Declartive knowledge (선언적 지식)

: 무엇이 어떻다는 식의 지식. 사실을 아는 것처럼 명확한 지식

Imperative knowledge (명령적 지식)

: 과정적 지식이나 방법적 지식으로 불리기도 하는데 학습한 결과를 기억x, 결과가 성립되는 과정을 증명하는 등 수행에 있어서의 과정이나, 단계, 절차에 대해 아는 것을 의미

Fixed program computers (프로그램 고정 컴퓨터)

: 컴퓨터 상에서 실행되는 명령어 사전에 결정되어 있으며, 실행은 자동적으로 행해지지만 프로그램의 변경은 불가능한 것

Ex) Atanasoff and Berry : a linear equation solver

Alan Truing : bombe machine

Calculator

Stored program computers (프로그램 내장식 컴퓨터)

: 컴퓨터 자신이 가지고 있는 내부 기억 장치에 프로그램을 기억시켜두고 이 프로그램에 의해서 제어되는 방식의 컴퓨터. 프로그램의 변경, 반복 자유롭게

Ex) Von Neumann machine

Computation is solving a problem with a program

Program is a realization of an algorithm on a computer

Algorithm is a finite sequence of instructions(imperative knowledge) to do a task

Algorithm should be refined enough to be easily translated into a program using a program language

Coding – a process of fighting with bugs

Top-down design, Divide and Conquer, Multi-level abstraction

( Decomposing a problem into smaller sub-problems )

처치 튜링 명제 (Turing-Church thesis)

: 인간이 계산할 수 있는 것은 기계도 계산할 수 있다.

: modern computers are essentially equivalent to a Turing machine(stored program computer)

D – T, UT ( 알고리즘을 추적할 수 있나 없나 )

UD { There exist problems that stored program computers cannot solve }

D UD ( 알고리즘을 만들 수 있나 없나)

Instruction set

Arithmetic, logical, relation, operator

Assignment, iteration, conditional

Input & output

Execution beginas at the first statement. Statements are executed one by one, top to bottom

Function definitions do not change the flow of execution but only define a function

Function calls are like detours in the flow of execution

Stepwise refinement

1. Start with a primitive program that solves a simple problem.
2. Make a small changes, one at a time to generalize the program
3. Make sure that each change does not invalidate what you have done before.
4. Add appropriate comments
5. Choose descriptive name

Simple object, complex object – user defined object!

Type(Robot()) – ‘cs1robots.Robot()

Type(load\_picture(“photos/geowi.jpg”)) – cs1media.Picture

The same name can be binded to different objects(of different types) in a program, e.g.

반대로 하나의 name이 여러 object를 지칭할 수 있다!!

Objects provide methods to perform these actions.

Expression (정의) : a combination of operators with operands including objects(values), variables and function calls

Function is a mapping from one set to another set

Positional Correspondence

A parameter is a local variable, that is, the name of an object which can only be recognized inside a function

Predicate functions : functions returning a True or False value

A variable is local if its scope is restricted inside a function : It appears when a function is called, exists during execution of the function, and disappears when the function is terminated

Variables defined outside of all functions are called global variables.

Tuple : an ordered sequence of data elements. It is an immutable object.

Tuple([2,4])

List[(2,4)]

[2, 4]

1. Parameter는 1개이다!
2. 형 변환의 형태이다!

Ex) [2]인 리스트를 만들고 싶으면 아무리 parameter가 한 개여도 list(2)이거 안된다!

List1 = [2] 이거나 list((2,))이렇게 해주어야!

dict또한 마찬가지! Dict([2,4], [3,6]) 이렇게 하면 에러 par가 1개이므로

dict([[2,4], [3,6]]) 이렇게 가야한다!

Slicing! 무조건 뒤에거 기준!

Tuple[1:3] – [1], [2]

뒤에거가 없을때만 앞에 거 기준

Tuple[1:] – [2], [3], …

Concatenation! – tuple1 + tuple2 = (1, 2, 3, 4) 가능! 쌉가능!

Slicing과 concatenation은 list, string도 동일!

String : an ordered sequence of characters. It is an immutable object.

List : an ordered sequence of data elements. It is a mutable.

