수집 – 관리 – 분석

computer System

언어마다 주소에 접근 하는게 다름

적합한 환경이 다르기 때문에 환경에 맞게 언어를 사용해야함

안드로이드는 플랫폼

임베디드 시스템은 cpu랑 memory를 원칩(1 chip)으로 만드는거 – 비용 때문에

Soc – System on chip. 단일 칩에 모든 기능이 집적된 집적회로

용도가 정해져 있음. PC 같은 거랑 다름

O/S->CPU/Memory?Device 제어

OS가 없으면 메모리주소, 장치 지정 다 해줘야함.

OS없는 상황을 펌웨어

OS를 잘다루면 시스템프로그래머

컴파일 형 언어는 한꺼번에 바이너리로 바꿔서 사용.. 통상적으로 속도가 빠름

Scrpit (Interpreter)언어는 문장 바로바로 실행하는거 컴파일 과정이 없음. 언어가 단순함 동시 통역

ex) HTML-컴파일러 따로 필용벗이 웹브라우저만 있으면됨.

파이썬-

작업속도 높고, 생산성의 향상, 유지보수 비용이 절감, 프로토타입을 만들기 좋다

프로토타입을 만들면서 개발하는걸 TDD(Test Driven Development)

확장성이 뛰어나서 다른 언어로 만든거 가져다가 쓸 수 있음

동적 자료형 지원

변수 – 기억장소

1 기억장소 크기,

2. 저장되는 값 형태

3. 저장방식

4. 값의 범위

C, JAVA는 정적 자료형 컴파일 할 때 ? 정해짐

파이썬은 런타임(실행) 때 결정됨-동적자료형

컴파일형 언어는 CPU에 맞춰서 컴파일? 됨. 바이너리 호환성이 없음

윈도우에서 컴파일한거를(실행파일?) 리눅스에서 하면실행안됨

자바는 가상머신이 있어서 OS에?CPU에?플랫폼에 영향 안받음

파이썬은 다 기본자료형:????????????????

stack-지역적

data-전역적

heap-동적, 실행시 생성/별도의 소멸 동작 필요

파이썬은 전부 heap에 만들고 메모리 소멸은 가비지콜랙터가 소멸시켜줌. 메모리관리 생각 할 필요 없음

파이썬은 라이브러리가 많다

라이브러리- 함수나 클래스 집합을 라이브러리라고 함

내가 만들지 않아도 다른 사람이 만들어 놓은 함수나 클래스가 많다

GPL은 오픈소스에서 많이 쓰는 라이센스

소스 공개 의무가 있음.

파이썬은 펌웨어 프로그램은 못함. OS에 종속된 시스템프로그래밍도 못함. 앱프로그램도 못함

우리가 보통 사용하는건 C 파이썬 – 씨언어로 작성된 컴파일러를 씀?

파이썬인터프리터 – 공식 사이트에서 배포, or 배포판 다운받아서 쓰기

배포판에서 유명한거 0아나콘다

파이썬은 3버전 2버전 있음

2버전은 앞으로 업그레이드 안함 근데 나온지 오래되서 라이브러리 많음

3버전은

2버전과 3버전은 호환이 안됨

리눅스에는 2버전이 기본 설치되어 있음

3버전 쓰려면 다시 다운받아야함

64비트에서 32비트 os 사용 가능 근데 속소

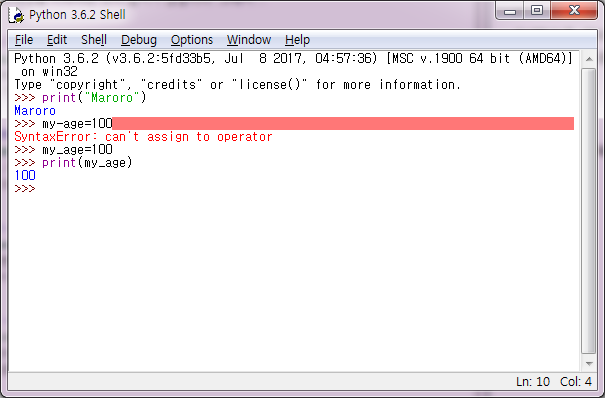
인터프리터 실행시키는 유형은 두가지

GUI형식 / CLI 형식

CLI는 Command Line Interface(명령 기반 인터페이스, 도스형식)

