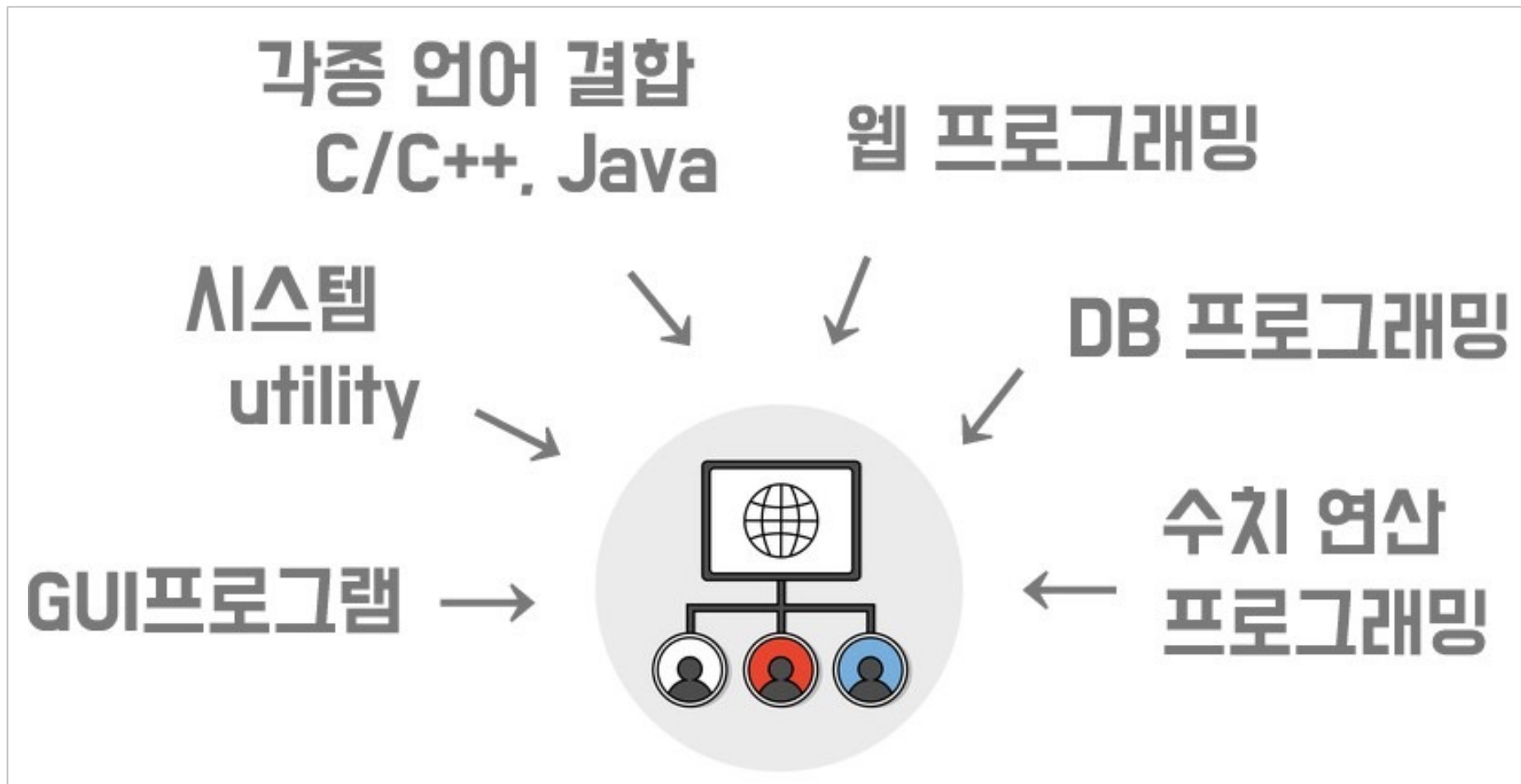
Brace 여부

```
print the integers from 1 to 9  
for i in range(1,10):  
    print i
```

