

Loops and Orbits - Week 1 - Day 3 - Computer Science

Last time we finished with Python lists (called arrays in most computer languages other than Python). Python has two more very convenient ways of holding and passing around multiple things at once: tuples and ranges.

Tuples

You've already seen tuples. When we have been putting t, x at the end of some Python or

times, positions

at the end of some Python, we've been telling the interpreter running in the kernel to pass back two things as a tuple.

It is very common (but unnecessary) to put parenthesis around a tuple. So at the end of someone else's Python you might see

(times, positions)

← returns times and positions as a tuple, and in our code each of these was a list, so this is a tuple of lists!

Unpacking Tuples — Multiple Assignments

When an expression is a tuple, you can "unpack" it into multiple individual variables. Here's a simple example:

$t_{\text{initial}}, v_{\text{initial}}, x_{\text{initial}} = 0.0, 0.0, 0.0$

Don't revel in making code compact. Your top priority is readability! The above statement is equivalent to

$t_{\text{initial}} = 0.0$ # Stopwatch starts when race starts
 $v_{\text{initial}} = 0.0$ # Drag racer starts at rest
 $x_{\text{initial}} = 0.0$ # Drag racer starts at start line

The latter is in most cases better because it is clearer, especially with the comments.

Ranges

It is very common to want a sequential list of integers. For example we might want

$[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$

or

$[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]$

more common to want 10 things from 0 to 9 than 11 things from 0 to 10, but obviously that depends on the situation

We have needed this in our while loops.

$i = 0$

$i_{\text{max}} = 10$

while $i < i_{\text{max}}$:

do something

$i = i + 1$ # increment i

Does something from $i=0$ to $i=9$

or

$i = 0$

$i_{\text{max}} = 10$

while $i \leq i_{\text{max}}$:

do something

$i = i + 1$

Does something from $i=0$ to $i=10$

which of the two you want depends
on the situation. The former is
more common.

Here is how you make a range that
goes from 0 to ?

range(0, 10) ← does not include 10
— usually what you want

For Loops

We could do everything we have been doing and much more with while loops, but there is another very popular way to do something 10 times in Python: the for loop. It looks like this:

```
for i in range(0, 10):  
    # do something
```

It is so common to start with 0 you are even allowed to abbreviate this as

```
for i in range(10):  
    # do something
```

You can also do

```
i_values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
for i in i_values:  
    # do something
```

The super-convenient thing about all three of these is that you don't have to create the variable i before you start writing the for loop and you don't have to include the $i = i + 1$ at the end of the loop.

Lists vs. Tuples vs. Ranges

It seems we have too many choices.
What guides your choice of
which to use?

It is somewhat stylistic, however, here
are some guidelines

* tuples

* * useful for putting a few
things together - often these
things don't have the same type -
example: ("Camaro", 1976)

* lists

* * usually contains large and variable
amounts of the same kind of
thing - example: [28868.80, 28634.88,
28703.38, 28583.68, 28745.09] -
those are the DJIA closings so far
this year, and after 1PM Pacific Time
another closing can be appended to the
list

* ranges

* * for fixed ranges of integers - very
efficient because they aren't actually
all stored in memory - goes from
example: range(0, 100) ← 0 to 9