

Python

<https://wikidocs.net/book/1>

소스 프로그램(원시 프로그램)→ 번역(인터프리터, 컴파일러)→ 실행→ 기계어→ 연계편집→ 로드 모듈(로드)→ 메모리(주기억장치) 실행

<컴퓨터 언어의 종류>

인터프리터 언어

컴파일러 언어

저장되어 있는 실행가능한 프로그램 -> 메모리(주기억장치)에서 실행

하드디스크는 프로그램이나 데이터를 저장하는 공간

로드 모듈(실행가능한 프로그램을 메모리로 가져가서 실행하는 것을 의미)

소스 프로그램 언어를 번역하여 기계어로 만들어야 함.

기계어로 만들어주기 위한 프로그램을 목적 프로그램이라고 함.

기계어로 만들어놓은 것을 연계 편집(linkage edit)함.

연계 편집을 하면 실행가능한 로드 모듈(*.exe, *.com 등 실행 가능한 파일)을 만듦

인터프리터 프로그램 바로 번역하여 실행함. 대신 한 번만 빠름

컴파일러 언어는 전체 번역을 하여 기계어로 만들어야 되기 때문에 단점이지만 한 번 만들어 놓고 바로 실행하면 빠름.

소스 프로그램의 값을 수정하면 인터프리터 언어는 바로 실행되지만 컴파일러 언어는 기계어를 다시 만들어야 함

많은 것들을 할 때는 인터프리터언어로 만듦

시장반응이나 피드백 등을 볼 때는 인터프리터 언어로 만든 후 컴파일링함.

프로토타입을 만들때는 파이썬같은 인터프리터 언어가 좋음.

안정적으로 유지 할때는 자바 같은 컴파일러 언어를 사용.

python 설치(Version에 따라 다르지만, 환경변수 추가해야 함 아래같이)

add python 3.10 to path -> install now



설치 후 close

`print('hi~')` 또는 `("hi~")` : 'hi~'가 출력됨

<vscode 한글>

`ctrl+shift+p` → configure language선택 -> other languages 한국어 선택
확장들어가서 python쓰고 위쪽 3개 다운로드 설치

powershell : Linux 터미널 같이 윈도우에서도 사용 가능하게 만듦

운영체제의 구성 : 하드웨어, 커널, 셸, 응용프로그램, 사용자

powershell 과 Linux의 차이

PS가 앞에 붙어있음 색상도 다름

컴퓨터에서 처리하기 위해 필요한 것을 만드는 것을 프로그램이라 지칭

다른 프로그램에 비해 python은 문법이 쉬운 편이라 빠르게 배울 수 있음

C언어 : 프로그램 언어 중 빠른편, 기계 친화적 언어

가장 빠른 프로그램 언어 : 어셈블리 언어

python 명령어의 대부분은 소문자로 사용

개발자 iOS, Linux 계열을 많이씀

들여쓰기와 대소문자 구분 잘해야 함.

`ctrl+l` = clear

`ctrl+k+s` = 바로가기 키 설정

바로가기 키 설정 Run Python → Ctrl+Shift+F9 엔터 입력

```
if a > 1:
```

```
    print("a is greater than 1")
```

:표시 if 안의 조건문을 나타냄

: 주석문 나타낼 때 사용(1줄단위)

"""(열고 닫기) 또는 '''(열고 닫기) : 여러 줄 주석

<VSCode 단축키>

블록지정 후

Ctrl+K+C = 주석처리

Ctrl+K+U = 주석해제

※ D2coding, 나눔코딩 폰트 → 코딩용 폰트

for 반복문 : 리스트 안에서 반복

while 반복문 : 초기값과 증가값 내에서 반복

함수 내가 만들고 불러쓰면 됨

def로 python내에서 사용자가 함수를 만들 수 있음

Character -> 문자 ex) 'a', 'b'

String -> 문자열 ex) 'apple'

Interger -> 정수

Flat -> 실수

이 것을 자료형 type이라고 함

C언어에서는 문자와 문자열을 별개로 취급한다.

python은 String, Charater 구분하지 않음

type 자료형을 나타내는 문법

예제) 사과가 1260개 있다. 1Box에는 사과 25개가 들어간다 필요한 Box수와 나머지는?

1260 // 25 = 50(몫)

1260 % 25 = 10(나머지)

<python에서 수를 나타내는 형태>

0o74 : 8진수, 0xAB : 16진수, 0b1011 : 2진수, 100 : 10진수

#, \, \\$, \ \, \ : 이스케이프 문자

\n : 개행문자(줄을 바꾸어주는 이스케이프 문자)
\r, \f

<특수문자의 명칭>

?, !, @(앳) #(크로스해치), \$, %, ^(서컴플렉스), &(앰퍼샌드), ~(틸드) *(아스터리스크), -(하이픈)
※ && : And를 나타냄

윈도우에서 리눅스 환경 조작하기

WSL : 윈도우에서 리눅스를 설치하여 사용하게 해 주는 것

※ vEthernet : wsl 랜카드(vmware 쓸 때 충돌되면 끄기)

a() : 함수

a[] : Index 번호

필요한 것을 뽑아낼 때 쓰는 것이 슬라이싱

■ 포매팅(포맷 스트링)

%d(숫자 대입), %s(문자 대입)

"문자열".format()

format : 품을 맞춰줌

<인덱스>

print("I ate {0} apples. so I was sick for {1} days." .format(number, day))

<인덱스 정렬하기>

< : 왼쪽 정렬

> : 오른쪽 정렬

^ : 가운데 정렬

<공백 채우기>

print("{0:*<10}" .format("hi"))

print("{0:?>10}" .format("hi"))

print("{0:=^10}" .format("hi"))