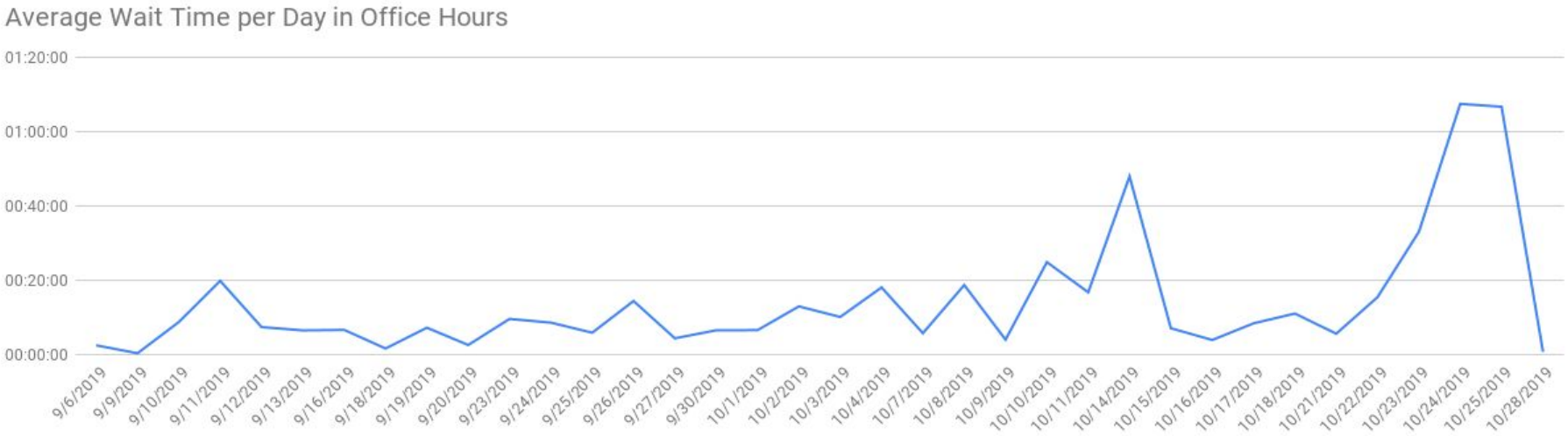
Dictionaries and other Things

October 29

Some Administrative Notes

- Project 2 is out. The Design is due THIS FRIDAY - November 1
  - The project is due NEXT FRIDAY - November 8
- Get the project done early!!



Project 2

We'll spend some time talking about this

- Understanding the game
  - See e.g.https://www.youtube.com/watch?v=_apsFik2JoQ
- Thinking about the design

More Administrative Notes

Test #2 is NEXT TUESDAY, November 5

The study guide has been updated. The test will include dictionaries.

We'll talk more about the test on Thursday

First, “Constants” vs. Variables

Constants are supposed to be just that - constant; unchanging - the same value for an entire program run.

We can change them in between program runs

Variables are things that allowed to change in a program.

True, unlike other languages, Python doesn’t really support Constants, but we construct them in 201

- Declare at the beginning of the program; use all caps; ensure you don’t change them; ...

What that means to your coding

Constants should be declared at the beginning of your code so they can be used throughout

- Constants are GLOBAL in scope

Variables should be used only in the scope you need them

- If you need a variable in the main program, instantiate & assign it in the main program
- If it's also needed in a functions, pass it as an argument/parameter in the function call

If a variable is only needed in a function, instantiate & assign it in the function, not the main program

DO NOT USE GLOBAL VARIABLES!!

Why not? They make programming more convenient!!

But they also make bugs more likely to happen, and harder to find

- You’re not sure where you’re manipulating that variable, so it’s hard to trace what’s happening

They also make your code less reusable

- You can’t pick up a function definition and use it in another program via “import” as easily - what if you don’t import all the code that impacts the variables?

Summary

Constants are GLOBAL in scope

Do not use global variables in this class!!

Mutability - an example

Credit to one of Dr. Johnson's students

```
def create_new_2d_list(height, width):
    row = []
    matrix = []

    for i in range(width):
        row.append(0)

    for i in range(height):
        matrix.append(row)
    return matrix
```

```
def pretty_print_matrix(matrix)
        for i in range(len(matrix)):
                print(matrix[i])
        return

if __name__ == '__main__':
    matrix = create_new_2d_list(8, 2)
    pretty_print_matrix(matrix)
    matrix[0][0] = 1
    matrix[4][1] = 5
    pretty_print_matrix(mat)
```

When would you use dictionaries?

When you have a limited number of keys that you want to uniquely map to a value or set of values.

- “Limited” doesn’t mean “small” - there might be a large number of keys.

Keys must be of an immutable type!!!

Values can be of any type