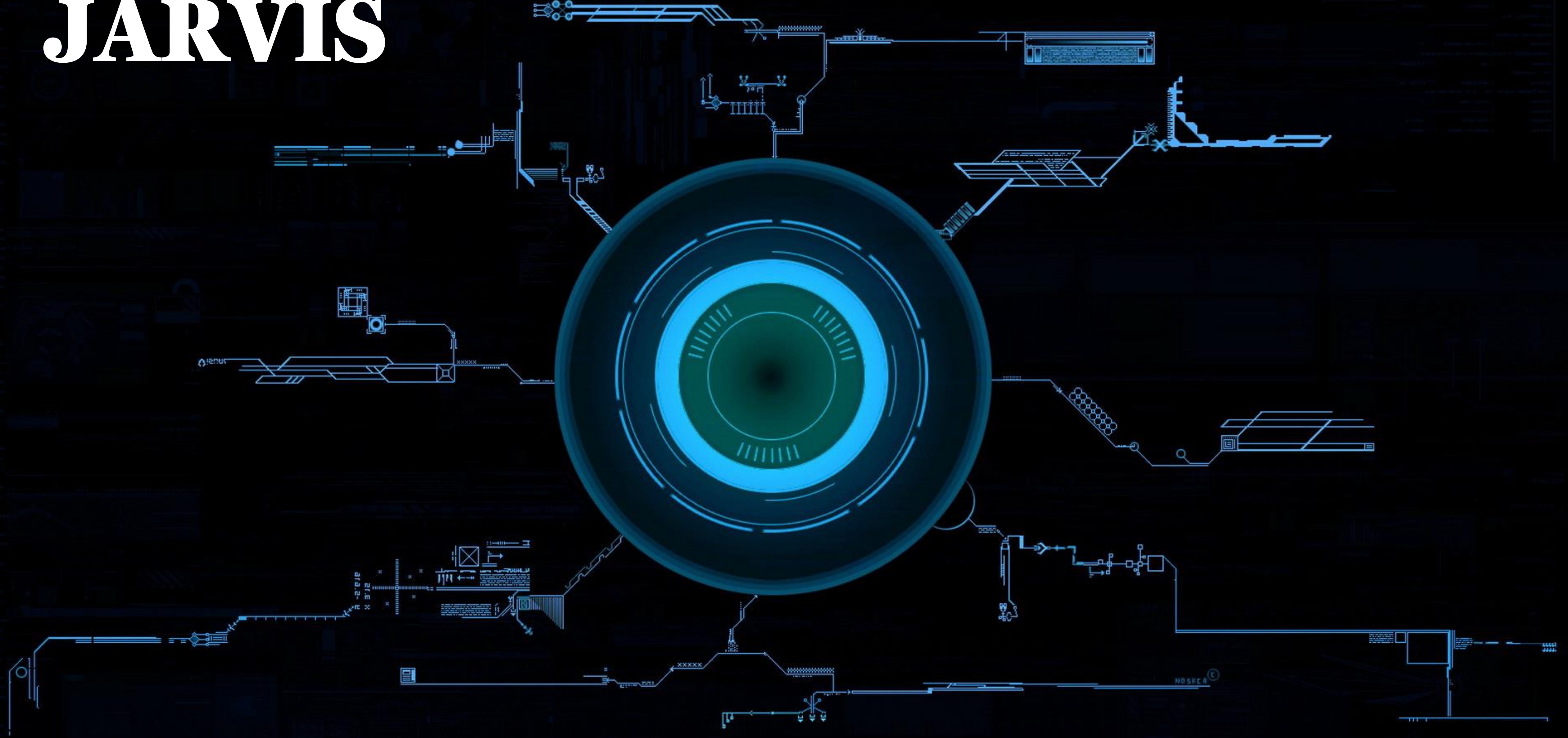
My First OSS Contribution

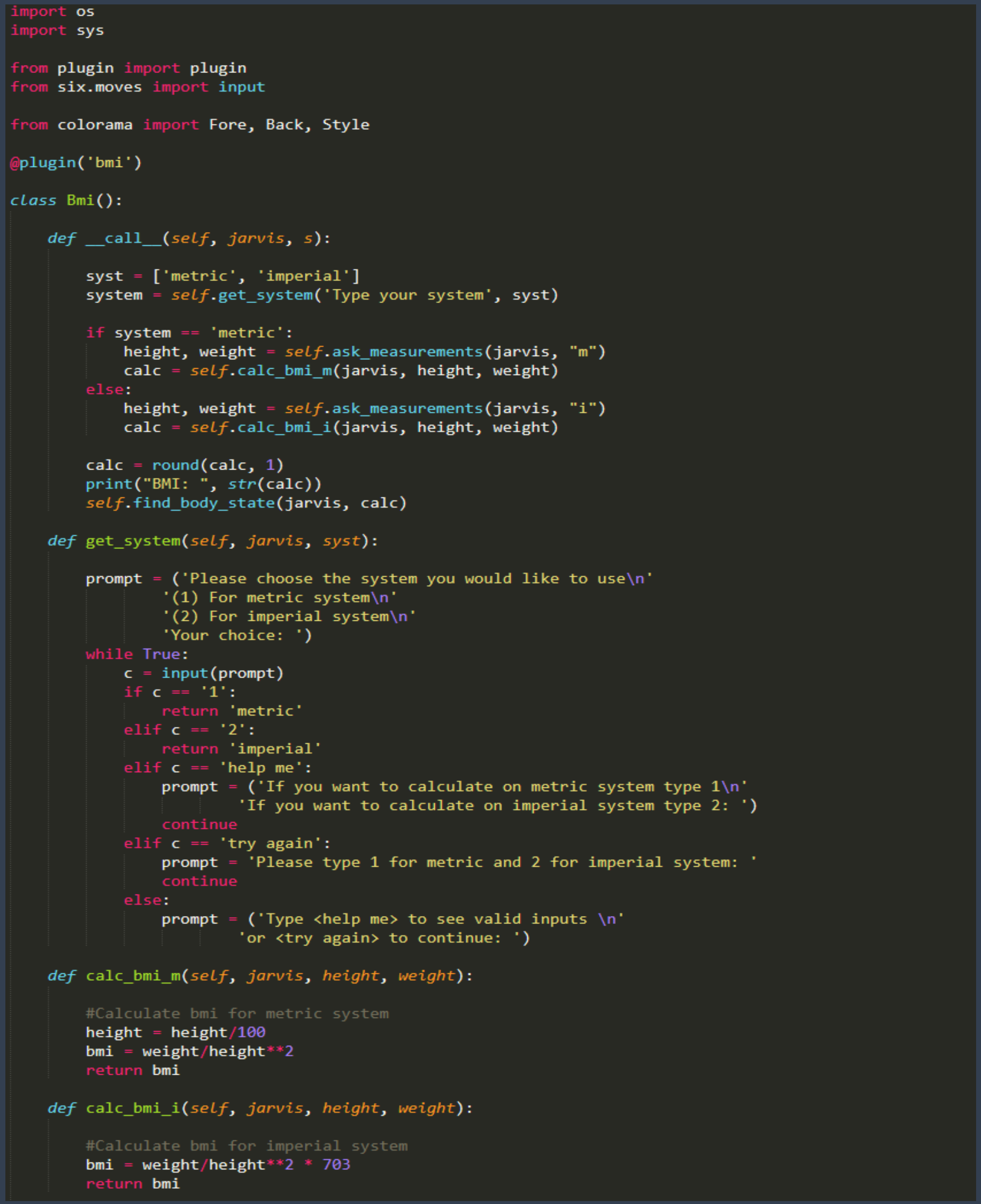
Final Presentation

Ιωάννης Πετρόπουλος, 8160107

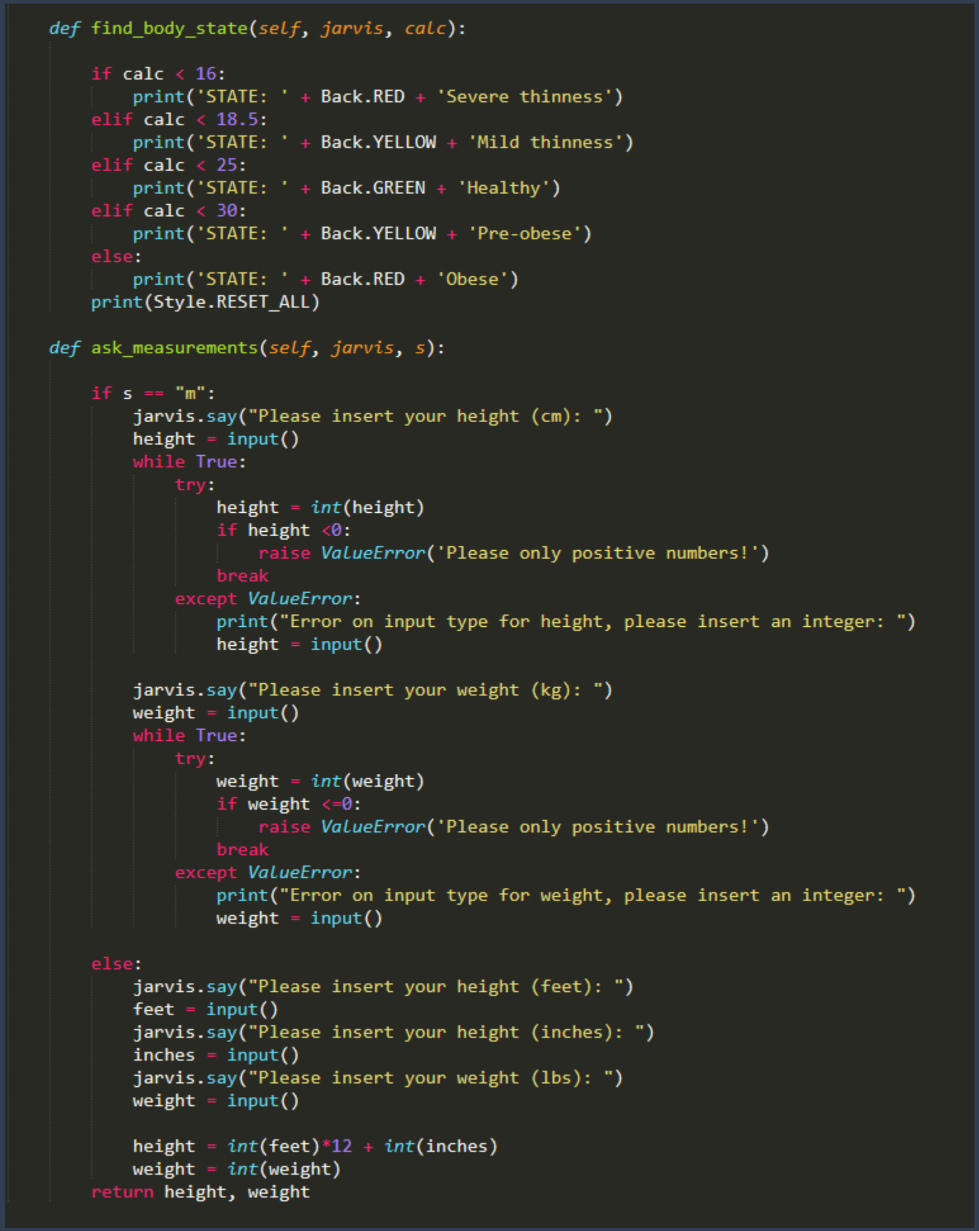
JARVIS



Main Plugin:



```python
import os
import sys

from plugin import plugin
from six.moves import input

from colorama import Fore, Back, Style

@plugin('bmi')

class Bmi():

    def __call__(self, jarvis, s):

        syst = ['metric', 'imperial']
        system = self.get_system('Type your system', syst)

        if system == 'metric':
            height, weight = self.ask_measurements(jarvis, "m")
            calc = self.calc_bmi_m(jarvis, height, weight)
        else:
            height, weight = self.ask_measurements(jarvis, "i")
            calc = self.calc_bmi_i(jarvis, height, weight)

        calc = round(calc, 1)
        print("BMI: ", str(calc))
        self.find_body_state(jarvis, calc)

    def get_system(self, jarvis, syst):

        prompt = ('Please choose the system you would like to use\n'
                  '(1) For metric system\n'
                  '(2) For imperial system\n'
                  'Your choice: ')
        while True:
            c = input(prompt)
            if c == '1':
                return 'metric'
            elif c == '2':
                return 'imperial'
            elif c == 'help me':
                prompt = ('If you want to calculate on metric system type 1\n'
                          'If you want to calculate on imperial system type 2: ')
                continue
            elif c == 'try again':
                prompt = 'Please type 1 for metric and 2 for imperial system: '
                continue
            else:
                prompt = ('Type <help me> to see valid inputs \n'
                          'or <try again> to continue: ')

    def calc_bmi_m(self, jarvis, height, weight):

        #Calculate bmi for metric system
        height = height/100
        bmi = weight/height**2
        return bmi

    def calc_bmi_i(self, jarvis, height, weight):

        #Calculate bmi for imperial system
        bmi = weight/height**2 * 703
        return bmi
```



