

1 장 파이썬 소개

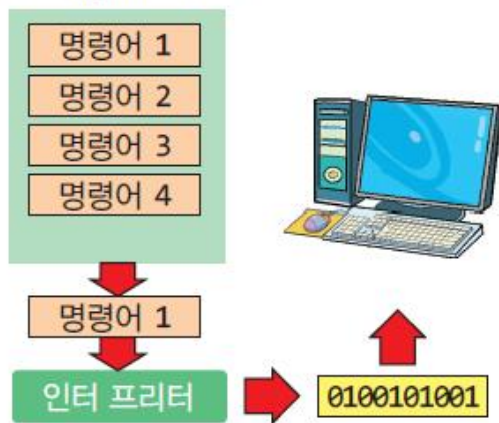




프로그램. p16

- 우리가 하고자 하는 작업을 컴퓨터에게 전달하는 역할
- 프로그램은 특정한 작업을 위한 작업 지시서
- 작업을 지시하려면 명령어(instruction)들을 나열해야 한다.

프로그램 소스 코드



프로그램 안의 명령어들이 하나씩
인터프리터에 의하여 기계어(이진수)로
변환되어서 컴퓨터에 의하여 실행됩니다.



프로그래밍 언어

- 기계어 – 컴퓨터가 바로 알아들을 수 있는 언어. 0과 1로 구성되어 있는 “001101110001010..”과 같은 이진수. 인간한테는 상당히 불편
- 프로그래밍 언어 – 기계어와 인간이 사용하는 자연어 중간쯤에 위치
- 인터프리터 - 인간이 프로그래밍 언어를 배워서 프로그램을 작성하면 컴파일러(또는 인터프리터)라고 하는 통역 소프트웨어가 프로그램을 기계어로 바꾸어 준다.



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

프로그래밍 언어



컴파일러

```
0100011010  
1010100101  
0101010101  
0010111111
```

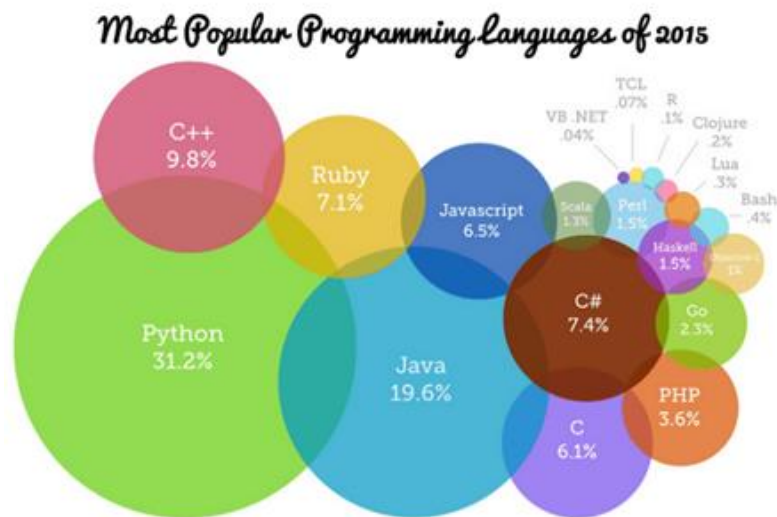
기계어





파이썬 소개. p17

- 많이 사용되는 언어들에는 '파이썬', 'Java', 'C', 'C++', 'Javascript' 등이 있다.



TIOBE
(the software quality company)

About us | Join TIOBE | News | Coding Standards | TIOBE Index | Contact

Products | Quality Models | Markets | [Schedule a demo](#)

When software quality is more than a code checker

Software quality does not arise from simply running a code checker: it requires strategy, oversight, and the right tools. TIOBE serves as a partner for organizations that want to be completely unburdened in the software quality domain. We adapt our solutions to your organization, tools and specific technologies.

TIOBE Quality Indicator

A

TIOBE Quality Models

The proverb "the proof of the pudding is in the eating" applies perfectly to

We focus on software quality.

- ✓ Quality models are based on ISO/IEC 25010.
- ✓ Fully automated software quality framework TICS.
- ✓ Supports all levels in your organization, from bit to board.
- ✓ Objective measurement of software quality.
- ✓ Checking more than 1 billion lines of code per day.

