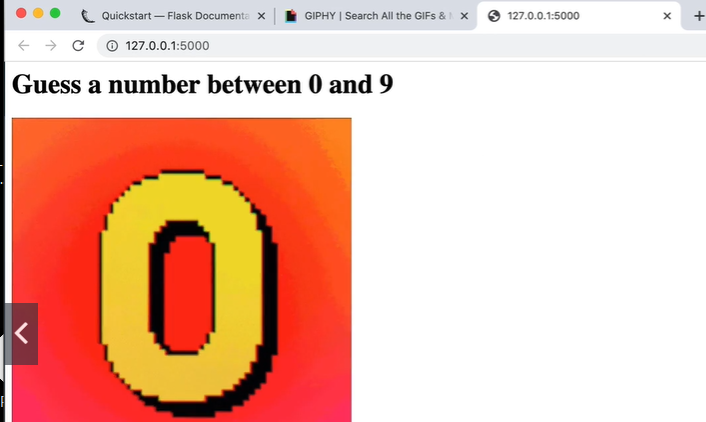
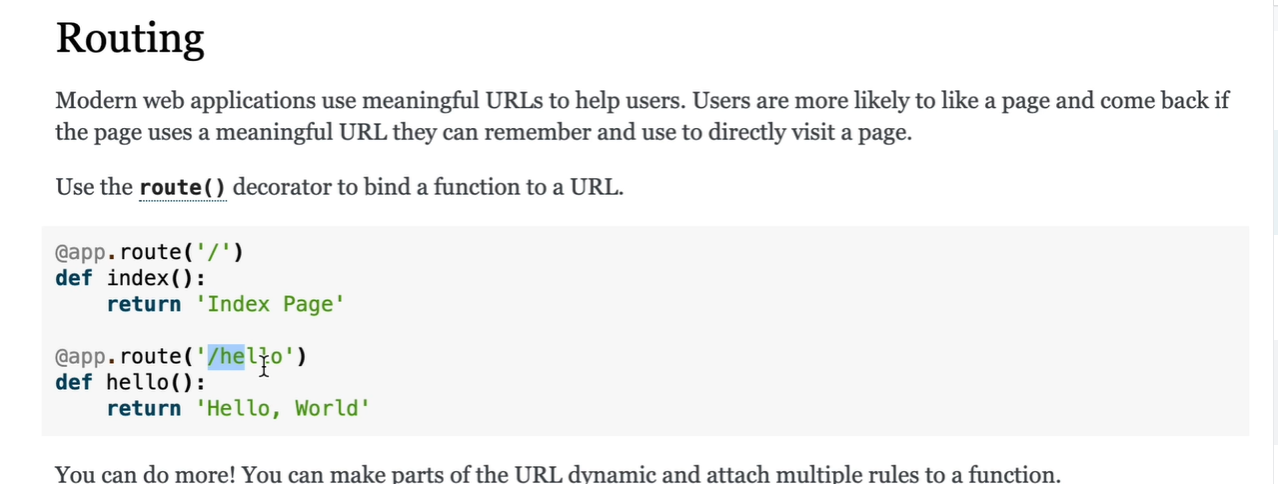
Day 55 HTML and URL parsing in Flask and the higher lower game