프로그래밍을 몰라도 해석이 가능한 문법의 언어

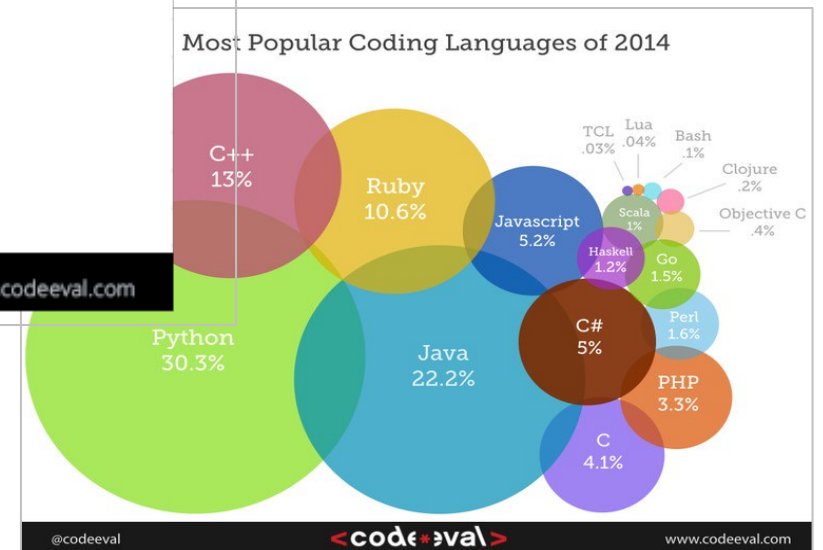
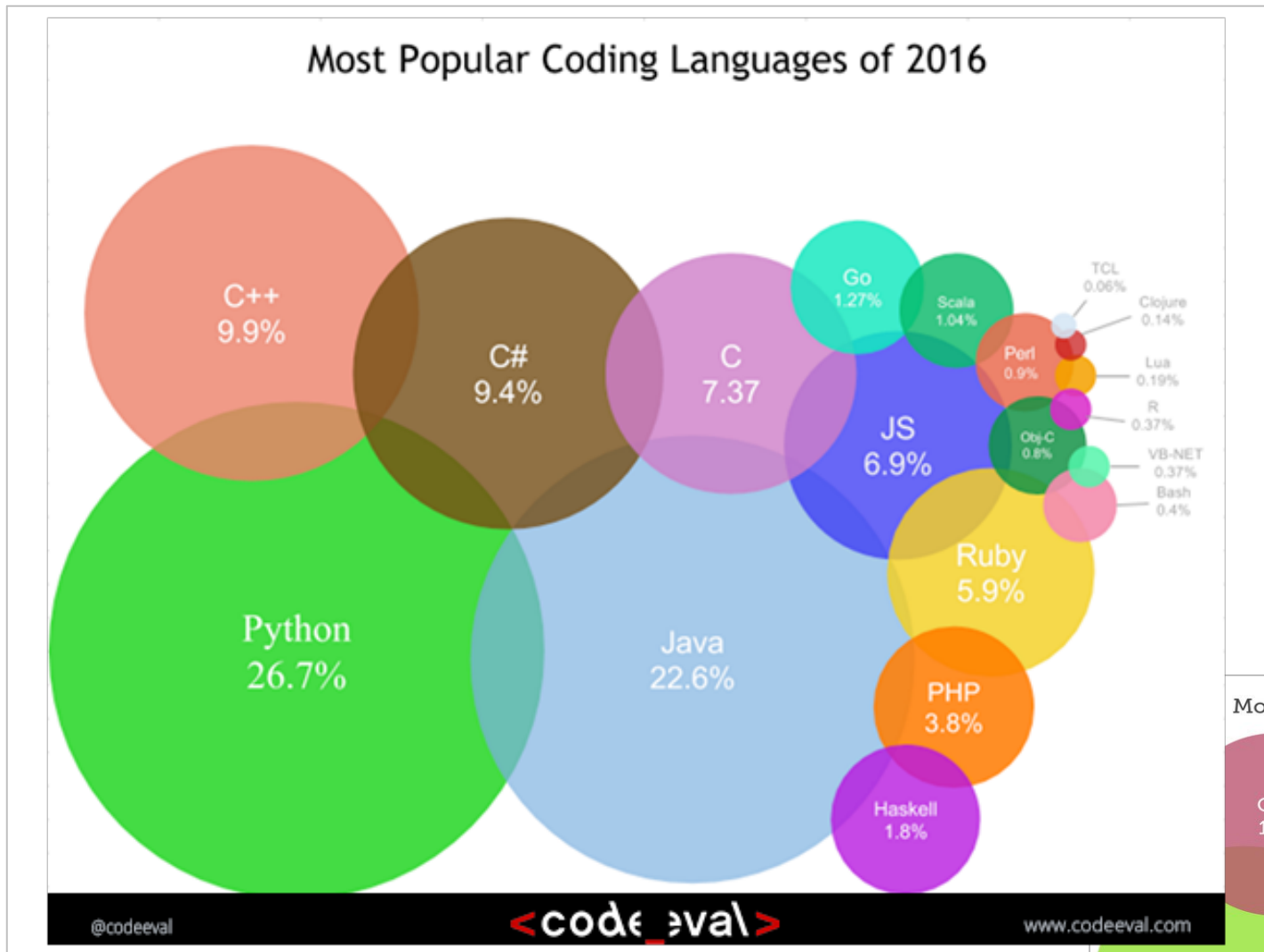
# 왜 파이썬인가 ?

