Essential Ref Ch. 6

/

What functional and declarative style programming tools does Python have?

-

functions

scoping rules

closures

decorators

generators

coroutives

(declarative)

list comprehensions

generator expressions

/

What must you be careful about regarding default function parameters?

-

Not to use mutable objects. No, f(x=[]):…

/

What should you use as a default for mutable objects?

-

None. Then check if it exists, if not create the object.

/

What is it called when a function mutates their input values or changes the state of other parts of the program and why should this be avoided/controlled?

-

Side effects.

Can lead to subtle bugs as program size/complexity increases.

/

How are variables in nested functions bound?

-

Using lexical scoping, innermost to outermost enclosing block.

/

What is a closure?

-

The statements that make up a function packaged with the environment in which they execute.

/

What is one use of closures?

-

lazy or delayed execution

/

What is a more efficient way to preserve state across function calls then a class?

-

A closure

/

What is the name for a function whose primary purpose is to wrap another function?

-

A decorator

/

What order are decorators applied in?

-

order listed

/

Can decorators be applied to class definitions?

What is the difference from function decorators?

-

Yes.

Decorator should return a class object.

/

What defines a generator object?

-

The yield keyword.

/

What is a generator?

-

A function that produces a sequence of values for use in iteration.

/

What is is a coroutine?

-

A function that executes in response to data being sent to it. Place the yield on the right side of the assignment operator.

/

What must be called to advance to the first yield statement in a coroutine?

What is called after that?

-

next()

sent(data)

/

Can coroutines also emit values?

-

yes.

/

What type of programming problems are generators and coroutines particularly effective for?

-

systems, networking, and distributed computation

/

What are generators similar to in unix?

What about coroutines?

-

pipe (a processing pipeline)

(coroutines would be an inverted pipe

/

What is the name for applying a function to all the items in a list?

What is the syntax?

-

list comprehension

[expression for item1 in iterable1 if condition1

for item2 in iterable2 if condition2

…]

/

What are list comprehensions roughly equivalent to?

-

Nested for loops:

For item1 in iterable1:

If condition1:

For item2 in iterable2:

…

/

Do sequences supplied to list comprehensions need to be the same length?

-

No.

/

Can list comprehensions return tuples?

-

Yes, but they must be enclosed in parenthesis.

/

What is a generator expression?

What is the syntax?

-

Generator expressions perform the same computation as list comprehensions but produce the results iteratively.

The syntax is the same except parenthesis are used instead of square brackets.

/

What is an advantage of generator expressions over list comprehensions?

-

Memory, generator expressions don’t create the whole list immediately, just parts as they are needed.

/

What are list comprehensions and generator expressions related too?

-

Declarative languages and mathematical set theory.

List comp: [x \* x for x in a if x > 0]

Set theory: {x^2 | x contained in a, x > 0}

/

What are two advantages of declarative code?

-

1) More compact

2) Runs faster

/

What is used to specify short anonymous callback functions?

-

lambda

/

What four things need to be kept in mind with Python recursion?

-

1) There is a limit on the depth of recursive function calls. Default is 1000.

2) Python does not perform tail recursion optimizations

3) Care must be taken with generators and recursion, so iterate over them and don’t just create new ones.

4) All inner recursive calls get passed through decorators

/

How must decorators be written so as to not break doc strings?

-

Propagate the function name and doc string.

/

Can functions have arbitrary attributes?

-

Yes.

/

Where are function attributes stores?

-

\_\_dict\_\_

/

What do you call to execute an expression string and return the results?

What do you call to execute a string of python code?

What can be used to precompile code for the above statements?

-

value = eval(string)

exec(string)

compile()