

# 고급프로그래밍및실습

## / 1. 파이썬 소개

-

이 정 진

조교수, 숭실대 글로벌미디어학부

[jungjinlee@ssu.ac.kr](mailto:jungjinlee@ssu.ac.kr), 정보과학관 623호



# 파이썬

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





# 파이썬의 특징

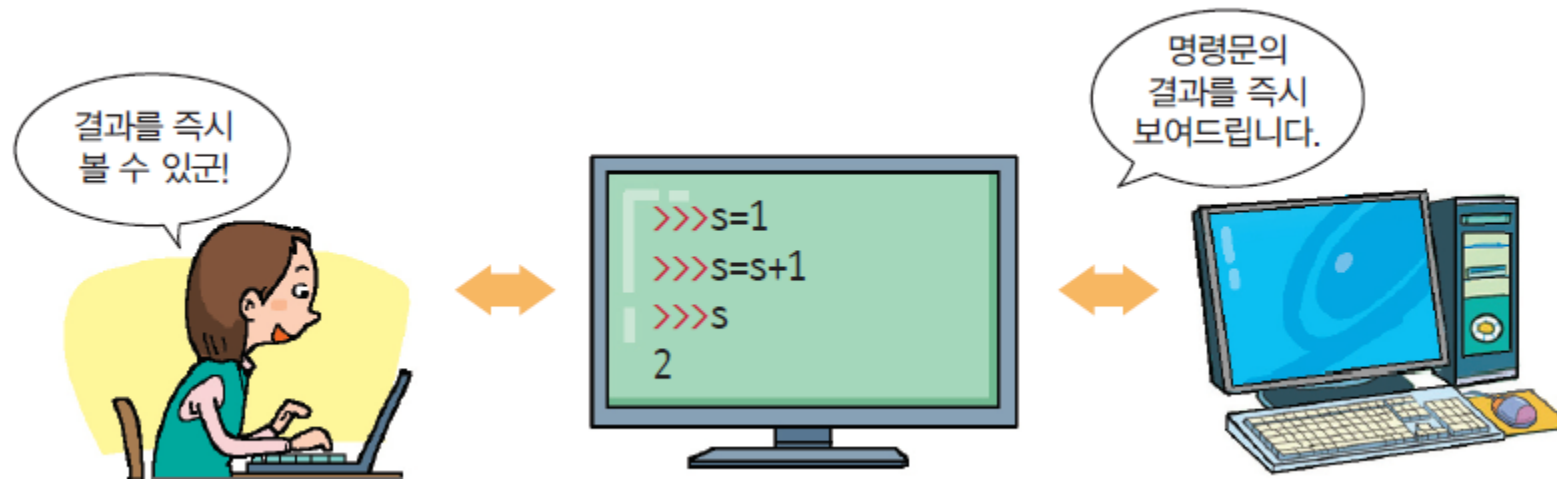
- 생산성이 뛰어나다.
- 간결하면서도 효율적인 프로그램을 빠르게 작성

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.

