

# 00.Intro

## ◆ 파이썬이란?

### ● 사전적인 의미

- 독이 없는 뱀으로 먹이를 몸으로 감아 압사시키는 큰 뱀
- 그리스 신화에 나오는 악마

### ● 프로그래밍 언어

- 1991년 프로그래머인 귀도 반 로섬(Guido van Rossum)이 발표한 고급 프로그래밍 언어
- 플랫폼에 독립적이며 인터프리터식, 객체 지향적, 동적 타이핑 대화형 언어
- 파이썬이라는 이름은 귀도가 좋아하는 코미디 <Monty Python's Flying Circus>에서 따옴

- 파이썬은 프로그래밍 언어 의 일종

. 컴퓨터가 알아 들을수 있도록 컴퓨터에게 개발자의 명령을 전달하는 통역가. ( JAVA , C# , JavaScript, C ..... )

. 오지의 원주민 에게 우리의 뜻을 전달 하려는데

통역가가 한국인 이면 우리가 한국어 문법을 알아야 하고 , 미국인 이면 영어문법을 , 중국인 이면 중국어 문법을 알아야 한다

- 컴퓨터 가 알아 들을 수 있는 명령어 들을 작성 하고 컴퓨터 에게 지시 하여 수행 할 수 있도록 하는 언어

- 컴퓨터 가 알아 들을 수 있는 명령어로 해석 해야 하므로 프로그래밍 언어 라고 하며 문법을 배워 컴퓨터에게 올바른 지시를 하려고 하는것.

## 파이썬 의 특징

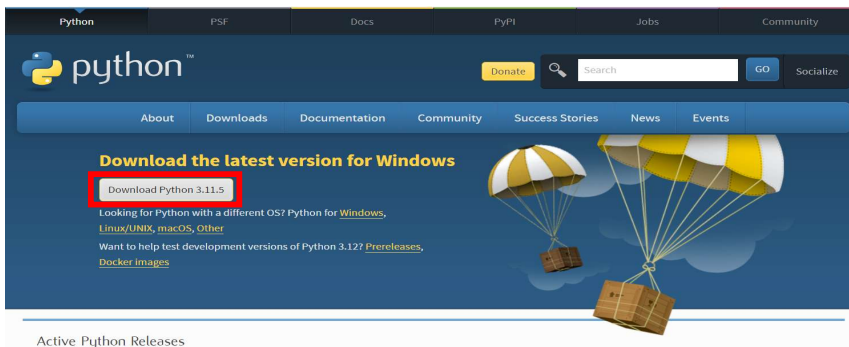
- 배우기 쉽고 사용하기 쉽다. 개발기간이 비교적 짧다.
- 운영체제에 구애 없이 사용할 수 있으며 이기종 통신 (맥, 리눅스.. 여러가지 프로그램 환경 ) 이 자유롭다
- 실무에 필요한 기능들이 많이 개발 되어 배포되고 있어 손쉽게 다양하고 복잡한 기능들을 사용 하여 개발을 완료 할 수 있다.
- 객체 지향적이며 클래스 를 지원한다.
- C 언어와의 호환성이 좋아 혼합하여 사용하기 용이하다.
- 비교적 문법이 쉽고 개발 과정이 단순 해 프로그래밍을 입문하는 입문자에게 접근성이 좋다
- 작은 단위의 프로젝트에는 성능이 효과적이나 대규모 프로젝트의 메인 언어로 는 다소 효율성이 떨어진다.

