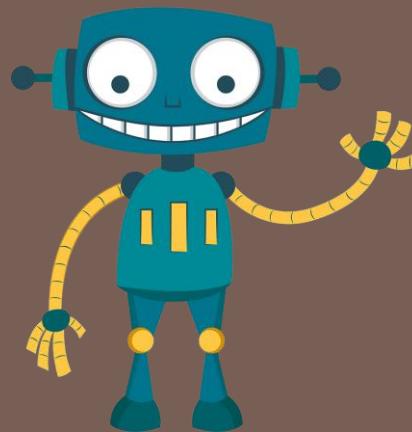
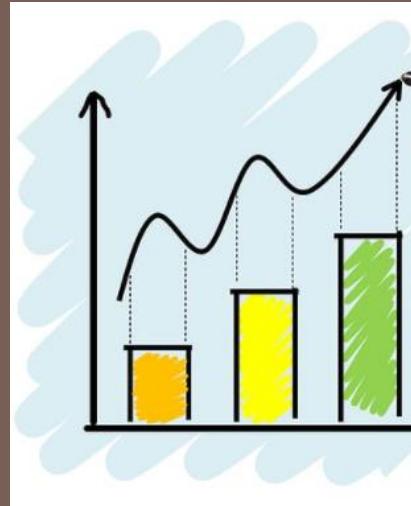


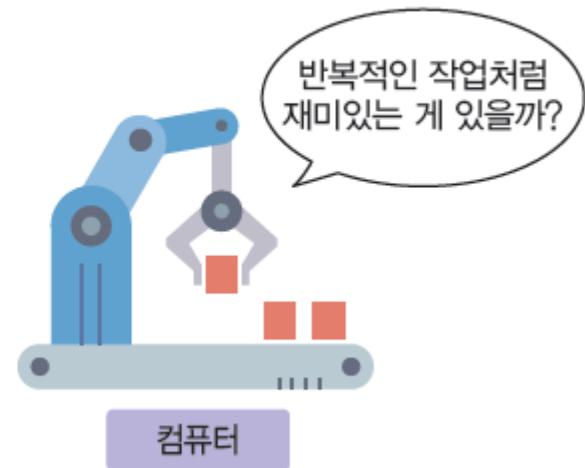
# 파이썬 이스프레스



1장 프로그래밍과 파이썬 소개

# 컴퓨터와 프로그램

## □ 컴퓨터와 현대 사회



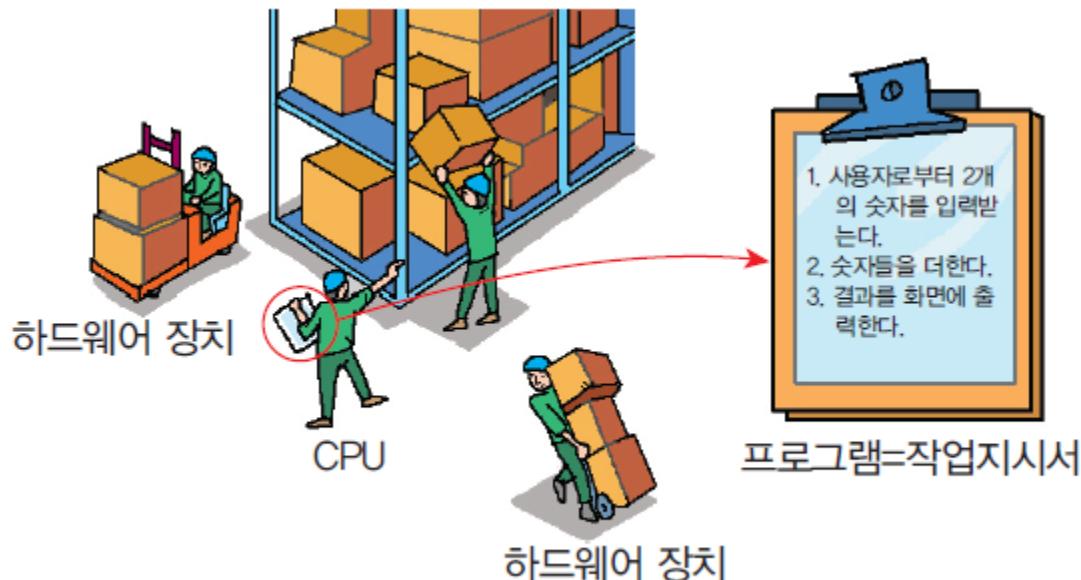
# 컴퓨터가 많이 사용되는 이유

- 컴퓨터는 대단히 유연한 기계
- 컴퓨터로 리포트를 작성할 수도 있지만, 게임도 할 수 있다



# 컴퓨터 프로그램

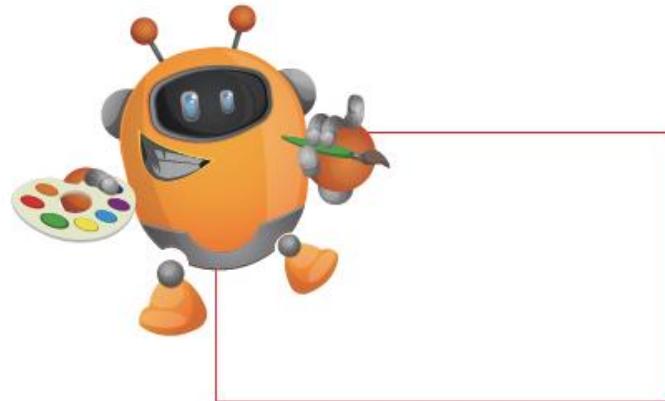
- 컴퓨터에 일을 시키려면 인간이 컴퓨터에게 자세한 명령어 (instruction)들의 리스트를 주어야 한다.
- 프로그램 (program) : 컴퓨터가 수행할 명령어를 적어놓은 문서



# 명령어들의 예

- 붓을 들고 있는 로봇에게 사각형을 그리게 하는 작업은 다음과 같은 지시사항들로 이루어질 수 있다.

- ▶ 100 픽셀만큼 앞으로 이동한다.
- ▶ 90도 회전한다.
- ▶ 100 픽셀만큼 앞으로 이동한다.
- ▶ 90도 회전한다.
- ▶ 100 픽셀만큼 앞으로 이동한다.
- ▶ 90도 회전한다.
- ▶ 100 픽셀만큼 앞으로 이동한다.



# 중간 점검

1. 컴퓨터의 장점은 무엇인가?
2. 왜 계산기는 컴퓨터라고 할 수 없는가?
3. 프로그램 안에는 무엇이 들어 있는가?



# 우리는 왜 프로그래밍에 대하여 알아야 할까?

- 우리가 어떤 일을 하던지 상관없이 프로그래밍은 필수적인 기술이 되었다. 이과 학생들에게는 물론, 인문사회계 학생들도 프로그래밍에 대하여 어느 정도는 알아야 한다

저는 인문계인데도  
프로그래밍에 대하여  
알아야 하나요?



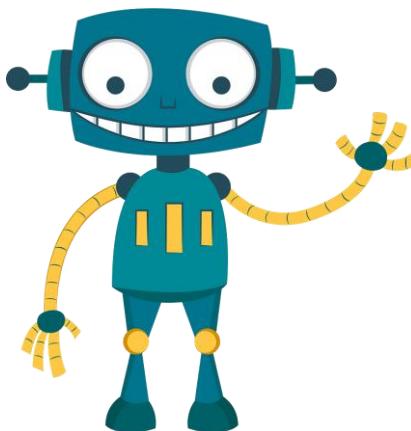
인문계 출신들이 주로 하는 각종  
업무도 파이썬을 이용하여  
자동화하는 경우가 많습니다.

앞으로 단순 반복 작업은  
컴퓨터가 맡게 될 것입니다.



# 우리는 왜 프로그래밍에 대하여 알아야 할까?

- 보고서 쓰는 인공지능 로봇
- 인공지능 알고리즘 주식 매매
- 로봇 프로세스 자동화(RPA)



# 인간 vs 컴퓨터

- 사람들이 아주 재미없고 지루하다고 생각하는 작업을 컴퓨터는 아주 잘한다.



# 중간 점검

1. 컴퓨터의 장점과 인간의 장점은 무엇인가?
2. 왜 우리는 프로그래밍을 배워야 하는가?



# 프로그래밍 언어

- 컴퓨터는 사람의 언어를 이해할 수 없다!
- '프로그래밍 언어'는 컴퓨터가 이해하는 언어이다.



# 컴파일러(인터프리터)

- 인간이 프로그래밍 언어를 배워서 프로그램을 작성하면 컴파일러(또는 인터프리터)라고 하는 통역 소프트웨어가 프로그램을 기계어로 바꾸어준다.



```
if x > 0:  
    pos = pos + 1  
else :  
    neg = neg + 1
```

프로그래밍 언어



```
0100011010  
1010100101  
0101010101  
0010111111
```

기계어



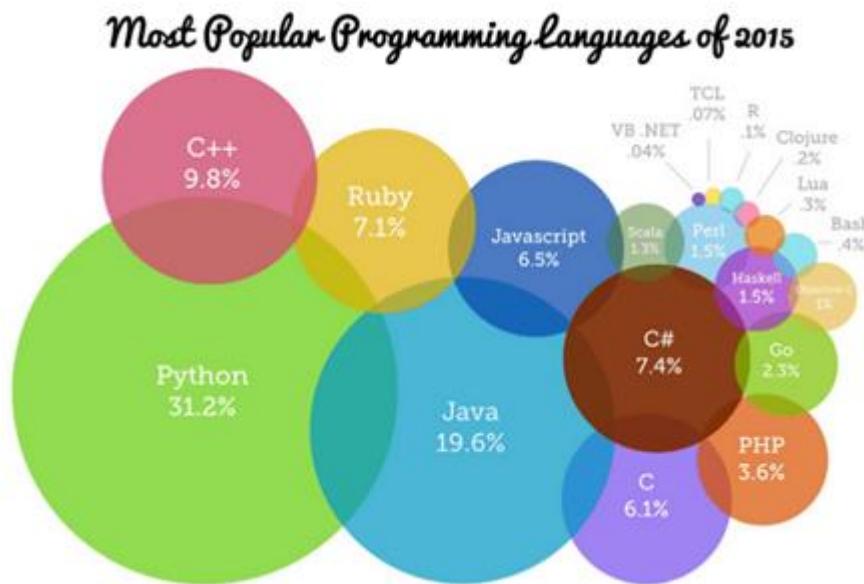
# 프로그래밍 언어

- 프로그래밍은 ‘프로그래밍 언어’로 작성된다. 프로그램을 만드는 사람을 ‘프로그래머’라고 한다.



# 프로그래밍 언어의 종류

- 많이 사용되는 언어들에는 '파이썬', '자바', 'C', 'BASIC' 등이 있다.