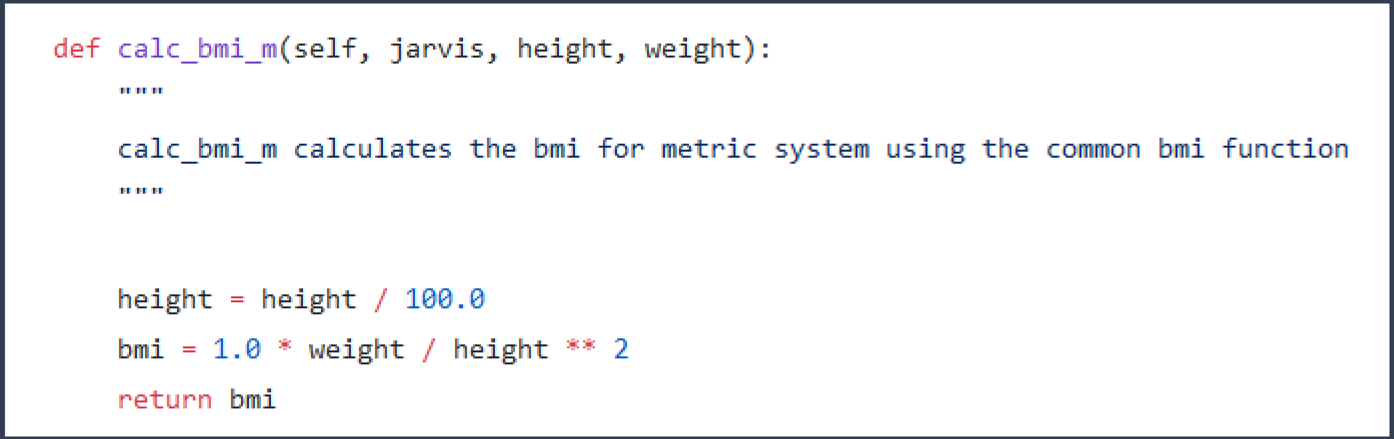
```python
def find_body_state(self, jarvis, calc):

    if calc < 16:
        print('STATE: ' + Back.RED + 'Severe thinness')
    elif calc < 18.5:
        print('STATE: ' + Back.YELLOW + 'Mild thinness')
    elif calc < 25:
        print('STATE: ' + Back.GREEN + 'Healthy')
    elif calc < 30:
        print('STATE: ' + Back.YELLOW + 'Pre-obese')
    else:
        print('STATE: ' + Back.RED + 'Obese')
    print(Style.RESET_ALL)

def ask_measurements(self, jarvis, s):

    if s == "m":
        jarvis.say("Please insert your height (cm): ")
        height = input()
        while True:
            try:
                height = int(height)
                if height <0:
                    raise ValueError('Please only positive numbers!')
                break
            except ValueError:
                print("Error on input type for height, please insert an integer: ")
                height = input()

        jarvis.say("Please insert your weight (kg): ")
        weight = input()
        while True:
            try:
                weight = int(weight)
                if weight <=0:
                    raise ValueError('Please only positive numbers!')
                break
            except ValueError:
                print("Error on input type for weight, please insert an integer: ")
                weight = input()

    else:
        jarvis.say("Please insert your height (feet): ")
        feet = input()
        jarvis.say("Please insert your height (inches): ")
        inches = input()
        jarvis.say("Please insert your weight (lbs): ")
        weight = input()

        height = int(feet)*12 + int(inches)
        weight = int(weight)
    return height, weight
```

PROGRESS

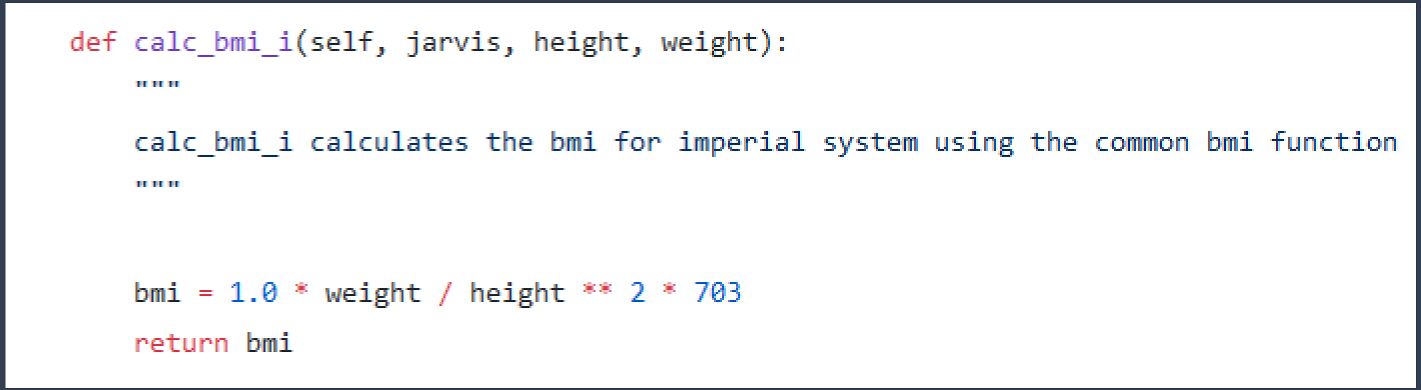
$\searrow$

UNIT TESTING

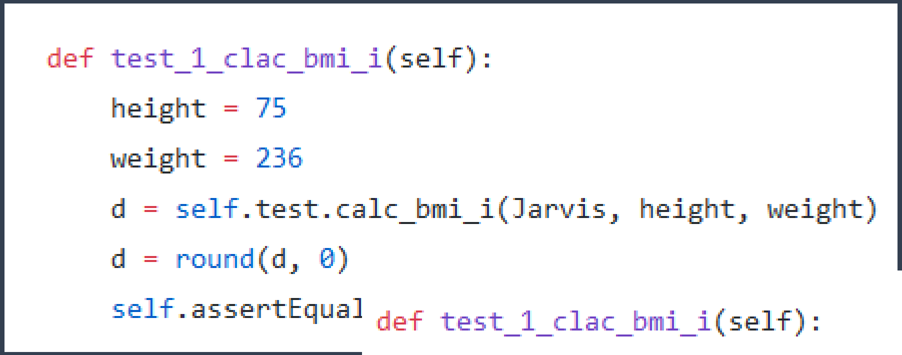
MAIN METHODS



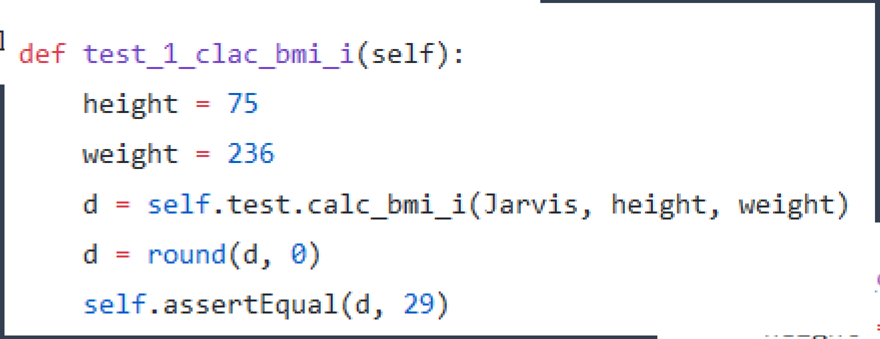
```python
def calc_bmi_m(self, jarvis, height, weight):
    """
    calc_bmi_m calculates the bmi for metric system using the common bmi function
    """

    height = height / 100.0
    bmi = 1.0 * weight / height ** 2
    return bmi
```