## 파이썬 시작 하기

- Python 의 설치

. <https://www.python.org/downloads/>

> 파이썬 다운로드

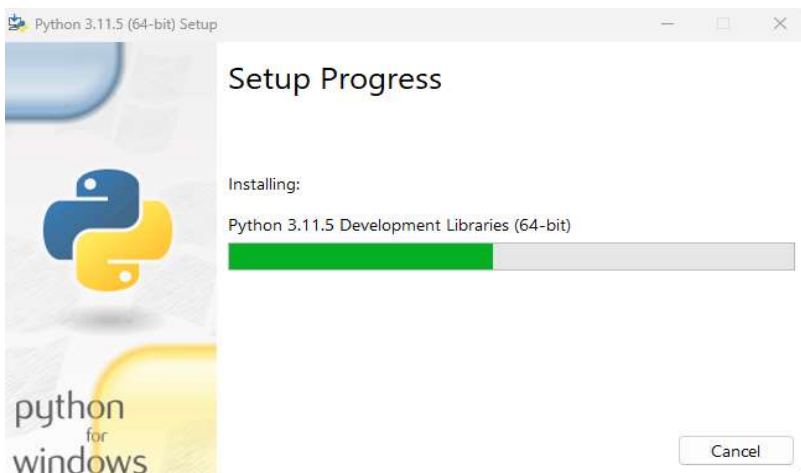


> Install Now

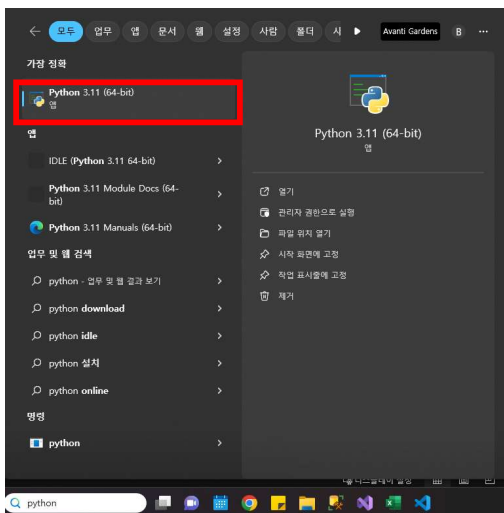
\* Add Python.exe to PATH 체크



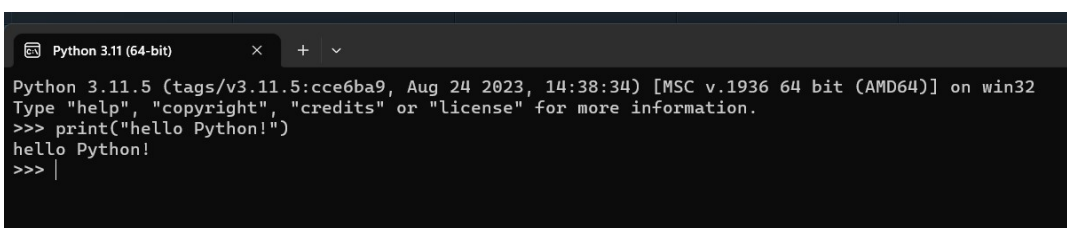
> Install



. 설치 확인 및 실행



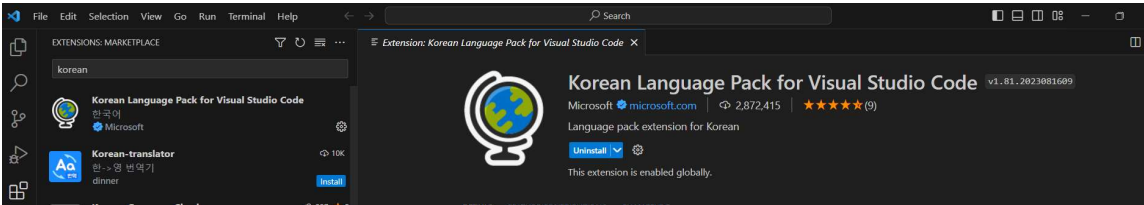
> 파이썬이 정상적으로 동작 하는것을 확인



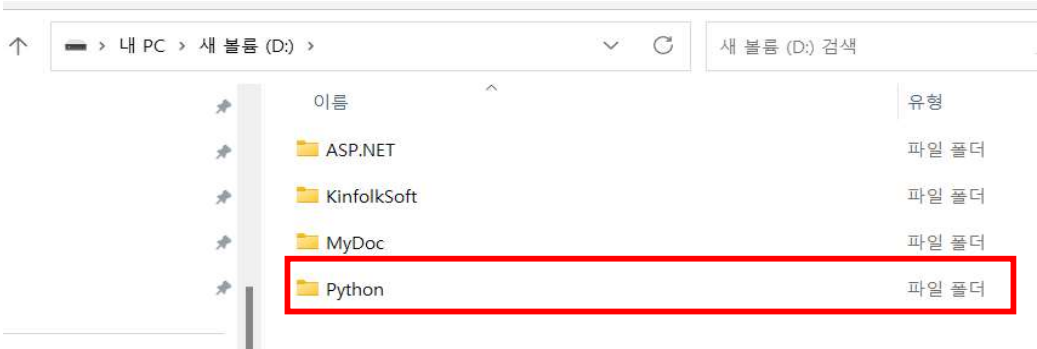
