

Review Lecture

The assignments have been pretty much entirely graded, we'll see the results soon. None were complete disasters.

We should have them this afternoon. We can email him with any questions we have.

Model View Controller

What It Is

MVC is a design pattern. It's a way of organising code.

It simplifies the coding process by having a name for a particular structure, rather than exhaustively describe the entire architecture. People can just say e.g. "I used an MVC design pattern for this code", instead of explaining the whole thing.

MVC is mainly for UIs. It separates an application into a Model, a View, and a Controller.

The Model

The model contains all of the data-related logic of a program. The calculate class in the calculator assignment was the model. It knew about operands and operators, and knew how to use these values to generate answers to simple calculations.

The View

The view is everything the user sees – the buttons and so on.

The Controller

The model should never interact directly with the view. The controller knows about the model and view and how they should interact in a particular application.

Advantages

When we organise code like this, we can replace the view (if we want to change the layout) without breaking the model. If we also keep all method calls in the view the same, we won't have to rewrite the controller either.

Using MVC makes our code simpler, modularised, and reusable.

What You Need to Know

You won't be asked at any point to write code that implements the MVC design pattern. Rather, they'll be questions looking for explanations about MVC.

Everything above this point is stuff we need to know, everything below isn't.

What You Might Like to Know

A typical way of arranging a program using an MVC design pattern is with a class for the model, a class for the view, and a class for the controller.

These classes would often be stored in separate files, so that you could replace import statements with new import statements in order to change the view or model or controller.

Regular Expressions

There are two parts to it that we need to know:

- The regular expression itself
- The Python implementation of regex

Regex

The regex itself won't be super complicated. A good resource would be tutorialspoint.com.

The Python Part

The Python part would be very similar to the examples we've seen, if it came up.

Compiling the regular expressions is recommended because it gives nice

exceptions if there are errors.

Example

Find all words ending in "ly".

```
import re

line = 'He was cautiously disguised but found rapidly by the
authorities'

pattern = re.compile(r'\w+ly\b')
matches = pattern.findall(line)

for match in matches:
    print(match)
```

Note: the 'r' before the string in the `compile` call makes it a raw string – special meanings (in Python) for characters are ignored.