GUI방식은 IDLE 작성하고 컴파일하고 결과 확인하고 할 때 유리

CLI는 명령 프롬프트에서 해도 되고 python3.6으로 해도됨



이게 GUI

>>> 이거는 프롬프트

cmd에서 path에서 Python추가된거 확인하고

python쓰면 실행됨 CLImode

종료 할 때는 ctrl+z 하고 엔터

아니면 바로 python 아이콘 실행시켜져도됨

Python Program 작성 형태

1. Interpreter에서 code입력/실행
2. 별도의 Editor Program으로 Python Script file 작성 후 실행

IDE (총합 개발 환경)

ex) eclipse

IDLE-file-new file

확장자는 .py

소스편집한거 위에 보면 run-run module

cmd에서 py파일(스크립트) 있는데 가서 python myname.py

이런식으로 실행해도 됨

C기본구조

#include<stdio.h>

int main(){

return 0;

}

JAVA기본구조

class Test{

pulic static void main(String []a)

{

}

}

이렇게 구조가 있는데 파이썬은 무조건 위에서 아래로 차례대로 실행함

별도의 시작 함수가 필요없음. C나 JAVA는 필요함

문장 끝을 지정하는 기호가 없음

콜론 다음에는 스페이스나 탭으로 빈공간을 줘야함 indentation

일정한 간격만 유지하면됨. 같은 함수 내에서만

예를 들어 스페이스 세번 눌렀으면

다음에도 세번눌러야함

코드 다음부터가 코드 불럭이다(C나 JAVA에서 {})

파이썬은 모든 문자가 유니코드

아스키는 1바이트 유니코드는 2바이트

한글로 변수 만들 수 있기는 하지만 영어로 쓰는거를 권장

\n 같은거는 못씀

예약어들 있음

알아내는 방법은

>>> importkeyword

>>> keyword.kwlist

>>> len(keyword.kwlist)

import : 모듈 추가. 외부 라이브러리(모듈) 추가 할 때

keyword가 모듈명

주석은 #주석내용요요요요용

\쓰면 다음 줄이랑 이어짐

표준 입출력 – 장치를 지정하지 않아도 자동으로 입출력 받아오는?

input함수는 무조건 문자열로 리턴 함

a=input()

줄바꿈이 기본적으로 일어남

end=’ ‘ 라면 줄바꿈 알일어남

sep=’;:주면 구분자로 :이 생김

print( ‘{0:<10}{1:5.2f}’.format( ‘apple’, 7.77 ) )

왼쪽 꺽쇠는 왼쪽 정렬 10자리

format메소드 동작 결과

format은 프린트 함수 안에서 써야함

출력형식 지정 때는 format쓰면 됨

>>> print(format(1.234567,"10.3f"))

1.235

>>> print('{0}{1}'.format('apple',7.7))

apple7.7

>>> print('{0}'.format('apple',7.7))

apple

>>> print('{0:<10}{1:5.2f}'.format('apple',7.77))

apple 7.77

>>> print('{0:>10}{1:5.2f}'.format('apple',7.77))

apple 7.77

>>> print('{0:>10}{1:4.2f}'.format('apple',7.77))

apple7.77

>>> '{0:=^10}'.format('hi')

'====hi===='

>>> 'Name:{0},Phone:{1}'.format("hong",5432)

'Name:hong,Phone:5432'

>>> '{0:=^10}{1:<10}'.format('hi','apple')

'====hi====apple '

>>> '{0:=^10}{1:=<10}'.format('hi','apple')

'====hi====apple====='

>>> '{{and}}'.format()

'{and}'

pprint

>>> import pprint

>>> numbers=[[1,2,3],[4,5],[6,7,8,9]]

>>> print(numbers)

[[1, 2, 3], [4, 5], [6, 7, 8, 9]]

>>> print(\*numbers)

[1, 2, 3] [4, 5] [6, 7, 8, 9]

>>> print(\*numbers,sep='\n')

[1, 2, 3]

[4, 5]

[6, 7, 8, 9]

>>> pprint.pprint(numbers)

[[1, 2, 3], [4, 5], [6, 7, 8, 9]]

>>> pprint.pprint(numbers,width=20)

[[1, 2, 3],

[4, 5],

[6, 7, 8, 9]]

>>> pprint.pprint(numbers,width=20,indent=4)

[ [1, 2, 3],

[4, 5],

[6, 7, 8, 9]]

list, dict, tuple, set은 다른 언어에 없는 자료형. 자료구조?