- VisualStudio Code 의 설치
  - . Python 을 효율적으로 프로그래밍 할 수 있도록 도와주는 도구
  - > <https://visualstudio.microsoft.com/ko/downloads/>



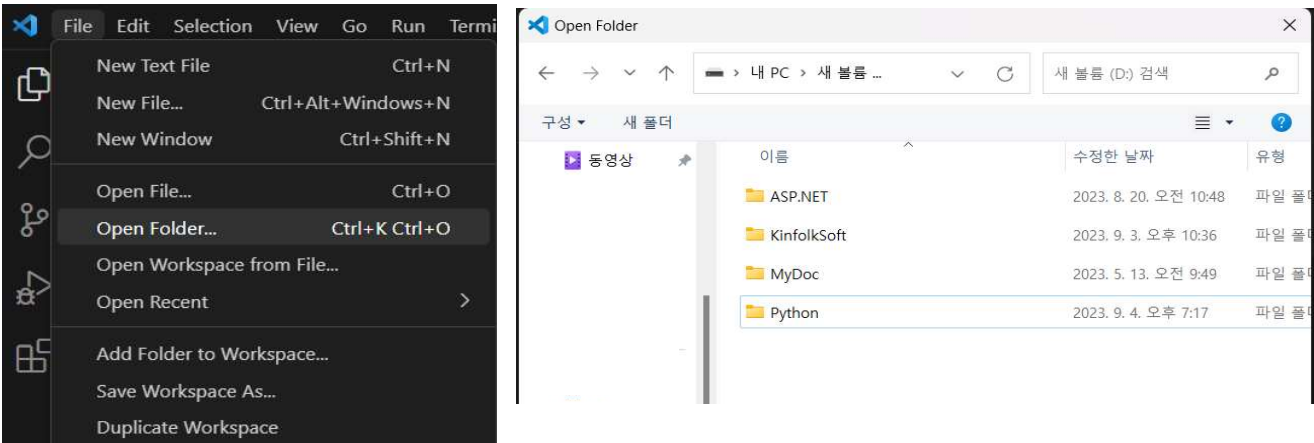
- . VSCode 의 장점
  - 1) 수많은 확장성을 가진 라이브러리 ( 개발을 도와주는 기능성 프로그램 패키지 ) 가 많아 개발을 쉽게 도와준다.
  - 2) 코드 작성의 자동완성 및 문법 오류 검출 (인텔리전스) 기능 으로 인한 개발 효율성 증가
  - 3) 중단점 지정 을 통한 디버깅( 오류 검출 작업 ) 이 용이
  - > Korean Language 선택 가능 (필수 사항 아님)