# 파이썬

- 1991년에 귀도 반 로섬(Guido van Rossum)이 개발한 대화형 프로그래밍 언어



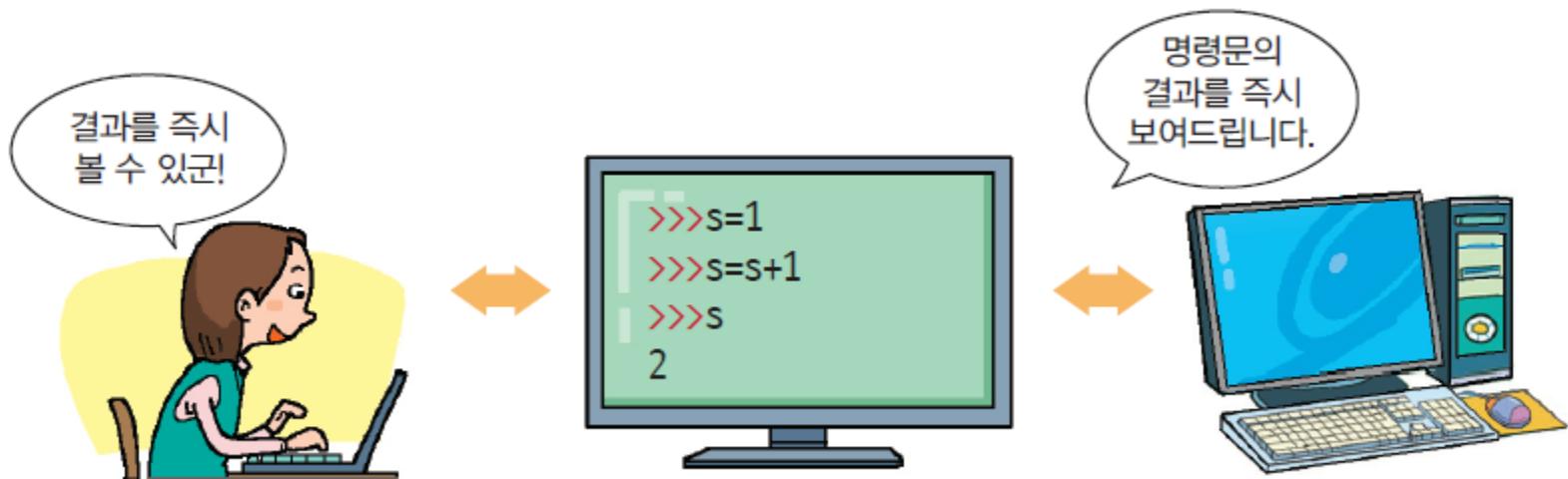
# 파이썬의 특징

- 생산성이 뛰어나다.
- 간결하면서도 효율적인 프로그램을 빠르게 작성

C 언어	파이썬
<pre>#include &lt;stdio.h&gt;  int main(void) {     printf("Hello World! \n");     return 0; }</pre>	<pre>print("Hello World!")</pre>

# 파이썬의 특징

- 인터프리터 언어: 파이썬 프로그래머는 자신이 작성한 명령문의 결과를 즉시 볼 수 있기 때문에 초보 프로그래머한테는 아주 바람직



# 파이썬의 특징

- 라이브러리가 풍부
- 라이브러리 설치가 쉽다.

