Lecture 5

0 Queries

1 More Loops

Let's write more loops! In particular, let's write a loop that will continue until the user wants it to stop. (Ask them how to do it!)

```
def do_you_want_to_continue():
    answer = input("Do you want to continue (y/n)? ")

    while answer == 'y':
        answer = input("Do you want to continue (y/n)? ")

    print("Goodbye!")

if __name__ == '__main__':
    do_you_want_to_continue()
```

Now let's write a function that takes a number as input. It then asks the user for values until they total more than that number. Then we'll print the total.

```
def count_until(num):
    total = 0
    while num > total:
        total += float(input("Enter a number: "))

    print("Total was", total)

if __name__ == '__main__':
    count_until(20)
```

2 Loops Practice

Give loops worksheet! Go over worksheet! For 5, spend some time with padding options.

3 Keyword Arguments (for print)

There have been many questions about how to print without the line break at the end. Let's talk about that! By default, Python always goes onto the next line after a print. But we can tell Python not to print the newline character at the end.

```
print("This has no line break!", end="")
```

Here I'm telling Python to print nothing (an empty string) after it prints what I told it to print. If I set `end` to be a newline character (as seen in Lab 1), I would get the same behaviour as before.

4 Nested Loops

We can have loops inside other loops! Let's look at a few and see if we can figure out what they do.

Lazy Import

There's another way of importing.

```
from math import sqrt
```

Upside: less typing, downside: pollute namespace

Project 2

Go over spec, show them what they get, show them executable. Talk about I/O testing.