- 프로그램 소스를 담을 폴더의 생성
- . Python 폴더 생성
  - > 소스 폴더의 경로는 한글 또는 공백을 사용하지 않는것을 추천

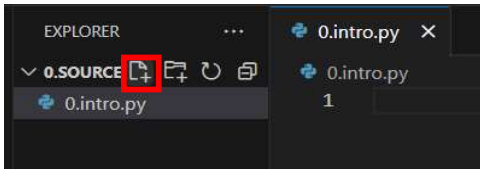


- . Python 폴더 Open
  - > File -> Open Folder




. Python 파일 생성

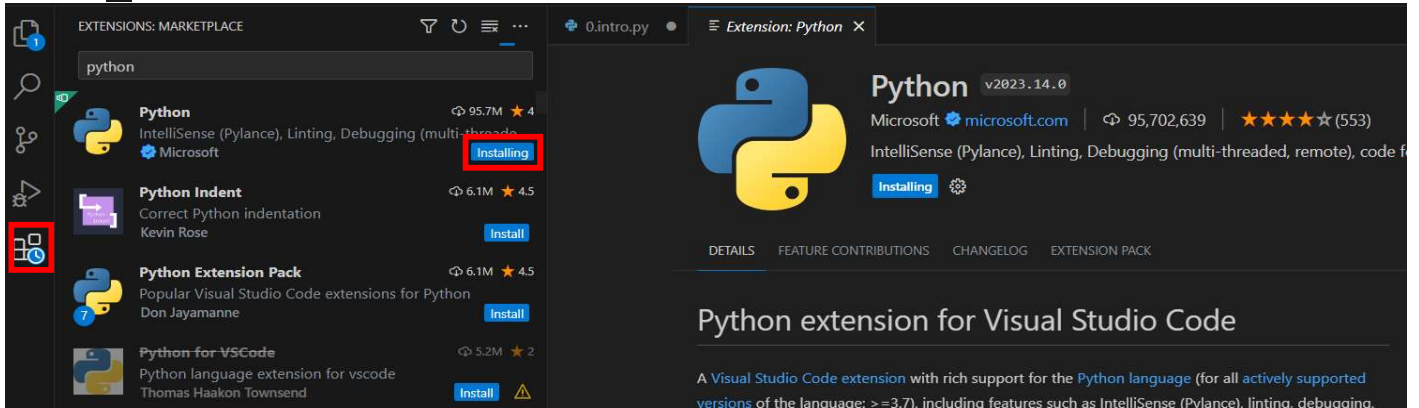
> 새 파일 만들기  클릭 후 0.intro.py 입력 후 엔터



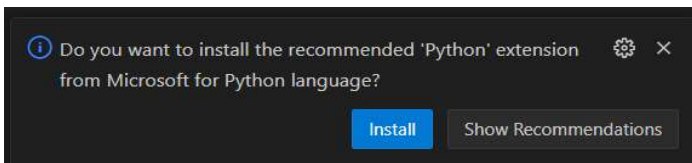
- Python Intellisense 설치

. Python 을 쉽게 개발 할 수 있도록 도와 주는 기능이 모여있는 응용 패키지

. 좌측의  를 클릭 후 Python 입력 및 install



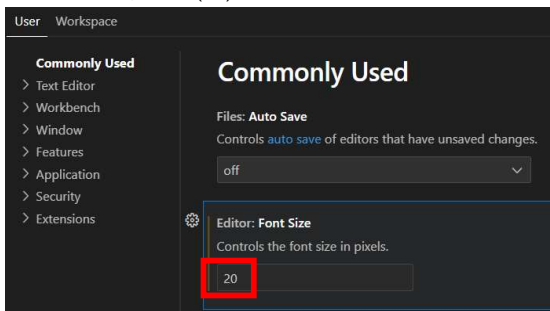
. Recommend Extension Install (필수 사항은 아님)



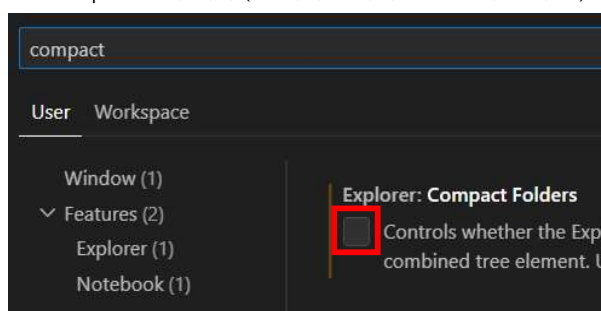
- 간단한 개발 환경 설정

. Ctr + ,

> 폰트 크기 변경 (20)