matplotlib

K Keras  
A deep learning library

OpenCV

파이썬의 막강한  
라이브러리

Requests

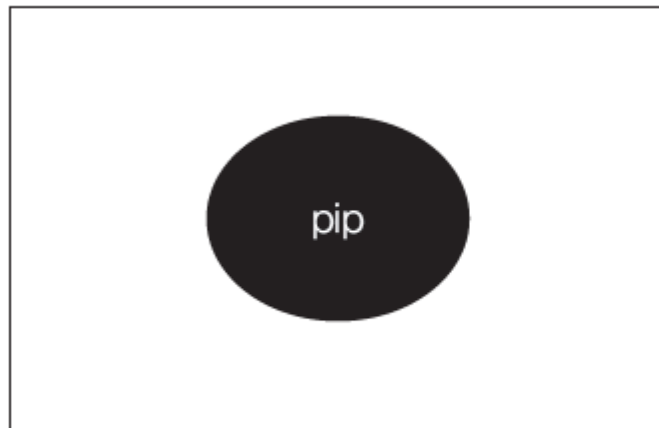
scikit  
learn

BeautifulSoup

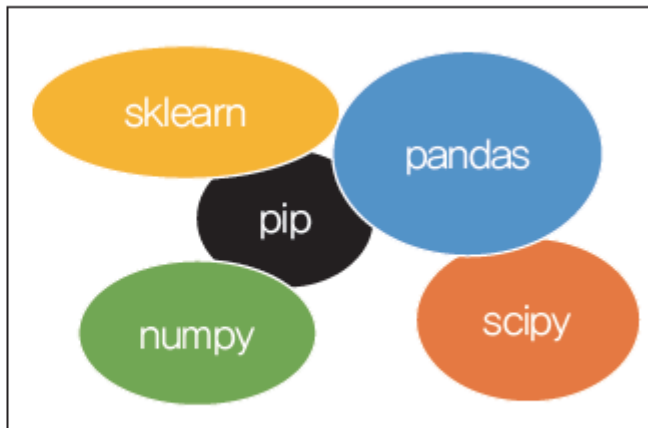


# 아나콘다

- 인기 있는 라이브러리가 거의 모두 포함된 배포판



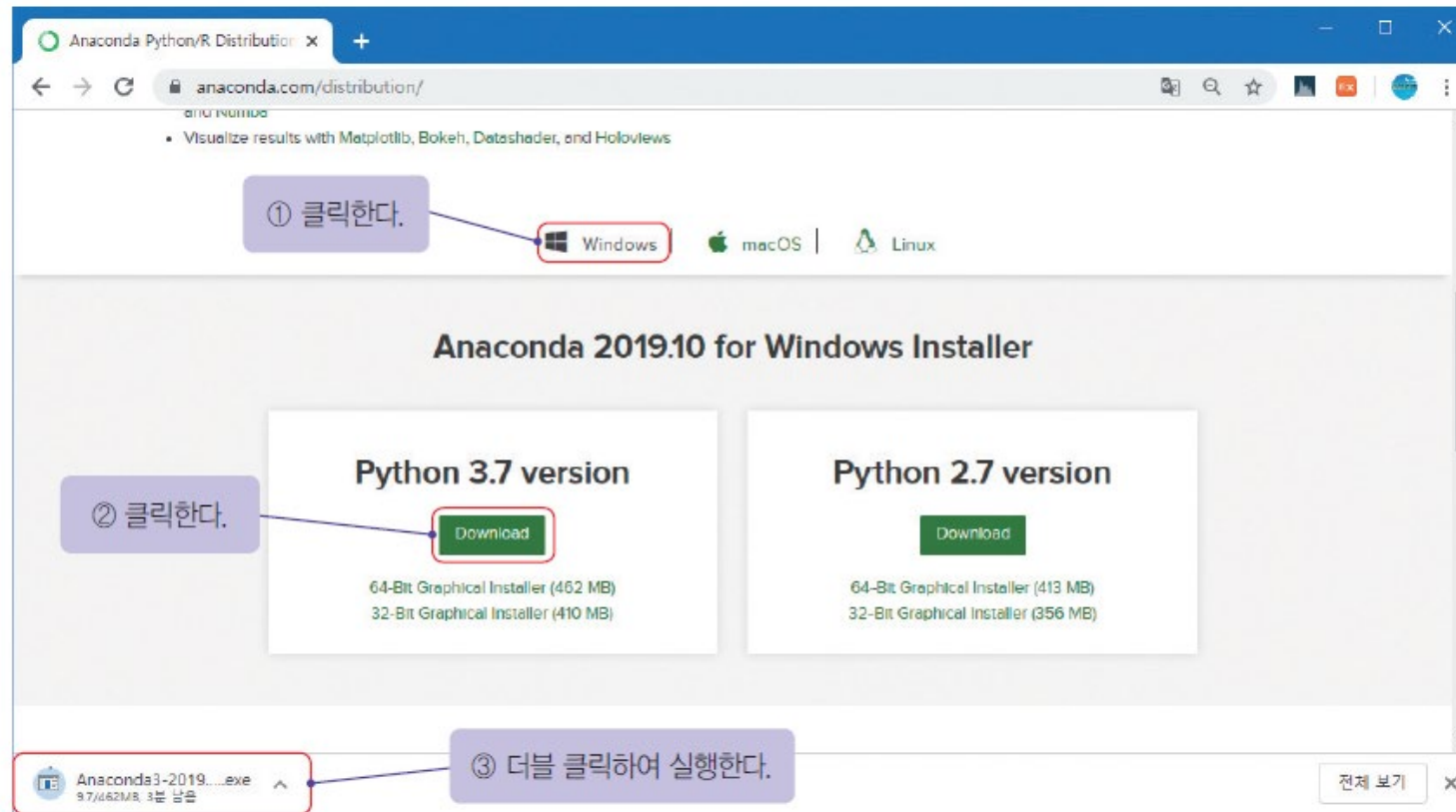
파이썬



아나콘다

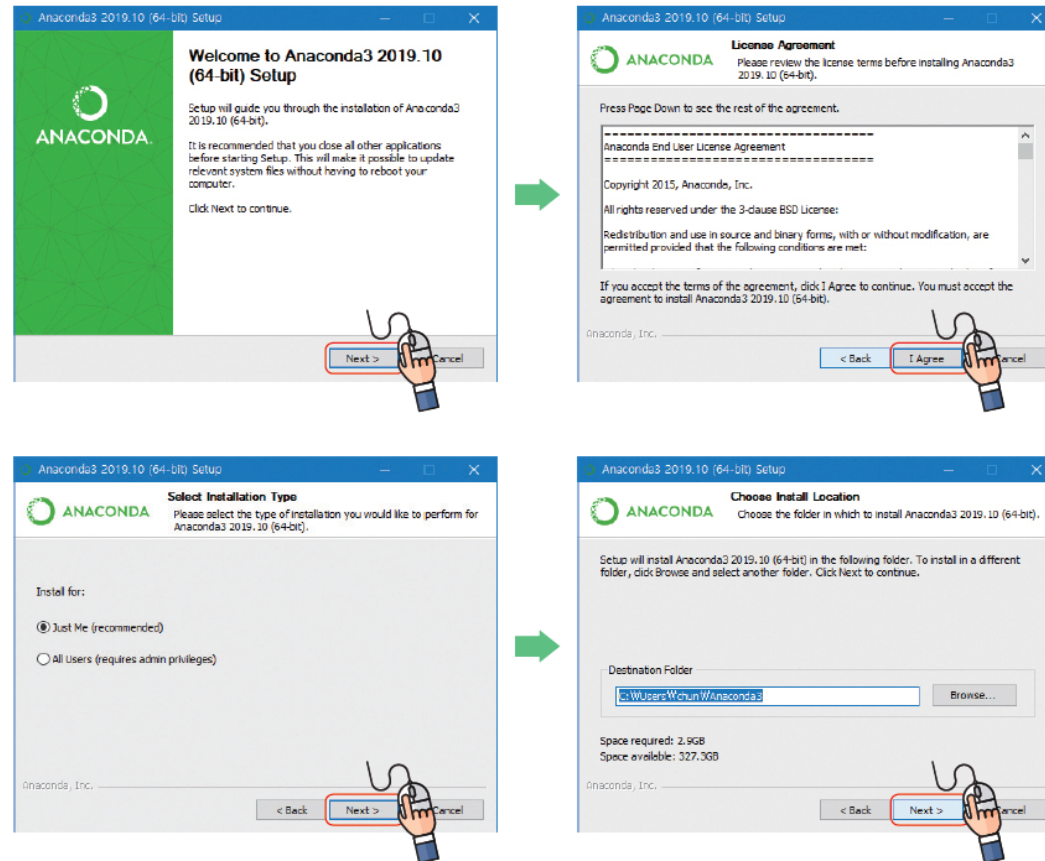


# 아나콘다 다운로드





# 아나콘다 설치







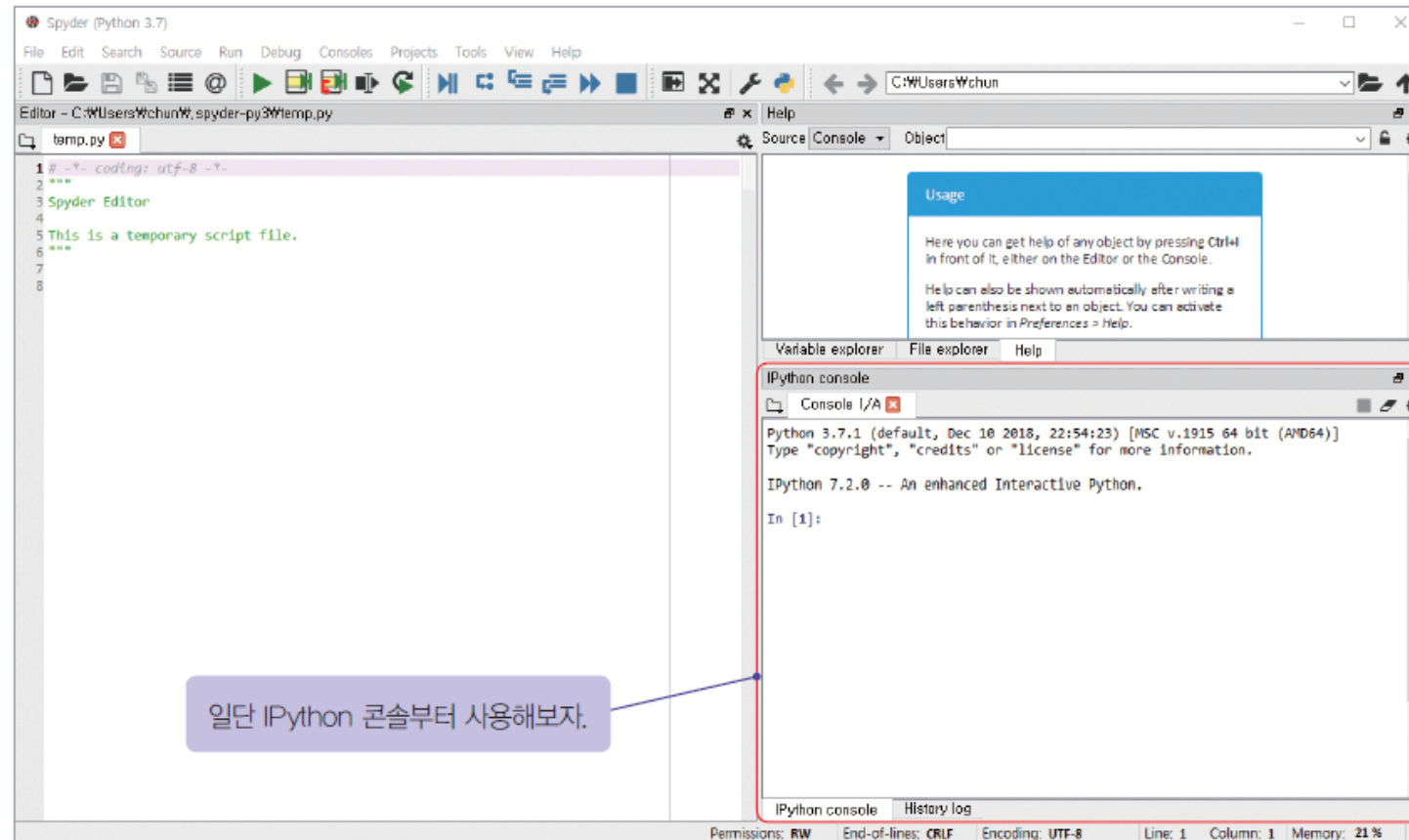
# 스파이더

- 스파이더는 파이썬으로 작성된 파이썬 개발 도구

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

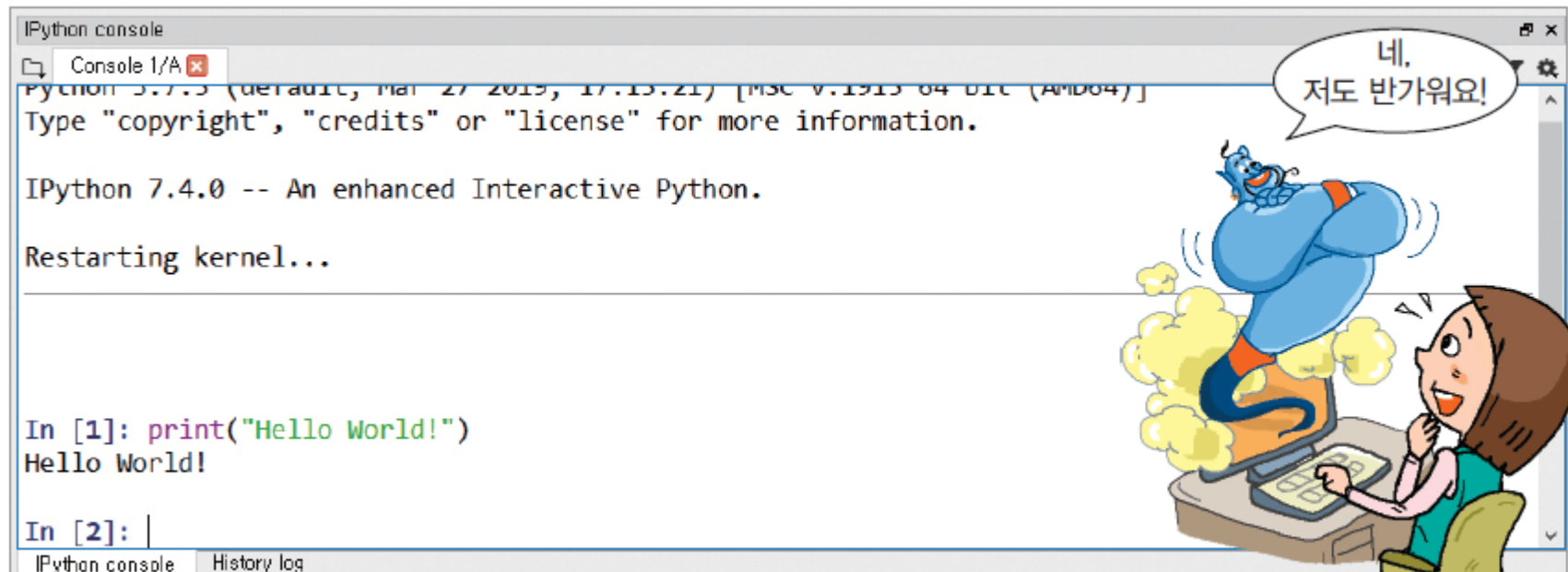


# Ipython 콘솔





# Ipython 콘솔





# 계산하기 #1

- 반지름이 10cm인 피자 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   