<https://www.tiobe.com/>



파이썬. p18

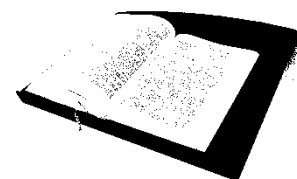
- 1991년에 귀도 반 로섬(Guido van Rossum)이 개발한 대화형 프로그래밍 언어
- 범용 프로그래밍 언어
- 생산성이 뛰어나다 – 간결하면서도 효율적인 프로그램을 빠르게 작성할 수 있다.
- 배우기가 쉽다.
- 시스템간 이식성이 뛰어나다. 동일한 파이썬 프로그램이 윈도우즈, 유닉스, 리눅스, 매킨토시 등에서 변경없이 실행된다.
- 인터프리터 언어(interpreted language)
- 기계학습, 영상 처리, 컴퓨터 비전, 통계, 데이터 시각화 등과 같은 다양한 분야의 수많은 라이브러리가 풍부하다.





컴파일러 vs 인터프리터

- 컴파일러는 책 전체를 번역하는 것과 같다.



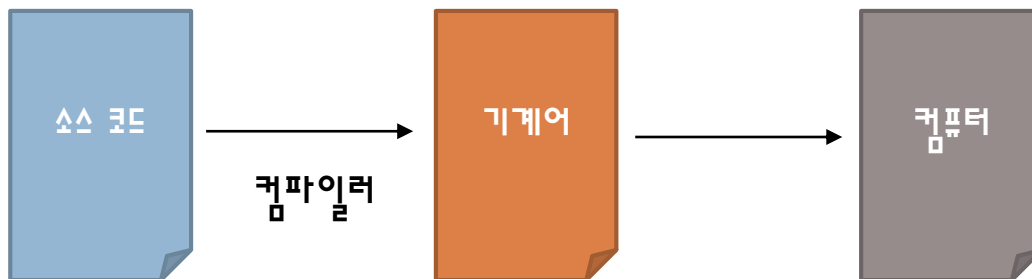
- 인터프리터는 한 문장씩 번역하는 동시 번역과 같다.





컴파일러 vs 인터프리터

- 컴파일러는 전체 코드를 전부 번역한 후에 실행한다.



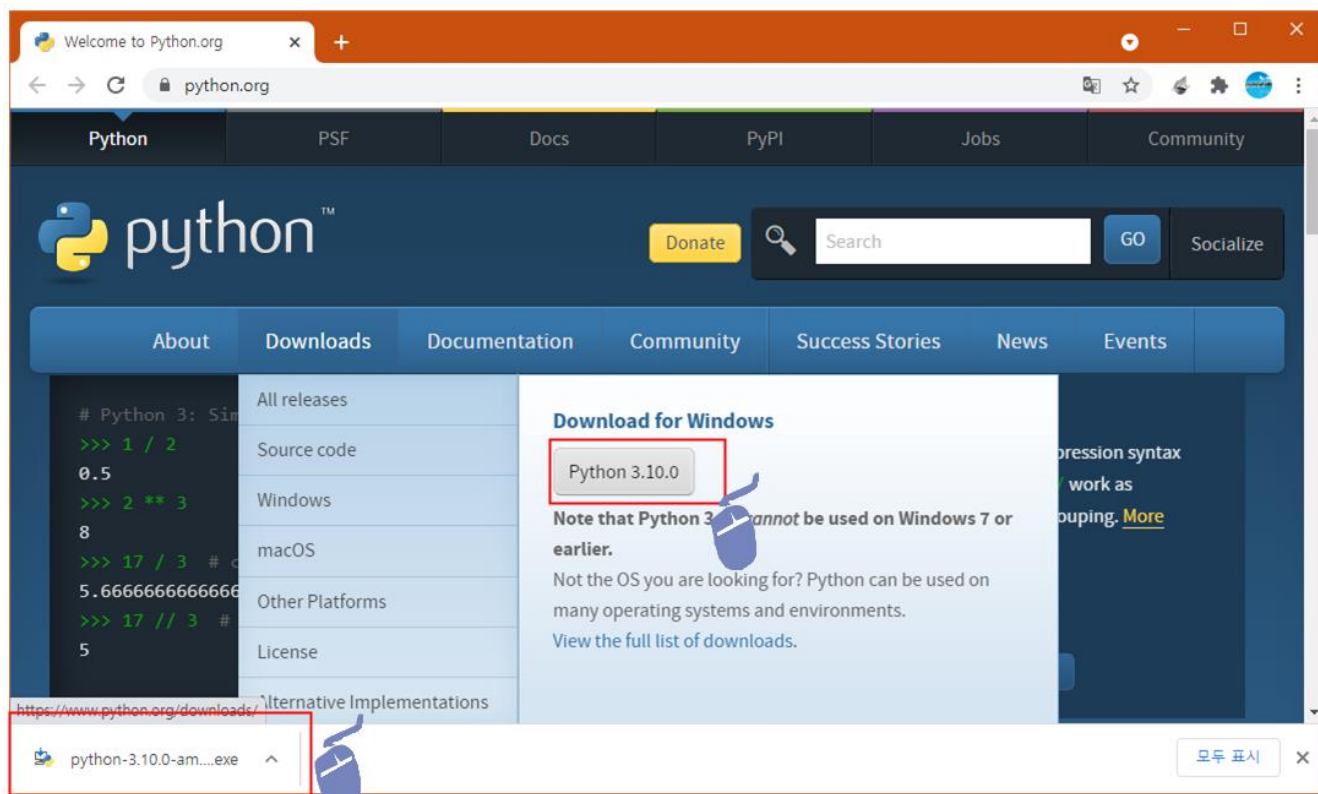
- 인터프리터는 코드 한줄 씩 기계어로 번역하여 실행한다.