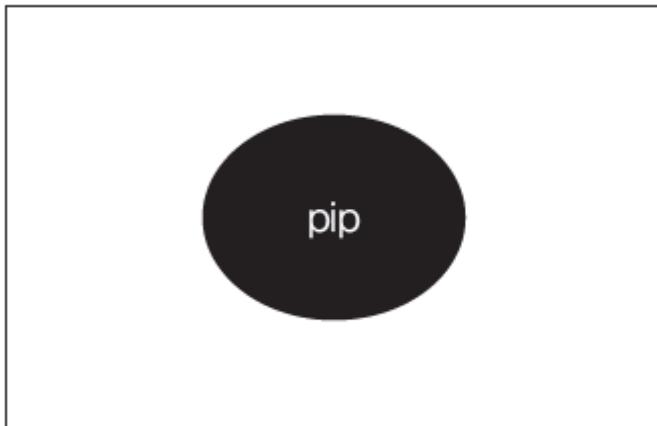


파이썬의 막강한  
라이브러리

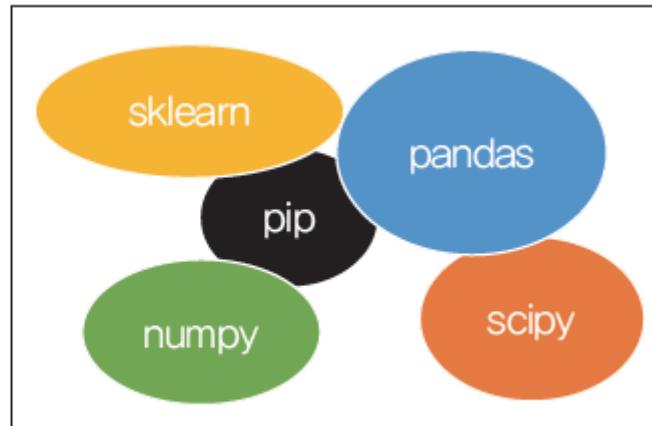


# 아나콘다

- 인기 있는 라이브러리가 거의 모두 포함된 배포판

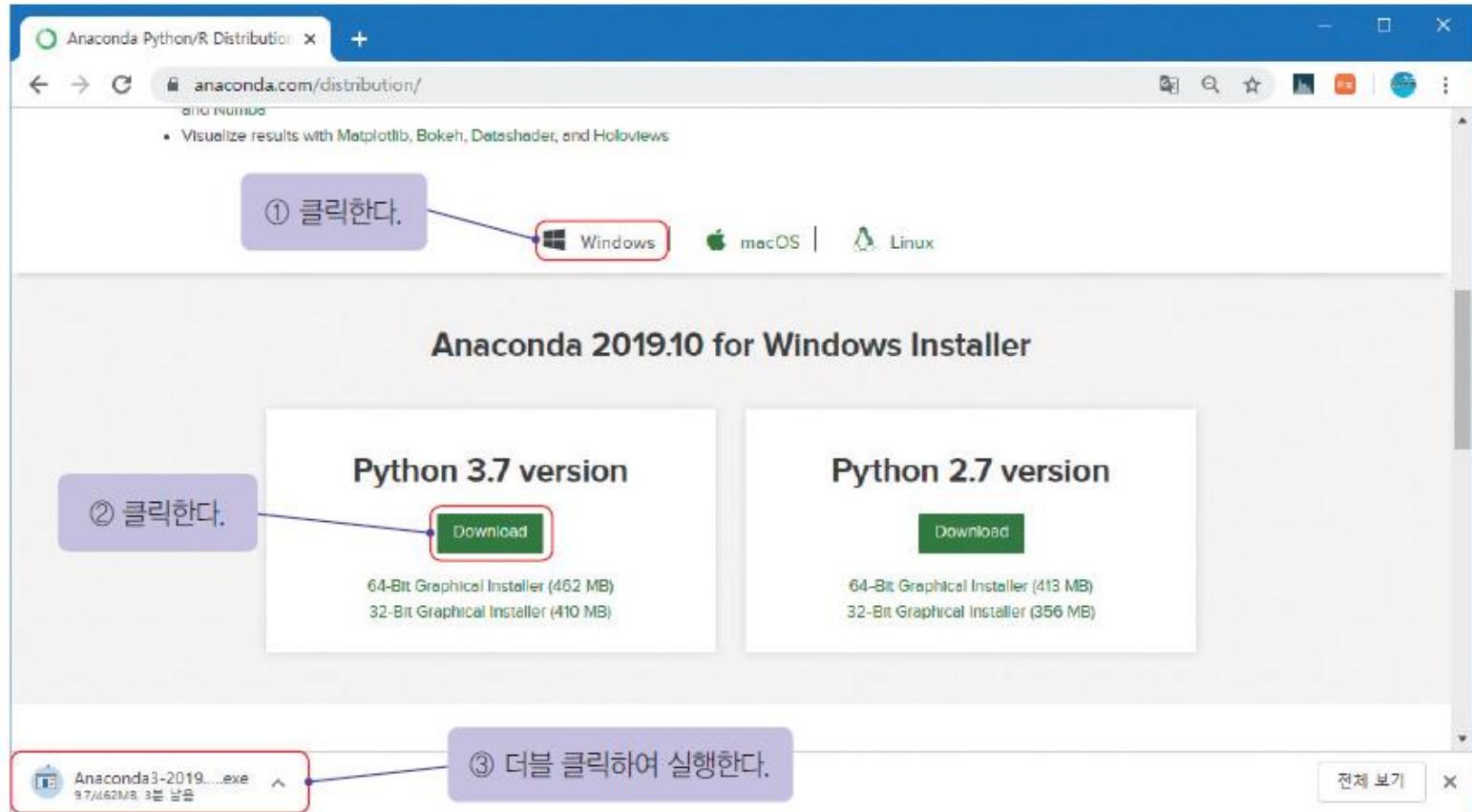


파이썬

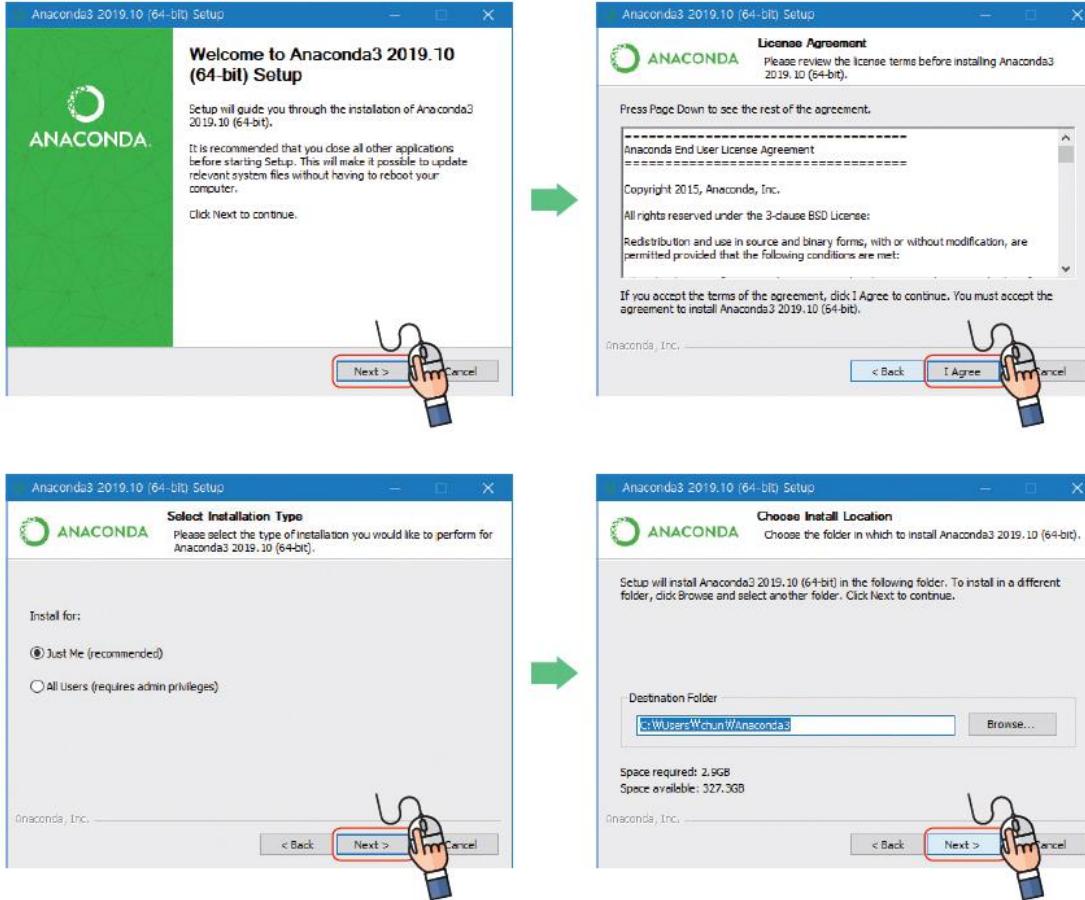


아나콘다

# 아나콘다 다운로드

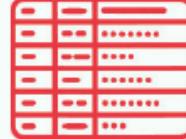


# 아나콘다 설치

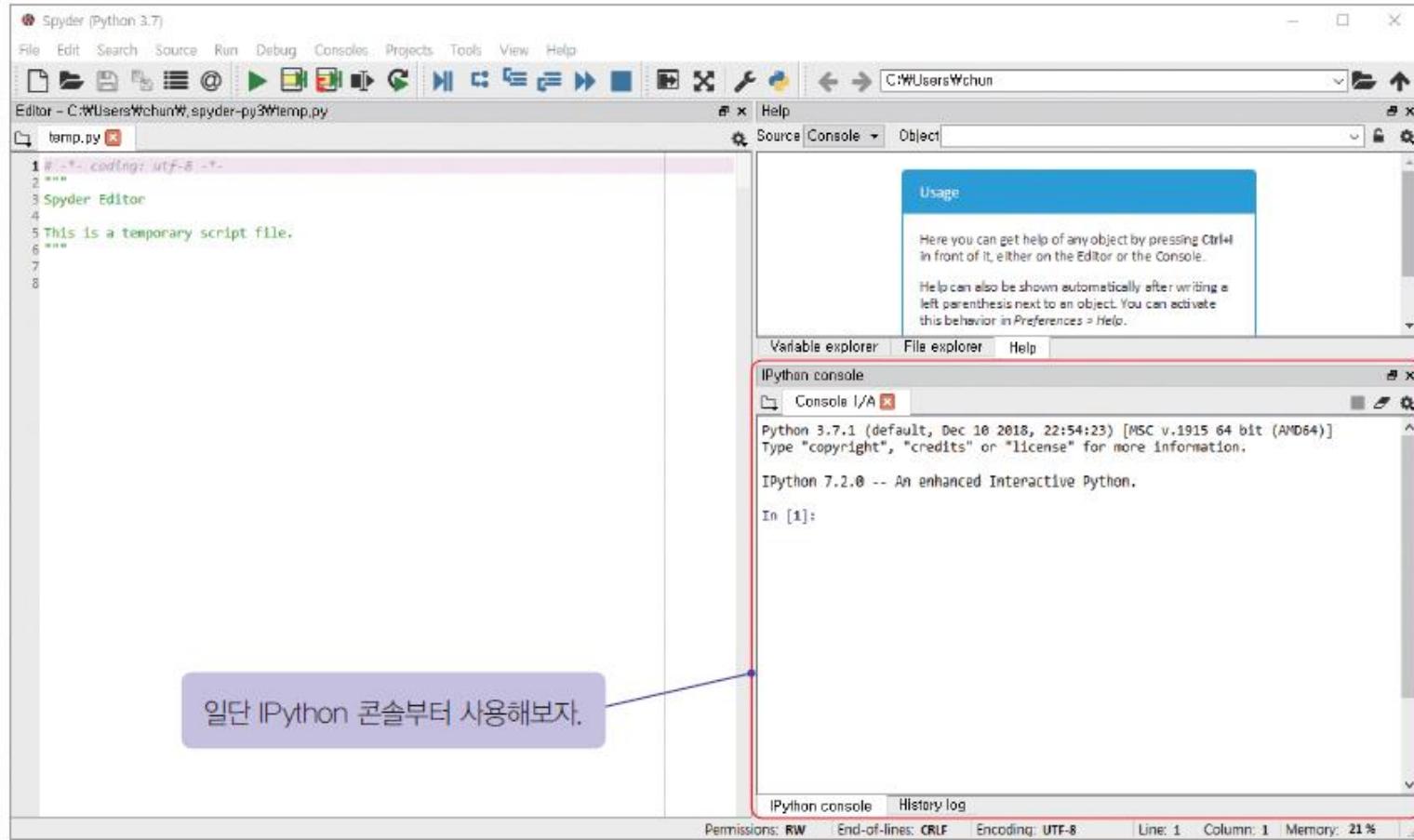


# 스파이더

## ▣ 스파이더는 파이썬으로 작성된 파이썬 개발 도구

 Editor	 IPython Console	 Variable Explorer
함수 / 클래스 브라우저, 코드 분석 도구, 자동 코드 완성, 수평 / 수직 분할을 사용하여 효율적으로 작업이 가능하다.	GUI 인터페이스 안에서 코드를 라인별로 실행하거나 인라인으로 차트를 그릴 수 있다.	변수와 상호작용하거나 변수를 변경할 수 있다. 히스토그램을 그리거나 데이터 프레임을 편집하고 컬렉션을 정렬할 수도 있다.
 Profiler	 Debugger	 Help
코드에서 가장 시간을 잡아먹는 부분을 바로 알아내서 제거할 수 있다.	디버거를 사용하여 코드를 한 줄씩 실행 할 수 있다.	클래스에 관한 도움말을 즉시 볼 수 있다.

# Ipython 콘솔



# Ipython 코스

IPython console

Console 1/A

Python 3.7.3 (default, Mar 27 2019, 17:15:21) [MSC v.1913 64 bit (AMD64)]

Type "copyright", "credits" or "license" for more information.

IPython 7.4.0 -- An enhanced Interactive Python.

Restarting kernel...

In [1]: `print("Hello World!")`

Hello World!

In [2]: |

IPython console History log

# 계산하기 #1

- 반지름이 10cm인 피자의 면적을 계산해보자.

```
In [ ]: 3.14 * 10 * 10 Enter ↵
```

```
Out[ ]: 314.0
```

```
>>> 3.14 * 10**2 Enter ↵
```

```
314.0
```

```
>>> 3.14 * 20**2 Enter ↵
```

```
1256.0
```

# 계산하기 #2

- 삼각함수, 로그함수 사용하기

```
>>> import math [Enter ↴]  
>>> math.sin(math.radians(30)) [Enter ↴]  
0.4999999999999994
```

```
>>> math.sqrt(9.0) [Enter ↴]  
3.0
```

```
>>> math.log(1.0) [Enter ↴]  
0.0
```

# 문자열 출력하기

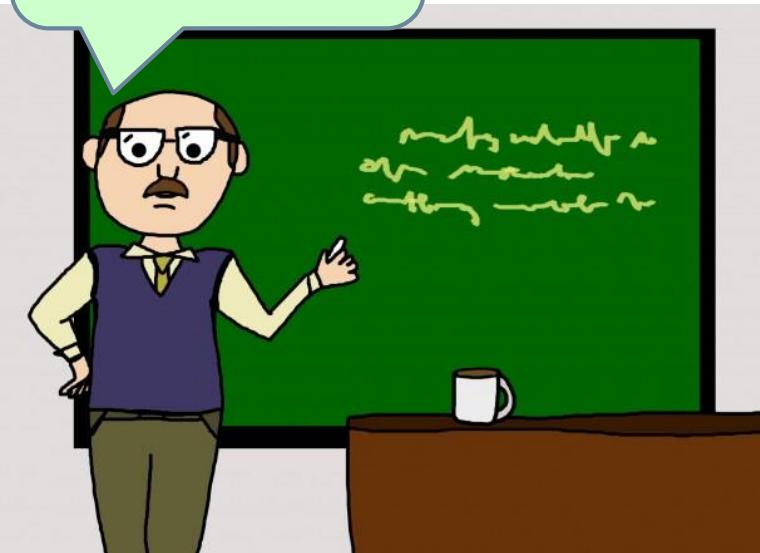
>>> "나의" + "고양이"

나의 고양이

>>>

문자열은  
어떻게  
구별하나요?

따옴표 ("...")가  
붙으면  
문자열입니다.



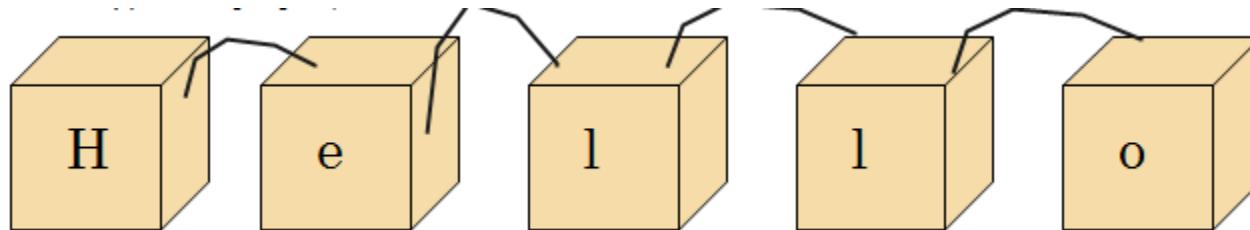
# 문자열 반복하기

```
>>> "Hello" * 10  
'HelloHelloHelloHelloHelloHelloHelloHelloHelloHello'
```



# 문자열

- 문자열(string) : 큰따옴표("...")나 작은따옴표('...') 안에 들어 있는 텍스트 데이터



- 반드시 따옴표가 있어야 한다.

```
>>> print(Hello World!)
      SyntaxError: invalid syntax
```

# 중간 점검

- 한글도 출력될까? 이번에도 따옴표를 올바르게 입력하여야 한다. “안녕하세요?”를 화면에 출력하여 보자.
- “programming에 입문하신 것을 축하드립니다.”를 출력하여 보자.
- “생일축하!!”를 10번 출력하는 명령문을 만들어보자. 문자열 반복을 사용한다.
- 다음과 같은 명령문을 실행하면 오류가 발생한다. 원인을 알아보자.  
`>>> print("Hello")`



# 대화형 모드와 스크립트 모드

- 대화형 모드(interactive mode): 콘솔에서 문장을 한 줄씩 입력하여 실행
- 스크립트 모드(script mode): 파일을 만들어서 저장한 후에 파이썬 인터프리터가 이 파일을 읽어서 한 번에 전부 실행

대화형 모드	스크립트 모드
<pre>&gt;&gt;&gt; print("Hello World!") Hello World! &gt;&gt;&gt; print(10+20) 30</pre>	<pre>print("Hello World!") print(10+20)</pre> <p>▶ 실행결과</p> <pre>Hello World! 30</pre>

문장이 즉시 실행되고 실행결과가 바로 나타난다.