dirct형은 데이터를 직접 표현하는 자료형 int, float, complex

seqeuence형은 다른 데이터를 포현하는 자료형 집합. 순서가 있음 list, str, tuple,bytes,bytearray,range

Mapping형은 순서가 없음. dict

set형은 집합형 중복도 없는 자료형 set, frozenset

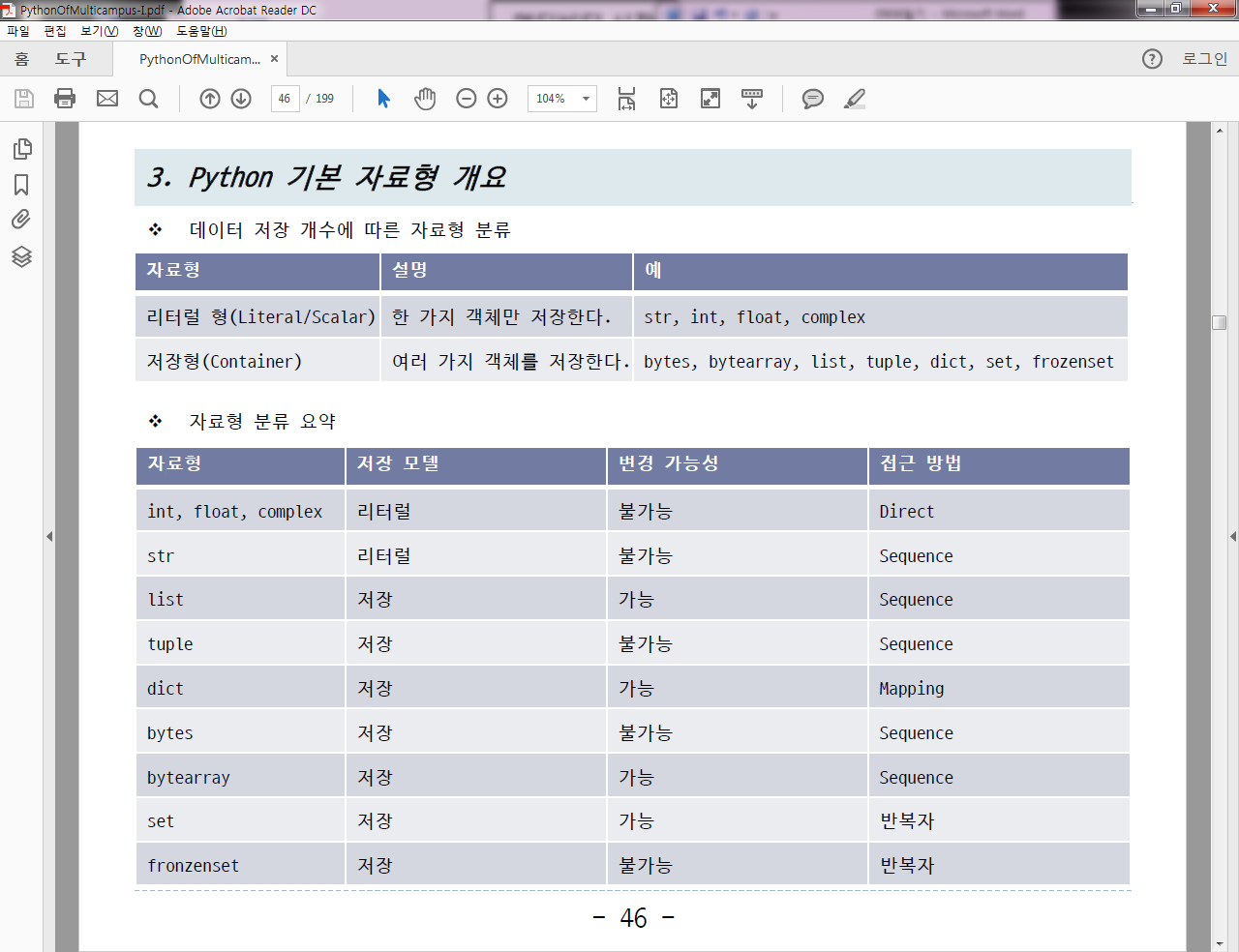
변경 가는 한 자료형 Mutable => list, dict, set, bytearray 데이터의 값을 변경 할 수 있음

변경 불가능=> 데이터 값 변경 불가 int float등

데이터 저장 개수

리터럴 형 – 한가지 객체만 저장 –str,int,float,complex

저장형 – container 여러가지 객체를 저장 bytes, bytearray, list, tuple, dict, set,frozenset



파이썬은 모든 내용이 다 객체

a=1

a=2

이렇게 해도 원래 대로면 에러나야함 int는 변경 불가능이니깐

근데 에러 안나는 이유는

a도 객체 1도 객체 2도 객체

a는 1이라는 객체의 레퍼런스만 갖음

레퍼런스는 포인터

파이썬에서는 각 객체마다 레퍼런스 카운트가 가지고 있음

레퍼런싱 할 때 마다 카운트가 증가하고 0이 되면 가비지 컬렉터가 객체를 제거함

>>> x=y=z=500

>>> print(x,y,z)

