1 Administrative Topics

- We take the quiz.

2 Cool things about Parameters

There are two neat things we can do with parameters:

1. When defining a function, we can assign default values to parameters.

2. When calling a function, we can supply arguments (i.e. parameter values) using keyword instead of positional notation.

2.1 Default Values

For example, I wrote a function that would let me place my turtle wherever I want it. The function also allows the caller to specify an initial heading (which I call orientation). But it doesn’t require an initial heading. If none is given, it will use 0 as its default value.

Here is the code:

```python
# place the turtle at (x, y), heading towards the given angle.
def place(x, y, orientation=0):
```

```
turtle.up()
turtle.goto( x, y )
turtle.seth( orientation )
turtle.down()

Here are three example calls:

1. place( 0, 100 )
2. place( 10, 15, 0 )
3. place( 200, 0, 90 )

In line 1, no orientation is supplied so the default value of 0 is used.

In line 2, 0 is supplied for the orientation. This happens to be the same as the default value. But that doesn’t matter.

In line 3, 90 is supplied for the orientation. So, 90 will be used.

2.2 Keyword Parameters

Until now, we have been using what is called “positional” notation. When we call a function, the first argument is used to set the first parameter, the second argument is used to set the second parameter, etc. So, the first value that we pass to place is used to set x, the second value is used to set y, and the third value (if there is one) is used to set orientation.

But keyword notation lets us specify the parameter directly, e.g.

place( x = 0, y = 10, orientation = 45 )

When you use keyword notation, you can put the parameters in any order, e.g.

place( orientation = 45, x = 0, y = 10 )

This is helpful when you are calling a function with a lot of parameters and you know their names (because the names were well-chosen), but not the order (because there is no obvious or natural order). Also, it is super useful when you are calling a function with lots of parameters with default values. You can pick and choose which parameters you want to set.

Also, you can mix positional notation with keyword notation, but once you being using keywords, you can’t stop.
place( 0, 0, orientation = 20 )
place( 0, y=100, orientation = 90 )
place( 0, orientation=90, y=100 )

3 Advice for Project 7

- To add leaves or berries to trees, add a symbol (e.g. L for leaf and B for berry) to the L-system. I put my leaves just before each ] because I assume that is when the branch is ending. But you can play around with leaf placement.

- Then, add support for each new symbol in drawString. This code can be simple or complex, but obey this rule: LEAVE THE TURTLE AS YOU FOUND IT! I mean this: if you change the line thickness or color, then record the line thickness or color in a variable first. Then do your drawing. Then, set the values back.

- I added place to interpreter.py. You could add position/orientation parameters to drawString and make it call place at the beginning (just like block in project 3)