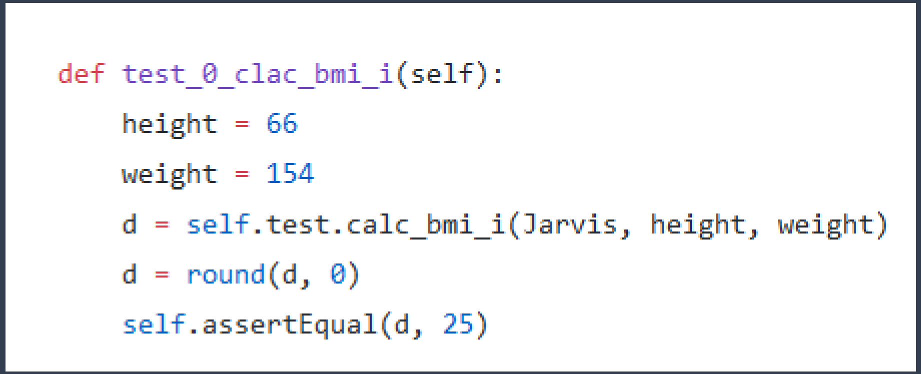
```python
def calc_bmi_i(self, jarvis, height, weight):
    """
    calc_bmi_i calculates the bmi for imperial system using the common bmi function
    """

    bmi = 1.0 * weight / height ** 2 * 703
    return bmi
```



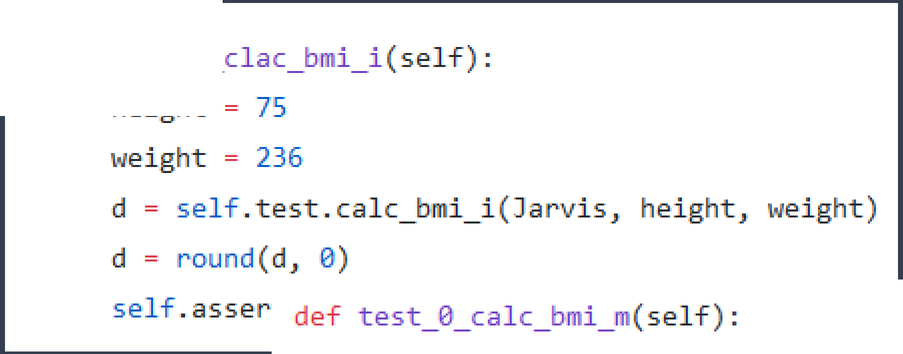
```
def test_1_clac_bmi_i(self):
    height = 75
    weight = 236
    d = self.test.calc_bmi_i(Jarvis, height, weight)
    d = round(d, 0)
    self.assertEqual
```



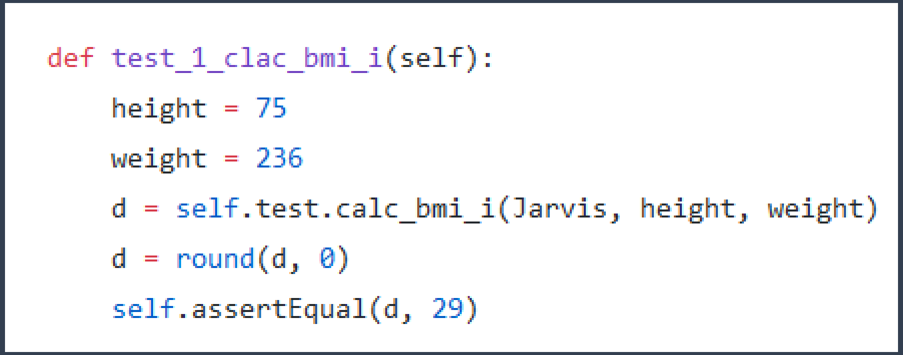
```
def test_1_clac_bmi_i(self):
    height = 75
    weight = 236
    d = self.test.calc_bmi_i(Jarvis, height, weight)
    d = round(d, 0)
    self.assertEqual(d, 29)
```



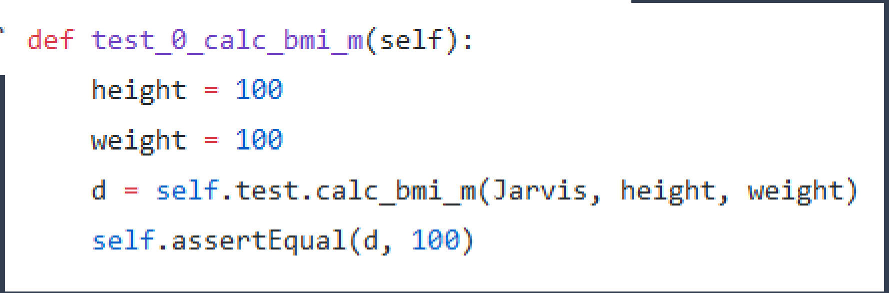
```
def test_0_clac_bmi_i(self):
    height = 66
    weight = 154
    d = self.test.calc_bmi_i(Jarvis, height, weight)
    d = round(d, 0)
    self.assertEqual(d, 25)
```



```
              clac_bmi_i(self):
              = 75
    weight = 236
    d = self.test.calc_bmi_i(Jarvis, height, weight)
    d = round(d, 0)
    self.asser
```



```
def test_1_clac_bmi_i(self):
    height = 75
    weight = 236
    d = self.test.calc_bmi_i(Jarvis, height, weight)
    d = round(d, 0)
    self.assertEqual(d, 29)
```

