Introduction to Python

Exercises

Andres Mendez-Vazquez

November 5, 2016

Outline

Outline

Exercises

Play with the reference counter

- Assign four references an integer
- Add some stuff to one of the references and see what happened

Standard Data Types

- add key/value to build a dictionary
- Use the dictionary instead of cases

Exercises

Play with the reference counter

- Assign four references an integer
- Add some stuff to one of the references and see what happened

Standard Data Types

- add key/value to build a dictionary
- Use the dictionary instead of cases

Outline

Exercises

Instantiate values

- Complex
- float
- Do a conversions
  - From float to complex
  - From complex to float

Mathematical Functions

- Play with
  - Absolute value
  - Compare references
  - exponential function from math package
  - etc

Exercises

Instantiate values

- Complex
- float
- Do a conversions
  - From float to complex
  - From complex to float

Mathematical Functions

- Play with
  - Absolute value
  - Compare references
  - exponential function from math package
  - etc

Outline

Playing Random Functions in Lists

import random

- Create a List
- Shuffle the elements
- Do a random choice

Some other instructions to play with them

- randrange(a,b,step) - A unique number from a random number
- uniform - Uniform Distribution
- randint(a,b)
- sample()
  - random.sample(range(100),5)

Playing Random Functions in Lists

import random

- Create a List
- Shuffle the elements
- Do a random choice

Some other instructions to play with them

- randrange(a,b,step) - A unique number from a random number
- uniform - Uniform Distribution
- randint(a,b)
- sample()
  - random.sample(range(100),5)

Outline

Exercises

Create two string

- Concatenate them

Make them into a list

- Using the conversion list

Extract stuff from a string

- Using the concept of a list

Exercises

Create two string

- Concatenate them

Make them into a list

- Using the conversion list

Extract stuff from a string

- Using the concept of a list

Exercises

Create two string

- Concatenate them

Make them into a list

- Using the conversion list

Extract stuff from a string

- Using the concept of a list

Outline

Exercises

Generate a List

- Change some values in the list
- Concatenate Tree Lists
- Play with the Range Function
- Slice a List

Generate a dictionary

- add key/value to the dictionary

Exercises

Generate a List

- Change some values in the list
- Concatenate Tree Lists
- Play with the Range Function
- Slice a List

Generate a dictionary

- add key/value to the dictionary

Some Extra Stuff

Using None and *

This is a special one

- [None]*10
- [1]*10
- etc

Outline

Exercises

Do some operations using

1. a+b
2. a-b
3. b/a
4. a**b
5. a%b

For Example

Define

A polynomial and evaluate it

$$a_0 + a_1 r + a_2 r^2 + ... + a_n r^n$$

Define the Greatest Common Divisor Between Two Number A_1 and B_2, $A_1 > B_1 \neq 0$

1. $A_1 = kB_1 + A_2$
2. $B_1 = tA_2 + A_3$
3. etc

For Example

Define

A polynomial and evaluate it

$$a_0 + a_1 r + a_2 r^2 + ... + a_n r^n$$

Define the Greatest Common Divisor Between Two Number A_1 and B_2, $A_1 > B_1 \neq 0$

1. $A_1 = kB_1 + A_2$
2. $B_1 = tA_2 + A_3$
3. etc

Outline

Exercise

Play a little bit with

1. <less
2. <>not equal
3. ! =Not Equal
4. > Greater

Exercise

Do the following

1. $1<2$
2. $2<1$
3. $3!=4$
4. etc

Outline

Exercises

We have

- a = b+c
- c+=a
- c*=s
- c/=b
- c%=d
- c**=a

For Example

Do the Folllowing

1. A = 100
2. B = 100

Then Play with the Operators

1. C = A+B
2. C+=A
3. C*=A
4. C/=2
5. C**=2

For Example

Do the Folllowing

1. A = 100
2. B = 100

Then Play with the Operators

1. C = A+B
2. C+=A
3. C*=A
4. C/=2
5. C**=2

Outline

Exercises

Play with the Operators to Compare Two Numbers

1. And &
2. Or |
3. Xor ^
4. Negation ~
5. shift to the left <<
6. shift to the right >>

For Example

Do the following

1. $(3 \,\&\, 1 == 0)$
2. $(4 \,\&\, 1 == 0)$
3. $(3 \,\&\, 1 == 0)$

What is this

A simple even-odd detector

Now $\sim$ is the Complement of $x \approx -x - 1$

1. $\sim 100 + 100$

For Example

Do the following

1. $(3 \,\&\, 1 == 0)$
2. $(4 \,\&\, 1 == 0)$
3. $(3 \,\&\, 1 == 0)$

What is this

A simple even-odd detector

Now $\sim$ is the Complement of $x \approx -x - 1$

1. $\sim 100 + 100$

For Example

Do the following

1. $(3 \,\&\, 1 == 0)$
2. $(4 \,\&\, 1 == 0)$
3. $(3 \,\&\, 1 == 0)$

What is this

A simple even-odd detector

Now $\sim$ is the Complement of $x \approx -x - 1$

1. $\sim$100+100

Power

Using <<

1. 100<<2
2. 1<<1
3. 1<<2
4. 1<<3

Basically a Multiplication By 2

Thus what is >>?