>>> math.sin(math.radians(30))   
0.49999999999999994
```

```
>>> math.sqrt(9.0)   
3.0
```

```
>>> math.log(1.0)   
0.0
```



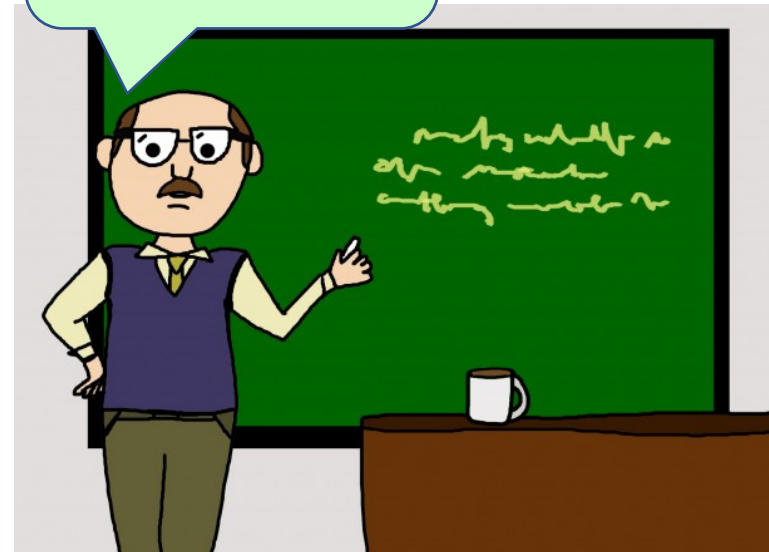
# 문자열 출력하기

```
>>> "나의 " + "고양이 "  
나의 고양이  
>>>
```

문자열은 어떻게 구별하나요?



따옴표("...")가 붙으면 문자열입니다.





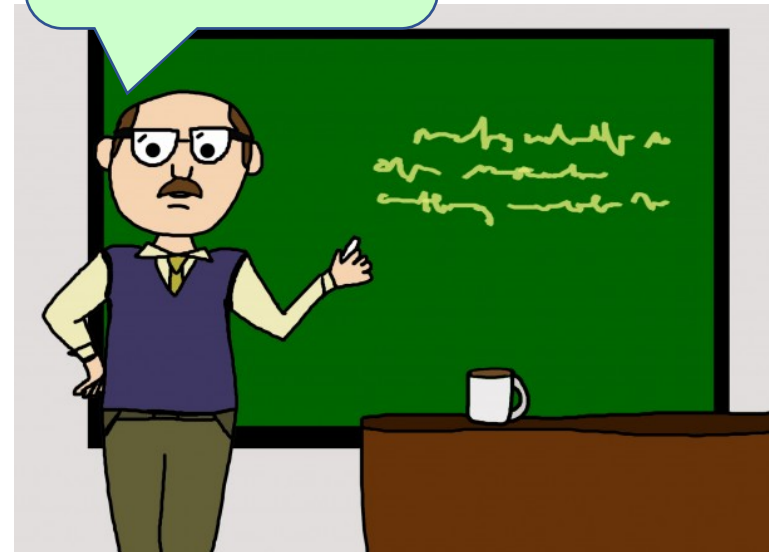
# 문자열 반복하기

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

문자열이 반복  
되네요!



문자열에 \*을 적  
용하면 반복을  
의미 합니다.





## 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 실행결과가 콘솔에 나타난다.



# 스크립트 모드

- 코드가 복잡해지면 인터프리트 모드는 번거롭다.

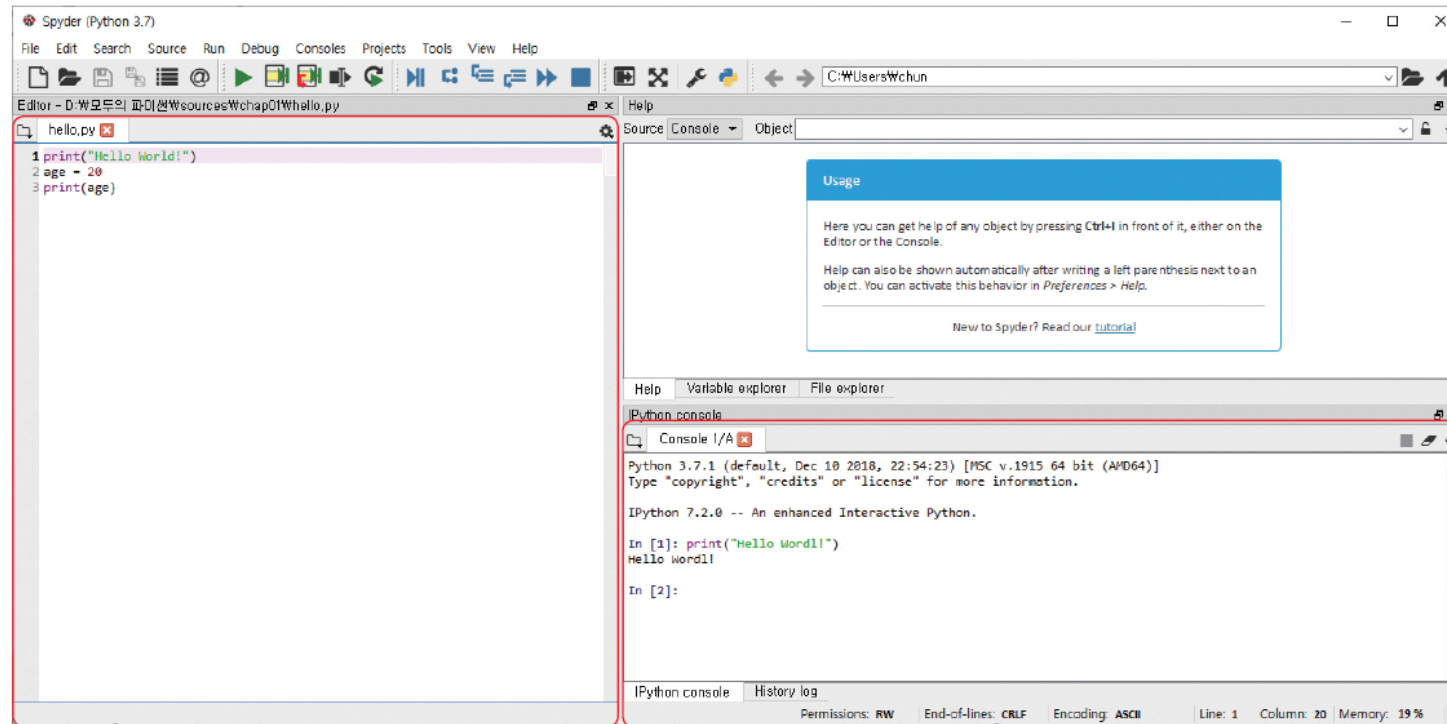
이걸 한 줄씩 입력하라고?