▶ 버튼을 눌러야 문장들이 실행되고 실행결과는 콘솔에 나타난다.

# 스크립트 모드

- 코드가 복잡해지면 인터프리트 모드는 번거롭다.

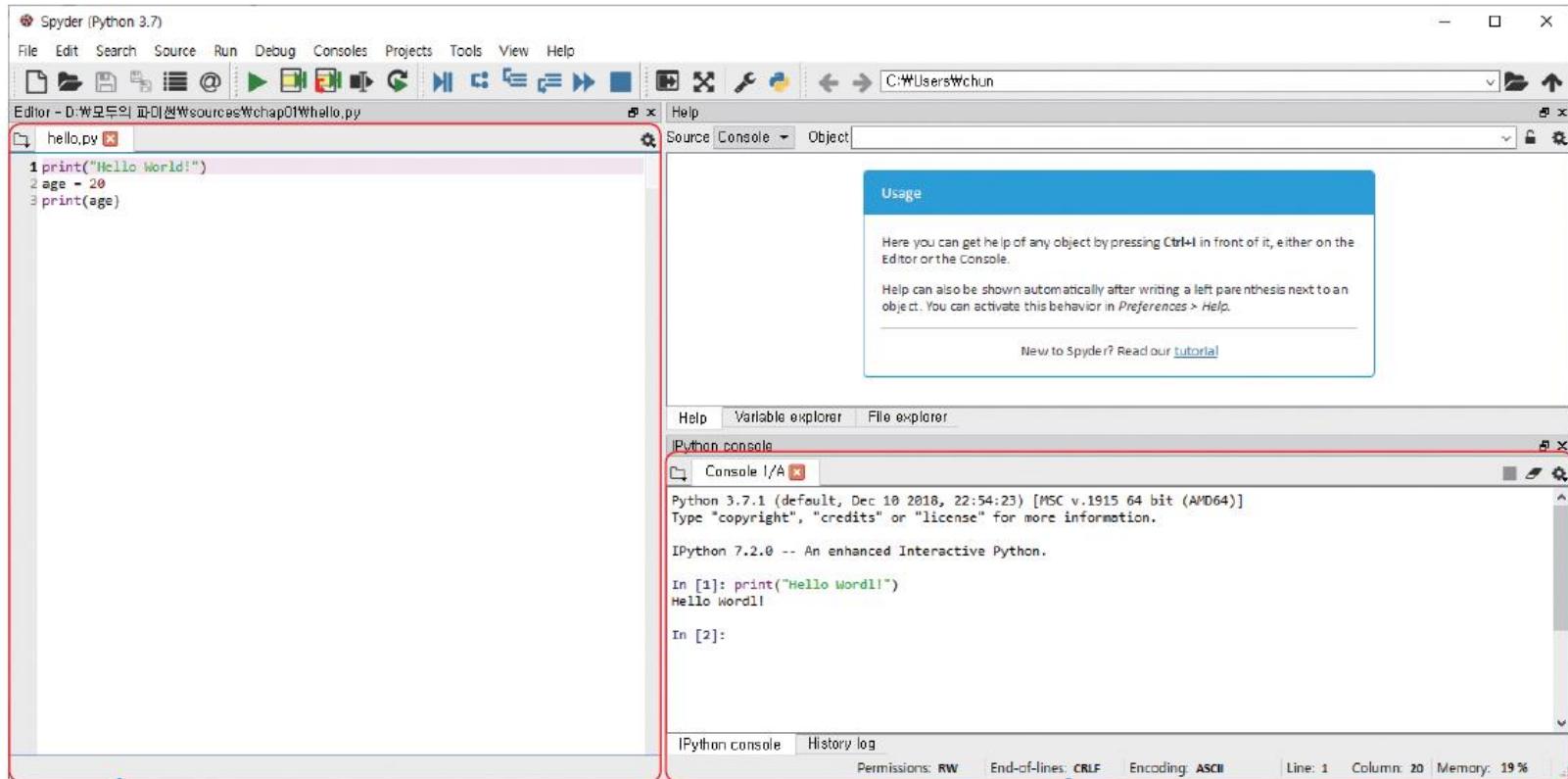
이걸 한 줄씩 입력하라고?



```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodeName()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print '%s [%label=%s]' % (nodename, label)
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '=' % ast[1]
        else:
            print ''
    else:
        print '';
        children = []
        for n, childenumerate(ast[1:]):
            children.append(dotwrite(child))
        print '%s -> (%s)' % (nodename, children)
        for name in children:
            print '%s' % name,
```

