Designing with loops

Work from the *inside out:*

1. What should this loop do **every time**?
2. How do we start and finish?

Example: summation

Project Euler

Example:

1. Ask a user to enter some numbers.
2. Add them up until the user enters an empty string.
3. Print the total.

After today's lecture, you will be equipped to tackle problem 16 in Project Euler. It's a puzzle, so expect to spend some time thinking about different ways to tackle it, but you will have everything you need to know in order to do it!

And remember: don't think like a *mathemetician*, think like a *programmer*.

Iteration

Walking through a bunch of things, *one at a time*

IRL: design *iterations*

Computing: loops!

- get "next" letter, use it
- ... until no more letters

```
for letter in 'Jonathan Anderson':
    print(letter)
```

Later: lists, arrays and `iter()`

Iteration problems

1. Count frequency of "e" in a string (upper- and lower-case)
2. For each letter in a string, count instances of that letter
3. Count upper-case characters in a string **(??)**

More string detail

What is a string?

What is a character?

A, B, C... 🧙‍♀️?

Some history: ASCII and Unicode

Python: `chr()` and `ord()`

How do you represent language to others?

- sounds
- writing

How would you transmit letters over a distance?

https://en.wikipedia.org/wiki/Telegraph_code#Electrical_telegraph_codes

Iteration problems

1. Count frequency of "e" in a string (upper- and lower-case)
2. For each letter in a string, count instances of that letter
3. **Count upper-case characters in a string**
4. Does a string have more upper- or lower-case characters?
5. Add up a string's letter values (A=1, B=2, etc.)
6. Compare two strings by alphabetical order