다양한 라이브러리, 넓은 활용범위



<http://blog.naver.com/hbi2hyunjung/220611246872>

# 가장 인기있고 대중화된 Python



# Start Python !

# Python 설치

<https://www.python.org> 에서 다운로드 설치 설치  
할 때 Path 자동으로 등록되도록 체크

The screenshot shows the Python.org website. The header includes the Python logo, a 'Donate' button, a search bar, and a 'Socialize' button. The navigation menu has links for 'About', 'Downloads', 'Documentation', 'Community', 'Success Stories', 'News', and 'Events'. The 'Downloads' section is active, showing a list of links: 'All releases', 'Source code', 'Windows', 'macOS', 'Other Platforms', 'License', and 'Alternative Implementations'. The 'macOS' link is highlighted with a red wavy line. The 'Download for macOS' section displays 'Python 3.9.7' and provides instructions for installation, including a link to 'View the full list of downloads.' At the bottom, a message states: 'Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)'.



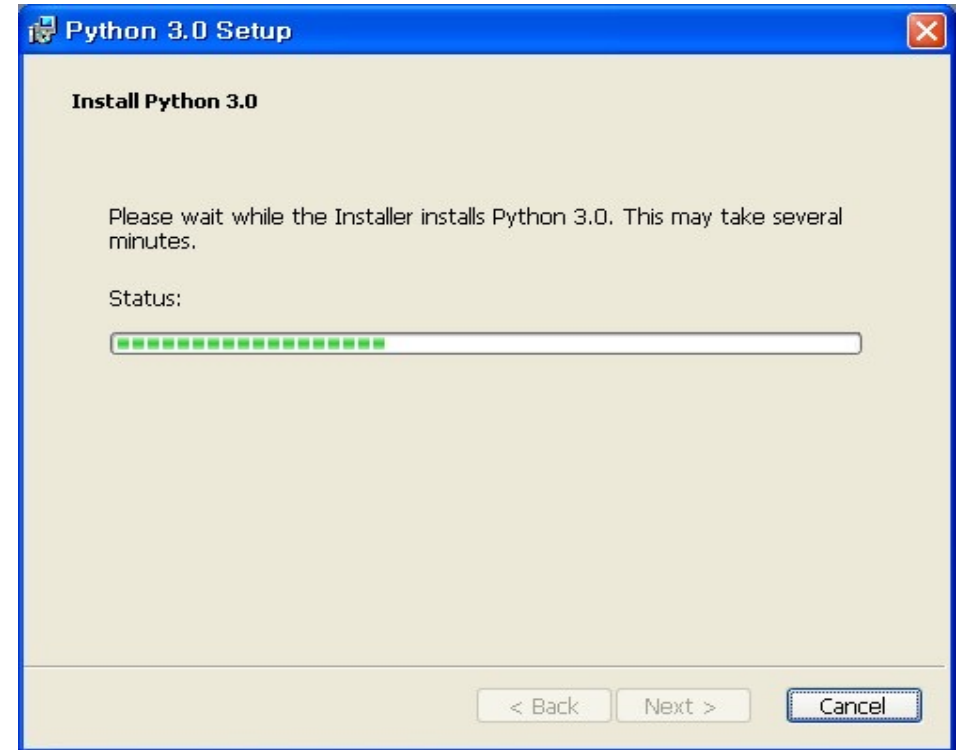
# 설치 및 개발환경

---

- <http://python.org/download>

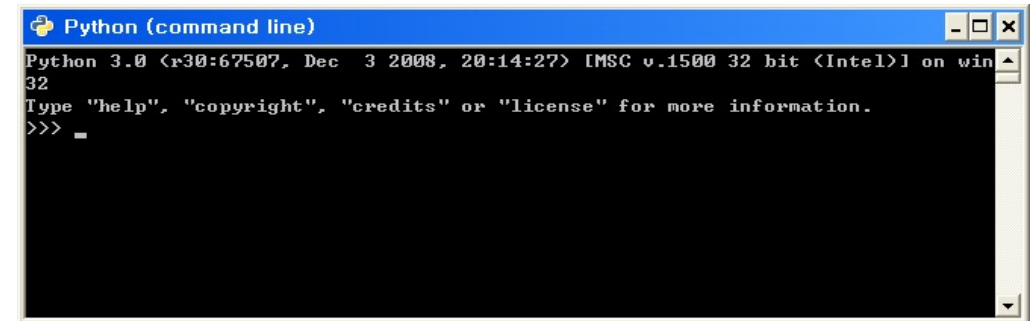


# 설치 및 개발환경

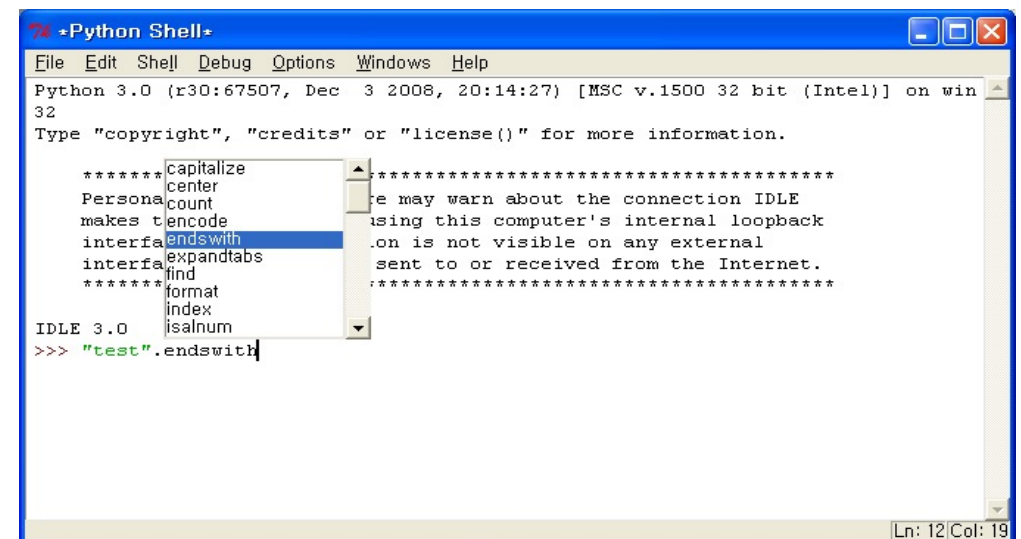


# 설치 및 개발환경

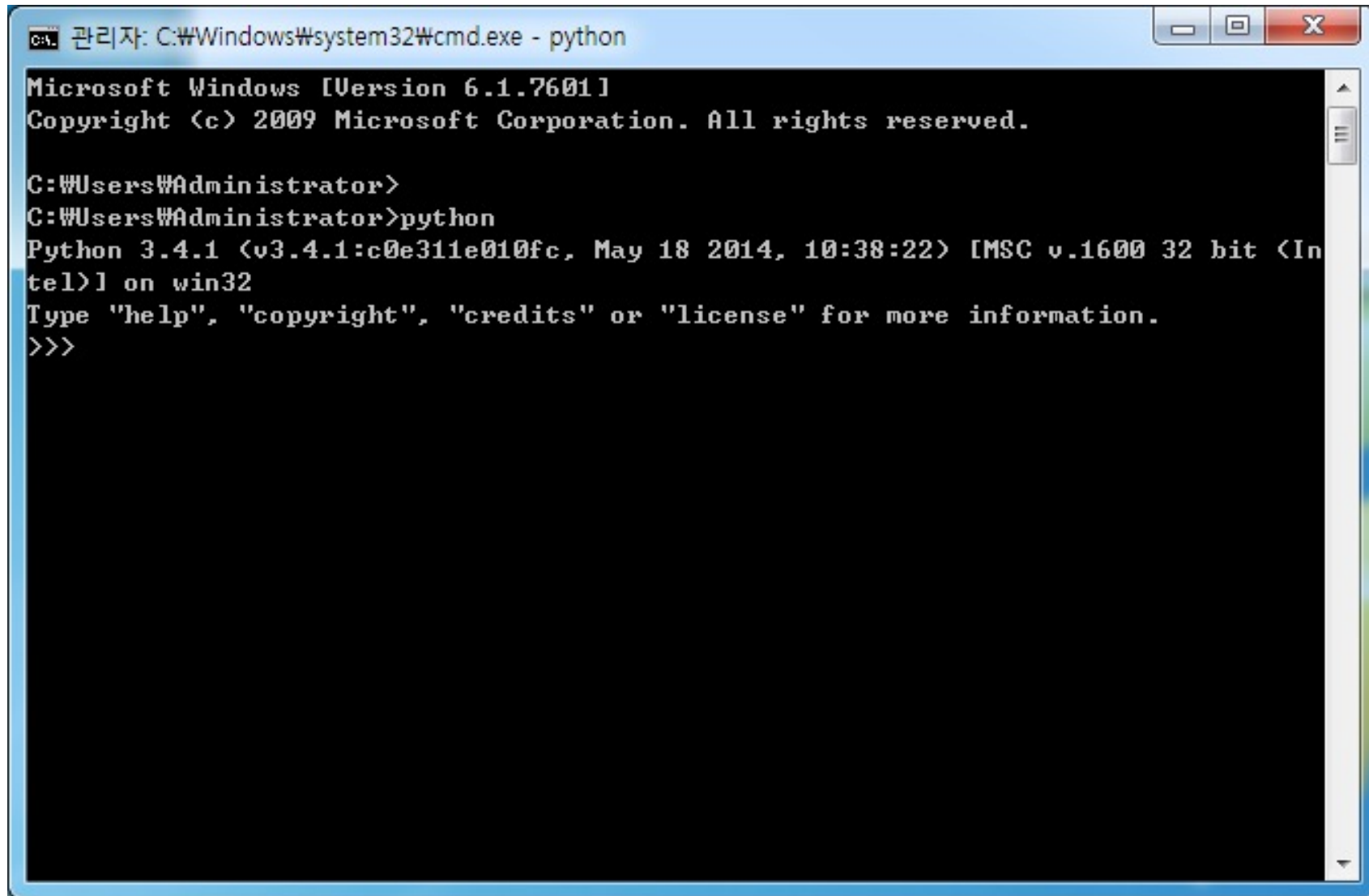
- Python command line



## IDLE



# CLI 환경에서 "python" 파이썬 실행



```
C:\> 관리자: C:\Windows\system32\cmd.exe - python

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>
C:\Users\Administrator>python
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

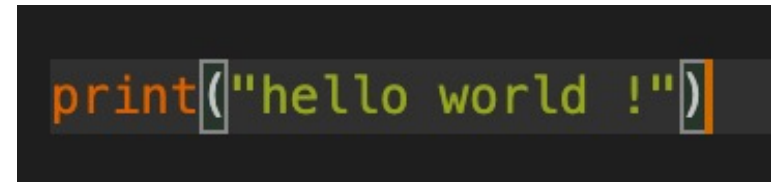
# Visual Studio Code : VSC 시작

VSC 란 ?

- 소스코드편집기