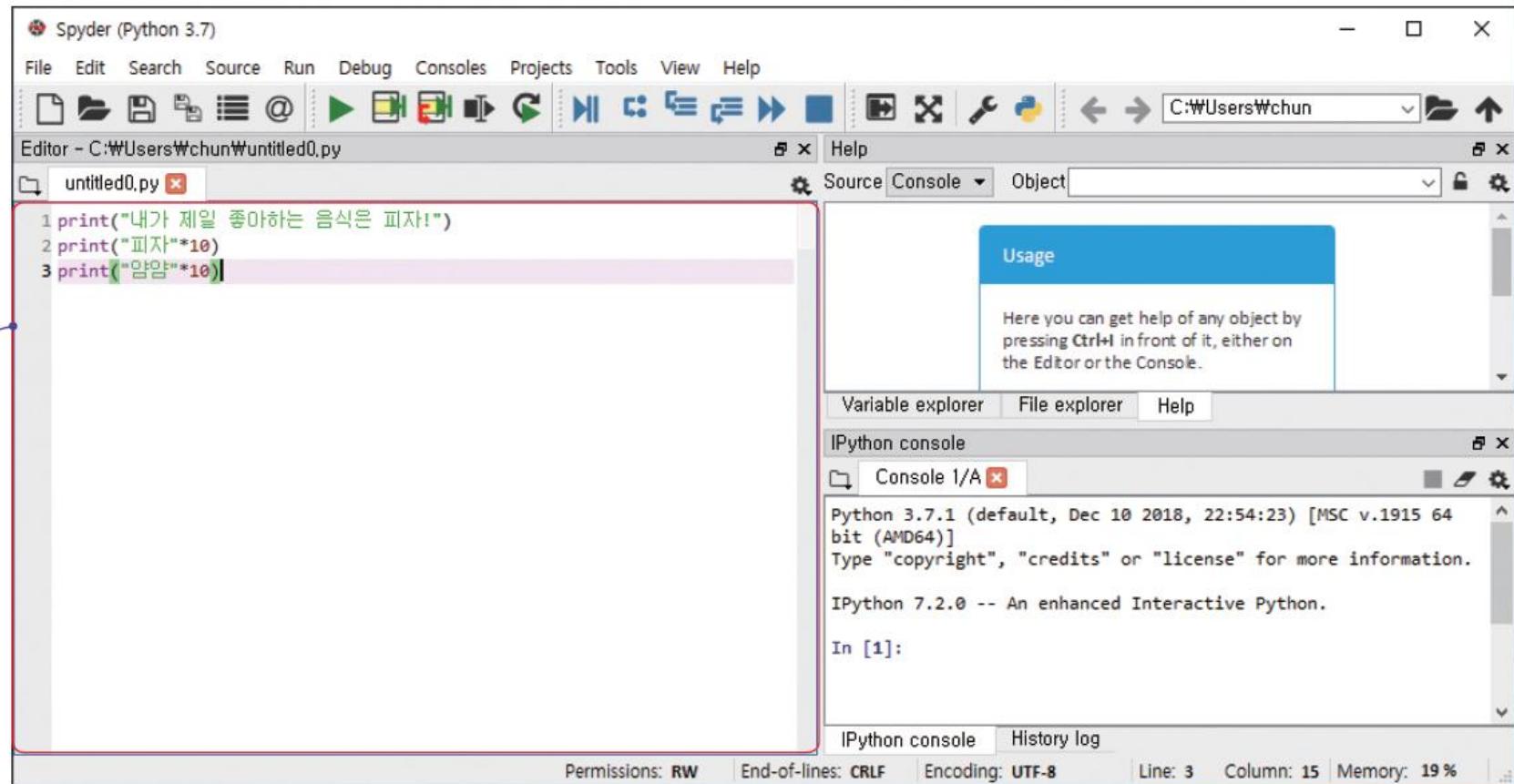
# 스파이더에서 대화형 모드와 스크립트 모드



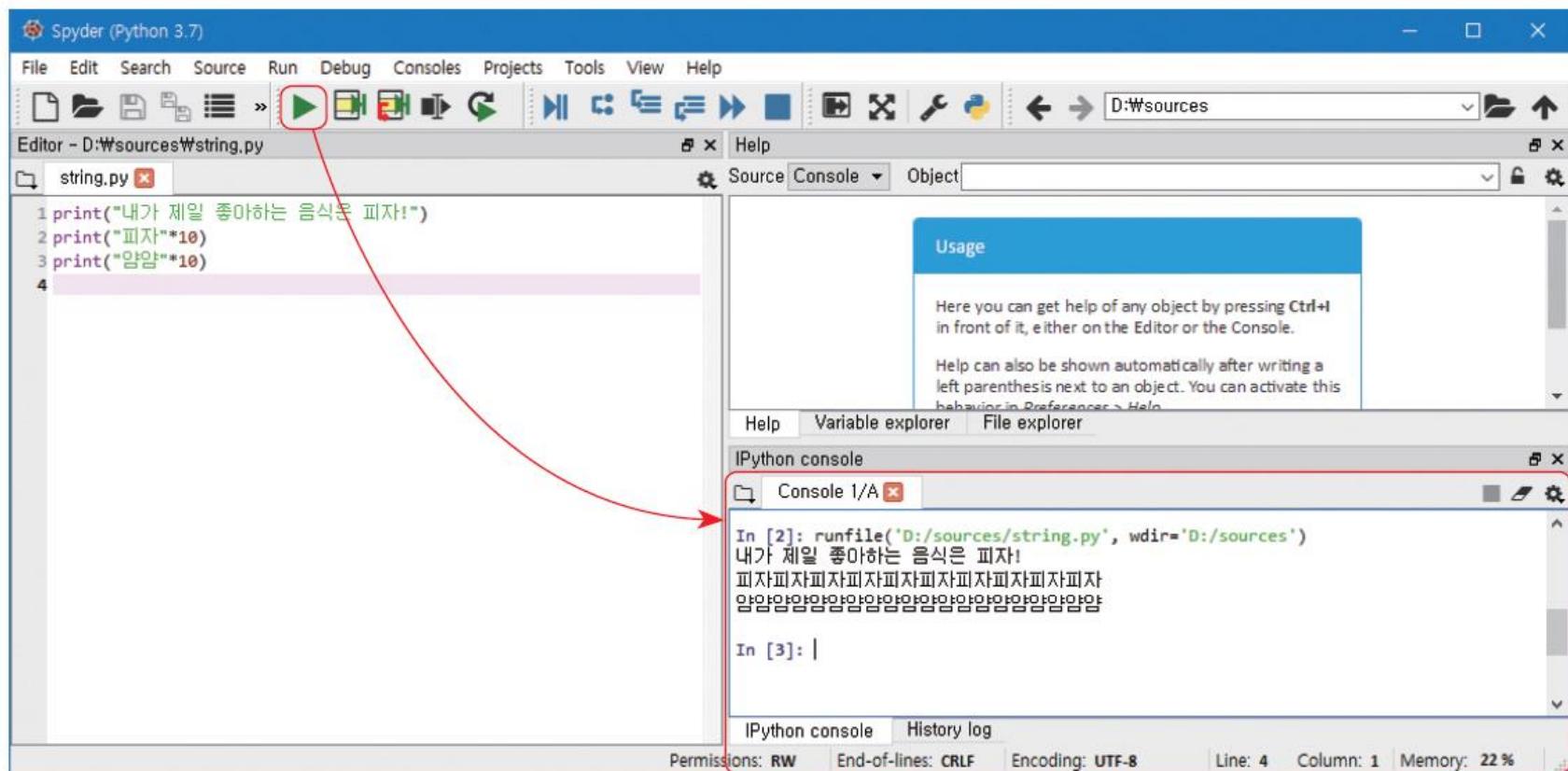
여기다  
입력하면  
스크립트  
모드로  
실행된다.

여기다  
입력하면  
대화형  
모드로  
실행된다.

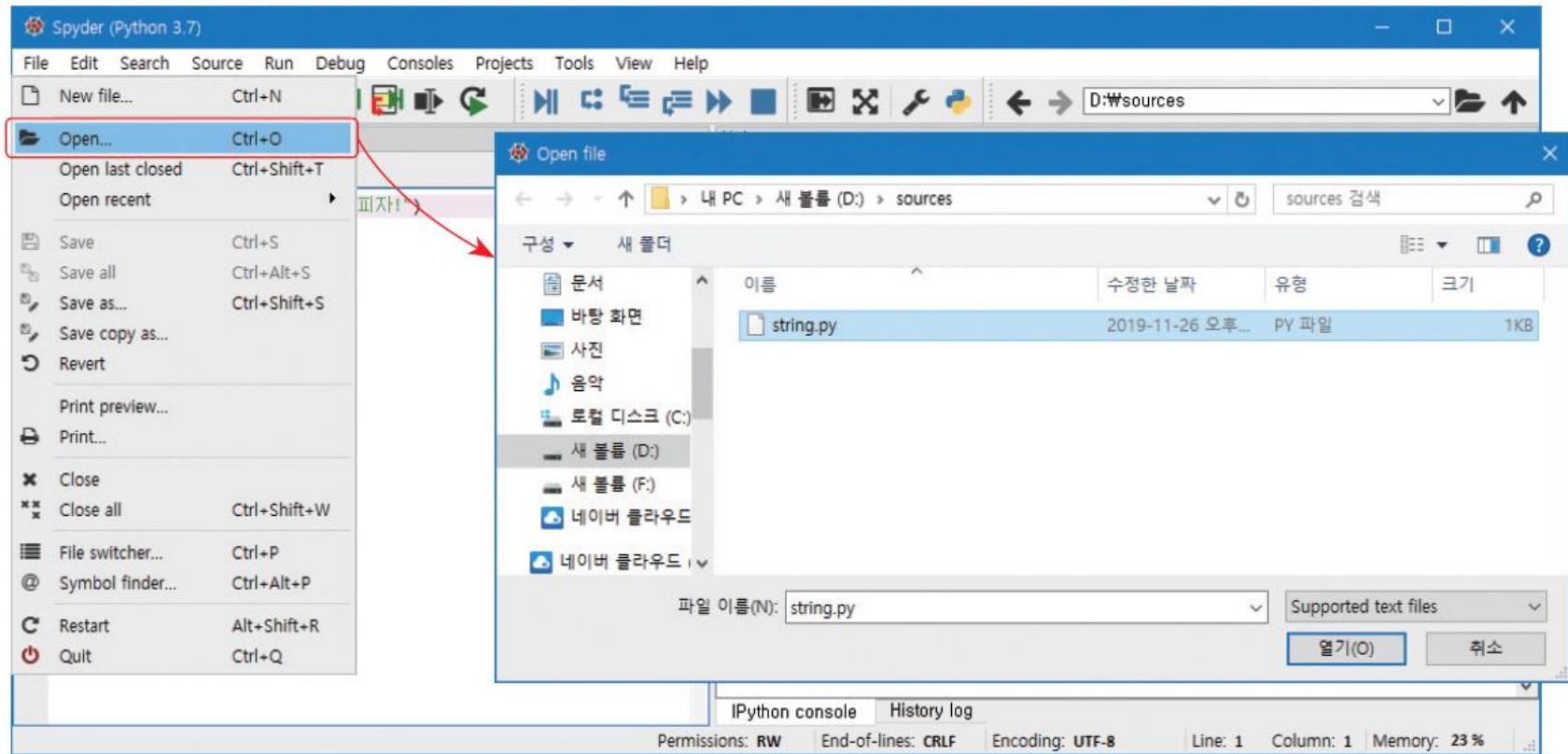
# 스크립트 모드



스크리트 모드



# 소스 파일 다시 열기



# 중간 점검

1. 스파이더는 두 가지 모드로 사용할 수 있다. 두 가지 모드에 대하여 설명해보자.
2. 파이썬으로 프로그램을 작성할 때 대문자와 소문자를 구분할까?
3. 파이썬 소스 파일의 확장자는 무엇인가?



# print() 함수

- 파이썬 프로그램은 여러 줄의 명령문(statement)들로 이루어진다.
- 함수(function)는 특별한 작업을 수행하는 명령어들의 모임이다.

Syntax: print() 함수 #1

형식

print(**문자열**)

print() 함수는 텍스트를 화면에 출력한다.

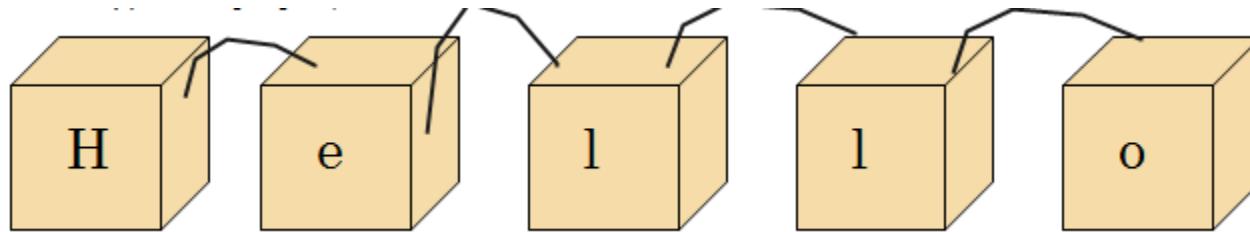
예

print("Hello World!")

"Hello World!"가 콘솔에 출력된다.

# 문자열

- 문자열(string) : 큰따옴표("...")나 작은따옴표('...') 안에 들어 있는 텍스트 데이터



- 반드시 따옴표가 있어야 한다.

```
>>> print(Hello World!)
      SyntaxError: invalid syntax
```

# print() 함수

- 여러 개의 값을 화면에 차례대로 출력할 수 있다.

```
>>> print("결과값은", 10, "입니다.")  
결과값은 10 입니다.
```

# 중간 점검

1. 다음과 같이 3단 구구단의 일부를 출력하는 프로그램을 작성해보자.  
 $3*1, 3*2, 3*3$ 의 수식을 계산하여서 결과를 출력한다.

$$3 * 1 = 3$$

$$3 * 2 = 6$$

$$3 * 3 = 9$$



# Lab: print() 함수 실습

파이썬에 오시 부로 환영합니다.

파이썬은 부습니다.

파이썬으로 부데어터, 부록그래프 작성할 수 있습니다.

# Sol: print() 함수 실습

```
print("파이썬에 오신 것을 환영합니다.")  
print("파이썬은 쉽습니다.")  
print("파이썬으로 빅데이터, 인공지능 프로그램을 작성할 수 있습니다.")
```

# Lab: 간단한 계산

$$2+3=5$$

$$2-3= -1$$

$$2 * 3 = 6$$

$$2/3= 0.6666666666666666$$

# Sol: 간단한 계산

```
print("2+3=", 2+3)
print("2-3=", 2-3)
print("2 *3=", 2 *3)
print("2/3=", 2/3)
```

# Lab: 오류를 처리해보자.

```
print(안녕하세요?)  
Print("이번 코드에는 많은 오류가 있다네요")  
print("제가 다 고쳐 보겠습니다.)
```

```
File "D:/모두의 파이썬/sources/chap01/hello.py", line 1  
    print(안녕하세요?)  
          ^  
SyntaxError: invalid syntax
```

# Sol: 오류를 처리해보자.

bug.py

```
1 print(안녕하세요?)  
2 Print("이번 코드에는 많은 오류가 있다네요")  
3 print("제가 다 고쳐 보겠습니다.")
```

