PYTHON SECTION SEVEN

Introduction to Python| Tuples & Lists | Pragmatic Agility

TUPLES

- Create with () & tuple()
- Trailing comma is important
- ('one',)
- tuple(['one'])
    - tuple() should be used on other objects

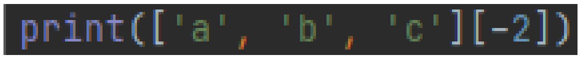
LISTS

- Create with [] & list()
- ['one', 'two']
    - No comma
- Again, use list() on other data types
- list(('one',))
    - Converts a tuple to a list

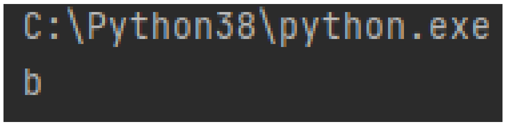
EXTRA

- The string split function creates a list
    - 'th is'.split() -> ['th', 'is']
- List slice for items from a list
    - ['a','b'][0:1] -> ['a']
- List offsets. Negative numbers start from the end
    - ['a', 'b'][-1] -> 'b'

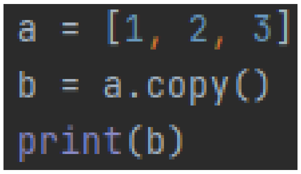
QUICK EXAMPLES



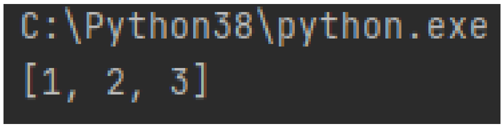
```
print(['a', 'b', 'c'][-2])
```



```
C:\Python38\python.exe
b
```



```
a = [1, 2, 3]
b = a.copy()
print(b)
```



```
C:\Python38\python.exe
[1, 2, 3]
```

Copy Function

The copy function creates a new data object with a copy of the values

Lists of lists

Lists can contain a lot of different objects, even other lists.

Deep Copy

Nested objects need a deep copy. If you do not use a deep copy, the nested items will share the same address. It will be two different variables that point to the same data object