

## Strings

Seq of chars

Can slice - returns a string

split = join

split on any whitespace

int(string), float(string)

S.strip() - whitespace before

S.rstrip() on right

S.find('a') returns index

is - digit, alpha, upper, lower

repr() for debugging

str() informal string rep

easier to read

Output w/ .format(vars)

"v:{0}, w:{1}".format(v,w)

names

{0:width} {0:2}

print('a', 'b', sep=' ')

Dictionaries {look up}

Keys

store any objects

not ordered

d = {'k1': 'v1', 'k2': 'v2'}

d.get('k1')

d.keys(), d.values(), d.items()

return views

d.get('k1')

used as Keys:

int, float, bool, complex, str

tuple, frozen set

## Control Flow

while boolean:

else

+ break or continue

if-elif-else:

No case, switch

for loops

for item in list:

do item.do.it

q = list(range(0,10,2))

for i in a:

print i \* 2

One lines

n = [i \* i for i in listx]

d = {i: i \* i for i in listx}

if i > 1

Only uses spaces in py code

False = None, 0, "", [], {}

and + or OK to use

functions

def func(self, a1, a2)

return v

func(self, v1=3, v2=4)

name default

def maxi(self, a\*, d\*\*\*)

tuple dictionary

func -> variable

label = func

label(32) -> value

Generator

yield value

makes an iterator

## Decorators

functions inside functions

## Modules

file w/ code

namespaces to avoid clashes

file: var

file: func(h)

import file

from file import f1, f2

from file import \* (avoid)

where?

import sys

sys.path

- func not imported

local -> global -> builtin

:- override, like --init--()

## Programs

if \_\_name\_\_ == '\_\_main\_\_':

argparser for input

#!/usr/bin/env python

.pyc - bytecode

distutil to distribute

py2exe win

py2app OSX

freeze for machine w/ copy

## The Filesystem

import os

os.getcwd()

os.listdir(os.getcwd())

os.path.join('dir', 'subdir')

os.path.isdir

os.name

os.path.exists, os.isfile, os.isdir

## Exceptions

object raise

program flow

list of exceptions

try:

except e-type as v1:

New exception

class MyError(Exception):

class Circle:

def \_\_init\_\_(self, var):

More to initialize than construct

Methods

def func(self, var):

Static method

def to\_full\_area():

class Circle(Shape):

def \_\_init\_\_(self, var):

More to initialize than construct

Methods

def func(self, var):

Static method

def to\_full\_area():

class Circle(Shape):

def \_\_init\_\_(self, var):

More to initialize than construct

Methods

def func(self, var):

Static method

def to\_full\_area():

class Circle(Shape):

def \_\_init\_\_(self, var):

More to initialize than construct

Methods

## Read & Write Files

f = open("fname", "r")

for line in f:

f.readlines() puts into memory

f.write("foo\n")

i = input("yes?")

stdin

write objects in/out file

import pickle

pickle.dump(a, f)

where a is anything

make a dictionary

def save\_data:

def restore\_data:

Exceptions

object raise

program flow

list of exceptions

try:

except e-type as v1:

New exception

class MyError(Exception):

class Circle:

def \_\_init\_\_(self, var):

More to initialize than construct

Methods

def func(self, var):

Static method

def to\_full\_area():

class Circle(Shape):

def \_\_init\_\_(self, var):

More to initialize than construct

Methods