Tuples, and more experiments with lists.


2. Tuples, more formally. A tuple is an immutable, ordered list of values.
   
   (a) Tuples are represented using parentheses:
       
       (4,2,3,1), ('north','north','east'), (3,), or ().
       
       (Notice that a singleton requires a comma to distinguish it, syntactically, from standard use of parentheses. Actually, you can always have an “extra” comma after a final list, tuple, or set element.)

   (b) When it’s not ambiguous, you can drop the parentheses!

       >>> a = 1,2
       >>> a
       
       (1, 2)
       
       Wowza.

   (c) The `tuple(i)` constructor will build a tuple from any iterable source of values.

   (d) As with lists, you can index and slice tuples.

   (e) You can concatenate them, with `+`; you can replicate them with `*`:

       >>> (1, 2)+(3, 4)
       (1, 2, 3, 4)

       >>> (1,) * 3
       (1, 1, 1)

   (f) You test for membership in a tuple with `in`, `t.find(item)` finds an item within a tuple, and `t.count(item)` counts occurrences of item within t.

3. Left and right values.

   (a) A left value or l-value is an assignable object. It is any expression that may occur on the left side of an assignment. Variables are obvious l-values, but so are items in lists.

   (b) A right value or r-value is any expression that has a value that may appear on the right of an assignment. In python, everything is an r-value.

   (c) Traditionally, the underscore (_) is used as a place-holder for an l-value when we don’t care about the result of the assignment.

   (d) In assignment, a tuple of l-values is, itself an l-value:
>>> (a, b, c) = (1, 2, 3)
>>> a, b, c = 1, 2, 3
>>> (a, b), _, c = (1, 2), 9, 3

Each of these effectively assigns a=1, b=2, and c=3.

(e) These complex assignments happen in parallel, so we can exchange values with:

```python
>>> a, b = b, a

Really: it’s tuple assignment!
```

(f) Here’s Euler’s algorithm for finding greatest common divisors:

```python
def gcd(a, b):
    while a > 0:
        a, b = (b, a) if a > b else (b % a, a)
    return b
```

(g) Wowza: when an asterisk precedes a variable name used as an l-value, it means assign this variable the remaining r-values as a list. This is very powerful:

```python
>>> car, *cdr = (1, 2, 3)
>>> cdr
[2, 3]
```

(From an old language, Lisp, the car of a list is the first element, and the cdr (“could-r”) is what remains.) This asterisk notation can be used to assign the last formal parameter all the actual parameters that remain:

```python
>>> def min(first, *args):
...     m = first
...     for x in args:
...         if x < m: m = x
...     return m
>>> min(31, 331, 21)
21
```

(h) Challenge: add key=func.

4. Recall: list comprehensions.

```python
>>> def factors(n):
...     return [f for f in range(1, n+1) if n%f == 0]
>>> primes = [n for n in range(100) if len(factors(n)) == 2]
>>> primes
```

5. Tuples have comprehensions, too. They’re surrounded by parens and, in most ways, are similar to list comprehensions.

Wednesday: Dictionaries.

*