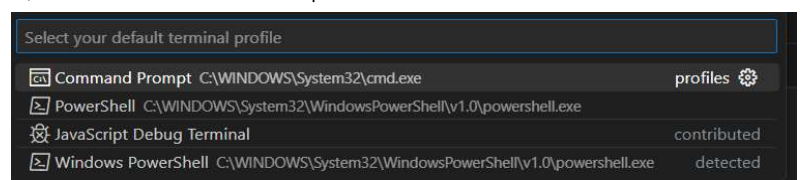
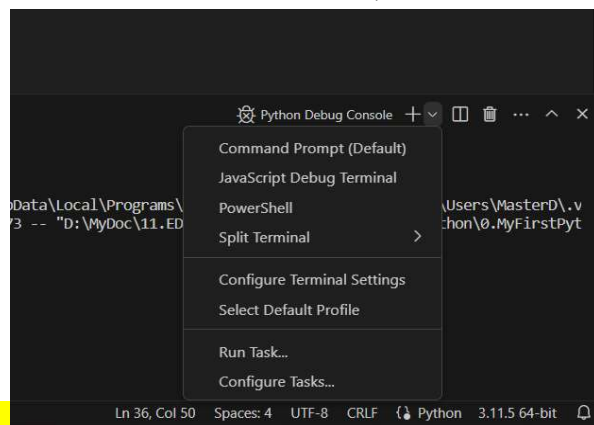




> Compact 표시 해제 ( 탐색기 폴더의 경로 폴더 형태로 )

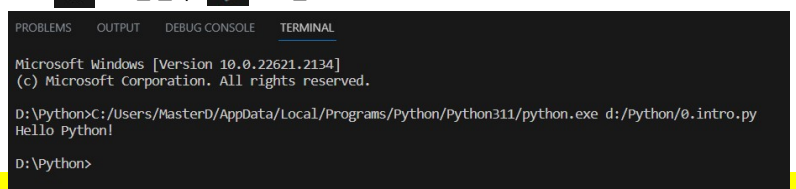


> Terminal 의 Default Profile 설정 (F5 실행 후 실행 창에서 선택)

> Command Prompt 선택



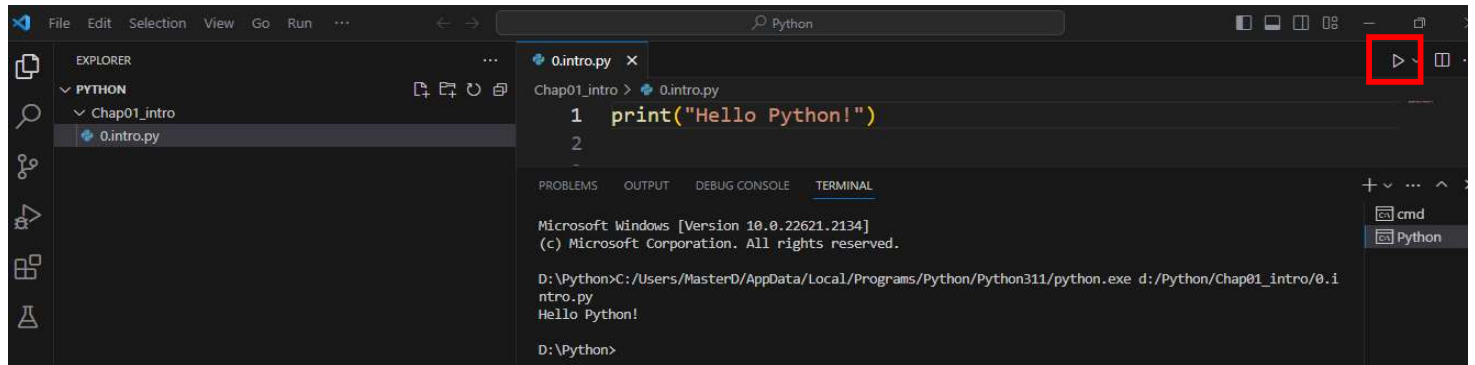
>  로 닫은 후  또는



프로그램을 설치하고 환경을 구축해 보시다.