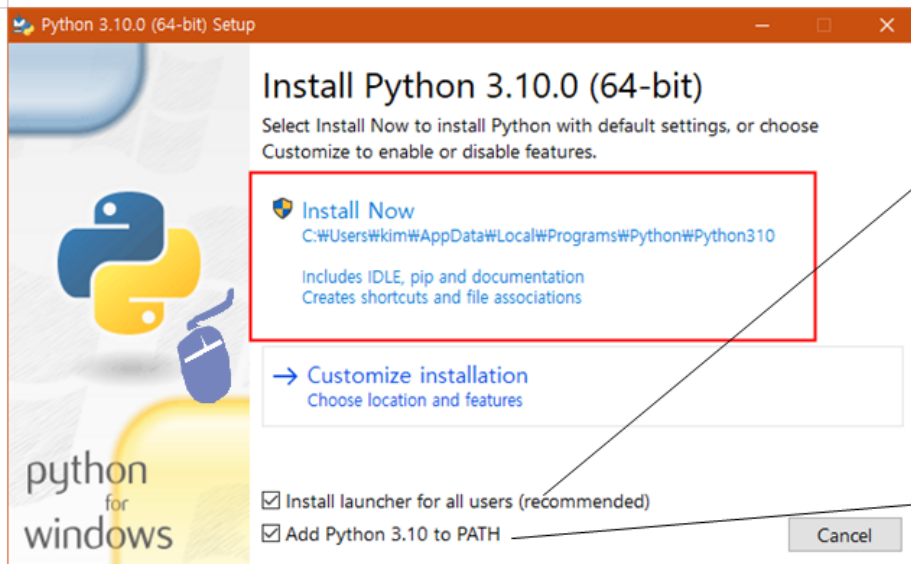
파이썬의 설치. p19

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





파이썬의 설치

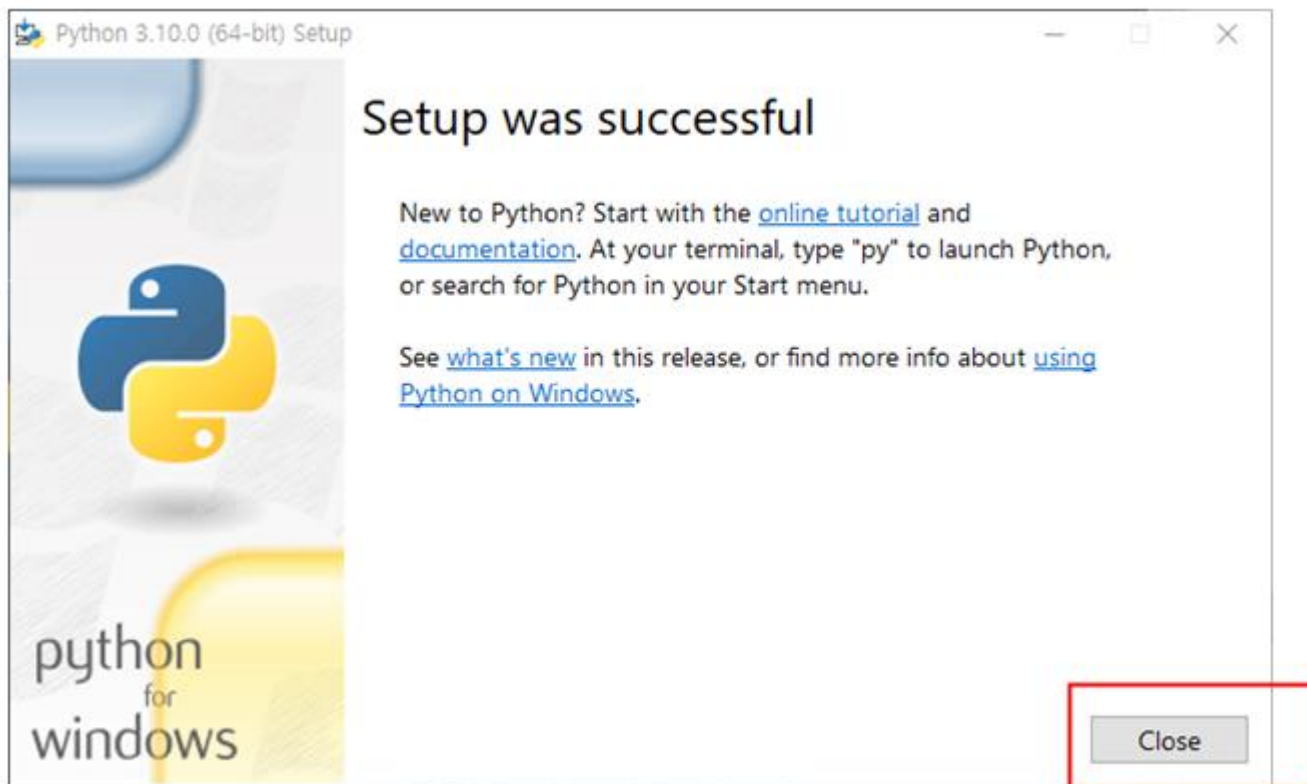


위 화면의 아래에 있는 첫 번째 체크 박스는 컴퓨터 사용자 전체가 사용하게 할 것인지, 아니면 현재의 사용자만 사용할 것인지를 묻는 것이다. “Install launcher for all users”를 체크한다.

두 번째 체크 박스 “Add Python 3.5 to PATH”는 PATH 환경 변수에 파이썬을 추가할 것인지를 묻는다. 이 부분을 설치 시 체크하지 않으면 나중에 직접 환경 변수 PATH를 변경하여야 한다. PATH에 파이썬이 들어 있어야 편리하다. 반드시 체크하도록 하자.



파이썬의 설치





파이썬의 버전. p20



참고 사항: 파이썬의 메이저 버전

파이썬을 처음 학습할 때 우리가 결정해야 하는 중요한 문제가 있다. "어떤 버전의 파이썬을 써야 할 것인가?" 이다. 파이썬에는 2.x와 3.x 버전이 있다. 문제는 이 버전들이 전혀 호환되지 않는다는 점이다. 2.x 버전 프로그램은 3.x 버전 인터프리터에서 실행되지 않는다. 따라서 개발자들은 어떤 버전을 사용해서 개발할 것인지를 미리 결정하여야 한다! 일반적으로 이전에 개발된 많은 라이브러리들을 사용하려면 2.x 버전을 사용하는 것이 좋다고 한다. 하지만 미래를 대비하는 파이썬 개발자라면 3.x 버전을 사용해야 할 것이다. 이 책에서는 3.x 버전을 사용한다.