Power

Using <<

1. 100<<2
2. 1<<1
3. 1<<2
4. 1<<3

Basically a Multiplication By 2

Thus what is >>?

Outline

Exercises

Belonging Operators

1. “in”
2. “not in”

Generate a List

For Example

1. L = [1,2,3,4]
2. Then (5 in L)
3. (4 in L)
4. (3 not in L)

Outline

Exercises

Play with

- a is b
- a == b
- a is not b

Outline

Exercises

Combine the if elif else

To select given a List of numbers from 0 to 100 which are a power of two

For This

Remember the following pattern in bitwise

1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0

Exercises

Combine the if elif else

To select given a List of numbers from 0 to 100 which are a power of two

For This

Remember the following pattern in bitwise

1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0

Exercise

Solve the following using a While Loop

1. Given a List of numbers from 0 to 100 with one not there
2. Set one of the numbers to zero using random
3. Find which one is not there using
    1. $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Then

Print the one is not there!!!

Exercise

Solve the following using a While Loop

1. Given a List of numbers from 0 to 100 with one not there
2. Set one of the numbers to zero using random
3. Find which one is not there using
    1. $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Then

Print the one is not there!!!

Exercise

Do the following

Given a list of random numbers

- Print the multiples of 3
- Print the multiples of 4
- Print the multiples of 7

Hint

Use List+= [Number]

Exercise

Do the following

Given a list of random numbers

- Print the multiples of 3
- Print the multiples of 4
- Print the multiples of 7

Hint

Use List+= [Number]

Outline

Give a Geometric Sequence Efficiently

Given the following function

$$S_n = \sum_{k=0}^{n} r^k$$

Implement this in linear time $O(n)$

Hint accumulate a power!!!

Give a Geometric Sequence Efficiently

Given the following function

$$S_n = \sum_{k=0}^{n} r^k$$

Implement this in linear time $O(n)$

Hint accumulate a power!!!

Outline

Implement Fibonacci

Implement two versions of the function

$$F_N = \begin{cases} 1 & \text{if } N = 1, 2 \\ F_{N-1} + F_{N-2} & \text{if } N > 2 \end{cases} \tag{1}$$

The Recursive One

Simply interpret the function!!!

Iterative

Use extra memory by building a list

F_1	F_2	F_3	...	F_N

Implement Fibonacci

Implement two versions of the function

$$F_N = \begin{cases} 1 & \text{if } N = 1, 2 \\ F_{N-1} + F_{N-2} & \text{if } N > 2 \end{cases} \quad (1)$$

The Recursive One

Simply interpret the function!!!

Iterative

Use extra memory by building a list

F_1	F_2	F_3	...	F_N

Implement Fibonacci

Implement two versions of the function

$$F_N = \begin{cases} 1 & \text{if } N = 1, 2 \\ F_{N-1} + F_{N-2} & \text{if } N > 2 \end{cases} \quad (1)$$

The Recursive One

Simply interpret the function!!!

Iterative

Use extra memory by building a list

F_1	F_2	F_3	...	F_N

Outline

Define the Polynomial Function

Implement the polynomial function

1. Use a While Loop for it

Similar to the Geometric Series

Do so...

Define the Polynomial Function

Implement the polynomial function

1. Use a While Loop for it

Similar to the Geometric Series

Do so...

Outline

Implement the Matrix Multiplication

Using a List of List

Implement the Matrix Multiplication of 3 by 3 matrices

1. L = [[1,2,3,],[4,5,6],[7,8,9]]
2. Then Remember

$$C = AB$$

$$[c_{ij}] = \left[\sum_{k=1}^{N} a_{ik} b_{kj} \right]$$

Outline

Give me an in-line function of the Fibonacci Numbers

For This, we can try the following

t = lambda x: 1 if x==1 or x==2 else t(x-1)+t(x-2)

Then using this as basis

Give me the recursive version of the factorial

$$n! = n \times (n-1)!$$

Give me an in-line function of the Fibonacci Numbers

For This, we can try the following

t = lambda x: 1 if x==1 or x==2 else t(x-1)+t(x-2)

Then using this as basis

Give me the recursive version of the factorial

$$n! = n \times (n-1)!$$

Outline

Now, Using a Dictionary

Select Different Functions Like

1. %
2. **
3. /
4. A<<2
5. B>>2

For This, you can use a dictionary

1. FDictionary = {'Power': lambda x: x**2,... }
2. Then you can use the FDictionary['Power'](5)
3. To evaluate the lambda functions

Now, Using a Dictionary

Select Different Functions Like

1. %
2. **
3. /
4. A<<2
5. B>>2

For This, you can use a dictionary

1. FDictionary = {'Power': lambda x: x**2,... }
2. Then you can use the FDictionary['Power'](5)
3. To evaluate the lambda functions

Outline

Now Try to use the Dictionary in a Numerical List

Given

- $L = range\,(10)$
- Use map to apply one of your dictionary functions
  - map(FDictionary['Power'],L)

Play with it

It is fun

Now Try to use the Dictionary in a Numerical List

Given

- $L = range\,(10)$
- Use map to apply one of your dictionary functions
  - map(FDictionary['Power'],L)

Play with it

It is fun