Essential Ref: Ch. 1-5

/

What holds the results of the last operation in the shell?

-

An underscore \_

/

What is python’s type system?

-

Dynamic, variables are bound at runtime.

/

How do you get input from the prompt?

-

date = raw\_input(“Input: “)

\*p3 use input()

/

Tuple?

-

Best viewed as a single, immutable object.

Lists are for adding/removing items.

/

Set?

-

Unordered and can’t be indexed (unlike lists and tuples)

/

Dict?

-

Most finely tuned data type

Use instead of rolling your own data structure

Use as a container for fast lookups on unordered data.

/

How to create a list of numbers?

-

range(i, j [, stride])

\*p3 returns generator (p2 call xrange() for one)

/

What is variable scope inside a function?

-

Local, use global keyword to get out.

/

What is a generator?

-

Powerful way of writing programs based on processing pipelines, streams, or data flows.

yield line

/

What is a coroutine?

-

Function written to operate as a task that processes a sequence of inputs.

Useful for concurrent programs based on producer/consumer.

Use with generators.

Line = (yield)

/

On whom do methods act?

-

Normally apply to instances. Can use @static to apply to class/type.

/

How do you list contents of a module?

-

dir(module)

/

What is the escape character for string literals?

-

The dastardly Backslash ‘/’

/

What is u’ at the front of a string?

-

Unicode prefix.

\*not necessary in p3

What is r’ at the front of a string?

-

A raw string prefix.

In raw strings backslash is treated like a normal character.

Use raw string for regular expressions.

/

Which data are objects in python?

-

Everything, all data are objects.

/

What three things make up an object?

-

Each object has an identity, a type/class, and a value.

The identity can be viewed as a pointer to a location in memory.

The type describes the objects internal representation, its methods and operations.

Value – the created object, an instance. Its type and identity can no longer be changed. If the value itself is either mutable or immutable.

/

What are objects that contain other objects called?

-

Containers or Collections

/

What two things characterize an object?

-

Attributes – values associated with the object.

Methods – functions that perform operations on the object.

/

How does object assignement work?

-

i.e. a\_obj = b\_obj

For immutable data types (numbers, strings, tuples, set) by copy.

For mutable data types (lists and dicts) by reference.

To receive a copy of mutable data types use either:

shallow copy: new\_list = list(old\_list)

deep copy: import copy; new\_list= copy.deepcopy(old\_list)

/

What does it mean that all python objects are first class?

-

That all objects have equal status, so a dict could hold numbers, functions, modules, exceptions, and so on.

/

What are python’s built-in types (for data representation)?

-

None

Numbers: int/long/float/complex & bool

Sequences: str/Unicode/list/tuple/exrange

(support iteration and slicing)

Mapping: dict

(arbitrary collection of objects indexed by a nearly arbitrary collection of objects - the latter being immutable and thus hashable)

Sets: set/frozenset

(unordered collection of unique items)

/

What operations are available to all sequences?

-

indexing/slicing: s[i], s[i:j], s[i:j:stride]

truth checks: all(s), any(s)

concatenation: s + t

make copies: s \* n

variable unpacking: s, t = v

membership: x in s, x not in s

iteration: for x in s

also: len(s), min(s), max(s), sum(s)

/

What operations are available to all mutable sequences?

-

slice/index assignment (i.e. s[i:j] = t)

slice/index deletion (i.e. del s[i])

What methods are available to lists?

-

Create/shallow copy: new\_lst = list(seq)

Add to: lst.append(element), lst.extend(list)

lst.count(occurrence), lst.index(occurrence [,start [,stop]])

lst.insert(I, element), lst.pop([i]), lst.remove(element)

reverse/sort in place: lst.reverse(), lst.sort([key [, revers])

/

String method for fast character substitution?

-

str.translate(table [,deletechars])

/

Split a string?

-

str.split([sep [, maxsplit]])

/

Combine items of list with a string, resulting in a string?

