Python for Advanced Beginners

Fernando Pineda 140.636 Oct. 11, 2017 version 0.1

6 Mutable vs Immutable objects

The Python Data Model defines *mutable* and *immutable* objects. Objects whose value(s) can change are said to be mutable. Objects whose value(s) are unchangeable (after they are created) are called immutable. The values associated with objects are stored as the attributes of the objects.

The following built-in classes are immutable

- int
- float
- decimal
- complex
- bool
- string
- tuple
- range
- frozenset
- bytes

The following built-in classes are mutable

- list
- dict
- set
- bytearray

User defined classes are mutable by default but can be made immutable by declaring that the class has no attributes.

6 lists, tuples, arrays and strings

1. A list object is what you would normally think of as a list -- i.e. an ordered list of items. It is created by surrounding the items with square brackets.

```
['Sun','Mon','Tue,'Wed','Thu','Fri','Sat']
type('Sun','Mon','Tue,'Wed','Thu','Fri','Sat')
```

2. A tuple object is just like a list except that it is immutable. It is created by surrounding the items with parentheses.

```
('Sun','Mon','Tue,'Wed','Thu','Fri','Sat')
type(('Sun','Mon','Tue,'Wed','Thu','Fri','Sat'))
```

3. An array just refers to either a list or a tuple that has had a name bound to it. We then just refer to the list or tuple by name. An array is mutable or immutable depending on whether the underlying object is a list or a tuple respectively.

```
mutable_array  = ['Sun','Mon','Tue,'Wed','Thu','Fri','Sat']
imutable_array = ('Sun','Mon','Tue,'Wed','Thu','Fri','Sat')
```

In python strings are a subclass of lists. Internally strings are represented as arrays with each character in the string being a separate item in the list. When you display a string the characters are concatenated together. If you want to see the string displayed as a list just cast it into a string using the `list()` function.

```
"taatattatagcccaggacacagg"
type("taatattatagcccaggacacagg")
list("taatattatagcccaggacacagg")
```

This is how you iterate through a list, tuple or string

```
def array_iterator(x):
  for c in x:
      print(c)

x = ('Sun','Mon','Tue','Wed','Thu','Fri','Sat')
array_iterator(x)

x = "taatattatagcccaggacacagg"
array_iterator(x)
```

list, tuple and string objects all have methods for manipulating them

```
dir(list())
dir(tuple())
dir(str())
```

Next Lecture: Dictionaries and Python by example

The examples in this section are taken from various sources

1. The Python language reference https://docs.python.org/3/reference/index.html
2. Mutable vs immutable presentation from CODEHABITUDE https://codehabitude.com/2013/12/24/python-objects-mutable-vs-immutable/
3. Illustrating Python via Bioinformatics Examples http://hplgit.github.io/bioinf-py/doc/pub/html/index.html

In python strings are internally represented as arrays

enterythe following