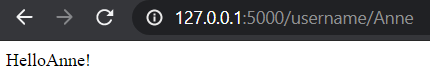
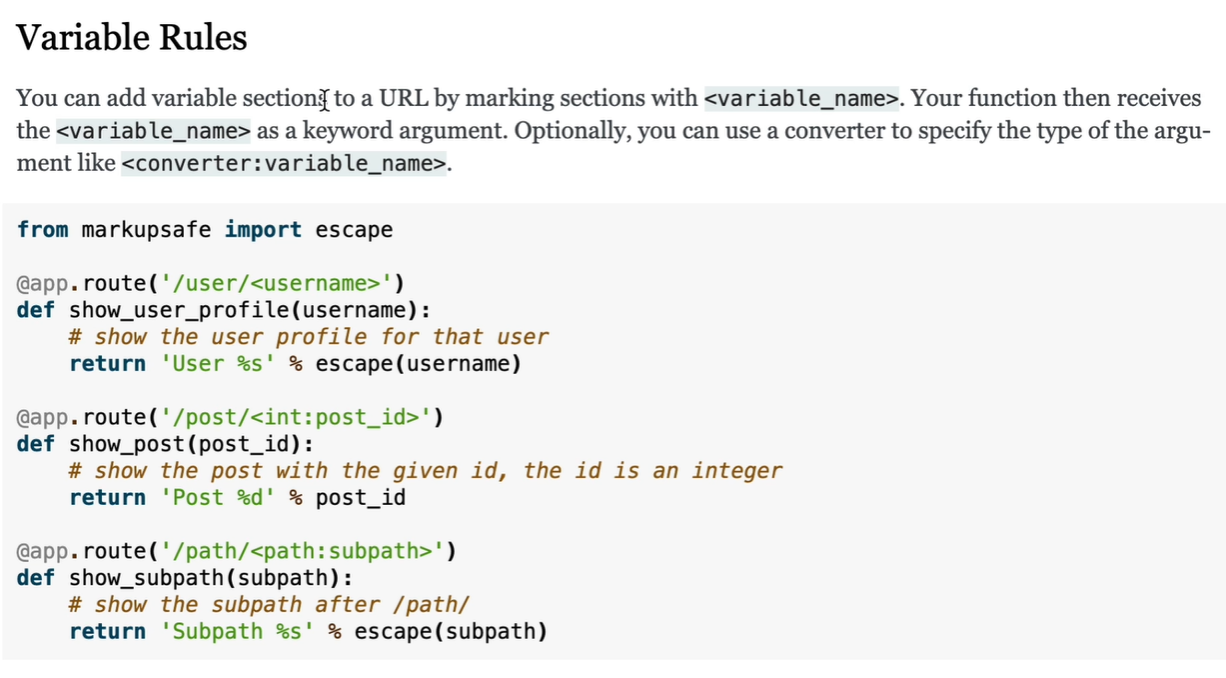


Parsing URL –

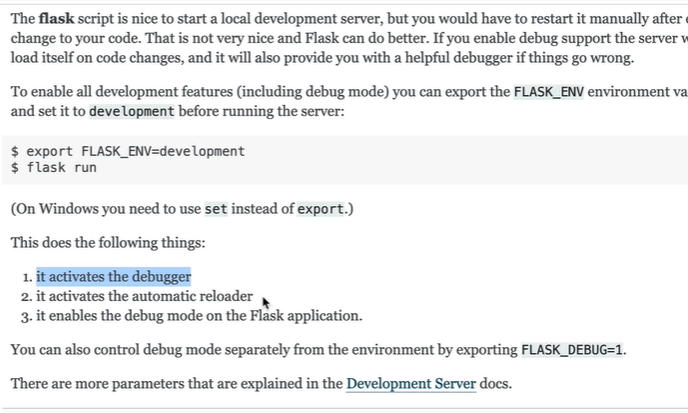
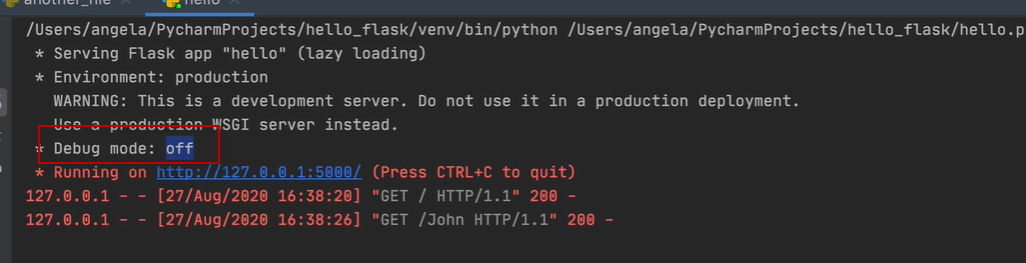
<https://flask.palletsprojects.com/en/1.1.x/quickstart/#routing>