```
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 '= %s';' % ast[1]  
        else:  
            print ']'  
    else:  
        print '];'  
        children = []  
        for n, child in enumerate(ast[1:]):  
            children.append(dotwrite(child))  
        print ', ' % ast[1] -> (' % nodename  
        for n, child in enumerate(children):  
            print '%s' % name,
```



# 스파이더에서 대화형 모드와 스크립트 모드

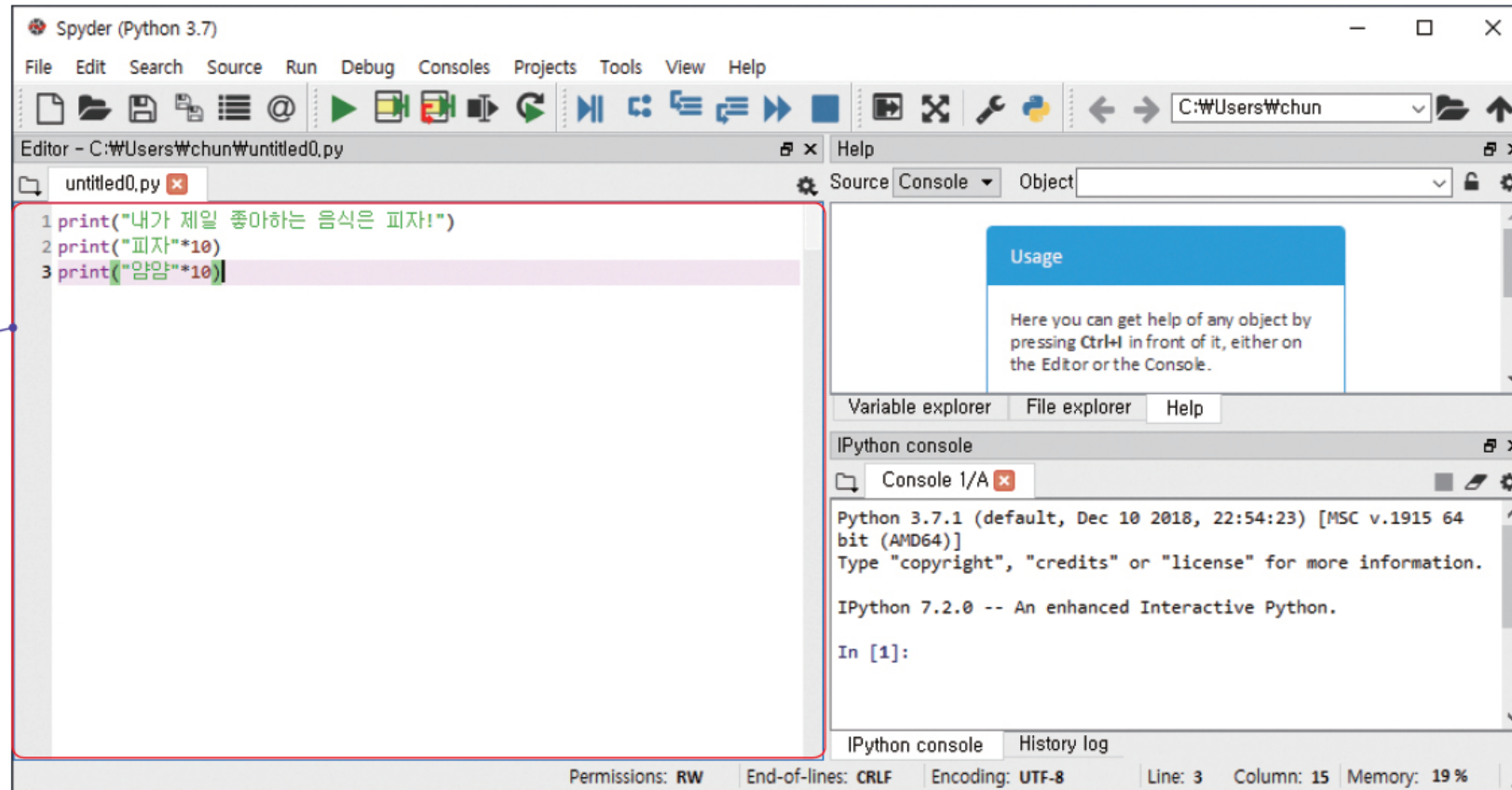


여기다 입력하면 스크립트  
모드로 실행된다.

여기다 입력하면 대화형  
모드로 실행된다.