참고 사항: 파이썬의 마이너 버전

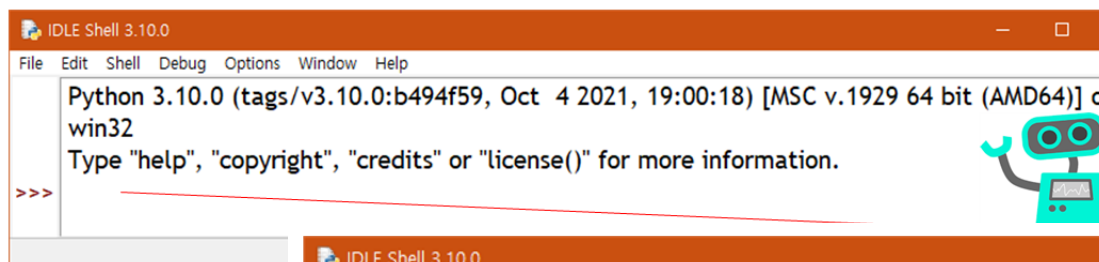
파이썬의 버전은 계속하여 업그레이드되고 있다. 이 책을 집필할 때는 파이썬의 최신 버전이 3.10.0이었지만 여러분이 1장을 읽을 때는 버전이 달라질 수 있다. 따라서 뒤에 붙는 숫자는 신경 안 써도 된다. 또 한 가지 만약 외부 라이브러리를 설치할 때, 버전에 맞는 라이브러리를 찾지 못한다는 오류가 발생하면 파이썬의 버전을 낮추어야 한다. 예를 들어서 최신 버전은 3.10이지만 3.8을 설치해야 하는 경우도 종종 발생한다. 3.8도 파이썬 홈페이지에서 여전히 제공된다.



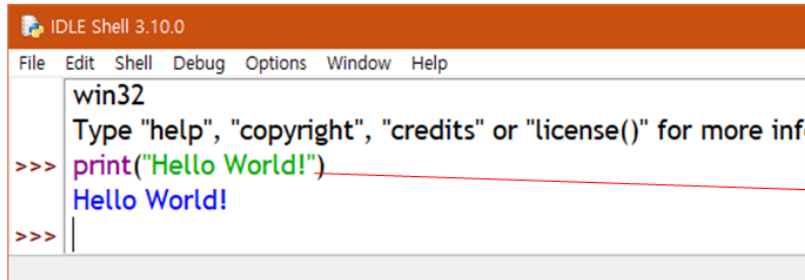
파이썬의 실행. p21



윈도우의 시작 버튼을 누르고 [모든 프로그램] -> [Python 3.10] -> [IDLE(Python 3.10 64-bit)]을 클릭한다.



명령을 기다리고 있습니다.



print("Hello World!")를 입력하고 엔터키를 누른다. 이 문장은 화면에 "Hello World!"를 출력하라는 의미이다.



첫 번째 프로그램 분석하기. p23

- 파이썬 프로그램은 여러 줄의 명령어로 이루어진다. 한 줄의 명령어를 문장(statement)라고 부른다.
- 함수(function)는 특별한 작업을 수행하는 명령어들의 모임이다.

print("Hello World!")

print() 함수는 문자열을 화면에 출력한다.

문자열은 따옴표로 둘러싸인 문자들의 모임이다. 텍스트 데이터를 나타낸다.

- 26쪽. 직접 실습해보면서 이해하자.



스크립트 모드. p26

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

대화형 모드	스크립트 모드
<pre>>>> print("Hello World!") Hello World! >>> print(10+20) 30</pre>	<pre>print("Hello World!") print(10+20)</pre> <div><p>→ 실행 결과</p><p>Hello World! 30</p></div>



