Python Workshop P1

ACM/WiCS Workshop 11.5.18

Andrew Rojas and Claudia Moncaliano

HelloWorld.java

```
class HelloWorld {
  static public void main( String args[] ) {
    System.out.println( "Hello World!" );
  }
}
```

helloworld.c

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
  puts("Hello World!");
  return EXIT_SUCCESS;
}
```

helloworld.py

```
print("Hello World")
```

0

Variables, numbers, strings

No semicolons!

File Edit View Search Terminal Help

```
[amrojas@SLAVE-I ~]$ python3
Python 3.7.0 (default, Sep 15 2018, 19:13:07)
[GCC 8.2.1 20180831] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 5 # assign int value 5 to x
>>> y = 2.4 # assign float value 2.4 to y
>>> z = x + y # add x and y, then assign that to z
>>> z
7.4
>>>
```

File Edit View Search Terminal Help

```
[amrojas@SLAVE-I ~]$ python3
Python 3.7.0 (default, Sep 15 2018, 19:13:07)
[GCC 8.2.1 20180831] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 'hello'     # assign string "hello" to x
>>> space = ' '     # assign string " " to space
>>> message = x + space + 'world'
>>> message
'hello world'
>>>
```

identity operators

```
a = ‘keep it real’
b = ‘keep it real’

if a is b:
    print(‘twins‘)
else:
    print(‘fraternal’)

if id(a) == id(b):
    print(‘twins again!‘)
else:
    print(‘nope!‘)
```

Output:

twins

twins again!

1

Control Flow

Indentation!

logical operators

```
t, f = True, False

if t and t:
  print(‘both true’)
elif t and f:
  print(‘f is false’)
else:
  print(‘sad’)
```

```
if not t:
  print(“Brojas”)

if not f:
  print(“Clauds”)

if t or f:
  print(“a truth exists”)
```

Output:

Clauds

a truth exists

```java
int x = 7;

if (x > 5) {
  System.out.println("bigger");
}
else {
  System.out.println('smaller');
}

while (x != 3) {
  System.out.println(x)
  x--;
}
```

```python
x = 7

if x > 5:
  print('bigger')
else:
  print('smaller')

while x != 3:
  print(x)
  x -= 1
```

```
int x = 1;

for (int i = 0; i < 3; i++) {
  x *= i;
}
```

In Python this becomes:

```
x = 1

for i in range(3):
  x *= i
```

range()

Represents sequence of numbers, replaces the 3 statements of for loop

```
range(5)
  0, 1, 2, 3, 4
range(5, 10)
  5, 6, 7, 8, 9
range(10, 5, -1)
  10, 9, 8, 7, 6
```

2
Lists

[‘this’, ‘is’, ‘a’, ‘subtitle’]

list() or []

creates a list (python ‘arrays’)

```
a = list(1, 2, 3, 4)
b = [1, 2, 3, 4]

if a == b:
  print(‘a equals b’)
```

Accessing elements

```
>>> listy = [1, 2, 3, 4]
>>> listy[1]
2
>>> listy[5]
IndexError: list index out of range
>>> len(listy)
4
>>> 3 in listy
True
```

len()

Returns the length of a sequence or collection

```
len([1, 2]) = 2
len([]) = 0
len(‘word’) = 4 # works on strings!
```

in

Checking that an element exists in a sequence or collection

```
>>> 5 in [1, 1, 2, 3, 5, 8, 13]
True
>>> 3 in [1, 1, 1, 1]
False
>>> ‘w’ in ‘word’ #works on strings
True
```

Iterating over elements

```
acm_board = [‘CC’, ‘JB’, ‘AR’,
‘AL’, ‘CM’, ‘ER’, ‘BP’]

for p in acm_board:
  if p == ‘AR’:
    print(‘Is really cool’)
  else:
    print(‘Is lame’)
```

lists as stacks