Range(a, b, c) : a에서 b까지 c간격으로..

List!!의 method

Append(x), insert(x, y), remove(x), pop(y), sort(), reverse()

이외의 생소한 method들!

List.index(x) – x의 index값 산출, List.count(x) – x가 몇 개 있는지 산출, List.extend(L) – L은 모든 squence들인데 이 ‘요소’들을 다 리스트에 넣어라!

sort 의 경우 return값이 따로 존재하지 않기에!! 저장해줄 값을 바로 갖지 않고 sort라는 함수에서 따로 return을 취해주어야 한다!

Mdals = medals.sort(x, y..) 이런식으로 안된다는 소리

함수를 따로 선언해서 medals.sort()해주고 return medals를 해주어야 저 저장이 가능하다!

reverse()도 마찬가지이다!

반면 pop은 바로 return을 갖기에 gas = nobles.pop(1) 이런식으로 바로 저장이 가능하다!

list1 = ['a', 'b', 'c'] list자체가 par가 될 수 있는 내재되어 있는 세가지의 함수. max, min sum

print(min(list1))

print(max(list1))

print(sum(list1))

a == b : 동등하냐? a is b : 일치하냐? ( tuple, list의 경우 요소들이 같으면 동등하나 일치하지는 않지만, string의 경우는 둘 다 일치로 간주 )

List1 = [a. b, c]

List2 = List1 : 둘은 일치한다

List1 = [a, b, c]

List2 = [a, b, c]

List1 != List2 : 둘은 일치하지 않다. 다만 동등할뿐…

String, tuples, and lists all represents sequences of elements – 이 말은 참이다. 괜히 string때문에 혼동 nono

In fact, string is not really data structure.

Dictionary : a collection of items(or elements). Key is a value of any immutable! Type. And used as an infex of the item. 순서가 없는게 중요 특징!

Dict의 pop의 경우 해당 key의 value값을 뽑아서 다른 값에 저장

Ex) num = dict1.pop(“nine”)

Print(num) – 9

Dict의 몇가지 method – dict.keys() : key값들 리턴, dict.values() - value리턴, dict.items() – (key,value)리턴

Named parameter = Default parameter 라고도 한다.

By using the names of parameters in a function call, the order of arguments does not matter!

File!

Readline() 줄 하나 읽어

Read() 다 읽어 read(28) ; 길이 28까지 다 읽어! 그냥 순수하게 읽어

Readlines() 다 읽어! 단, 한 리스트를 구성해서! Planet.txt를 예로 들면,

[‘mercury\n’, ‘venus\n’, ‘earth\n’ … ] 이런 식으로

file에서 “a”는 추가하는 것으로

f.write(“pluto\n”)을 추가하면 맨 마지막행에 pluto가 추가된다!!

Print(“-2%s %3%d” % (a, 2)) – 이 구성요소를 반드시 기억하라!

Format string – Every element in the tuple has its corresponding place holder in the format string

String에도 in method가 있다! 연속하는 string 이 다음 string안에 있는가?

‘abc’ in ‘acbdefg’ – True , ‘abc’ in ‘abdfgc’ – False

반드시 알아야 할 String의 method들!!

Upper(), lower(), capitalize()

Isalpha() – 문자냐?, isdigit() – 숫자냐?

Startswith(prefix) – 이걸로 시작하는지, endswith(suffix) – 이걸로 끝나는지

Find(str) : str를 찾아라, find(str, start) : start부터 str찾아라, find(str, start, end) : start에서 end까지 str찾아라 str가 없으면 -1반환

있으면 해당 index값 반환!

‘abc’에서 string1.find(‘a’) = 0 반환, string1.find(‘h’) = -1반환

Replace(str1, str2) ; str1을 str2로 반환하라

Strip()은 양쪽만 제거!!!!!! 과제떄 봤지?

Split(기준치) – 기준치 기준으로 쪼개서 리스트 만들어라!

Ex) split() – 공백 기준, split(“,”) – 콤마 기준

Join(list1) – 해당 리스트를 공백을 스페이스바로 두고 string으로 전환!!

List와 string을 전환해주는 split와 join은 필수로 알아두자!!