대화형 모드

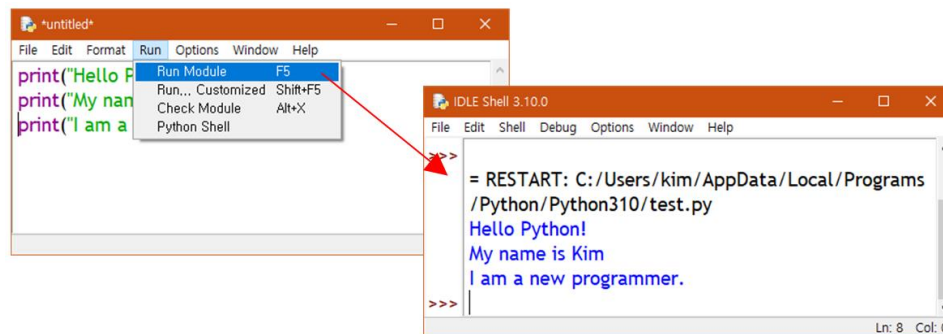
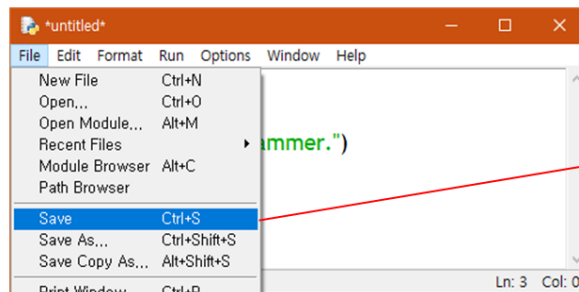
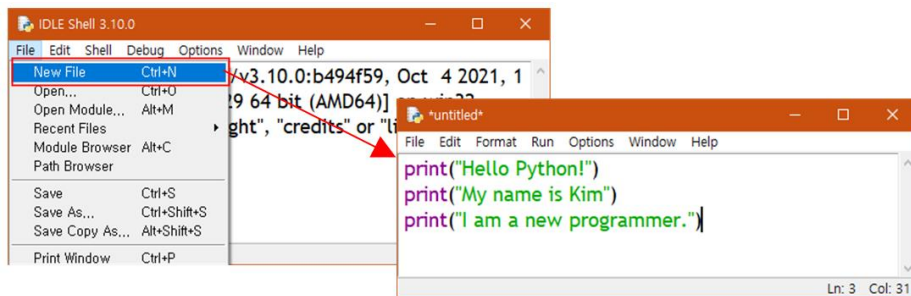