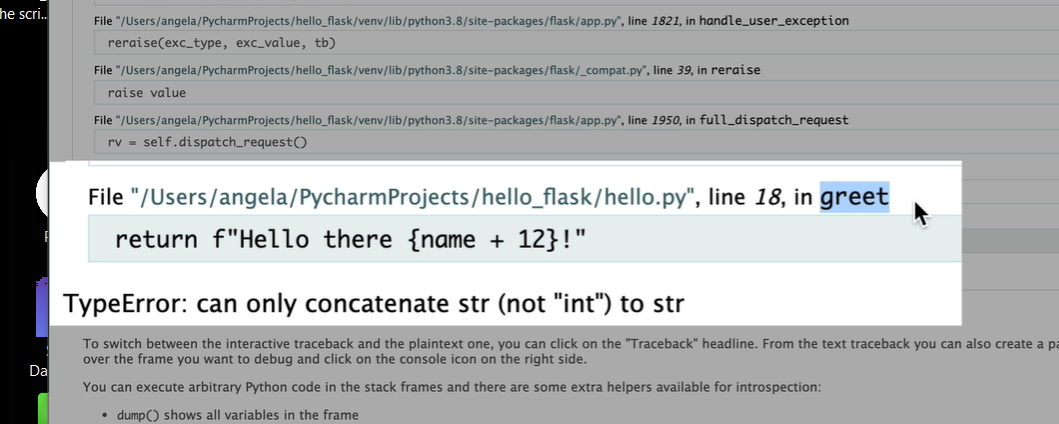
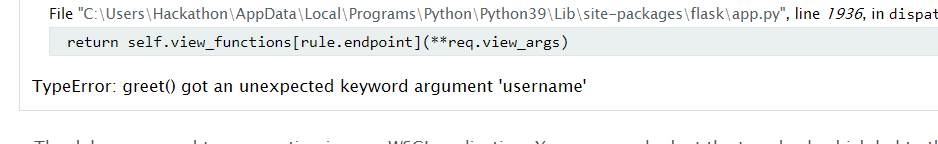
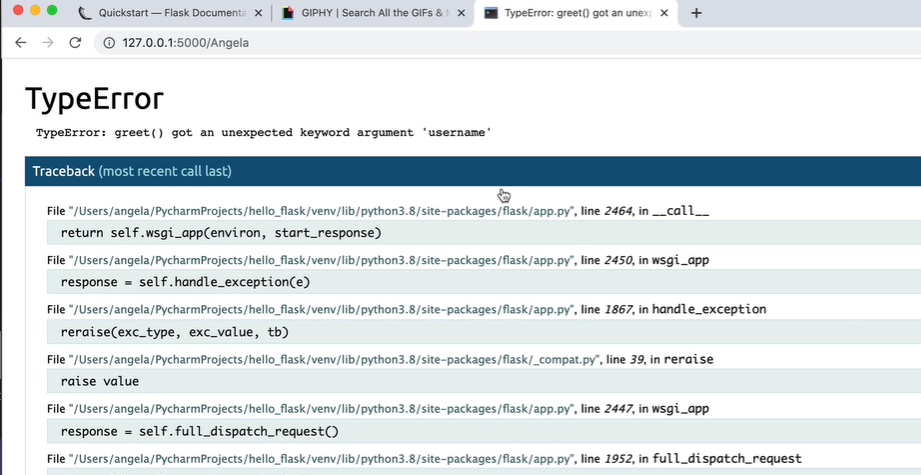
<https://flask.palletsprojects.com/en/1.1.x/quickstart/#variable-rules>



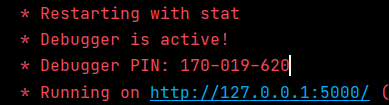
Using debug mode to not have to keep re starting server