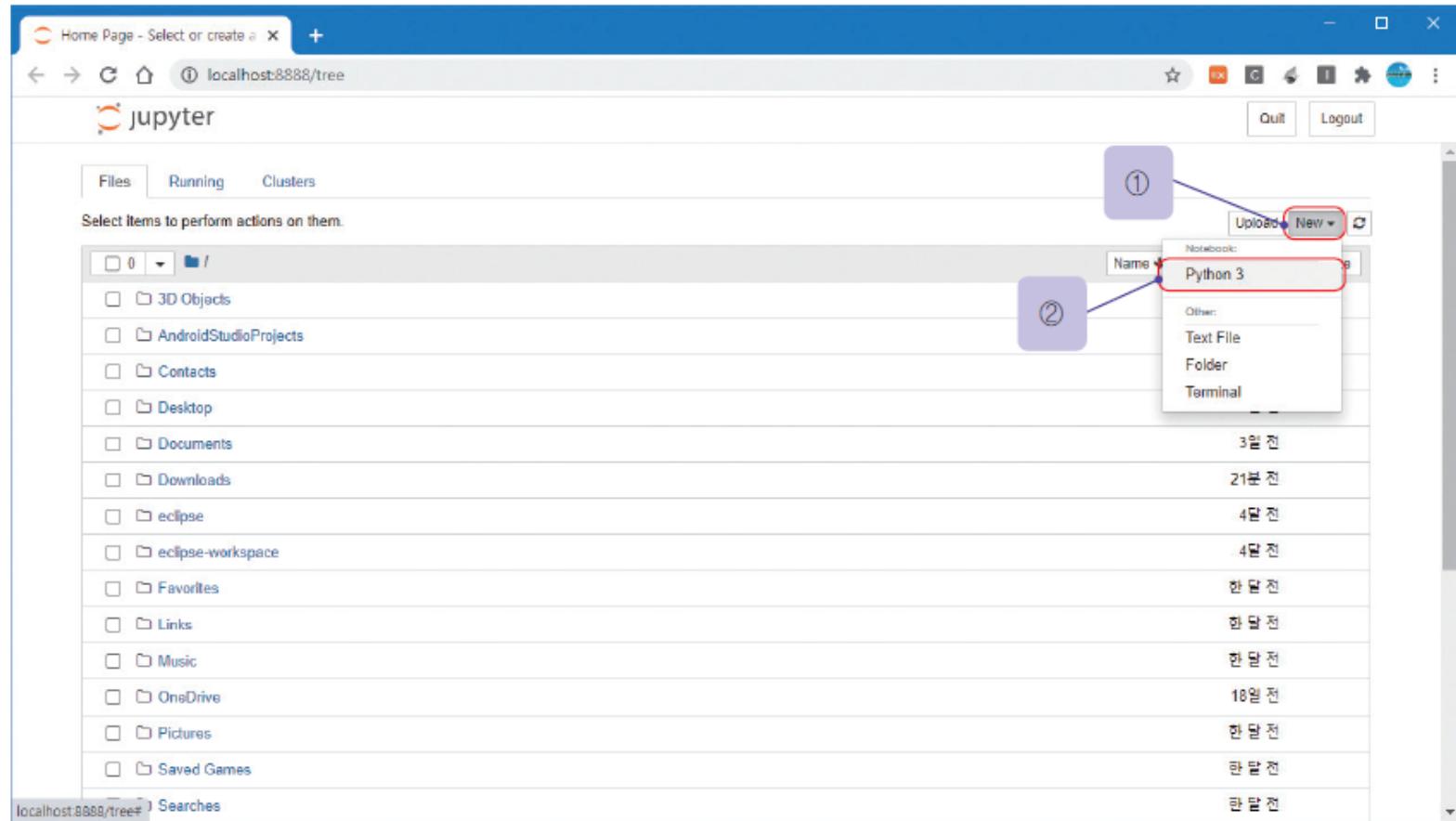
문자열인데 따옴표가 없다.

P가 대문자이다.

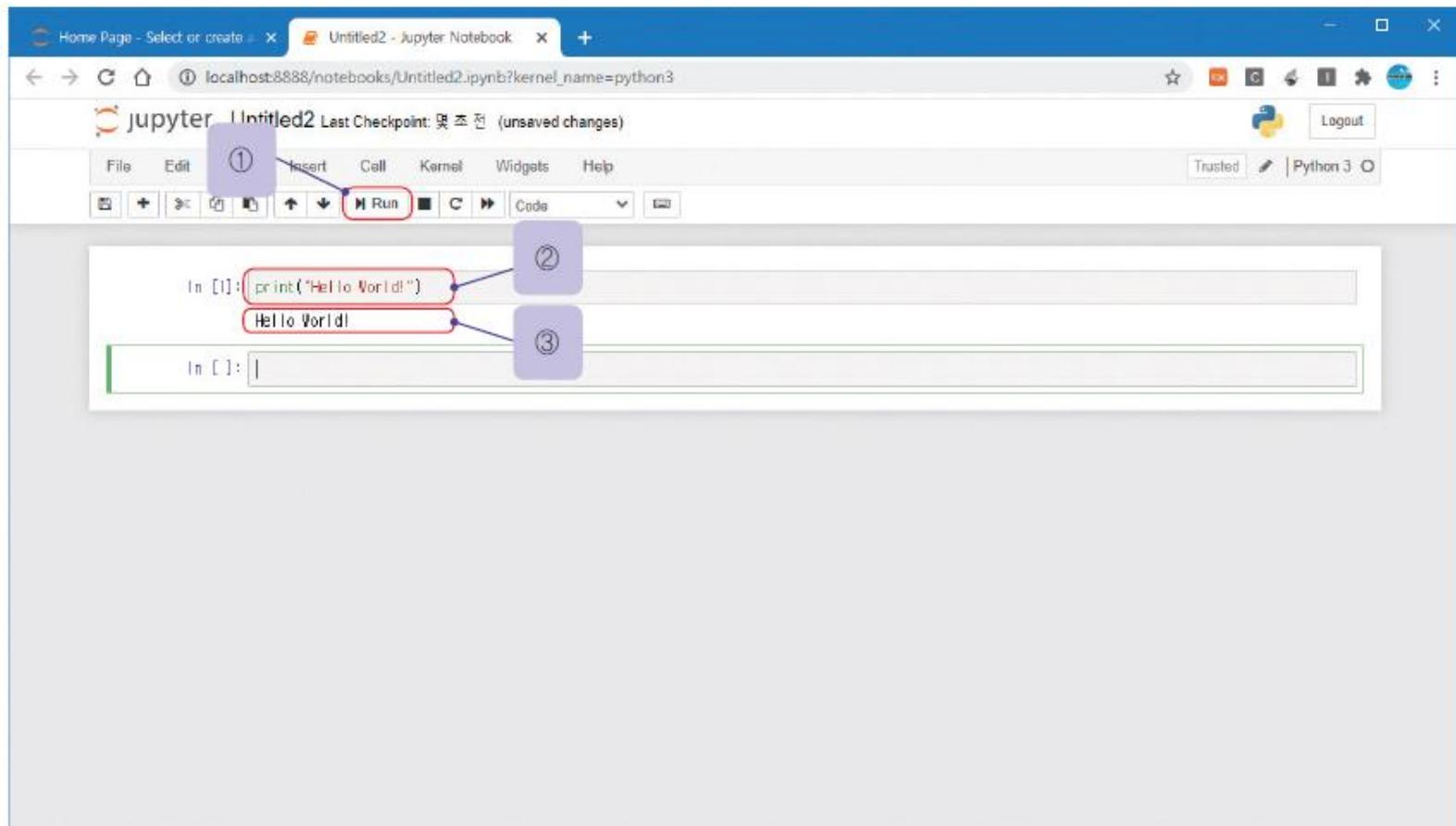
끝나는 따옴표가 없다.

```
print("안녕하세요?")  
print("이번 코드에는 많은 오류가 있다네요")  
print("제가 다 고쳐 보겠습니다.")
```