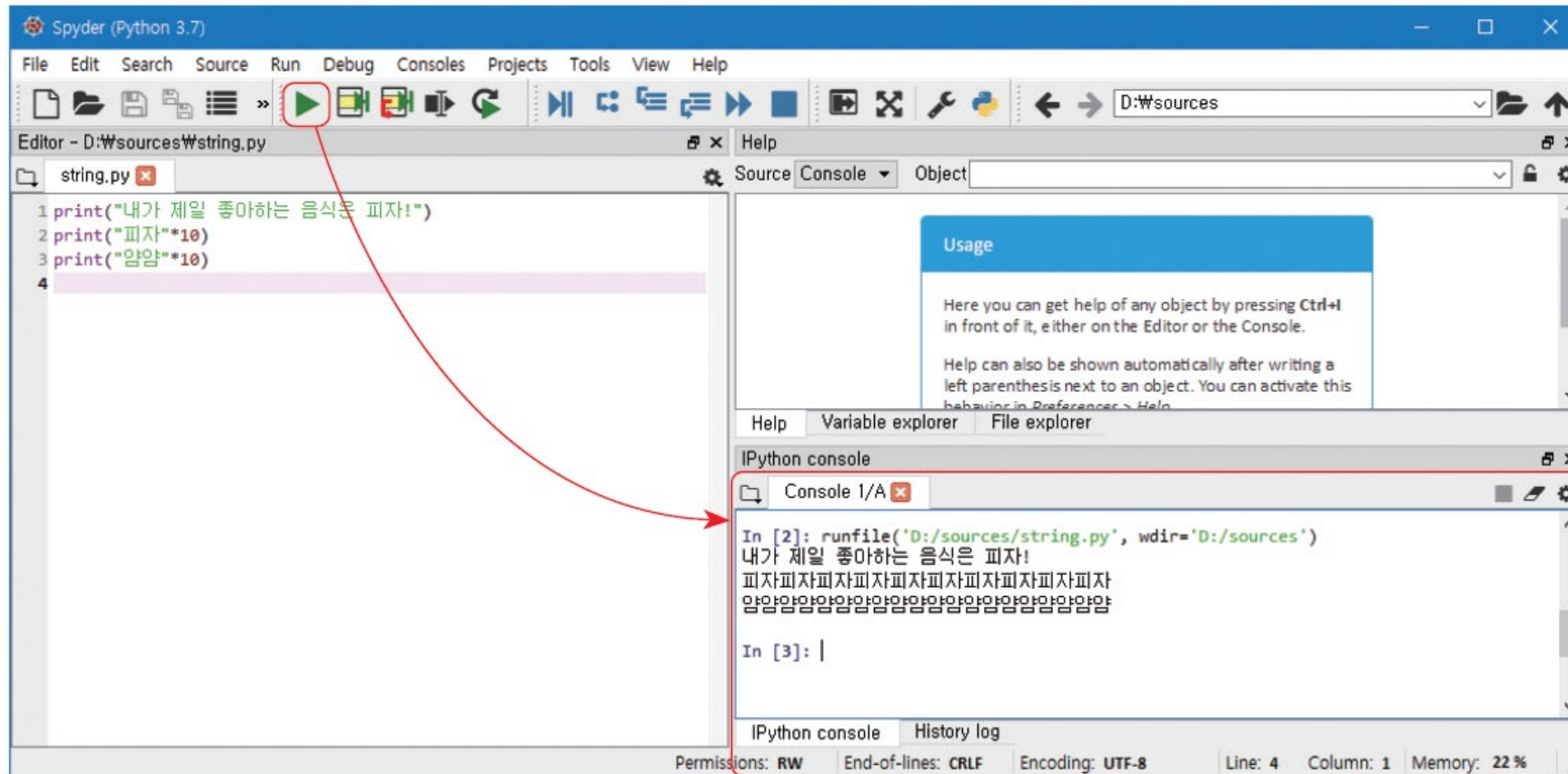


# 스크립트 모드



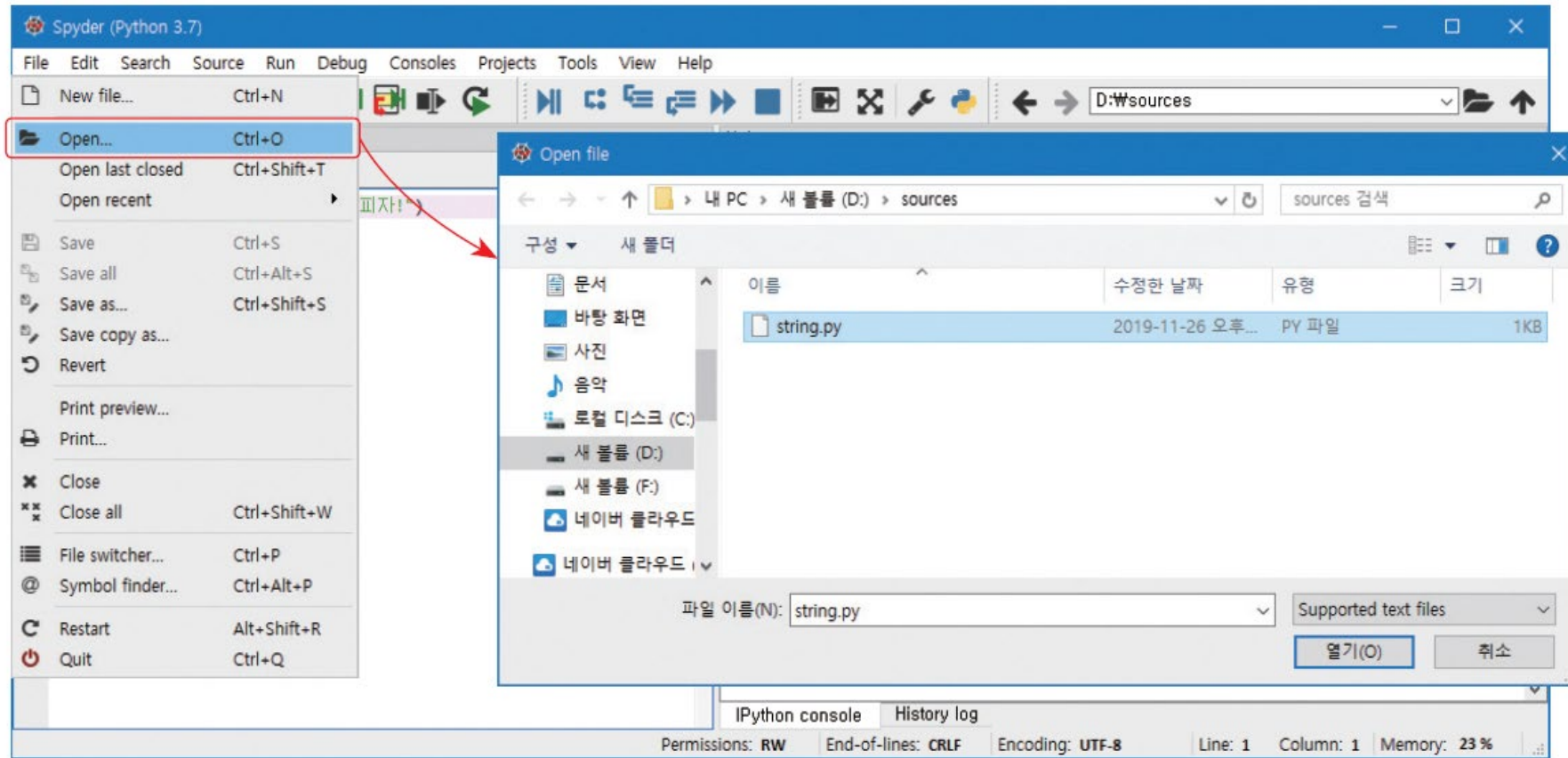


# 스크립트 모드





# 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.

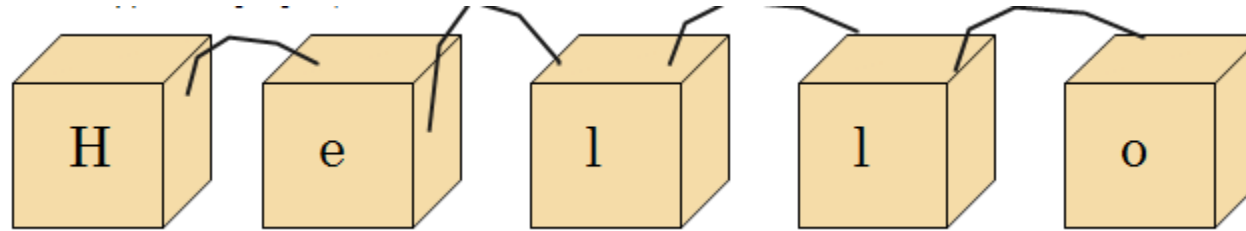
**예** print("Hello World!") "Hello World!"가 콘솔에 출력된다.





# 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 값들을 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\*3의 수식을 계산하여서 결과를 출력한다.

$$3 * 1 = 3$$

$$3 * 2 = 6$$

$$3 * 3 = 9$$





# Lab: print() 함수 실습

파이썬에 오신 것을 환영합니다.  
파이썬은 쉽습니다.  
