Big head

- A person’s hat size is obtained by measuring the circumference of the head just above the ears in inches and dividing by $\pi$.
- Up to now, a program to take in circumferences and output hat sizes would consist of two files, e.g.:
  - head.html: an HTML web page that contains a form
    ```html
    <form action="hat.py" method="get">
      <input type="text" name="circumference" />
      <input type="submit" />
    </form>
    ```
  - hat.py: a Python program that receives the data, carries out the calculation, and prints the result into a new web page.

Self-processing page

- We can instead write this using a self-processing page:
  - a single Python program, e.g., head_to_hat.py
  - when requested without user data, it outputs the form
  - when requested with user data, it receives the data, carries out the calculation, and outputs the form again but with the result included.
Pseudocode

if the user has not sent us any data:
    output a web page containing the form
else:
    calculate the person's hat size
    output a web page containing the form and the hat size

Or:

set the hat size to 0
if the user has sent us some data:
    calculate the person's hat size
    output a web page containing the form and the hat size

head_to_hat.py: a self-processing page

Complete the program:

```python
#!/usr/local/bin/python
from cgitb import enable
enable()
from cgi import FieldStorage
from math import pi
print('Content-Type: text/html')
print()
form_data = FieldStorage()
hat_size = 0.0
if
    print('**
<DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <title>Big head</title>
</head>
<body>
<form action="head_to_hat.py" method="get">
    <label for="circumference">Circumference: </label>
    <input type="text" name="circumference" id="circumference" />
    <label for="hat_size">Hat size: </label>
    <input type="text" name="hat_size" id="hat_size" value = "0" disabled />
    <input type="submit" />
</form>
</body>
<html>
```
Sticky forms

- A sticky form
  - If you submit some data using the form and the form is then re-displayed, the text fields are re-displayed with your data still in them
- Modify head_to_hat.py to make the form a sticky form

Private study

Modify head_to_hat.py so that it also validates the user's input