

CS10


The Beauty and Joy of Computing


Lecture #20

Python II

2014-11-12


OBAMA URGES FCC TO TAKE CONTROVERSIAL APPROACH TO NET NEUTRALITY






UC Berkeley EECS
Lecturer
Gerald Friedland

<http://www.pcmag.com/article2/0,2817,2471914,00.asp>




Peer Instruction




How's your Python experience so far?

- a) I'd rather stay with Snap!
- b) Neutral
- c) I love Python!
- d) Not sure yet.



UC Berkeley CS10 "The Beauty and Joy of Computing": Python II (2)





Python II: Let's get comfortable...




<http://www.thechobble.com/2012/06/meet-julius-huge-yellow-pet-python-18.html>




UC Berkeley CS10 "The Beauty and Joy of Computing": Python II (3)







Python II


- Defining a function
- List and dictionaries
- HOFs in Python
- APIs
- A game in Python





UC Berkeley CS10 "The Beauty and Joy of Computing": Python II (4)






Defining a Function


A function/method/procedure/block is defined with def


```
def function1():
    print "Hello World"

def function2(parameter1, parameter2):
    print parameter1+parameter2
```



UC Berkeley CS10 "The Beauty and Joy of Computing": Python II (5)






Defining a Function


Remember this?

```
def fib(n):
    a,b=0,1
    while a < n:
        print(a)
        a,b=b, a+b
```

Let's give it a try...



UC Berkeley CS10 "The Beauty and Joy of Computing": Python II (6)



Iterators

For iterates over lists

```
list = [2, 4, 6, 8]
sum = 0
for num in list:
    sum = sum + num
print("The sum is:", sum)
```

Let's give it a try...

UC Berkeley CS10 "The Beauty and Joy of Computing" : Python II (7)

How to iterate over a range...

...with range:

```
list = range(1,10)
sum = 0
for num in list:
    sum = sum + num
print("The sum is:", sum)
```

Let's give it a try...

UC Berkeley CS10 "The Beauty and Joy of Computing" : Python II (8)

Speaking of list...

Lists can contain anything

```
list = [1,2,3]
list = ["Hello", "World"]
list = ["Hello", 1, "World", 2.3]
list = [[1,2], [3,4], [5,6]]
list = [(1,2), [3,4], 5, 6]
```

Let's give it a try...

UC Berkeley CS10 "The Beauty and Joy of Computing" : Python II (9)

List Comprehension

How to work with Lists

```
list = range(1,5)+range(5,10)
elem = list[3]
elems = list[3:5]
list2 = list[4:]
list3 = list[:9]
```

Let's give it a try...

UC Berkeley CS10 "The Beauty and Joy of Computing" : Python II (10)

List Comprehension (more)

How to work with Lists

```
fruits = ['Banana', 'Apple', 'Lime']
loud_fruits = [fruit.upper() for fruit
in fruits]

list(enumerate(fruits))
```

Let's give it a try...

UC Berkeley CS10 "The Beauty and Joy of Computing" : Python II (11)

List Comprehension (more)

How to work with Lists

```
List=[0,1,2,3,4,5,6,7,8,9]
list.reverse()
list.sort()
list.count(10)
list.insert(5,123)
```

Let's give it a try...

UC Berkeley CS10 "The Beauty and Joy of Computing" : Python II (12)

lbc **Dictionaries**

```
stuff = {'name': 'Zed', 'age': 39,
        'height': 6 * 12 + 2}

print stuff['name']
```

Let's give it a try...

UC Berkeley CS10 "The Beauty and Joy of Computing" : Python II (18)

lbc **HOFs**

HOFs are part of Python too!

```
map(upper, ['sentence', 'fragment'])

def is_even(x):
    return (x % 2) == 0
filter(is_even, range(10))
```

Let's give it a try...

UC Berkeley CS10 "The Beauty and Joy of Computing" : Python II (14)

lbc **APIs**

APIs = Application Programming Interface

- APIs allow blackbox use of pre-programmed elements
- Provide abstraction, save work


```
import pygame
pygame.init()
```

Let's give it a try...

UC Berkeley CS10 "The Beauty and Joy of Computing" : Python II (16)

lbc **A Game!**

- Python very productive language
- Pacman game in under 1000 lines of code.
- Works platform independent (Linux, MacOS X, Windows, iOS, etc...)



Let's give it a try...

UC Berkeley CS10 "The Beauty and Joy of Computing" : Python II (16)