파이썬으로 빅데이터, 인공지능 프로그램을 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.

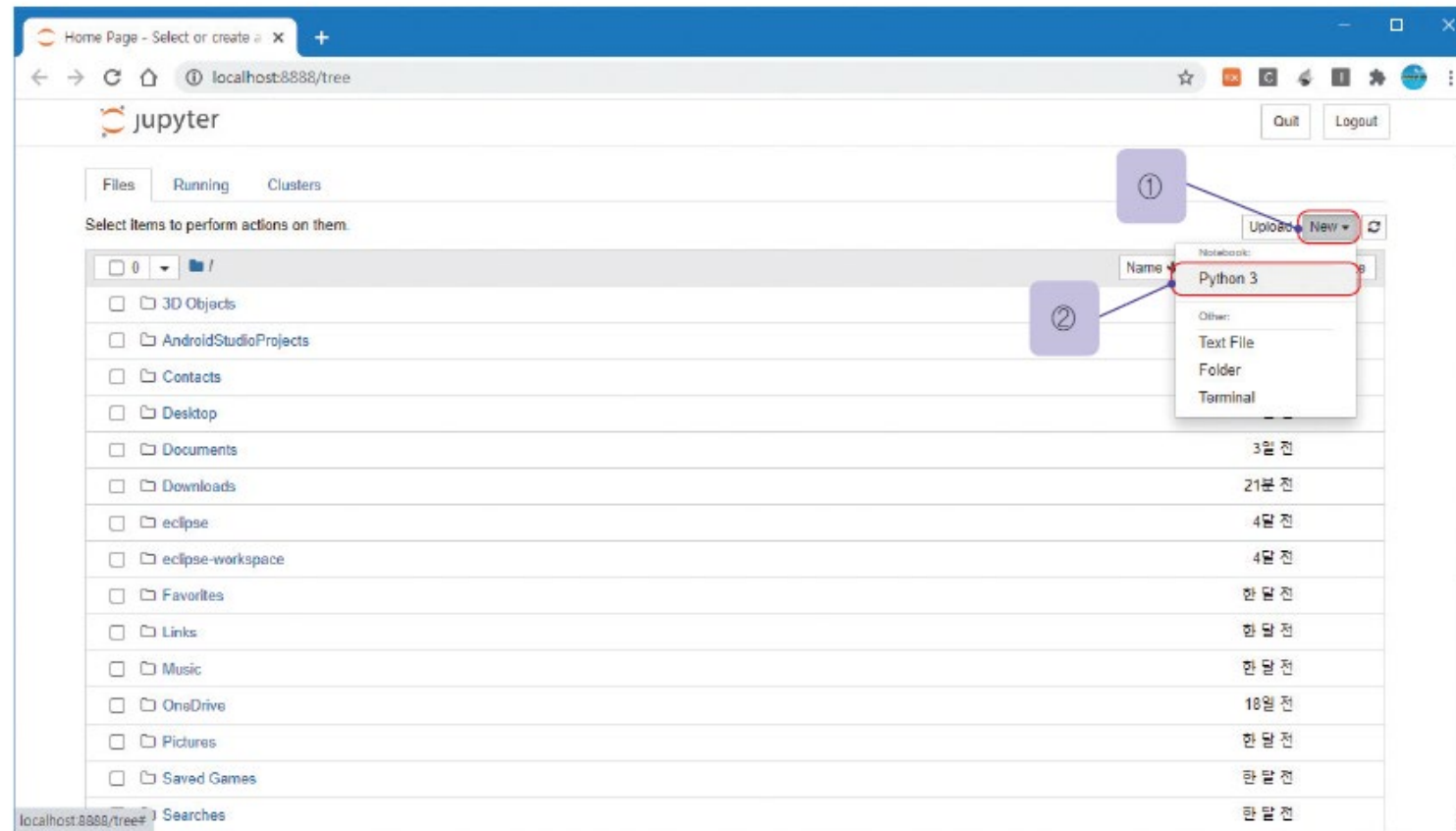
P가 대문자이다.

끝나는 따옴표가 없다.

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

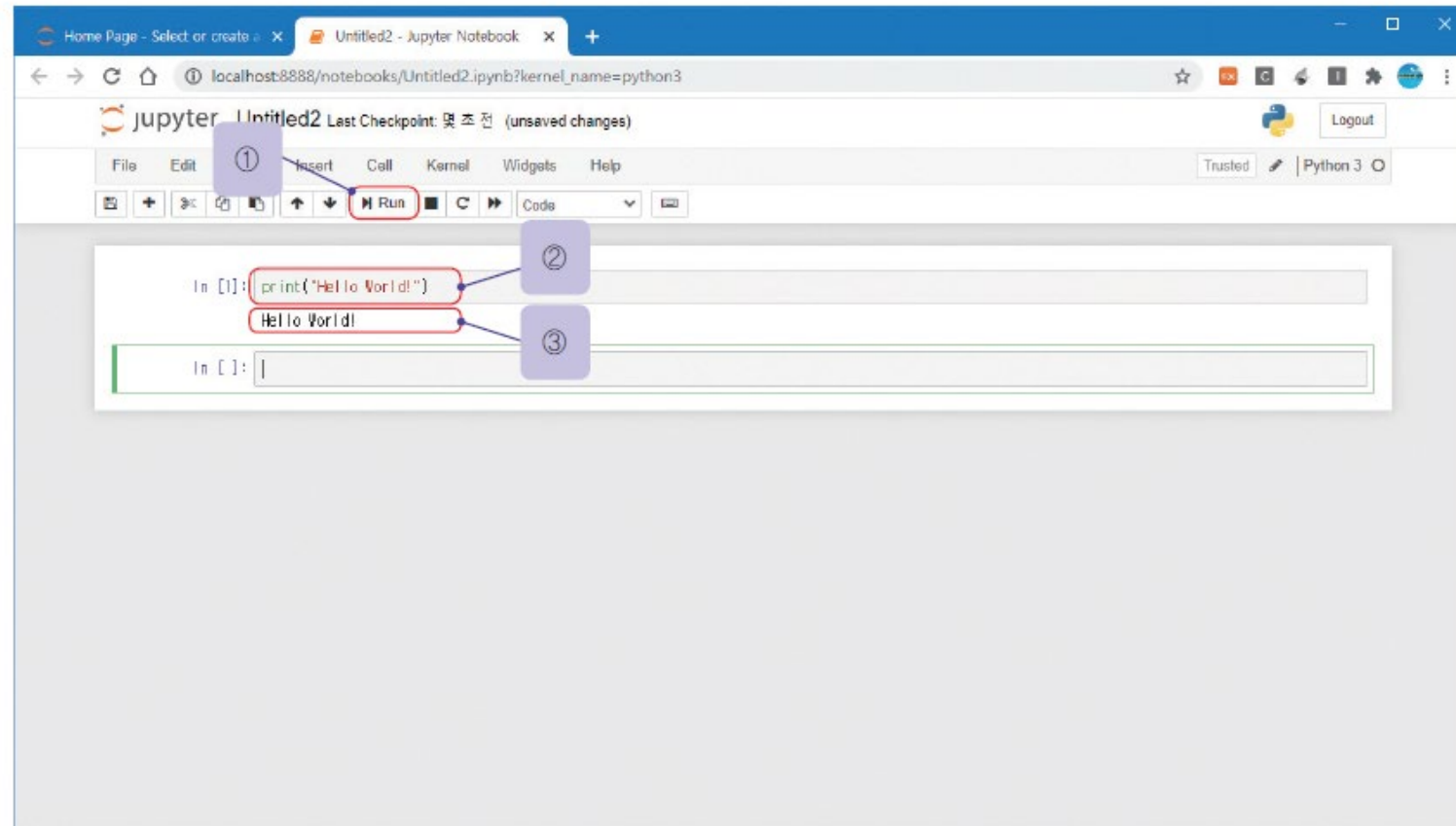


# 주피터 노트북



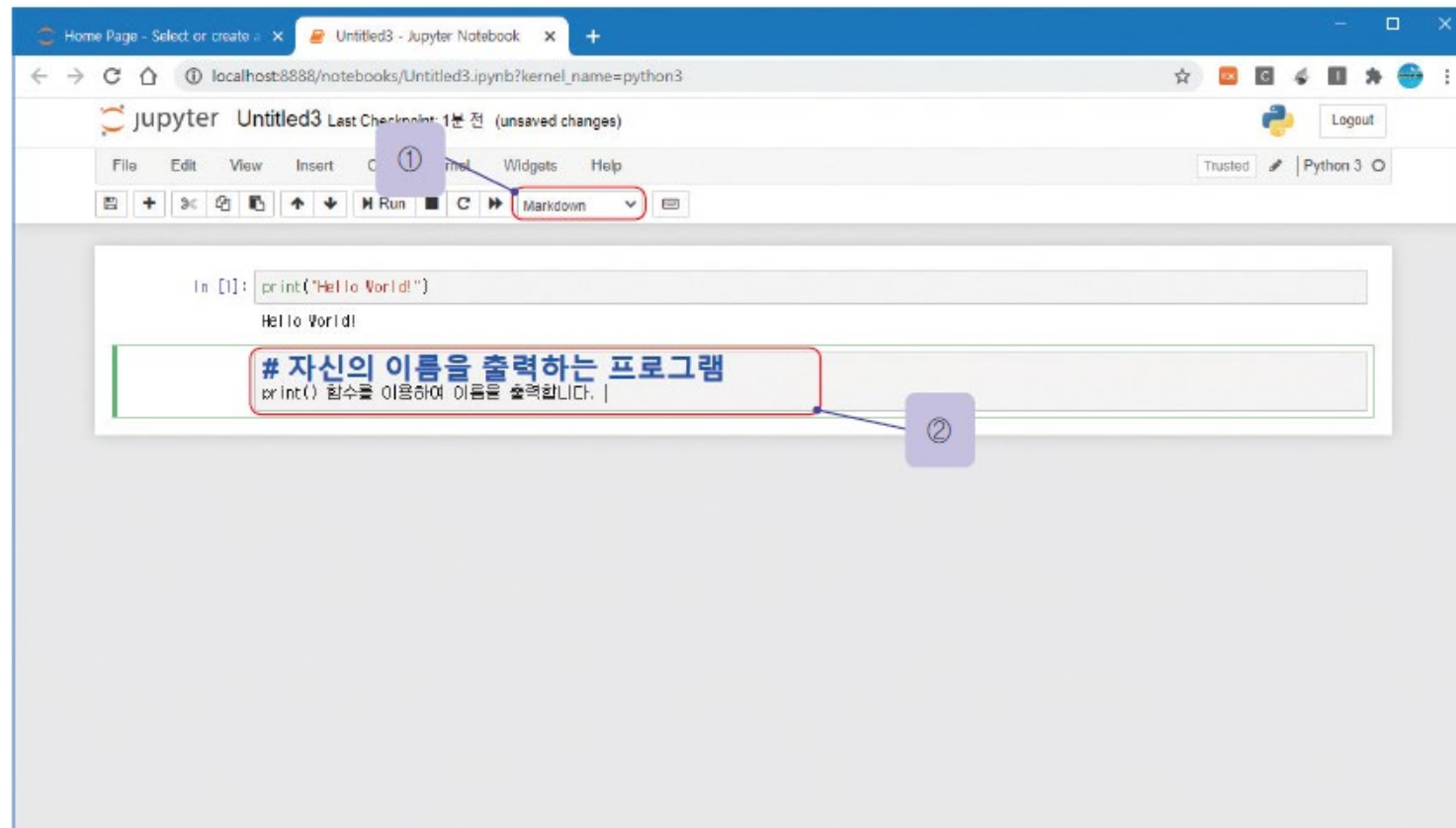


# 코드 추가



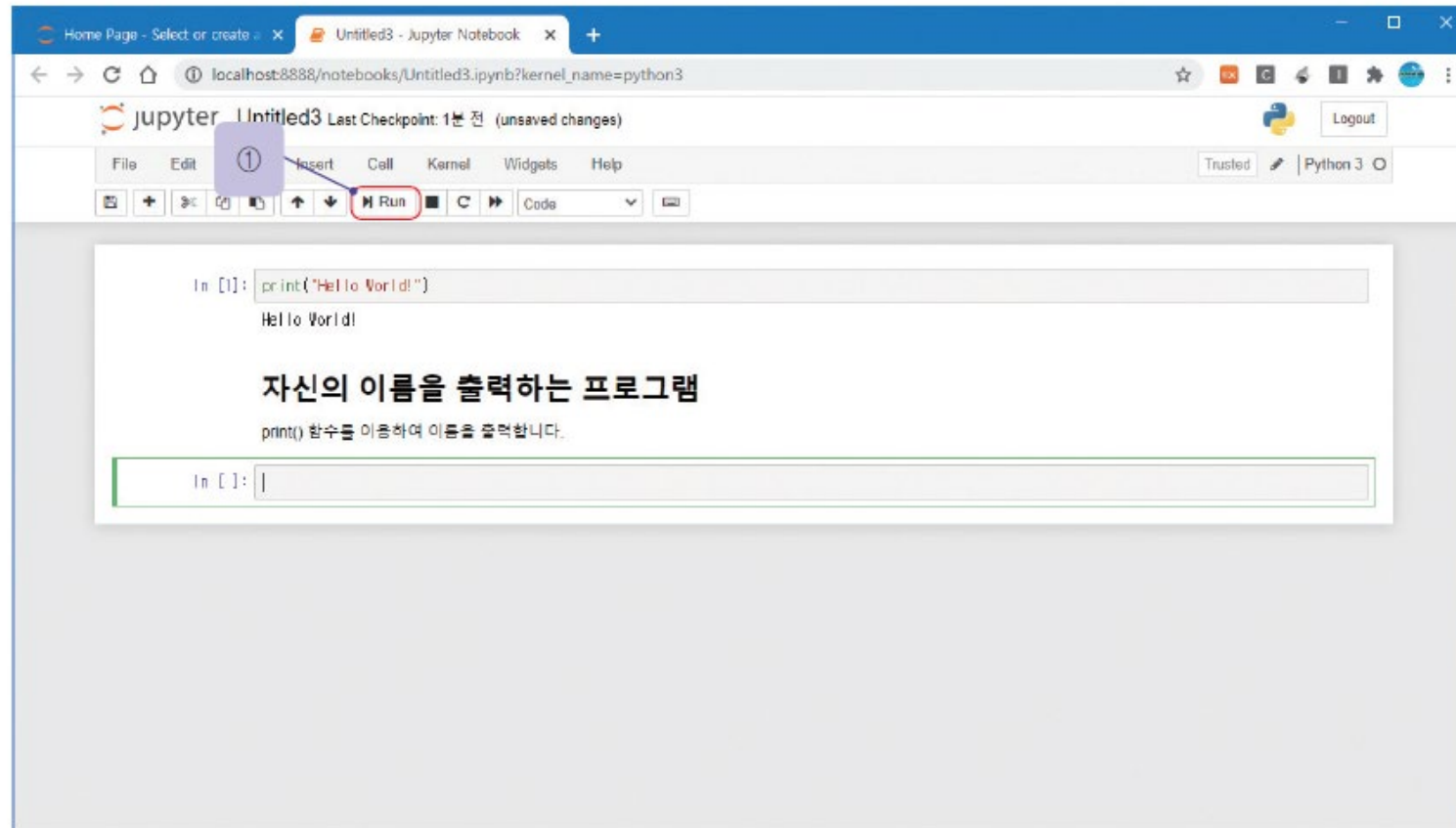


# 설명 추가하기





# 설명 추가하기





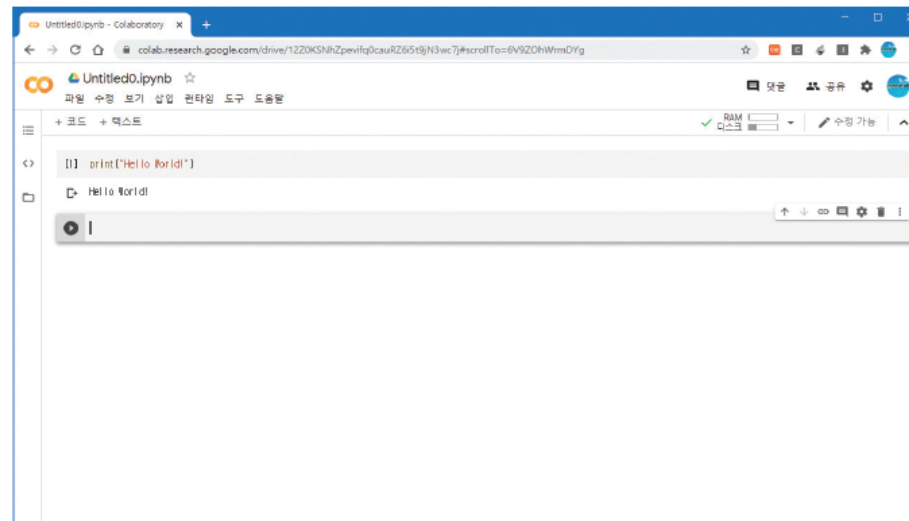
# 구글 Colab

## NOTE



### 구글 Colab

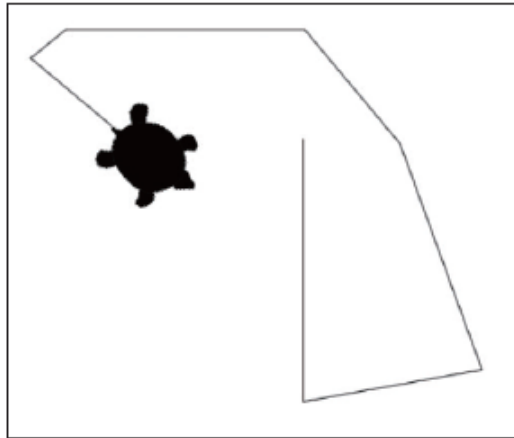
