CS1116/CS5018

Web Development 2

Dr Derek Bridge

School of Computer Science & Information Technology
University College Cork

Body Mass Index (BMI)

- A controversial measure, defined as:

$$bmi = \frac{mass\_kg}{height\_m^2}$$

- Even more controversially, for Irish adults:

BMI range	Category
less than 18.5	underweight
18.5 to 25	'normal'
more than 25	overweight

`bmi.html`: a page with a form

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <title>BMI</title>
    </head>
    <body>
        <form action="bmi.py" method="get">
            <label for="mass_kg">Mass (kg): </label>
            <input type="text" name="mass_kg" id="mass_kg" />
            <label for="height_m">Height (m): </label>
            <input type="text" name="height_m" id="height_m" />
            <input type="submit" value="Calculate BMI" />
        </form>
    </body>
</html>
```

bmi.py: a program with bugs

```
#!/usr/local/bin/python3

from cgi import FiledStorage

print('Content-Type: text/html')

form_data = FieldStorage()
mass_kg = form_data.getfirst('mass_kg')
height_m = form_data.getfirst('height_m')
bmi = mass_kg / heiight_m * height_m
category = ''
if bmi < 18.5
    category = 'underweight'
elif bmi > 25
    category ='overweight
else
category = 'normal'

print("""
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <meta charset="utf-8" />
            <title>BMI</title>
        </head>
        <body>
            <p>
                Your mass in kg is %s. Your height in m is %s.
                Your BMI is %s. You are %s.
            </p>
        </body>
    </html>""" % (mass_kg, height_m, bmi, category))
```

Fixing the program

- One problem is that we can't see the error messages
  Q: Why?
- To remedy, insert after the first line:

```
from cgitb import enable
enable()
```

  I want you to **always** include these!
- In a real environment, you would comment them out just before putting the program onto your production Web server:

```
# from cgitb import enable
# enable()
```

  Q: Why?
- Now fix the program!

Form data always arrives as strings

- Suppose the user enters 82.1 and 1.65 resp.
  - `mass_kg` contains '82.1' (str), not 82.1 (float)
  - `height_m` contains '1.65' (str), not 1.65 (float)

  — even if you use HTML5's <input type="number">
- Python is a **strongly-typed** language
- It won't calculate with strings — they need to be converted