ZIP/LISTS/TUPLES

9.12.2018

TUPLES

* Tuples are almost exactly like lists, but they can’t be changed (no append, insert, or delete)

  * iow, tuples are “immutable”

* (1,2,3) is a tuple

* 1,2,3 is also a tuple (the parentheses are implicit)

ZIP

* given (for example) two lists:

  `[1,2,3] and [‘a’, ‘b’, ‘c’]`

  zip returns three tuples:

  `[(1,’a’), (2,’b’), (3,’c’)]`

* *(ok that’s a lie it actually returns a generator but that’s complicated and we’re not going to talk deeply about it yet or maybe ever)*

ZIP

* `list(zip("zpi mzn", “i saaig”)) => ???`

ZIP

* zip can operate on an arbitrary number of inputs:

* list(zip([1,2,3], [4,5,6], [7,8,9], [10,11,12]))

* => [(1, 4, 7, 10), (2, 5, 8, 11), (3, 6, 9, 12)]

ENUMERATE

* Given a list (e.g. [“occipital”, “parietal”, “frontal”, “temporal”]),

  enumerate returns four tuples:

  [(0, “occipital”), (1, “parietal”), (2, “frontal”), (3, “temporal”)]

* *(again, kind of a lie but deal with it)*

ENUMERATE

* This is very useful if you need access to both an element of a list and to its index. For example:

*
```
filenames = [“file1.txt”, …]
for fi, fname in enumerate(filenames):
    print(“loading file ” + str(fi))
    …
```

UNPACKING

* l = [“scotts pine”, “larch”, “spruce”]
* *What does this do?*
  * a, b, c = l

UNPACKING

* l = [“scotts pine”, “larch”, “spruce”]
* *What about:*
  * a, b, c, d = l
  * a, b = l

LIST COMPREHENSIONS

* List comprehensions let you cram an entire for loop into one line of code (while staying really understandable!)

* e.g. `[x**2 for x in [1,2,3,4,5]]`

LIST COMPREHENSIONS

* List comprehensions can be nested

*
  ```
  list1 = [1,2,3]
  list2 = [4,5,6]
  [[a*b for a in list1] for b in list2]
  ```

* => ?

LIST COMPREHENSIONS

* List comprehensions can be nested

*
  ```
  list1 = [1,2,3]
  list2 = [4,5,6]
  [[a*b for a in list1] for b in list2]
  ```

* => [[4, 8, 12], [5, 10, 15], [6, 12, 18]]

LIST COMPREHENSIONS

* List comprehensions can also have more than one “for”

*
```
list1 = [1,2,3]
list2 = [4,5,6]
[a*b for a in list1 for b in list2]
```

* => ?

LIST COMPREHENSIONS

* List comprehensions can also have more than one “for”

*
```
list1 = [1,2,3]
list2 = [4,5,6]
[a*b for a in list1 for b in list2]
```

*
```
=> [4, 5, 6, 8, 10, 12, 12, 15, 18]
```

LIST COMPREHENSION

*
  ```
  ''.join([''.join(s) for s in
  list(zip("zpi mzn", "i saaig”))])
  ```

PUTTING IT ALL TOGETHER

* when zip, list comprehensions, and unpacking combine, they make *magic*

PUTTING IT ALL TOGETHER

* `list1 = [2,4,6,8,10]`
  `list2 = [1,3,5,7,9]`

* How do we compute the product of each pair of elements (e.g. `list1[0]*list2[0]`, etc.)?

PUTTING IT ALL TOGETHER

```
* list1 = [2,4,6,8,10]
  list2 = [1,3,5,7,9]

  [x*y for x,y in zip(list1, list2)]
  => [2, 12, 30, 56, 90]
```

END