```
>>> acm_board = ['CC', 'JB', 'BP']
>>> acm_board.pop()
'BP'
>>> acm_board
['CC', 'JB']
>>> acm_board.append('new_BP')
>>> acm_board
['CC', 'JB', 'new_BP']
```

slicing

```
>>> my_string = “make this short”
>>> my_string[10:]   # 10 to end
‘short’
>>> my_string[-5:]   # -5 to end
‘short’
>>> my_string[:4]    # 0 to 4
‘make’
>>> my_string[::-1]  # reverse
‘trohs siht ekam’
```

3

Dictionaries

Mapping out a solution

```
dict = {}
```

creates a dictionary {python maps}

```
snacks = {
    ‘insomnia cookies’ : 3,
    ‘Whole milk’ : 2,
    ‘skim’ : 1
  }
```

.items()

```
for k, v in snacks.items():
  print(k, v)
```

Output :

insomnia cookies 3

whole milk 2

skim 1

.keys() .values()

```
>>> print(snacks.keys())
[‘whole milk’,‘insomnia cookies’,
‘skim’]
>>> print(snacks.values())
[3, 1, 2]
```

4

Functions and Modules

It’s machine learning if
I import torch

(def)initely a function

```
def cookies(event_type):
    if event_type is ‘acm only’:
        return ‘oreos!’
    if event_type is ‘acm+wics’:
        return ‘insomnia!’
    return None    # Null in python

wkshp_snacks = cookies(‘acm+wics’)
```

```
import math

>>> math.ceil(5.2)
6
>>> math.factorial(6)
720
>>> math.copysign(-1, 5)
1

...and many more
```

Function	Description
ceil(x)	Returns the smallest integer greater than or equal to x.
copysign(x, y)	Returns x with the sign of y
fabs(x)	Returns the absolute value of x
factorial(x)	Returns the factorial of x
floor(x)	Returns the largest integer less than or equal to x
fmod(x, y)	Returns the remainder when x is divided by y
frexp(x)	Returns the mantissa and exponent of x as the pair (m, e)
fsum(iterable)	Returns an accurate floating point sum of values in the iterable
isfinite(x)	Returns True if x is neither an infinity nor a NaN (Not a Number)
isinf(x)	Returns True if x is a positive or negative infinity
isnan(x)	Returns True if x is a NaN
ldexp(x, i)	Returns x * (2**i)
modf(x)	Returns the fractional and integer parts of x
trunc(x)	Returns the truncated integer value of x
exp(x)	Returns e**x
expm1(x)	Returns e**x - 1
log(x[, base])	Returns the logarithm of x to the base (defaults to e)
log1p(x)	Returns the natural logarithm of 1+x
log2(x)	Returns the base-2 logarithm of x

5

User IO

You probably want to be
able to see things

print() ‘thank u, next’ <3

Lets you print things during execution! (with a newline)

```
stringy = ‘thank u,’
crazy = ‘ next’
print(stringy + crazy)
print(5)
print(3)
```

Output:

thank u, next

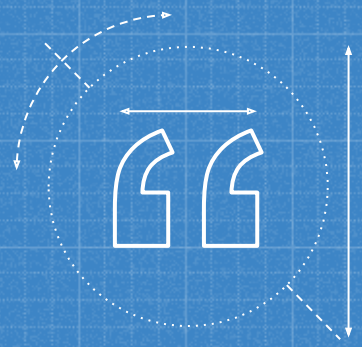
5

3

input()

```
name = input('give me your name: ')
reverse = name[::-1]
print(reverse)
```

```
PPL10205-AndrewRojas:~ andrew.rojas$ python3 test.py
give me your name: Andrew Rojas
sajoR werdnA
PPL10205-AndrewRojas:~ andrew.rojas$
```



thank you fam!

have fun coding and stuff

use python for:

data science

machine learning

fun scripts and tools

robust programs (i.e. instagram, compiler)

next week is python part 2!