구글 Colab은 주피터 노트북 개념을 클라우드로 확장한 것이다. 구글 Colab을 이용하면 브라우저에서 파이썬 프로그램을 작성하고 실행할 수 있다. Colab 메모장을 사용하면 실행 코드와 서식 있는 텍스트를 이미지, HTML, LaTeX 등과 함께 하나의 문서로 통합할 수 있다. Colab 메모장을 만들면 Google 드라이브 계정에 저장된다. Colab 메모장을 간편하게 공유하여 동료나 친구들이 댓글을 달거나 수정하도록 할 수 있다. 자세한 설명은 유튜브 영상 "<https://www.youtube.com/watch?v=inN8seMm7UI>"을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





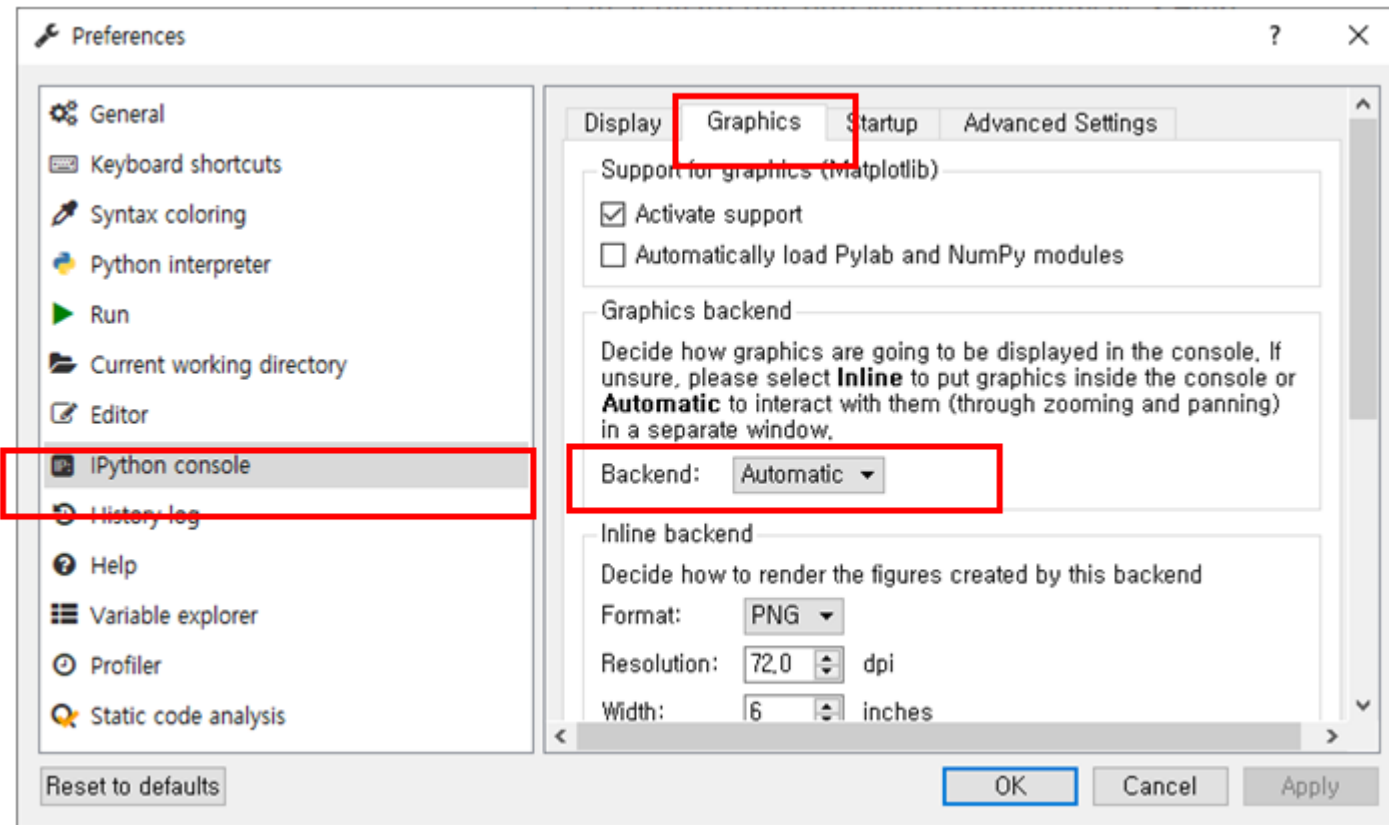
# 터틀 그래픽

- 터틀 그래픽은 화면에서 거북이를 이용하여서 그림을 그리는 기능이다.





# 스파이더 설정 변경







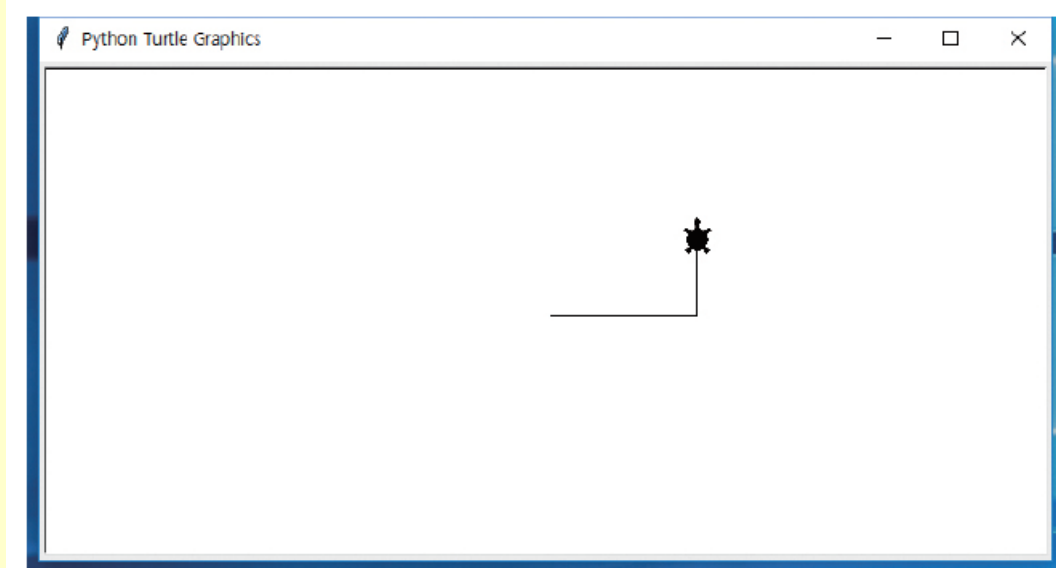
# 터틀 그래픽 시작