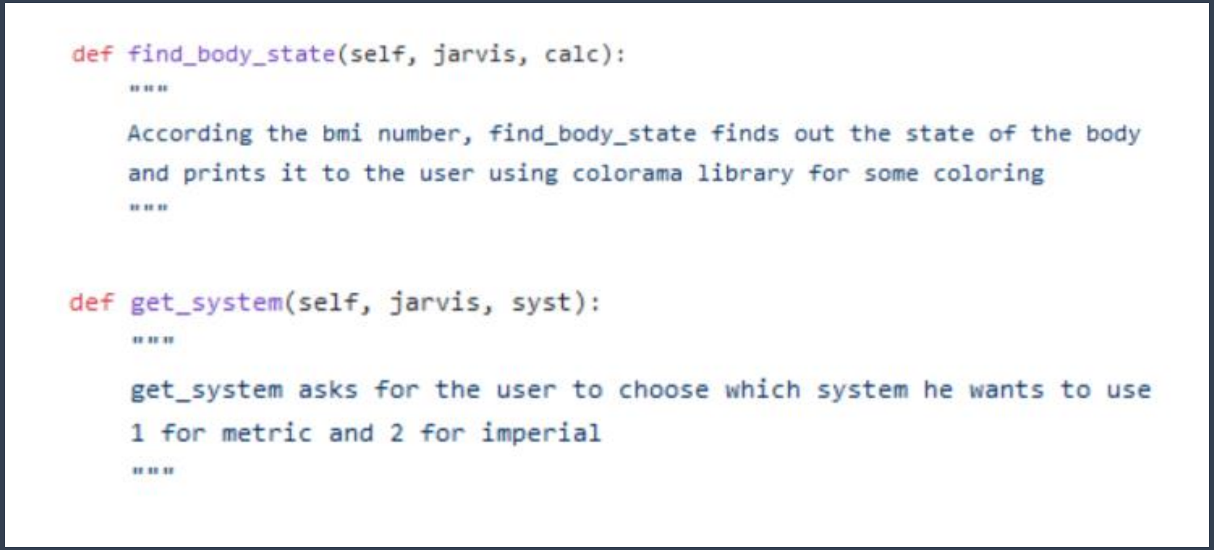


```
def test_0_calc_bmi_m(self):
    height = 100
    weight = 100
    d = self.test.calc_bmi_m(Jarvis, height, weight)
    self.assertEqual(d, 100)
```

IMPORT UNITTEST

QUALITY

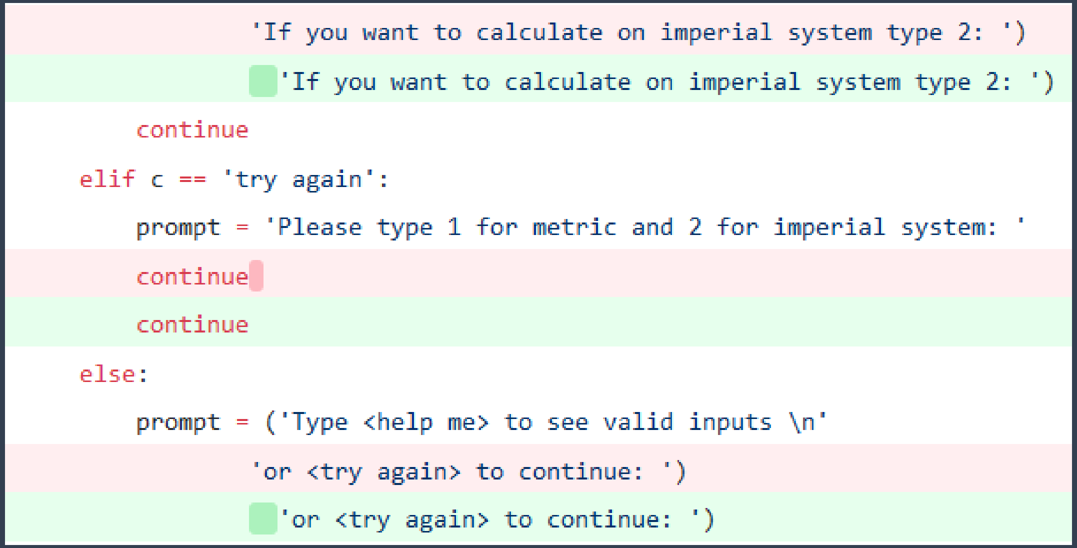
Documentation



```
def find_body_state(self, jarvis, calc):
    """
    According the bmi number, find_body_state finds out the state of the body
    and prints it to the user using colorama library for some coloring
    """

def get_system(self, jarvis, syst):
    """
    get_system asks for the user to choose which system he wants to use
    1 for metric and 2 for imperial
    """
```

Every Method is well documented

Fix linting errors

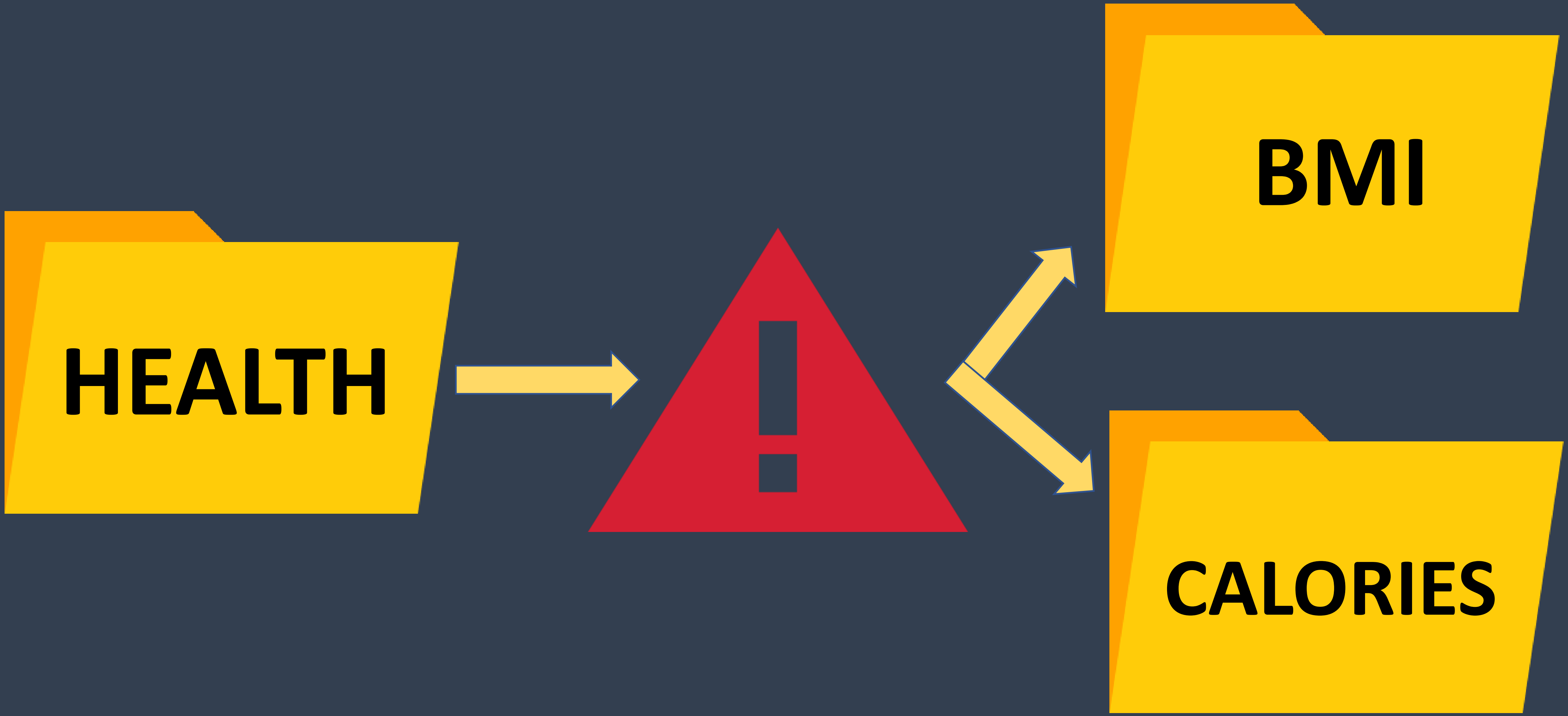


```
                   'If you want to calculate on imperial system type 2: ')
                    'If you want to calculate on imperial system type 2: ')
        continue
    elif c == 'try again':
        prompt = 'Please type 1 for metric and 2 for imperial system: '
        continue
        continue
    else:
        prompt = ('Type <help me> to see valid inputs \n'
                   'or <try again> to continue: ')
                    'or <try again> to continue: ')
```