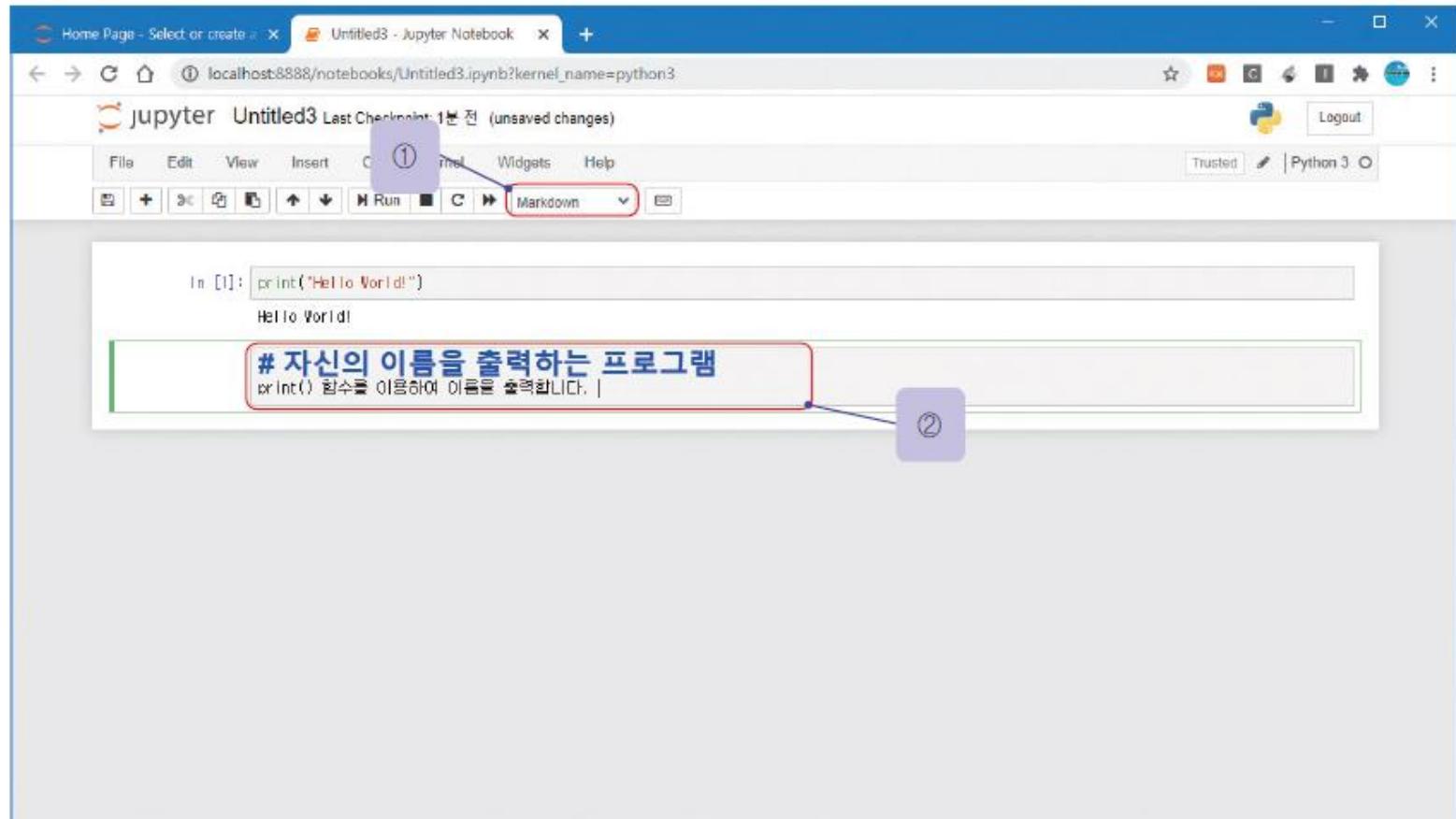
# 주피터 노트북



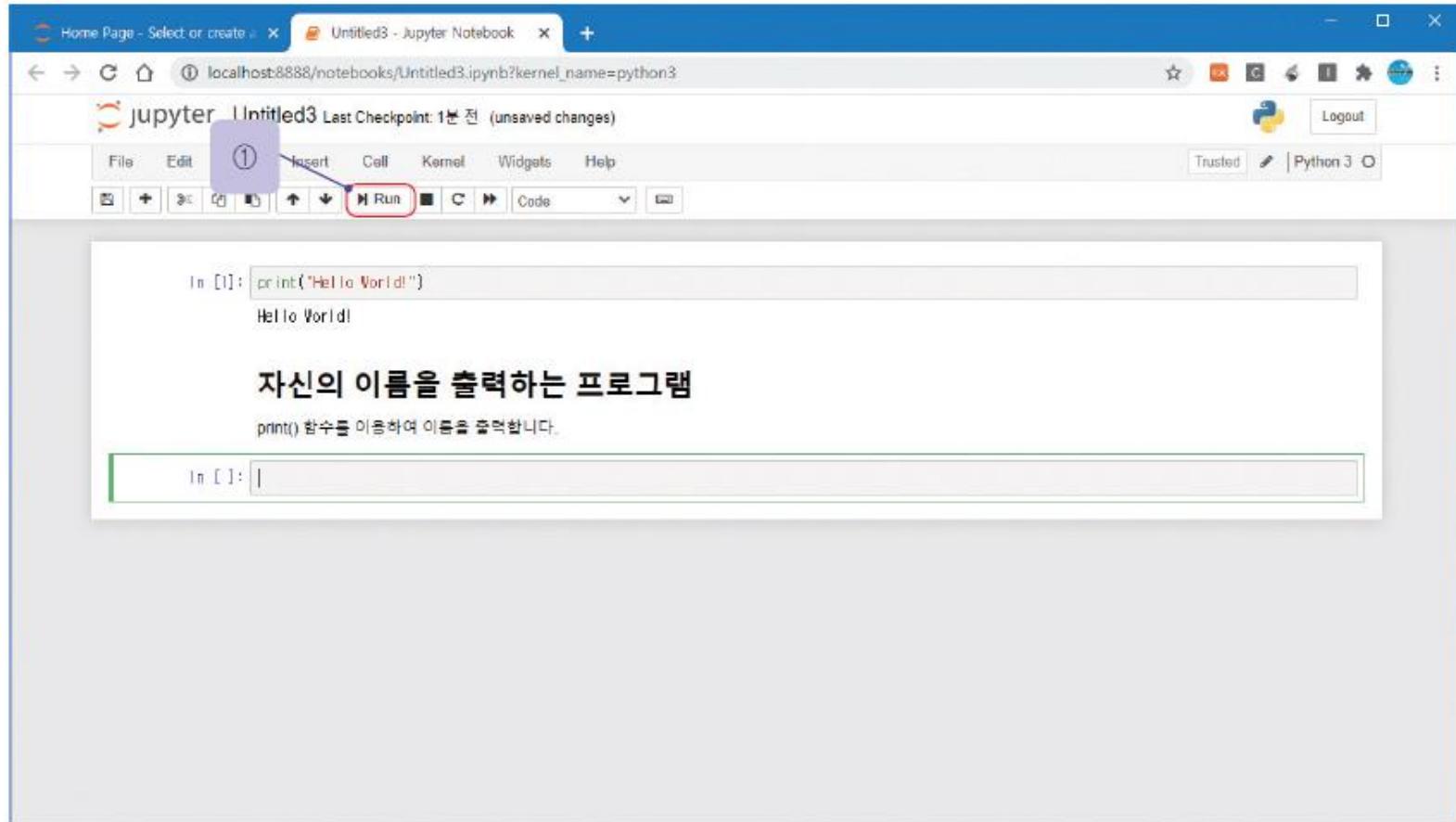
# 코드 추가



# 설명 추가하기



# 설명 추가하기



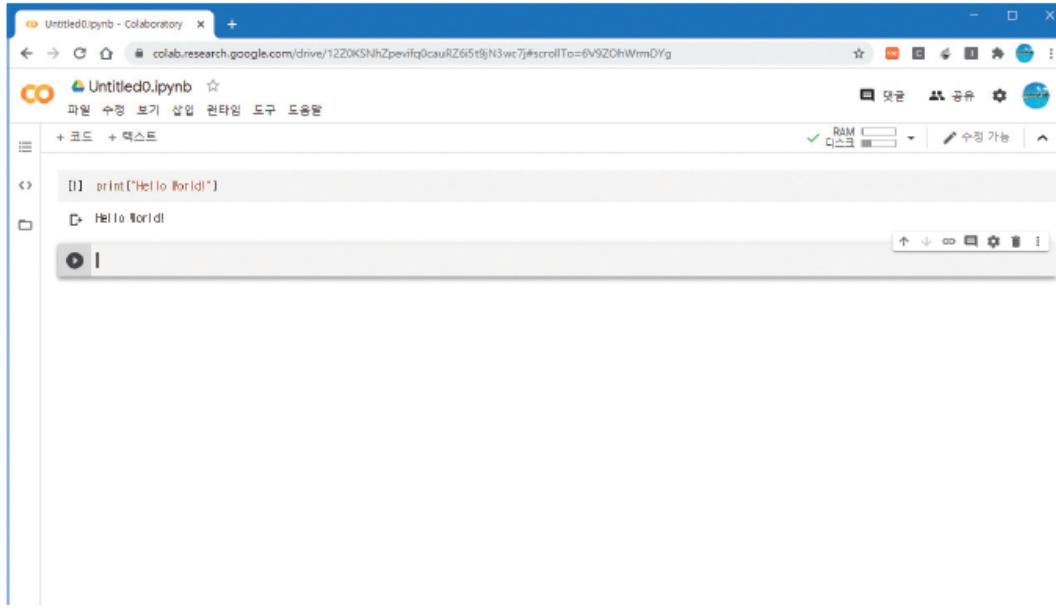
# 구글 Colab

## NOTE



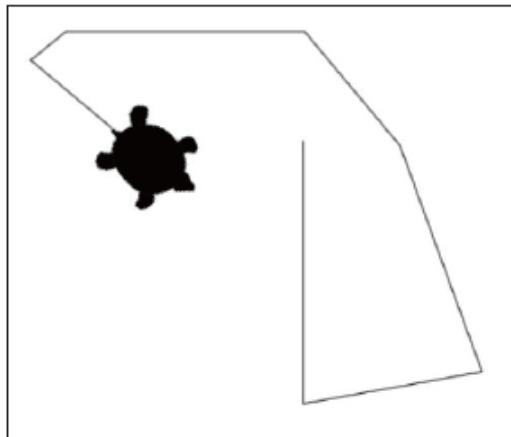
### 구글 Colab

구글 Colab은 주피터 노트북 개념을 클라우드로 확장한 것이다. 구글 Colab을 이용하면 브라우저에서 파이썬 프로그램을 작성하고 실행할 수 있다. Colab 메모장을 사용하면 실행 코드와 서식 있는 텍스트를 이미지, HTML, LaTeX 등과 함께 하나의 문서로 통합할 수 있다. Colab 메모장을 만들면 Google 드라이브 계정에 저장된다. Colab 메모장을 간편하게 공유하여 동료나 친구들이 댓글을 달거나 수정하도록 할 수 있다. 자세한 설명은 유튜브 영상 “<https://www.youtube.com/watch?v=inN8seMm7UI>”을 참조하자. 주피터 노트북과도 호환된다.



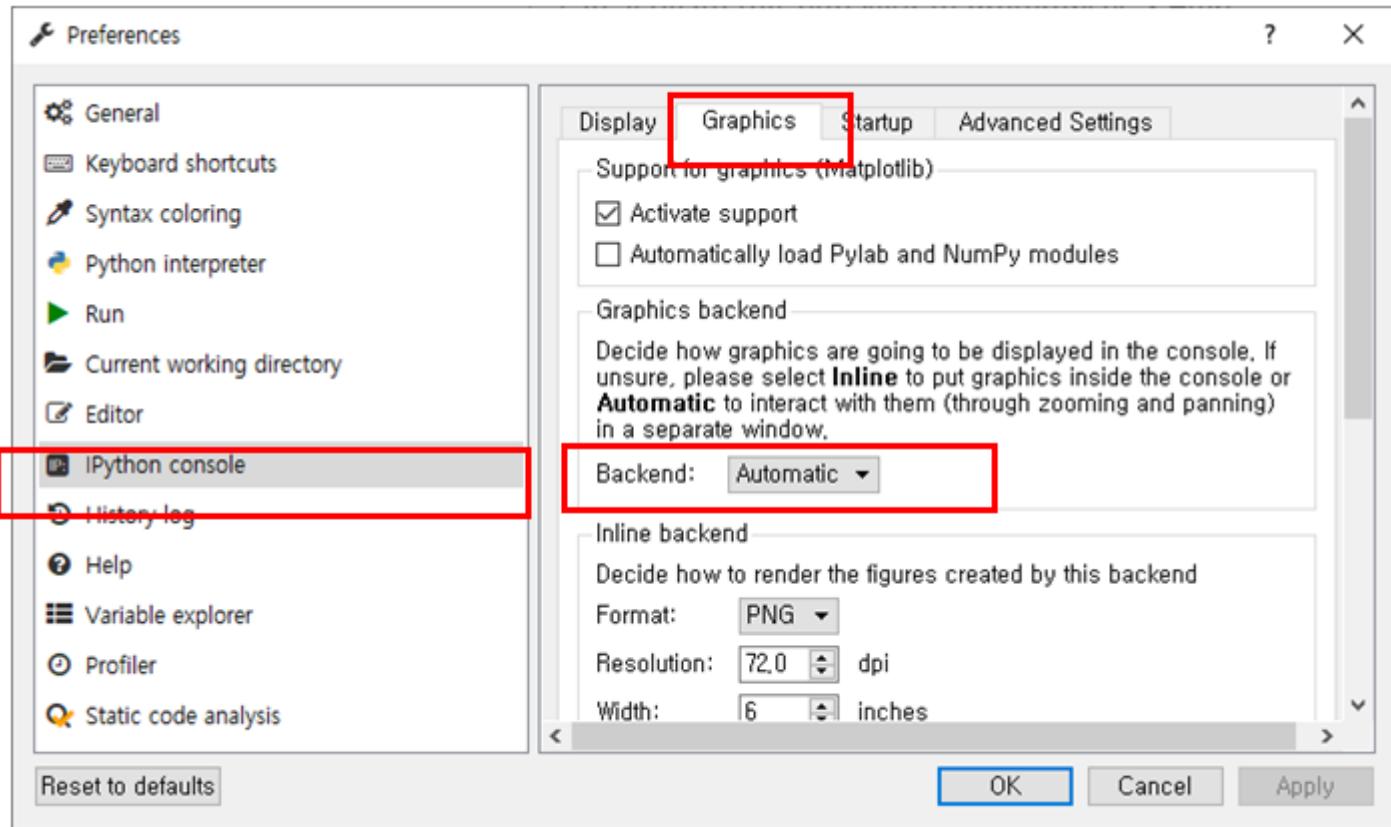
# 터틀 그래픽

- 터틀 그래픽은 화면에서 거북이를 이용하여서 그림을 그리는 기능이다.