```
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()
```



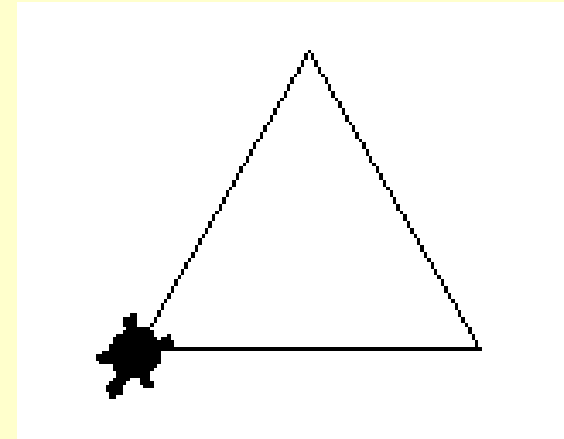


# Lab: 삼각형 그리기

```
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 무엇을 만들 수 있을까?

- 아래의 소스를 입력하고 실행해보자.

```
import turtle

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

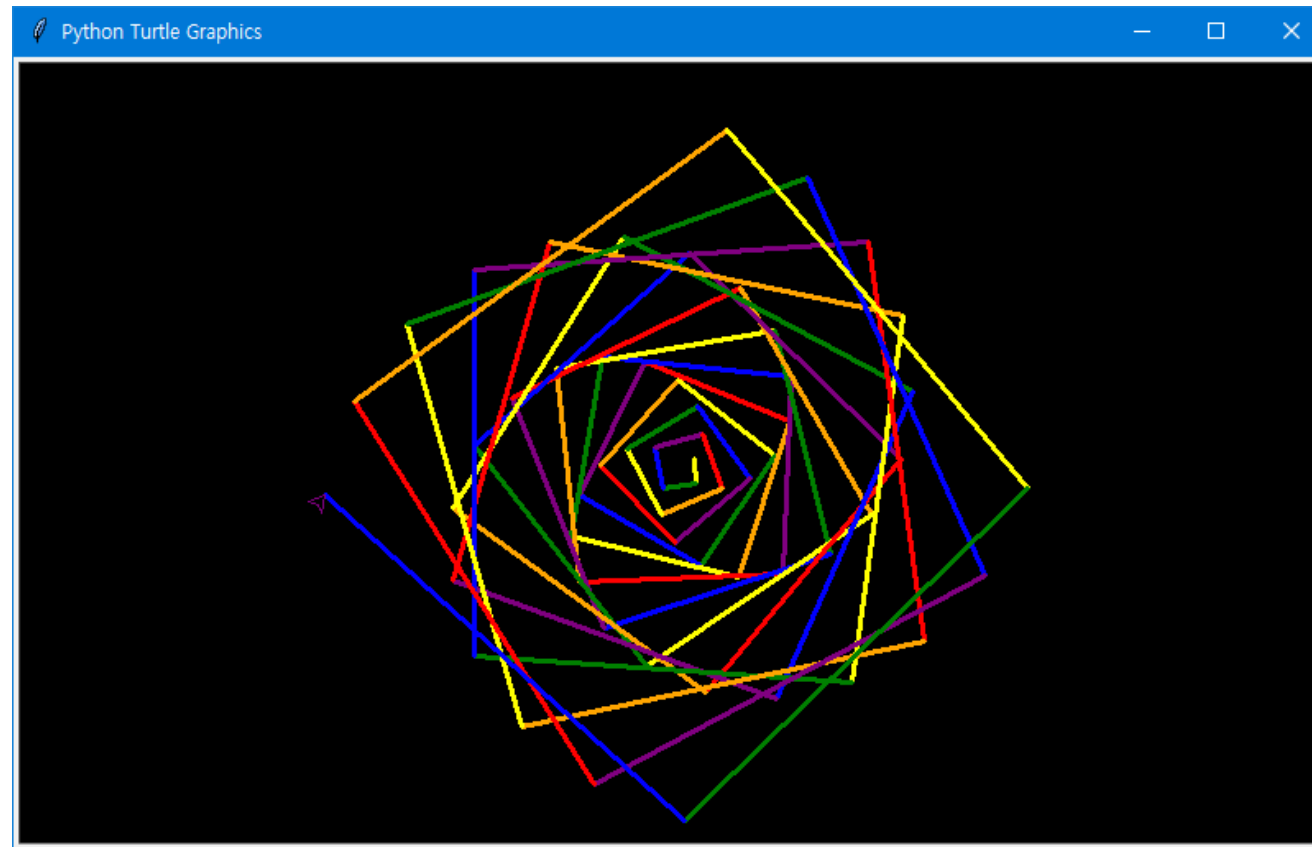
turtle.bgcolor("black")
t.speed(0)
t.width(3)
length = 10

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

- 빈칸을

```
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 `<Enter>` 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 한꺼번에 실행할 수 있다.





파이썬 소개

수고하셨습니다 😊

글로벌미디어학부 <고급프로그래밍및실습>, 이정진