500 500 500

>>> id(x)

49108592

>>> id(y)

49108592

>>> id(z)

49108592

500의 레퍼런스를 기억하고 있는거

id는 레퍼런스 값 확인하는거

변경불가능 가능은 레퍼런스가 가르키는 대상의 데이터를 바꿀 수 없냐 있냐. 변경 가능한 타입은 내용까지 바꿀 수 잇음

새로운 객체가 생기는거

is로 비교하면 레퍼런스로 비교

==로 비교하면 레퍼런시가 가지고 있는 대상의 내용 비교

bool은 참이냐 거짓이냐, 비어있느냐 이런거 비어있으면

내장함수는 .뭐뭐뭐 없이 쓰는거 점이 없어야

bytes타입은 아스키 코드 써야할 떄 1바이트로 됨 b로 시작해야함 a=b’Python rules’ 이런식

bytes타입은 변경 불가능한 타입

변경하고 싶을 때는 bytearray사용

>>> b=b'Python rules'

>>> b[1:5]

b'ytho'

>>> b'th'in b

True

>>> b.upper()

b'PYTHON RULES'

>>> b.split()

[b'Python', b'rules']

>>> b.startswith(b'Py')

True

>>> b.endswith(b'id')

False

>>> b=b'Python rules';b.decode()

'Python rules'

>>> b.decode('cp949')

'Python rules'

>>> s='홍길동';

>>> s.encode()

b'\xed\x99\x8d\xea\xb8\xb8\xeb\x8f\x99'

>>> b.decode()

'Python rules'

>>> b.decode('utf8')

'Python rules'

b’th’ in a은 내용이 포함되어 있는지 a에 th가

decode 바이트 타입을 문자열로 형변환 utf8이 문자에 따라서 1~4바이트까지 cp949는 윈도우에서 쓰는 encode규칙

encode는 문자열을 바이트 타입으로(코드 값)

>>> b=b'Python rules'

>>> ba=bytearray(b)

>>> ba[7]=ord('R')

>>> ba

bytearray(b'Python Rules')

>>> bytes(ba)

b'Python Rules'

>>> ba.decode()

'Python Rules'

ord는 코드값으로 바꿔주는거

decode하면 유니코드 문자열로 바뀐거

파이썬은 값의 범위가 없음

인스턴스는 메모리에 생성된 객체

높은 자리 값을 높은자리에 저장할거냐 낮은데 저장할거냥 big endian littler endian

endian은 memory에 data 저장 방식 높은 자리를 낮은 번지에 저장 헐건지

네트워크에선s big endian

일반적으로는 little endian

arm코어 같은거는 선택 할 수 잇음

(1024).to\_bytes( 2, byteorder= ‘big’ )

반대로 하려면

int.from\_bytes( b’\x04\x00’, byteorder= ‘big’ )

>>> import sys

>>> sys.float\_info

sys.float\_info(max=1.7976931348623157e+308, max\_exp=1024, max\_10\_exp=308, min=2.2250738585072014e-308, min\_exp=-1021, min\_10\_exp=-307, dig=15, mant\_dig=53, epsilon=2.220446049250313e-16, radix=2, rounds=1)

>>> sys.float\_info.max

1.7976931348623157e+308

실수의 최대 최소 값 확인

float(‘inf’)

//는 몫만 계산 /는 나누기 - 소수점까지 divmod는 몫과 나머지

파이썬은 증감연산자(++,--)가 없음 ++는 그냥 부호

문자열 관계연산자로 비교하면 길이로 비교하는게 아니라 abc 사전순으로 비교함

논리연산자는 x and y에서 true/false가 나오는게 아니라 x가 참이면 y의 결과가 나옴

x or y에서 x가 참이면 x 아니면 y계산

None은 거짓

파이썬은 제어문 안에 제어문(for)를 넣을 수 잇음

끝에가 seq형 자료형 이어야함. 순서가 있는 배열

ditc이 우선순위가 제일 높음

71p 연산자 우선순위

수치형? 반복가능????

1. 속도와 거리를 입력받아 시간을 계산하여 출력하는 프로그램 거리.속도
2. 길이와 너비를 입력받아 면적과 둘레를 께산하여 출력하는 프로그램
3. 화씨 온도를 입력받아 섭씨온도로 변환하는 프로그램 섭씨=(화씨-32)/1.8
4. 두수를 입력받아 덧셈, 뺄셈, 곱셈, 몫, 나머지를 출력하는 프로그램
5. 두수를 입력받아 최대값과 최소값을 출력하는 프로그램