MAGIC NUMBERS

This is in the Code Quality entry, but it's a such a common issue in student code that it gets its own separate entry.

**Avoid magic numbers or unnamed numerical constants in your code**.

Look at this example - what's wrong?

# Passwords have to be at least 7 characters

if len(password) < 7:

   print('your password is too short')

What does 7 mean? Letters, penguins, waffles, what? We can figure it out by looking at the code but code is hard to understand, and adding more overhead to reading and understanding is not desirable. This is better,

MIN\_PASSWORD\_LENGTH = 7

if len(password) < MIN\_PASSWORD\_LENGTH:

  print('your password is too short')

This approach also makes it easier to re-use the value somewhere else.

And if you use the same value in multiple places in your code, if you need to change it, you can simple change the value of the variable; avoid bugs by not changing it in all the places.

And you can search your code for this variable name. Lots of advantages!

Typing a variable name is longer than 7, but we have autocomplete - so it's not that taxing to use variables, and the extra typing is outweighed by the advantages.

There's some exceptions, usually where a number has meaning by itself.

0 (and maybe 1) are usually exceptions to the rule. Here's an example,

if number\_of\_cards == 0:

   print('out of cards, you lose')

Zero means there's none of something, so it's usually more obvious what the intent is, in this example.

Another example could be 100 when used with percentages, or testing for 1 result returned from a database when querying by primary key.

More examples and discussion: <https://en.wikipedia.org/wiki/Magic_number_(programming)#Unnamed_numerical_constants>  <http://stackoverflow.com/questions/47882/what-is-a-magic-number-and-why-is-it-bad>