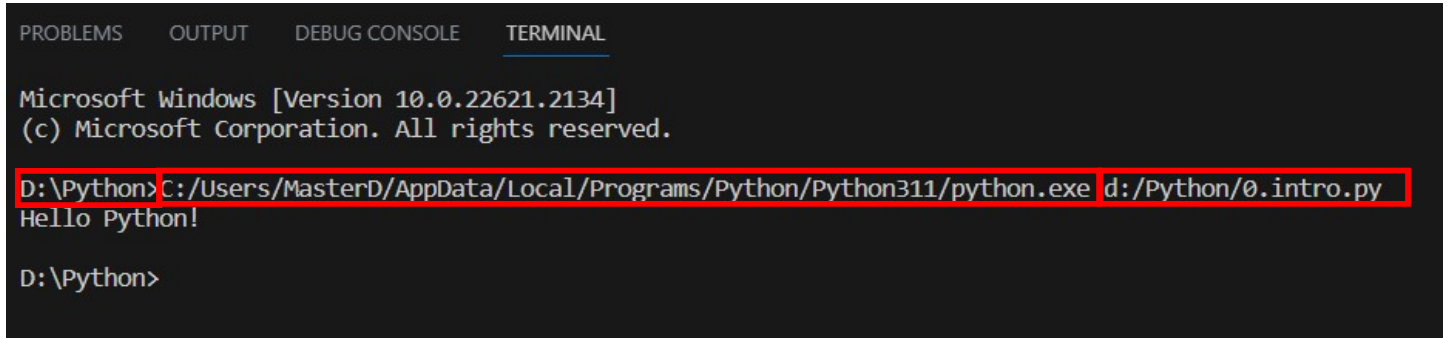
### 기초 문법

- Chap01.intro 폴더 추가
- . 탐색기 에서 오른쪽 마우스 클릭 -> New Folder 선택후 Chap01\_intor 생성 및 0.intro.py 파일 이동

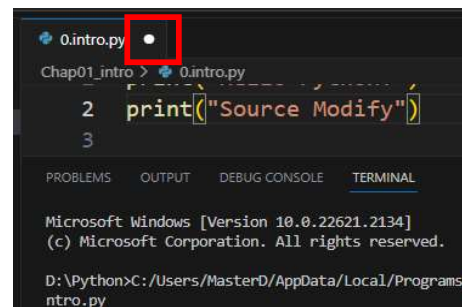


### Command 창 의 실행 결과 살펴보기

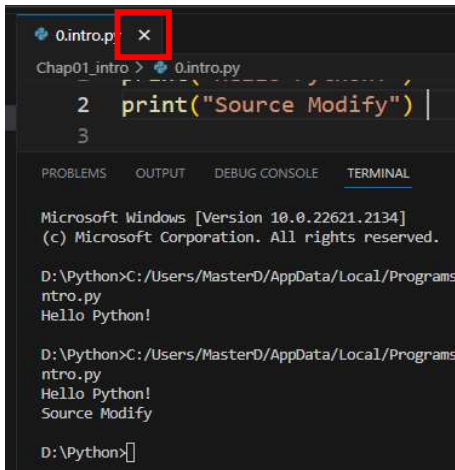
- Command 의 Terminal 실행 결과
- . D:\Python> : 경로에서
- . C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe : 파이썬 프로그램이
- . d:/Python/0.intro.py : 경로에 있는 0.intro 파이썬 모듈을 실행한다



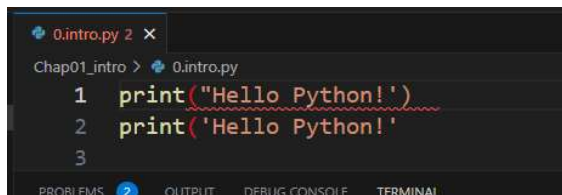
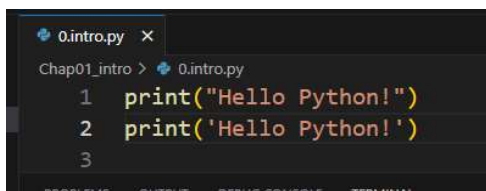
- 소스 의 수정
- . 수정 중일 경우 탭의 제목 우측 ● 로 표현



