함수 자체를 변수에 저장할 수 있다

그래서 다른 함수에 인수로 전달 가능

반환값으로도 사용 가능하고 변수나 자료 구조에 저장 가능

함수도 객체고 모든 객체는 레퍼런스를 가짐

일급함수 = 변수에 저장 가능한 함수

>>> def decorate(type='italic'):

def italic(s):

return '<i>'+s+'</i>'

def bold(s):

return '<b>'+s+'</b>'

if type=='italic':

return italic

else:

return bold

>>> dec=decorate()

>>> dec('hello')

'<i>hello</i>'

>>> dec=decorate('bold')

>>> dec('hello')

'<b>hello</b>'

여기 아래에서 cache는 quadratic안에 지역변수 인데 quadratic이 종료되면서 f에서는 사용할 수 없어야 하는데 f가 그 종료된 지역변수를 쓰고 있음. 이런걸 함수 클로저

함수의 주소를 받으면(리턴 받으면) 그 주소를 가지고 호출이 가능하다.

def quadratic(a,b,c):

cache={}

def f(x):

if x in cache:

return cache[x]

y=a\*x\*x+b\*x+c

cache[x]=y

return y

return f

>>> f1=quadratic(3,-4,5);f1(0.1)

4.63

>>> f2=quadratic(-2,7,10);f2(0.4)

12.48

함수 클로저가 되면 자기가 가지고 있지 않은 지역변수를 사용 가능. 바로 사라지지 않음

내장함수를 리턴해주면 함수 클로저가 됨

전역변수 사용을 낮추기 위해서

전역변수를 많이 사용하면 side effect- 간섭현상이 일어남 누가 어떻게 썼는지 몰라

누가 어떻게 사용했는지 모르니깐 워닝 발생시 하나씩 다 확인해봐야함

그리고 data은닉을 위해서

a, b, c, cache는 f의 입장에서는 전역변수라고 생각하면 됨

quadratic은 함수 클로저를 리턴해주는 함수

c1.\_\_closure\_\_

\_두개 들어가는거는 내부변수라고 함 attribute

c1.\_\_closure\_\_[0].cell\_contents

2

이러면 가지고 있는 값의 형태? 값?이 나옴

c2.\_\_closure\_\_[0].cell\_contents

1

>>> c1=makeCounter();c2=makeCounter()

>>> c1();c1();c2()

1

2

1

함수 크롤저 작성 방법

1. 중첩 함수 작성
2. 중첩 함수를 return

partial()함수의 리턴 값이 함수 클로저

f.\_\_code\_\_는 의사 컴파일

인터프리터는 컴파일 느림.

미리 컴파일 코드 만들고 필요하면 실행하는거

실행 가능한 파이썬 코드 – 바이너리형태.

코드객체를 직접 실행하는건 불가능???????????????

함수는 실행 가능???????????

코드 객체 - 코드접근은 가능하지만 실행은 안됨. 코드 통해서 정보 획득은 가능

가변인수는 0x04??????????????????

재귀적 프로그래밍

Iteration(반복), Recursive(재귀)

재귀로 짜면 더 간단한 부분이 있음

재귀-언젠가는 멈춰야함

>>> def sum(N):

if N==1: return 1

return N+sum(N-1)

>>> sum(10)

55

>>> sum(100)

5050

>>> sum(1000)

Traceback (most recent call last):

File "<pyshell#139>", line 1, in <module>

sum(1000)

File "<pyshell#136>", line 3, in sum

return N+sum(N-1)

File "<pyshell#136>", line 3, in sum

return N+sum(N-1)

File "<pyshell#136>", line 3, in sum

return N+sum(N-1)

[Previous line repeated 989 more times]

File "<pyshell#136>", line 2, in sum

if N==1: return 1

RecursionError: maximum recursion depth exceeded in comparison

1000은 에러남

스텍에 호출한 함수의 정보를(인수 목록, 함수 위치 등 stack frame) 담아두는데 공간, 스택 사이즈는 제한적

스택이 없으면 함수 사용 불가

C언어에서 스타트코드 부트코드에서 가장 중요한게 스택 사이즈 설정. 그래야지만 C코드가 실행됨

C언어 특징 – 함수 중심 언어

함수호출은 속도가 느림. 스택프레임 만들어서 푸쉬해야해서

함수는 이름이 있어야 호출 가능

함수 클로저는 레퍼런스만 있어도 됨

람다식 표기법

이름이 없는 함수. 정의 방법c++ 자바 등 있음

lambda <인수들>:<반환할 식>

이름 없이 한줄로 함수 정의

로이만방식???????????뭐지 펌웨ㅐ어 할 때

구조적 프로그래밍: 기능(함수)중심 – C

객체지향 프로그래밍: 객체중심 – C++,JAVA 등 파이썬도 지원은 함. 객체에 메시지 전달

함수적 프로그래밍: 함수 중심

상태변화에 따라서 반환 값이 다를 수 있음

눈에 안보이는 인수들이 있음?????????????????????인수 반환 값. 이것들이 결과에 영향 미치면>부작용

이런게 싱글코어에서는 크게 문제가 되지 않았는데 멀티코어에서는 문제가 됨

함수적 프로그램이에서는 부작용을 없앰. 즉 눈에 보이지 않는 인수 반환값 없음

무조건 리턴으로 나오고 모든걸 다 인수로 넘겨줌

내부적으로 상태변화가 존재하지 않음

재귀적 프로그램을 많이씀. 외부의 상태 변화를 활용하지 않게끔 프로그래밍

순수함수를 활용

반복자(iterator)를 활용. 중요한 역할

파이썬의 함수적 프로그래밍에서 사용한 순수 ㅏㅁ수는 출력으로 반복자를 반환

함수에서 iterator를 반환 받아서 활용 출력 값을 필요로 하는 시점에서 계산함. 게으른 계산

ex>map()함수의 결과인 map객체는 반복자

map()은 함수주고 인자 주면 언제나 같은 반복자를 반환해줌. 순수 함수

filter()도 순수 함수

예를 들면

함수 안에서 date()라는 함수 날짜 함수 같ㅇ느걸 부르면 내가 인자 전달을 안했는데 date에 영향ㅇ르 받는 경우.

또는 멀티쓰레드에서 접근 우선순위를 정할 수 없으니깐 한 개의 인자 같이 쓰면 결과가 어찌 될지 모름

인자 전달아냏쓴ㄴ데 new해서 쓰는거는 순수 함수가 아님. 부작용이 잇을 수 있다.