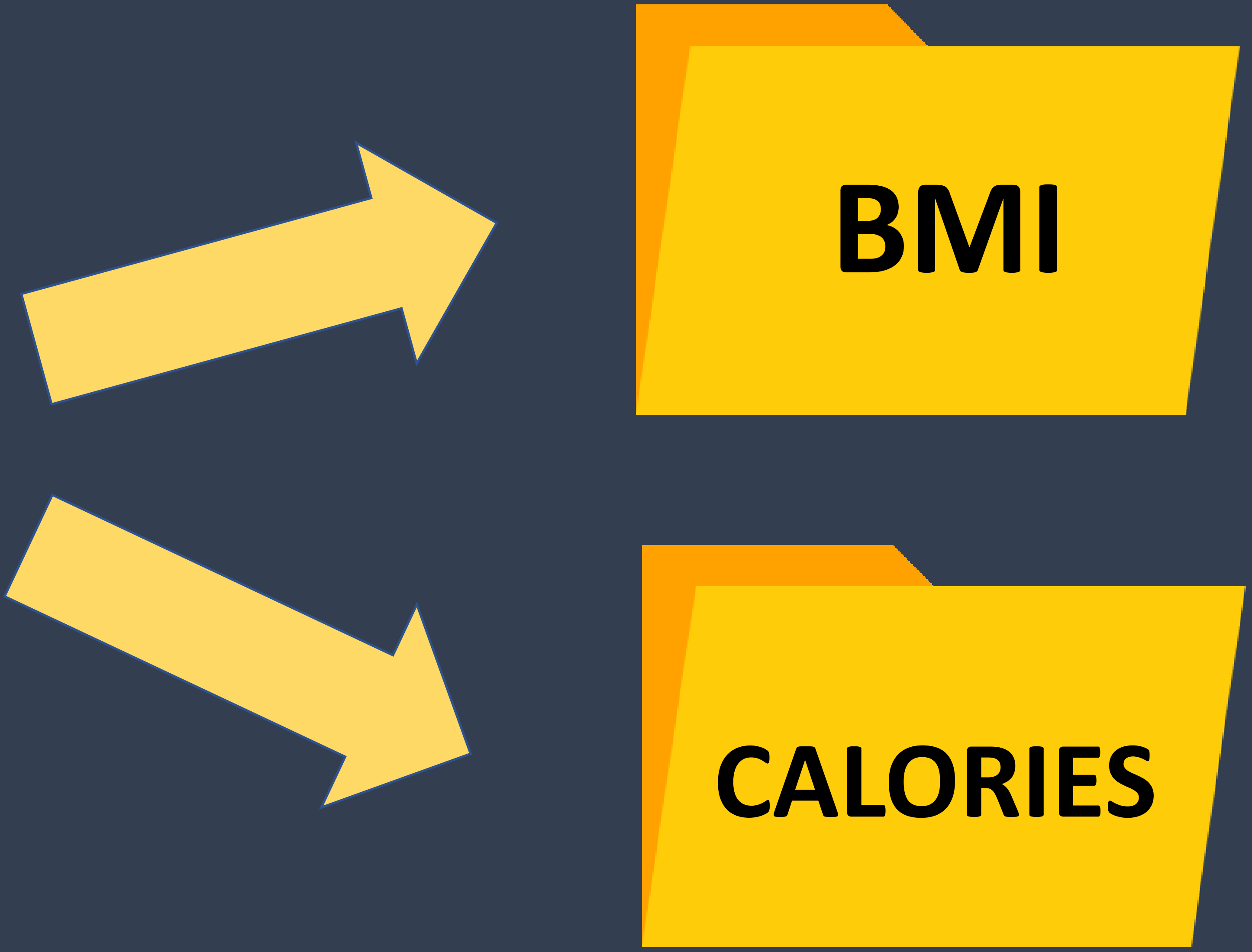
No linting errors found

Package Structure Refactoring

BEFORE

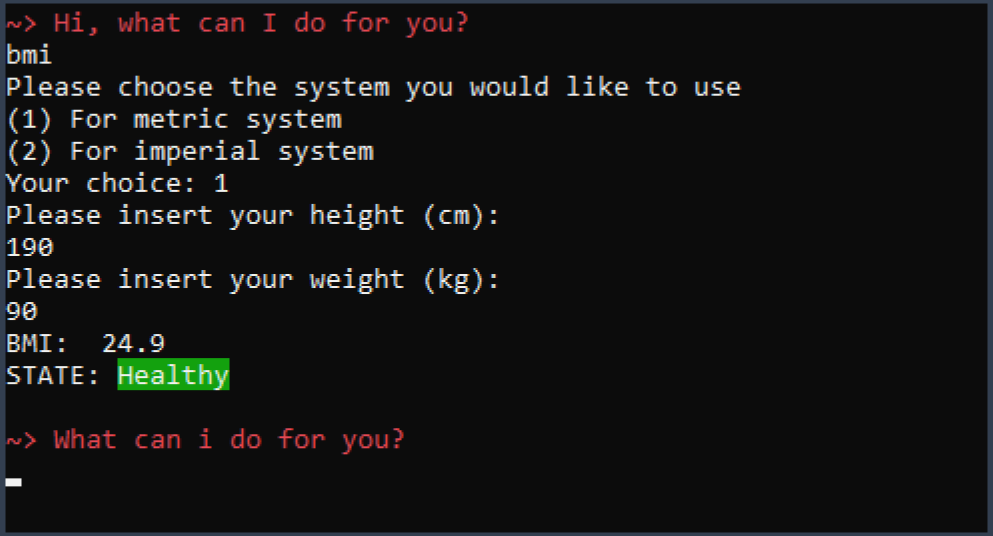


AFTER



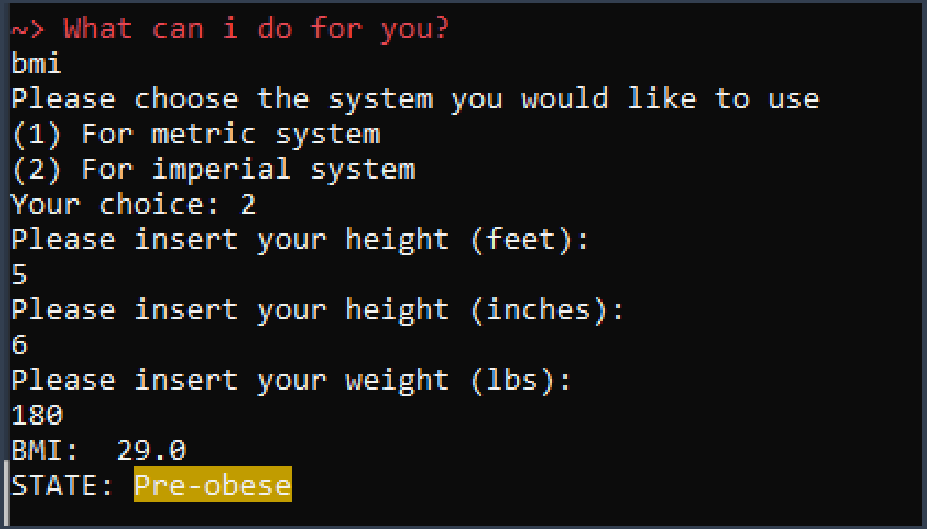
Results:

Metric System:



```
~> Hi, what can I do for you?
bmi
Please choose the system you would like to use
(1) For metric system
(2) For imperial system
Your choice: 1
Please insert your height (cm):
190