스크립트 모드



스크립트 모드

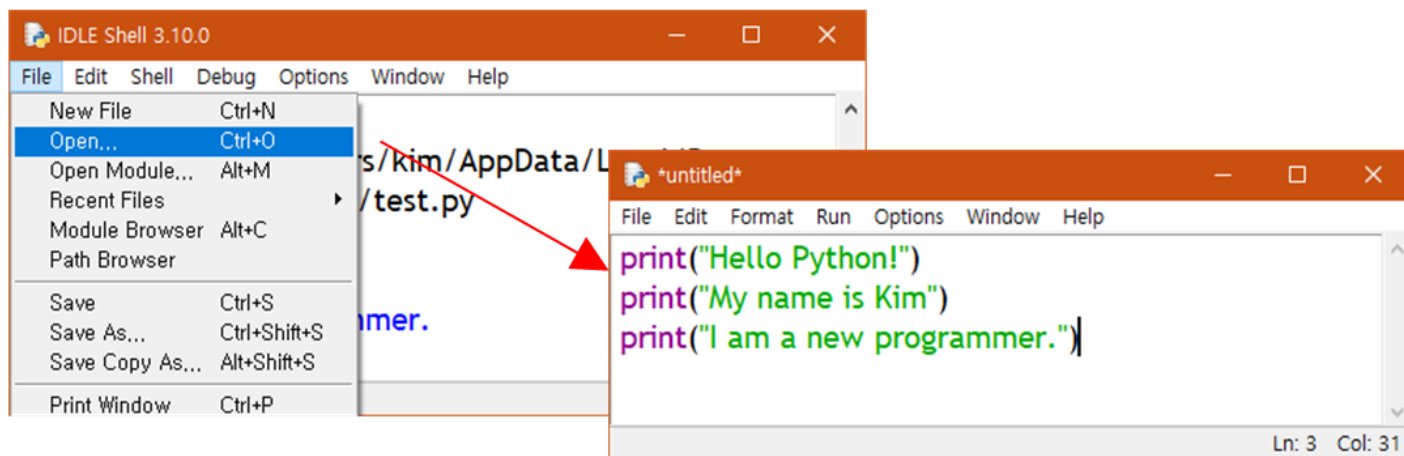
- IDLE의 [File] → [New File] 메뉴를 선택한다





소스 파일 다시 열기. p28

- 소스 파일(source file) : 코드가 저장된 파일





(추가) 교재에 없는 내용

- 파이썬 개발 도구 : Spyder, VSC(Visual Studio Code), jupyter Notebook, Atom, Pycharm
- Anaconda(아나콘다) 설치와 사용
- VSC 설치와 사용



Lab: 스크립트 모드로 계산을 해보자. p30

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



도전문제

다음 두 문장의 출력 결과를 예측해 보자. 어떤 점이 다른가?

```
print("2+3")
print(2+3)
```



Lab: 오류를 처리해보자. p31

bug.py

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

문자열인데 따옴표가 없다.

P가 대문자이다.

끝나는 따옴표가 없다.

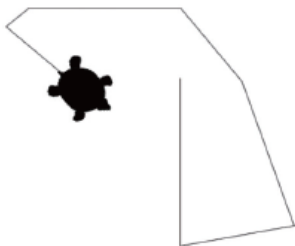
debugged.py

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



터틀 그래픽. p32

- 터틀 그래픽(turtle Graphic) : 화면에서 거북이를 이용하여서 그림을 그리는 기능이다.



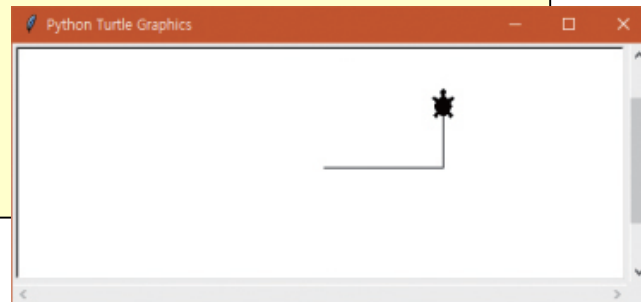
터틀 그래픽은 거북이가 화면 위에서 지나가면서 그림을 그리는 것이라고 생각하세요.



```
import turtle                # 터틀 모듈 포함
t = turtle.Turtle()          # 터틀 생성
t.shape("turtle")            # 터틀 모양을 거북이로 설정

t.forward(100)               # 100 픽셀만큼 전진
t.left(90)                   # 90도 회전
t.forward(50)

turtle.done()                # 터틀 그래픽 종료
```





터틀 그래픽. p34

- 픽셀(pixel) : 컴퓨터 화면을 이루는 점을 뜻한다. picture element
- 터틀 그래픽을 명령어를 단축하여서 쓸 수 있다. forward는 fd, left는 lt, right는 rt를 적어도 된다.

r.fd(100)

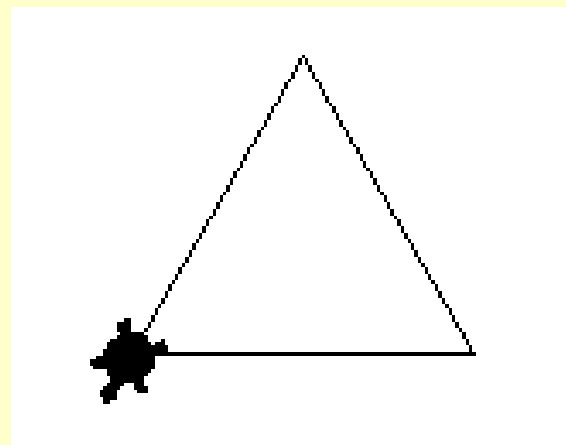
t.lt(90)

t.td(100)