```
Type "help", "copyright", "credits" or "help()" to see  
>>> print("hello World")  
hello World
```

- Terminal

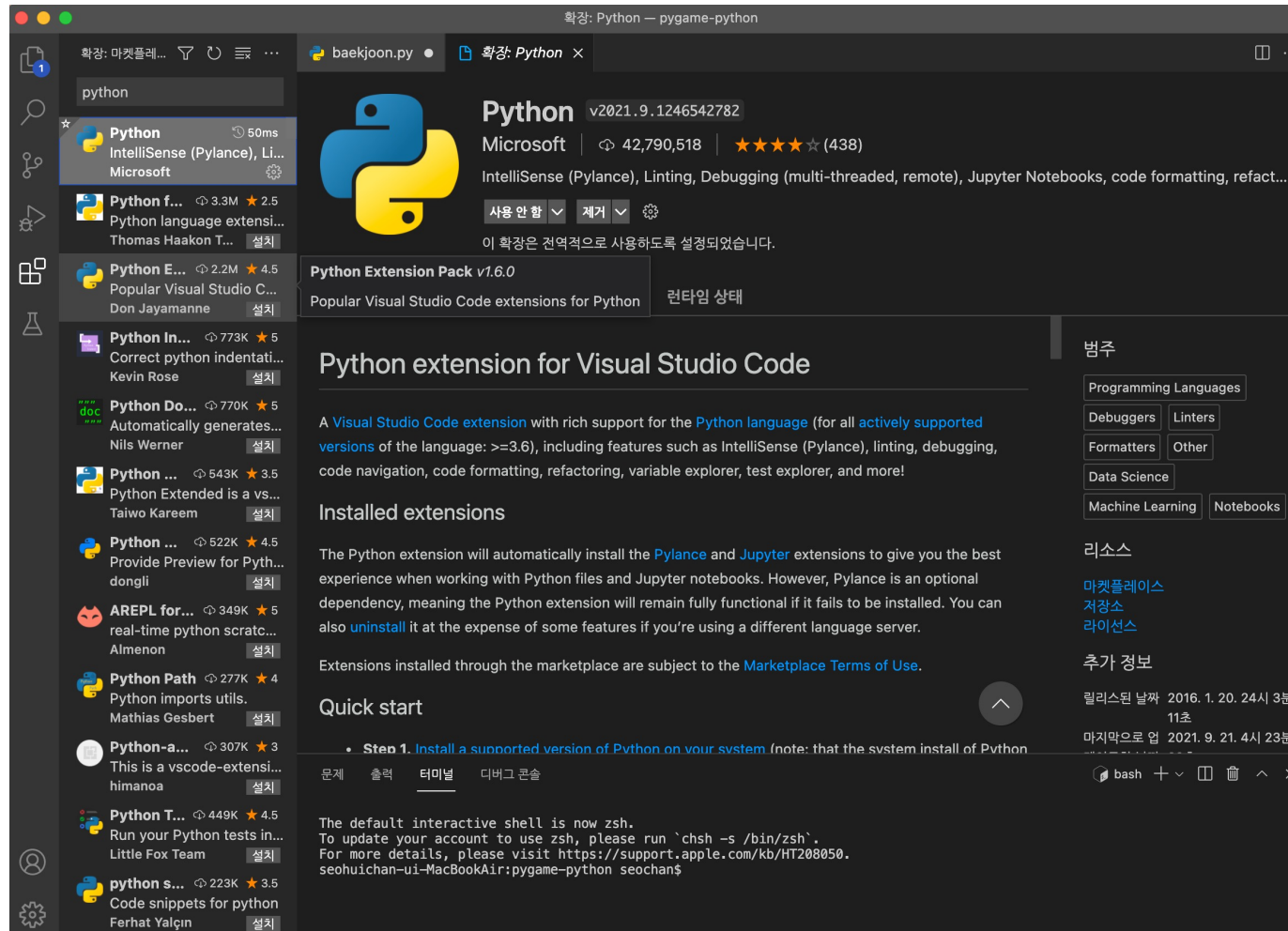
A screenshot of a code editor with a dark background. The text 'print("hello world !")' is written in a monospaced font. The word 'print' is orange, the opening and closing parentheses and the string content are green, and the closing parenthesis is followed by a vertical orange bar.

- VSC

설치 : <https://code.visualstudio.com/>

# Visual Studio Code : VSC Python 설치

## - Python 설치



# 코딩컨벤션

# 코딩 컨벤션

---

프로그래밍은 공동작업이 많기 때문에 프로그램을 작성하는 가이드라인 언어마다 정해져 있음

자신만의 방식으로만 프로그램을 작성할 경우, 다른 사용자의 이해가 떨어짐

컴퓨터가 이해할 수 있는 코드는 어느 바보나 다 짤 수 있다. 좋은 프로그래머는 사람이 이해할 수 있는 코드를 짤다.

- 마틴 파울러

이를 위해 기본적인 코딩 작성법을 코딩 컨벤션이라 함



# Python 코딩 컨벤션

---

들여쓰기는 공백 4 칸을 권장