Please insert your weight (kg):
90
BMI:  24.9
STATE: Healthy

~> What can i do for you?
```

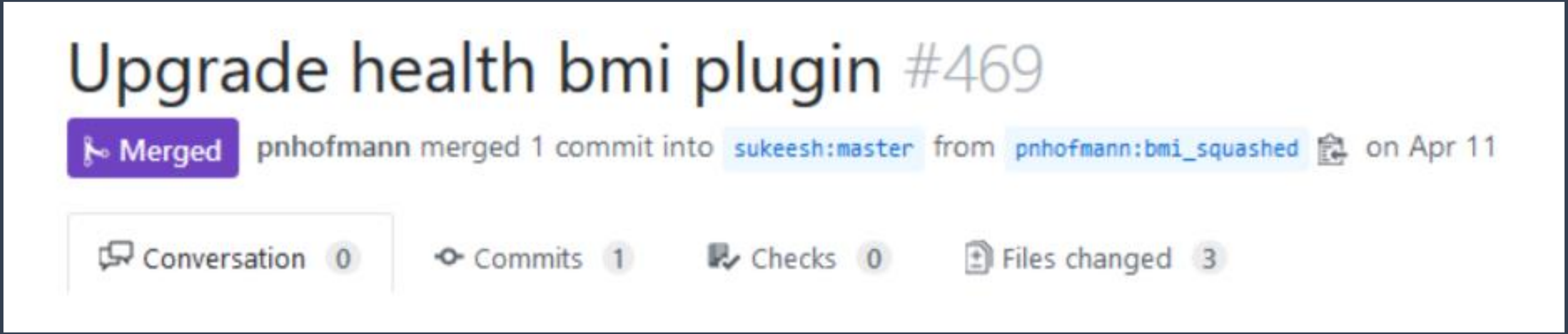
Imperial System:



```
~> What can i do for you?
bmi
Please choose the system you would like to use
(1) For metric system
(2) For imperial system
Your choice: 2
Please insert your height (feet):
5
Please insert your height (inches):
6
Please insert your weight (lbs):
180
BMI:  29.0
STATE: Pre-obese
```