Lab: 터틀 그래픽으로 삼각형을 그려보자. p35

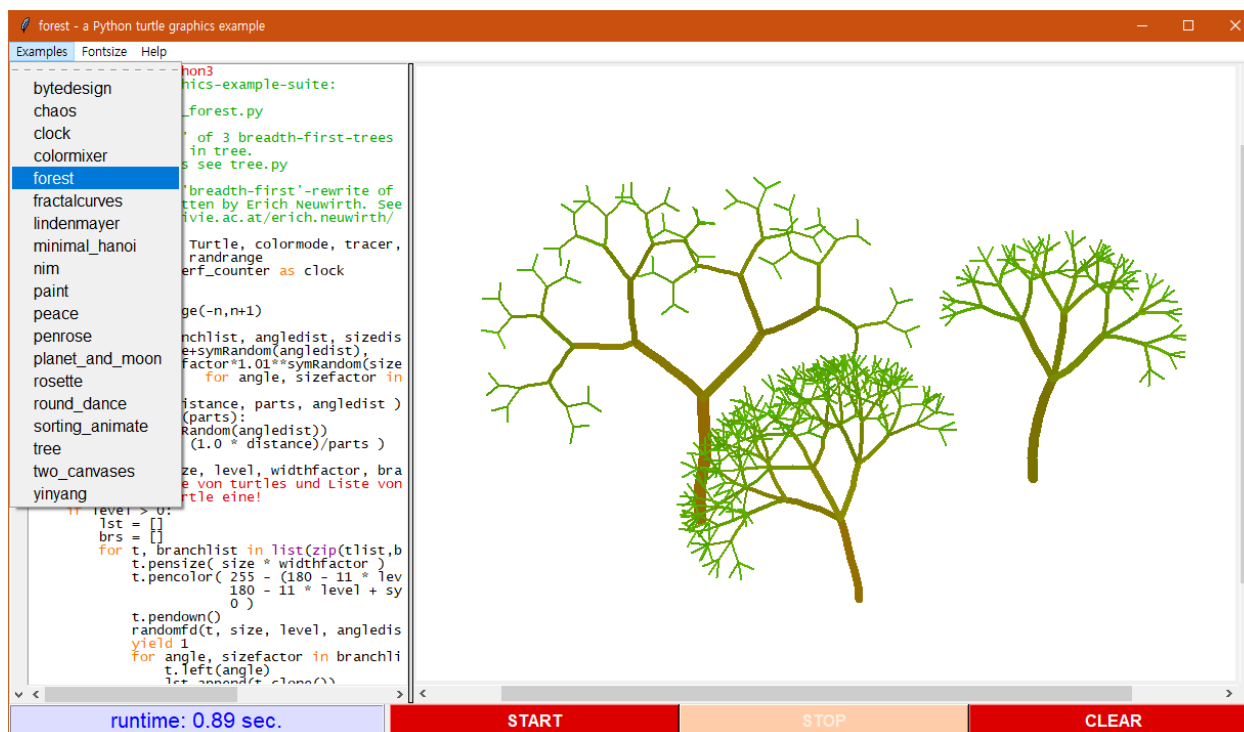
```
import turtle  
t = turtle.Turtle()  
  
t.shape("turtle")  
t.forward(100)  
t.left(120)  
t.forward(100)  
t.left(120)  
t.forward(100)  
  
turtle.done()
```





Lab: 예제 프로그램 실행해보기. p36

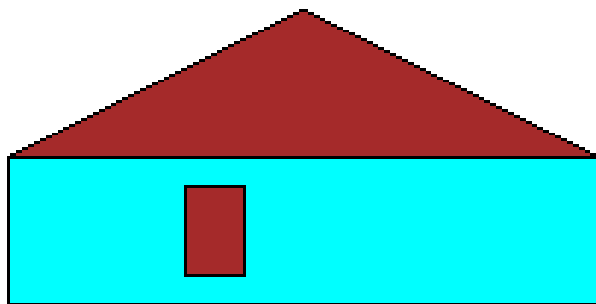
- IDLE에서 [Help]->[Turtle Demo]를 누르면 다음과 같은 터틀 그래픽 예제들이 등장한다. [Examples]에서 예제를 선택하고 화면 하단의 [START] 버튼을 눌러본다.





Mini Project: 터틀 그래픽으로 집 그리기. p37

- 터틀 그래픽으로 다음과 같은 집을 그려보자.



- 미완성 코드 -> 완성시키자.



연습문제. p39



Programming. p41