한 줄은 최대 79자까지

클래스 내의 메소드 정의는 1 줄씩 띄어쓰기

불필요한 공백은 피함 ( ([ ]) 와 ( ( ) ) 안, ( , ), ( : ) 과 ( : ) 앞 = 연산자는 붙여쓰

주석은 항상 갱신, 불필요한 주석은 삭제

소문자 l, 대문자 O, 대문자 I 는 변수명으로 사용하지 말 것 ( 가독성 문제 )

함수명은 소문자로 구성, 필요하면 밑줄로 나눔

# 데이터 타입

# 데이터 타입

## 기본데이터 타입

정수형은 소숫점을 갖지 않는 정수를 갖는 데이터 타입이며, float는 소숫점을 갖는 데이터 타입이다. bool 타입은 True 혹은 False 만을 갖는 타입이고, None은 아무 데이터를 갖지 않는다는 것을 표현하는 것으로 타 언어의 Null과 같은 개념이다. 정수형에 리터럴 데이터를 넣을 때, 10진수 이외에 16진수 혹은 8진수를 위의 예와 같이 사용할 수 있다.

타입	설명	표현 예
int	정수형 데이터	100, 0xFF (16진수), 0o56 (8진수)
float	소숫점을 포함한 실수	a = 10.25
bool	참, 거짓을 표현하는 부울린	a = True
None	Null과 같은 표현	a = None

## 정수형

정수형(Integer)이란 말 그대로 정수를 뜻하는 자료형을 말한다. 다음 예는 양의 정수와 음의 정수, 숫자 0을 변수 a에 대입하는 예이다.

```
>>> a = 123
>>> a = -178
>>> a = 0
```

# 데이터 타입

## 실수형

파이썬에서 실수형(Floating-point)은 소수점이 포함된 숫자를 말한다. 다음 예는 실수를 변수 a에 대입하는 예이다.