-

str.join(list)

/

Locate a substring?

-

str.find(sub [, start [, stop]])

-1 if none

can also count.

/

Strip white space?

-

str.strip([chars])

/

How do you create a dictionary?

-

new\_dict = {}

new\_dict = dict(seq\_key\_value\_tuples)

new\_dict = old\_dict.copy()

new\_dict.fromkeys([seq [, value\_to\_set])

/

How do you determine if a key is in a dictionary?

-

Check its length?

k in dct

(returns a Boolean)

dct.has\_key(key) is deprecated

len(dct)

/

How do you retrieve and set an item in a dictionary?

-

x = dct[key]

x = dct.get(key [, default\_value\_to\_return])

x = dct.setdefault(key [, default\_to\_set\_&\_return])

x = dct.pop(key [, default]) – removes item

x = dct.popitem() – removes random item

set: dct[key] = value

/

How do you remove items from a dictionary?

-

del dct[key]

dct.clear()

/

How do you combine dictionaries?

-

dict\_new.update(dict\_old)

/

What three ways can you return the elements of a dictionary as a sequence?

-

seq = dct.items()

seq = dct.keys()

seq = dct.values()

/

What methods are available to all sets?

-

len(s)

s.copy()

set1.difference(set2) or set1 - set 2

set1.intersection(set2) or set1 & set 2

set1.isdisjoint(set2)

set1.issubset(set2)

set1.issuperset(set2)

set1.symmetric\_difference(set2) or set1 ^ set 2

set1.union(set2) or set1 | set 2

/

How do you add/remove items from a mutable set?

-

s.add(item)

s.update(any\_iterable)

s.clear()

s.discard(item)

s.remove(item) – key error if not member

s.pop() – remove random

(can also do diff/intersect updates)

/

What are python’s built-in types (for program structure)?

-

Callable – built-in(c/c++), user-defined(def/lambda), type, object(with \_\_call\_\_), methods(instance, class, static)

Module

Classes – ancestor of all types/classes

Types – type of built-in types/classes

/

How do you retrieve the documentation string for functions, methods, classes, types, instances, or modules?

How do you determine the name?

-

obj.\_\_doc\_\_

obj.\_\_name\_\_

/

What does \_\_dict\_\_ hold?

-

user-defined function: function attributes

class/type: class methods and variables

instance: instance data

module: module’s namespace

/

How do you determine module’s file and package?

-

mod.\_\_file\_\_ and mod.\_\_path\_\_

/

How do you find a method or instances class?

A classes inheritance?

-

obj\_meth.\_\_class\_\_

class.\_\_bases\_\_

/

Find a class/type/instances module?

Find abstract methods?

-

obj.\_\_module\_\_

obj.\_\_abstractmethods\_\_

/

Determine function namespace/scope?

-

func.\_\_defaults\_\_

func.\_\_globals\_\_

func.\_\_closure\_\_

/

Convert object to string?

String to object?

Sequence to tuple? List? Set?

Sequence of (key, value) tuples to dict?

-

repr(object)

eval(string)

tuple(seq), list(seq), set(seq)

dict(seq\_key\_value\_tuple)

/

What is a conditional expression?

-

min\_value = a if a <= b else b

good with list comprehensions and generators

example:

values = [1, 100, 45, 23, 73, 37, 69]

clamped = [x if x < 50 else 50 for x in values]

print(clamped) # [1, 50, 45, 23, 50, 37, 50]

/

What does partial() from functools do?

-

Evaluates a function with some of its variable:

f = partial(foo, 1, 2)

and returns a function which can be called later with all the variables:

results = f(3)

/

Are loop iteration variables local to the loop?

How do you break from a deeply nested loop?

-

No, scope is visible outside loop and last value is saved.

Throw an exception, break/continue only do one level.

/

How do you loop with an index?

How do you loop in parallel?

-

for i, x in enumerate(seq)

for x, y in zip(seq1, seq2)

(use itertools izip() to get generator in p2)