- . 저장 또는 실행 시 x 로 변경



- 쌍따옴표와 홑따옴표
- . 쌍 따옴표 와 홑 따옴표 로 문자열 지정 가능
- \* 쌍따옴표 쌍 이 맞지 않거나 괄호의 쌍이 맞지 않으면 오류가 발생함





```
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>
```

```
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python
ntro.py
File "d:\Python\Chap01_intro\0.intro.py", line 1
    print("Hello Python!")
    ^
SyntaxError: unterminated string literal (detected at line 1)

D:\Python>
```

## 코드 의 실행 순서

- 위쪽에서 아래로 오른쪽에서 왼쪽으로 순차적으로 실행

```
print("Hello Python1!")
print("Hello Python2!")
print("Hello Python3!")
age2 = age = 11;
```

## 코드 블록 설정

- Python 의 코드 단위
- . 코드 블록 : 코드를 실행하는 최소 단위로서 코드의 묶음을 나타낸다.
- > 들여쓰기가 되어있는 구문은 if 명령어 이후 한 프로그램의 실행 단위 묶음

```
0.intro.py x
Chap01_intro > 0.intro.py > ...
1 age = 11;
2 if age > 10:
3     print("Hello Python1!")
4     print("Hello Python2!")
5     print("Hello Python3!")
6
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python
ntro.py
Hello Python1!
Hello Python2!
Hello Python3!

D:\Python>
```

> 3 개의 코드 블록 이 구성 되어있는 예제

```
0.intro.py •
Chap01_intro > 0.intro.py > ...
1 age = 11;
2
3 if age > 10:
4     print("Hello Python1!")
5     print("Hello Python2!")
6
7 print("Hello Python3!")
8
```

\* 코드 블록 if 절 안쪽으로 배치된 구문

```
print("Hello Python1!")
print("Hello Python2!")
```

은 age 가 10 이하라면 출력 되지 않는다.

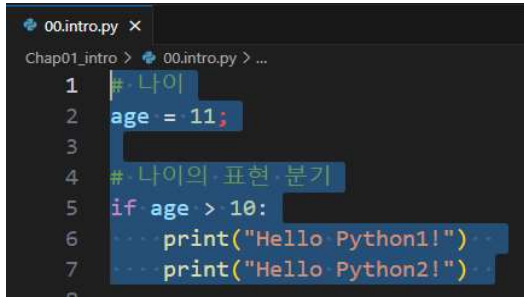
## 주석

- 개발자 의 생각을 나타내는 글
- . 컴퓨터는 알아보지 못하며 실행을 하지 않는다.
- \*\*\* 프로그래밍 도중 **개발자의 의도를 기록**으로 남기거나 **변수또는 함수 에 대한 설명**을 할때 사용한다. (개발자에게 중요한 습관)
- > # 을 통해 주석을 표현 한다.

```
00.intro.py x
Chap01_intro > 00.intro.py > ...
1 # 나이
2 age = 11;
3
4 # 나이의 표현 분기
5 if age > 10:
6     print("Hello Python1!")
7     print("Hello Python2!")
8
```

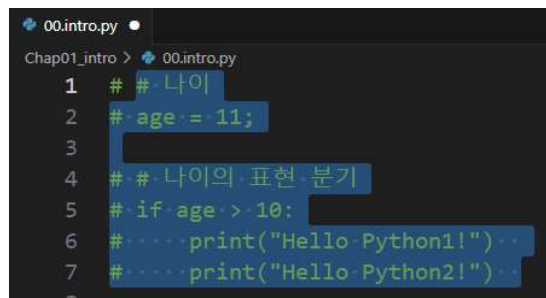
- 여러 줄의 주석

. 주석 처리 코드 선택



```
00.intro.py x
Chap01_intro > 00.intro.py > ...
1 # 나이
2 age = 11;
3
4 # 나이의 표현 분기
5 if age > 10:
6     print("Hello Python1!")
7     print("Hello Python2!")
```

. Shift + / 로 다중 행 주석 처리 및 해제 (Ctrl + k + c , ctrl + k + u)



```
00.intro.py
Chap01_intro > 00.intro.py
1 # # 나이
2 # age = 11;
3
4 # # 나이의 표현 분기
5 # if age > 10:
6 #     print("Hello Python1!")
7 #     print("Hello Python2!")
```