```
>>> a = 1.2
>>> a = -3.45
```

위의 방식은 우리가 일반적으로 볼 수 있는 실수형의 소수점 표현 방식이다.

```
>>> a = 4.24E10
>>> a = 4.24e-10
```

위의 방식은 "컴퓨터식 지수 표현 방식"으로 파이썬에서는 4.24e10 또는 4.24E10처럼 표현한다(e와 E 둘 중 어느 것을 사용해도 무방하다). 여기서 4.24E10은  $4.24 * 10^{10}$ , 4.24e-10은  $4.24 * 10^{-10}$ 을 의미한다.

## 8진수와 16진수

8진수(Octal)를 만들기 위해서는 숫자가 0o 또는 0O(숫자 0 + 알파벳 소문자 o 또는 대문자 O)로 시작하면 된다.

```
>>> a = 0o177
```

16진수(Hexadecimal)를 만들기 위해서는 0x로 시작하면 된다.

```
>>> a = 0x8ff
>>> b = 0xABC
```

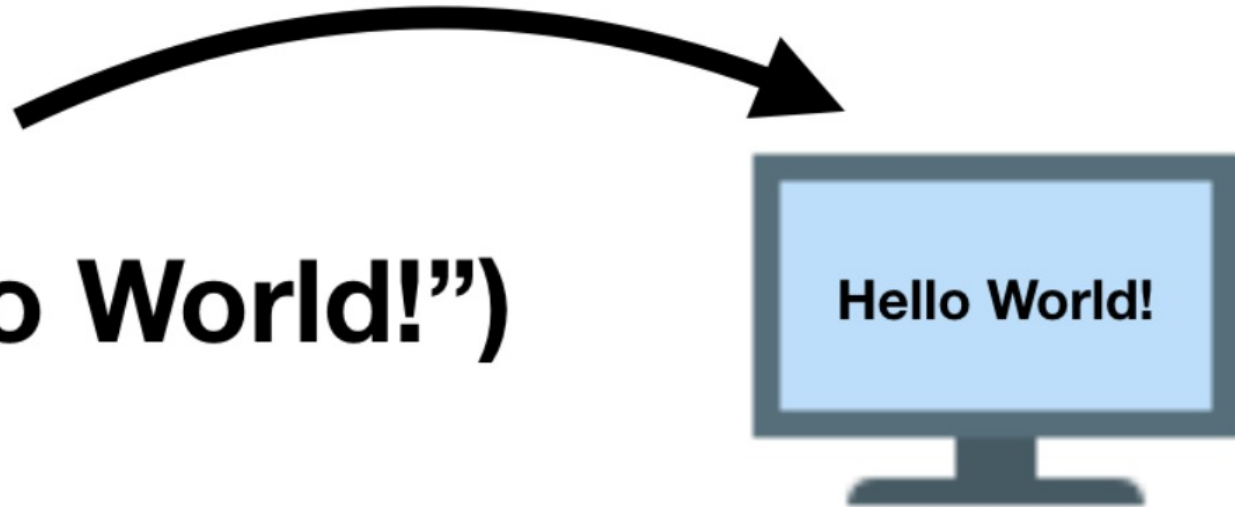
8진수나 16진수는 파이썬에서 잘 사용하지 않는 형태의 숫자 자료형이니 간단히 눈으로 익히고 넘어가자.

# Hello World !

# Print 문

'print'는 번역하면 '출력하다'라는 뜻이죠. 파이썬에서 print문을 사용하여 모니터에 우리가 원하는 문자들을 출력할 수 있습니다. 우리가 파이썬 파일에 print문을 쓰면 컴퓨터는 출력하라는 명령을 받아서 우리가 보는 모니터 화면에 print 문 안의 내용을 보여주게 됩니다.

```
print("Hello World!")
```



print의 사용

# Print 문

1207

1208 | `print("hello world !")`

```
def print(  
    *values: object, sep: str | None = ..., end: str | None = ..., file:  
    SupportsWrite[str] | None = ..., flush: bool = ...  
    ) -> None: ...  
  
_E = TypeVar("_E", contravariant=True)  
  
(function) print: (*values: object, sep: str | None = ..., end: str | None = ...,  
file: SupportsWrite[str] | None = ..., flush: bool = ...) -> None
```

```
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

Prints the values to a stream, or to sys.stdout by default. Optional keyword arguments:

file: a file-like object (stream); defaults to the current sys.stdout.

sep: string inserted between values, default a space.