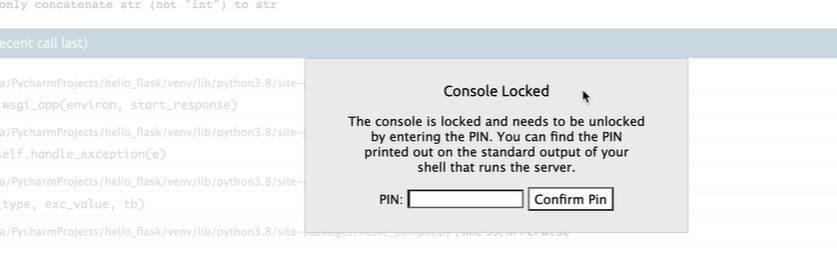


Flask debug mode



If site live on internet with debugger on prevents someone from messing with code since they will not have PIN



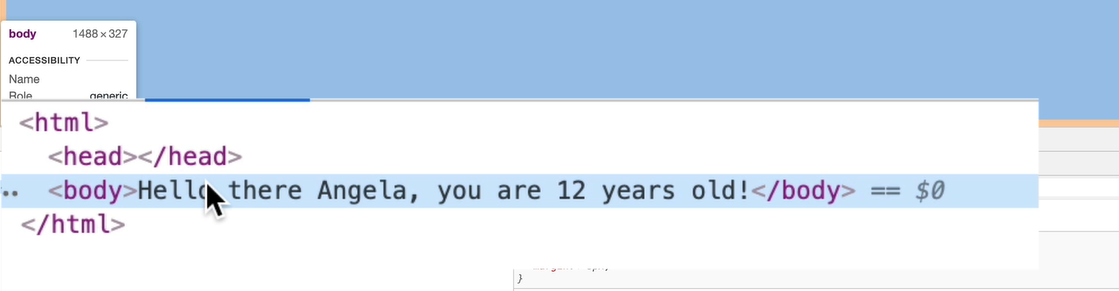


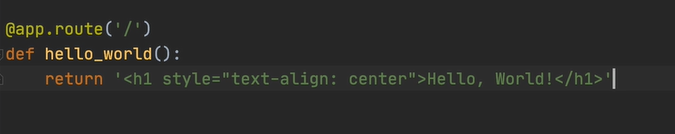
The converter changes data type to what is specified



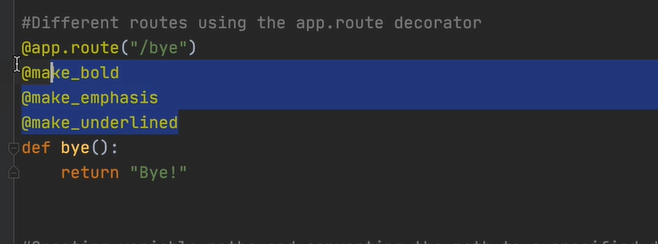
<https://giphy.com/>

This is what flask does by default



Can also put HTML and in line css  in the return

Use python decorators to style HTML Tags



HTML decorators

*def* make\_bold(*function*):

*def* wrapper():

        return "<b>" + function() + "</b>"

    return wrapper

*def* make\_emphasis(*function*):

*def* wrapper():

        return "<em>" + function() + "</em>"

    return wrapper

*def* make\_underlined(*function*):