✓ **FUNCTIONALITY**

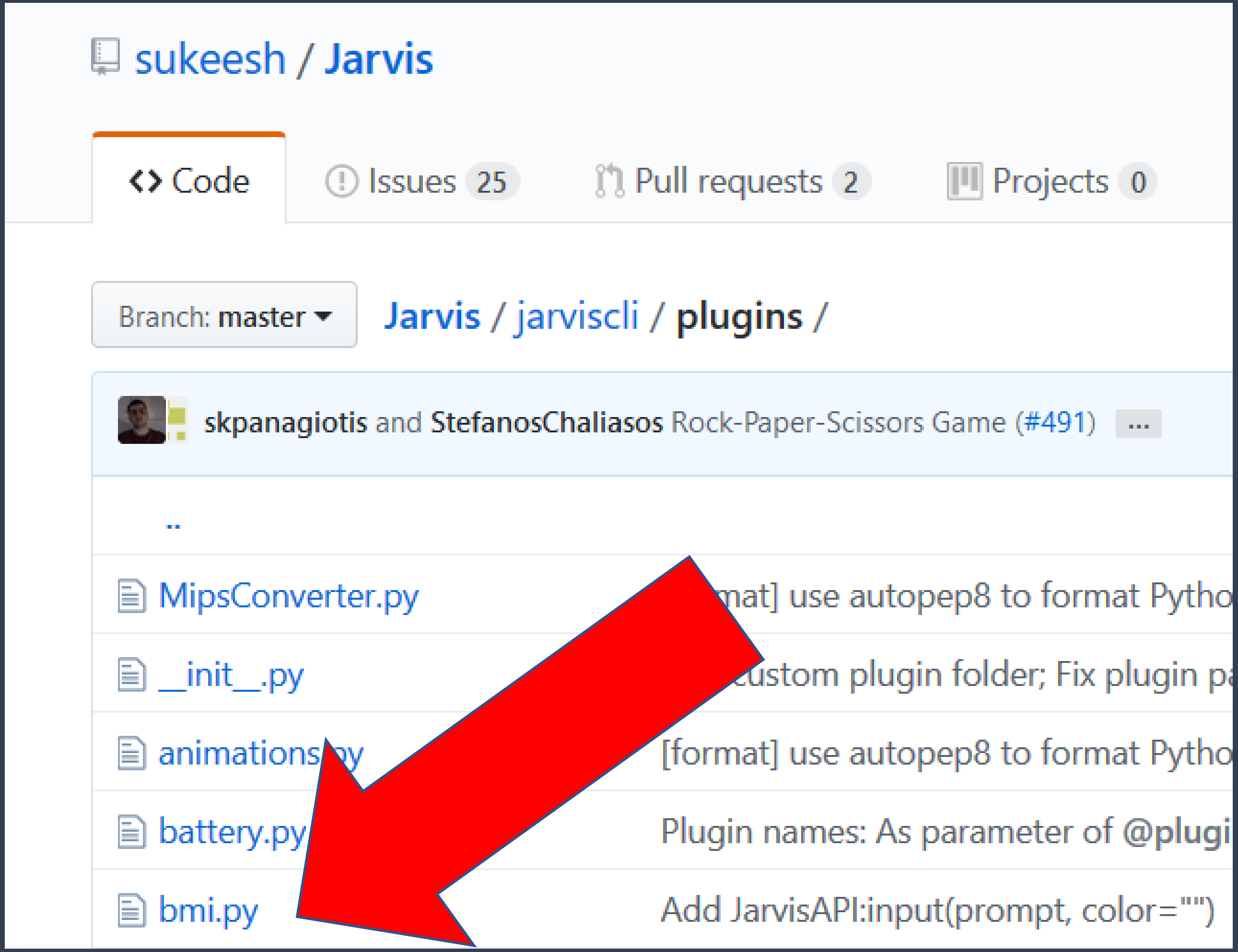
✓ **QUALITY**



Upgrade health bmi plugin #469

Merged pnhofmann merged 1 commit into `sukeesh:master` from `pnhofmann:bmi_squashed` on Apr 11

PROUD OF:



sukeesh / Jarvis

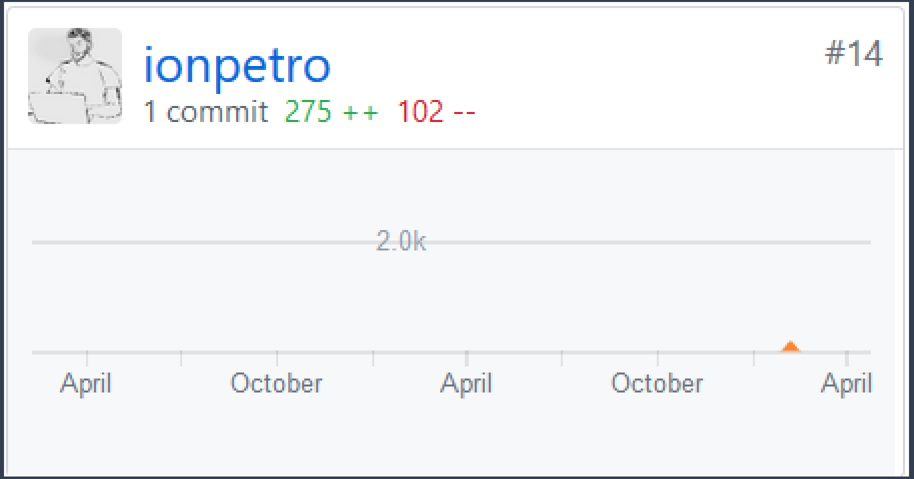
<> Code | Issues 25 | Pull requests 2 | Projects 0

Branch: master ▾ Jarvis / jarviscli / plugins /

skpanagiotis and StefanosChaliasos Rock-Paper-Scissors Game (#491) ...

..

MipsConverter.py	mat] use autopep8 to format Pytho
__init__.py	custom plugin folder; Fix plugin pa
animations.py	[format] use autopep8 to format Pytho
battery.py	Plugin names: As parameter of @plugi
bmi.py	Add JarvisAPI:input(prompt, color="")

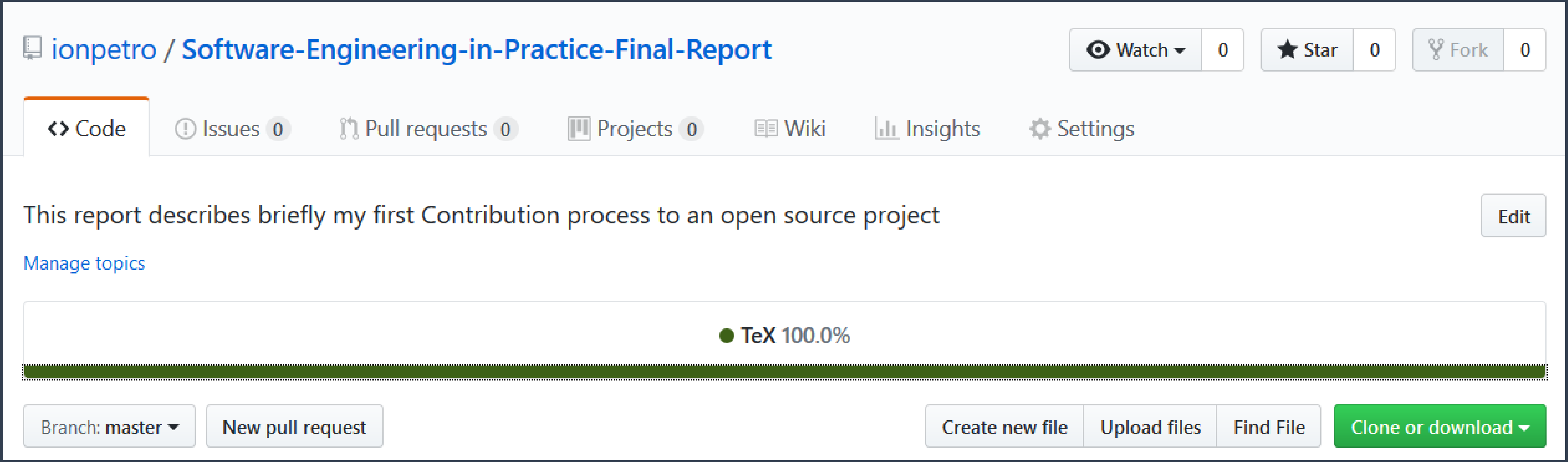


ionpetro #14

1 commit 275 ++ 102 --

Contributors of Jarvis

Read more about my contribution here:


ionpetro / Software-Engineering-in-Practice-Final-Report

This report describes briefly my first Contribution process to an open source project

Conclusion:

Software Engineering in Practice

Open Source Contributor

Thank you for your time!