## 실행하기

**연산자**



# 연산자

파이썬은 산술연산자, 비교연산자, 할당연산자, 논리연산자, Bitwise 연산자, 멤버십연산자, Identity연산자를 지원한다.

## 산술연산자

산술연산자에는 (1) 사칙연산자 `+`, `-`, `*`, `/` 와 (2) 제곱을 나타내는 `**`, (3) 나머지를 산출하는 `%` (Modulus), 그리고 (4) 나누기에 소숫점 이하를 버리는 `//` 연산자(Floor Division) 등이 있다.

```
1 | 5 % 2 # 1
2 | 5 // 2 # 2
```

## 비교연산자

비교연산자는 관계연산자로도 불리우는데, 여기에는 등호(`==`), 같지 않음(`!=`), 부등호(`<`, `>`, `<=`, `>=`) 등이 있다.

```
1 | if a != 1:
2 |     print("1이 아님")
```

## 할당연산자

할당연산자는 변수에 값을 할당하기 위하여 사용되는데, 기본적으로 `=` (Equal Sign)을 사용한다. 산술연산자와 함께 사용되어 할당을 보다 간결히 하기 위해 사용되는 `+=`, `-=`, `*=`, `/=`, `%=`, `//=` 등과 같은 연산자도 할당연산자에 해당된다.

```
1 | a = a * 10
2 | a *= 10 # 위와 동일한 표현
```

# 연산자

---

## 논리연산자

논리연산자에는 and, or, not 이 있는데, and 는 양쪽의 값이 모두 참인 경우만 참이 되고, or 는 어느 한쪽만 참이면 참이 된다. not 은 참이면 거짓으로 거짓이면 참이 된다. 아래 예제는 No가 출력된다.

```
1 x = True
2 y = False
3
4 if x and y:
5     print("Yes")
6 else:
7     print("No")
```

## Bitwise연산자

Bitwise연산자에는 & (AND), | (OR), ^ (XOR), ~ (Complement), <<, >> (Shift)가 있는데, 이 연산자는 비트단위의 연산을 하는데 사용된다.

```
1 a = 8      # 0000 1000
2 b = 11     # 0000 1011
3 c = a & b   # 0000 1000 (8)
4 d = a ^ b   # 0000 0011 (3)
5
6 print(c)
7 print(d)
```

# 연산자

---

## Identity연산자

멤버쉽연산자에는 in, not in 이 있는데, 이는 좌측 Operand가 우측 컬렉션에 속해 있는지 아닌지를 체크한다.

```
1 | a = [1,2,3,4]
2 | b = 3 in a    # True
3 | print(b)
```

## 멤버쉽연산자

Identity연산자에는 is, is not 이 있는데, 이는 양쪽 Operand가 동일한 Object를 가리키는지 아닌지를 체크한다.

```
1 | a = "ABC"
2 | b = a
3 | print(a is b) # True
```

**이스케이프코드**

# Print문 응용

## 이스케이프 코드 알아보기

이스케이프 코드란 출력할 때, 보기 좋게 하는 코드들을 문자 조합을 미리 지정해 둔 것입니다. 예를 들어, 줄바꿈, 탭 띄우기, 따옴표 출력 등이 있습니다. 대표적으로 많이 사용되는 이스케이프 코드에 대해 알아보겠습니다.

# Print문 응용

줄바꿈 문자 - \n

example

```
1 print("enter\nokay enter!\nyes enter!")
2 # enter okay
3 # enter!
4 # yes enter!
```

탭 띄우기 문자 - \t

example

```
1 print("\tthis\t is tab")
2 #      this      is tab
```

# Print문 응용

역슬래시 문자 - \

example

```
1 print("\\\\\\\\\\\\\\ this is back slash")
2 # \\\\\ this is back slash
```

따옴표 문자 - \'

example

```
1 print("\\'\\'\\'\\' single quote")
2 # ''' single quote
```

쌍따옴표 문자 - \"

example

```
1 print("\\\"\\\"\\\"\\\" double quote")
2 # \"\"\"\" double quote
```

# Print문 응용

홀따옴표 안의 쌍따옴표

example

```
1 print(""" double quote")  
2 # error
```



example

```
1 print('' single quote')  
2 # '' single quote
```





# Print문 응용

쌍따옴표 안의 홑따옴표

example



```
1 print('''' single quote')  
2 # error
```

example



```
1 print('""" double quote')  
2 # """ double quote
```

# Print문 응용

세 개의 따옴표

example

```
1 print('''
2 I love you
3     so so so much
4 but miss you~ bye~!
5 ''')
6 # I love you
7 #     so so so much
8 # but miss you~ bye~!
```