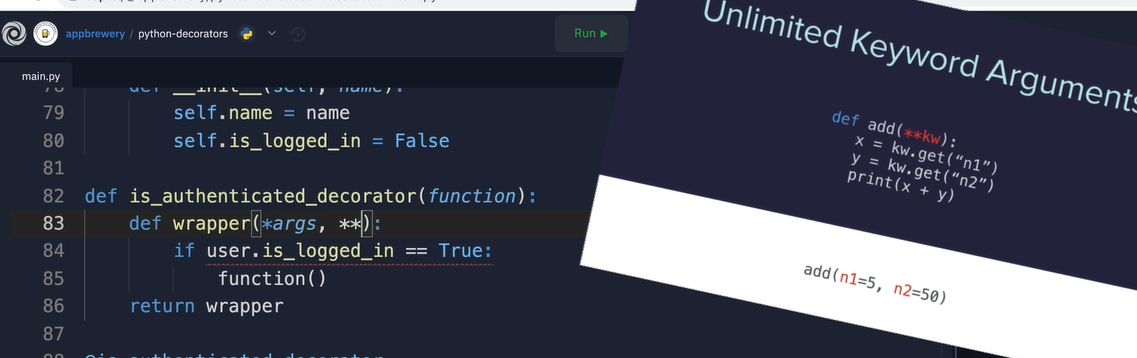
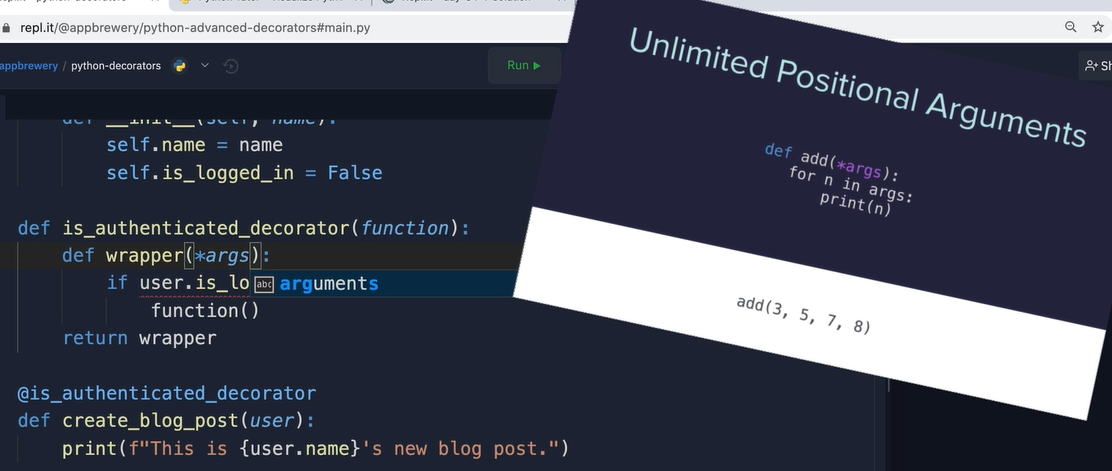
*def* wrapper():

        return "<u>" + function() + "</u>"

    return wrapper

Advanced decorators with \*args and \*\*kwargs

Get a hold of a function inside a decorator



Final Project - Higher or Lower URLs

Now it's time to complete the final project of the day, the higher lower game that we created in Day 14, but now with a real website.

1. Create a new project in PyCharm called **higher-lower**, add a **server.py** file.

2. Create a new Flask application where the home route displays an <h1> that says "Guess a number between 0 and 9" and display a gif of your choice from [giphy.com](https://giphy.com/).

Alternatively use the one I found on Giphy: https://media.giphy.com/media/3o7aCSPqXE5C6T8tBC/giphy.gif

3. Generate a random number between 0 and 9 or any range of numbers of your choice.

4. Create a route that can detect the number entered by the user e.g "URL**/3**" or "URL**/9**" and checks that number against the generated random number. If the number is too low, tell the user it's too low, same with too high or if they found the correct number. try to make the <h1> text a different colour for each page.  e.g. If the random number was 5:

3 is too low:

7 is too high:

and 5 is just right:

Here are the GIF URLs I used, but it's way more fun finding your own on giphy.com

High: https://media.giphy.com/media/3o6ZtaO9BZHcOjmErm/giphy.gif

Low: https://media.giphy.com/media/jD4DwBtqPXRXa/giphy.gif

Correct: https://media.giphy.com/media/4T7e4DmcrP9du/giphy.gif

[SOLUTION](https://gist.github.com/angelabauer/26eb9190a094761a9f49b22e8ee4c0fb)