터틀 그래픽은  
거북이가 화면 위에서  
지나가면서 그림을  
그리는 것이라고  
생각하세요.

# 스파이더 설정 변경



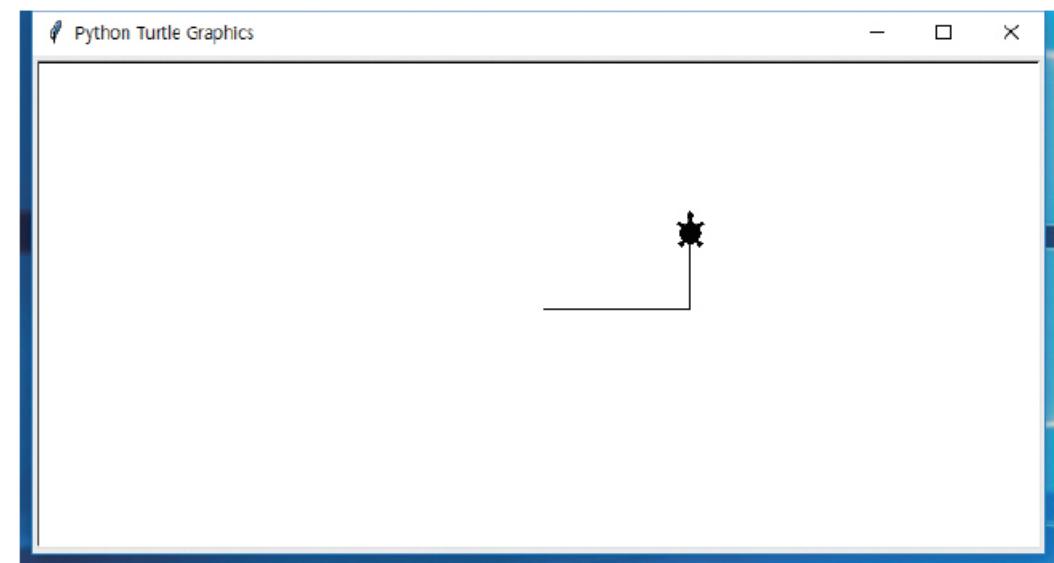
# 터틀 그래픽 시작

```
import turtle # (1)

t = turtle.Turtle() # (2)
t.shape("turtle") # (3)

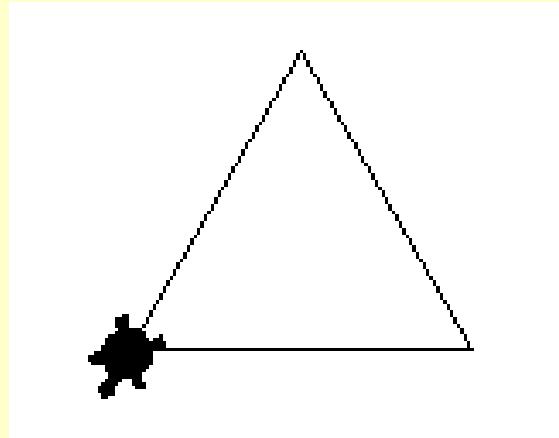
t.forward(100) # (4)
t.left(90) # (5)
t.forward(50)

turtle.mainloop() # (6)
turtle.bye()
```



# Lab: 삼각형 그리기

```
import turtle  
t = turtle.Turtle()  
  
t.shape("turtle")  
t.forward(100)  
t.left(120)  
t.forward(100)  
t.left(120)  
t.forward(100)  
  
turtle.mainloop()  
turtle.bye()
```



# 파이썬으로 무언가 만들 수 있겠다?

```
import turtle

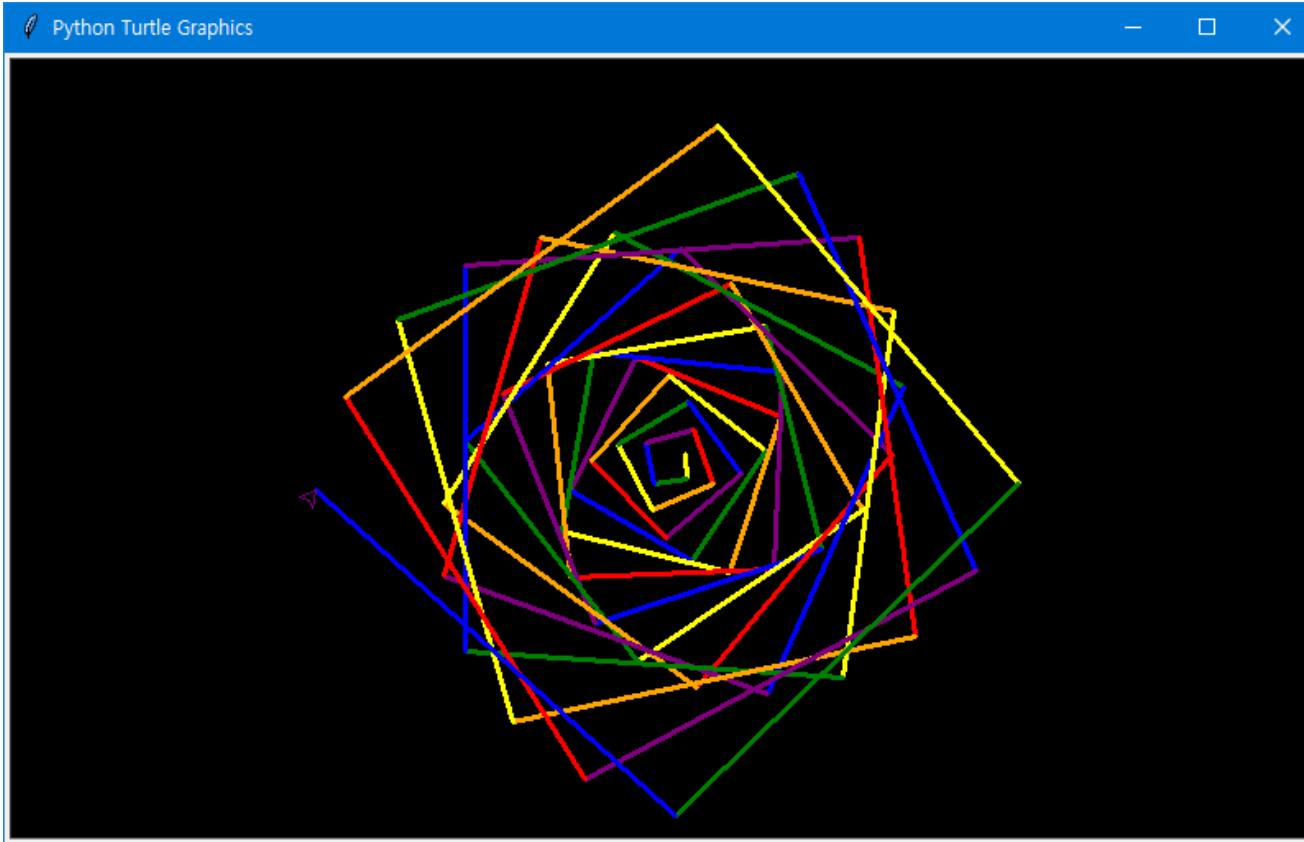
colors = ["red", "purple", "blue", "green", "yellow", "orange" ]
t = turtle.Turtle()

turtle.bgcolor("black")
t.speed(0)
t.width(3)
length = 10

while length < 300:
    t.forward(length)
    t.pencolor(colors[length%6])
    t.right (86)
    length += 5

turtle.mainloop()
turtle.bye()
```

# 실행 결과는?



# Lab: 파일 안의 단어 분석하기

```
from collections import Counter  
  
f = open("d://mobydick.txt", encoding="utf-8")  
count = Counter(f.read().split())  
print("단어 출현 횟수 :", count)
```

단어 출현 횟수 : Counter({'the': 13851, 'of': 6638, 'and': 6000, 'a': 4549, 'to': 4529, 'in': 3904, 'that': 2692, 'his': 2428, 'I': 1723, 'with': 1695, 'as': 1600, 'is': 1588, 'was': 1567, 'it': 1516, 'he': 1495, 'for': 1385, 'all': 1314, 'at': 1231, 'this': 1169, 'by': 1121, 'from': 1072, 'not': 1043, 'but': 1034, 'be': 991, 'on': 926, 'so': 785, 'you': 784, 'or': 763, 'one': 755, 'have': 752, 'had': 751, 'were': 645, 'But': 637, 'The': 635, 'their': 613, 'are': 586, 'an': 579, 'some': 571, ...}

# 이번 장에서 배운 것

- 프로그램은 컴퓨터에 내리는 \_\_\_\_\_로 이루어지는 작업 지시서이다.
- 고급 언어는 컴퓨터가 이해할 수 있는 언어이다.
- 다양한 종류의 프로그래밍 언어가 있고 파이썬도 프로그래밍 언어의 일종이다.
- 파이썬으로 프로그램을 작성하기 위한 개발 환경인 \_\_\_\_\_는 <https://www.anaconda.com/distribution/> 웹사이트에서 다운받아서 설치할 수 있다.
- IPython 콘솔에서는 프롬프트 다음에 코드를 입력하고 □ 를 누르면 코드가 실행된다.
- 산술 계산을 하는 파이썬 연산자에는 +, -, \*, /가 있다.
- \_\_\_\_\_는 화면에 문자열이나 계산 결과를 출력할 때 사용하는 함수이다.
- 스크립트 모드를 사용하면 코드를 파일에 저장하였다가 꺼번에 실행할 수 있다.



# Q & A

