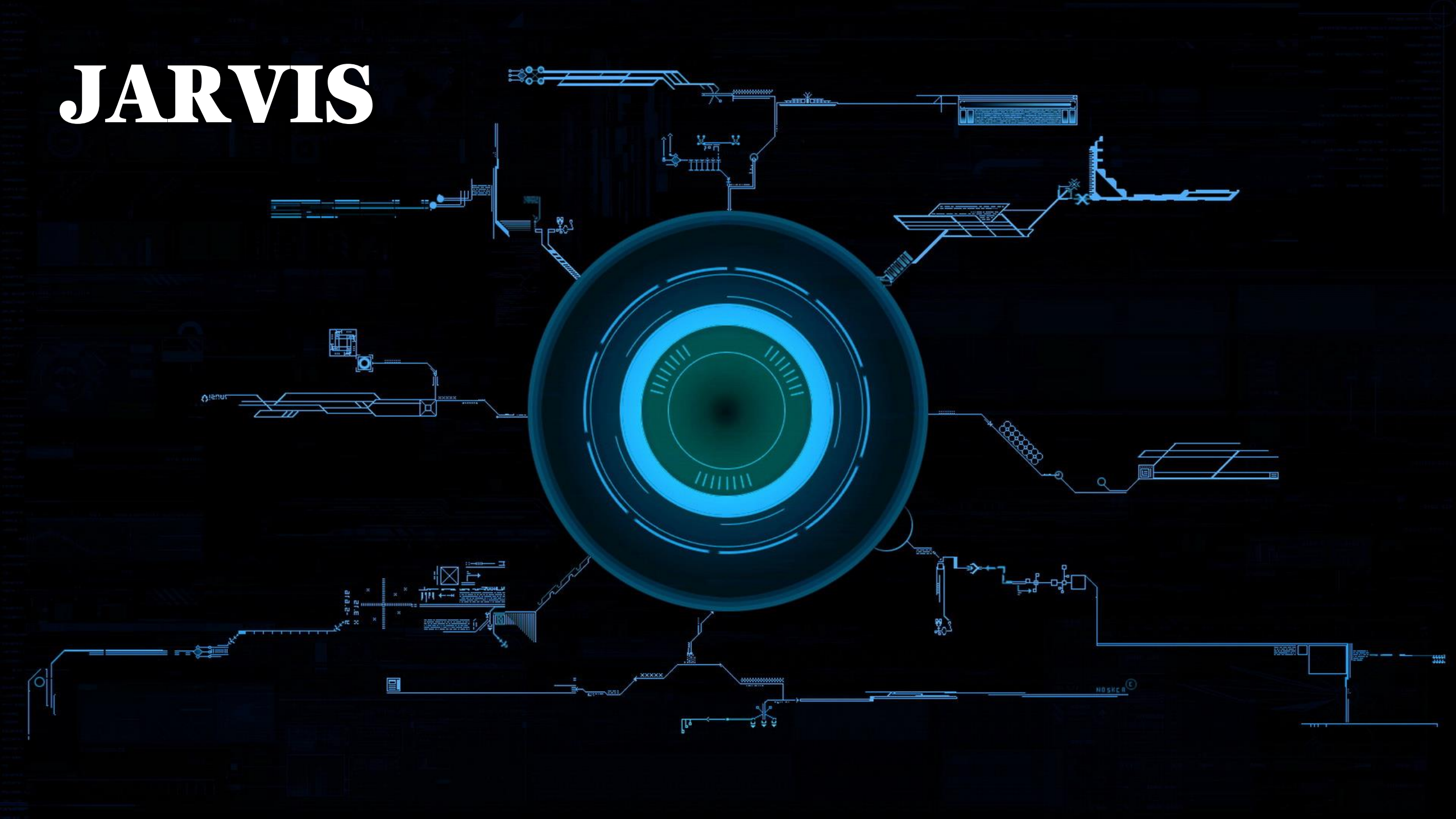
My First OSS Contribution

Part II

Ιωάννης Πετρόπουλος, 8160107

JARVIS

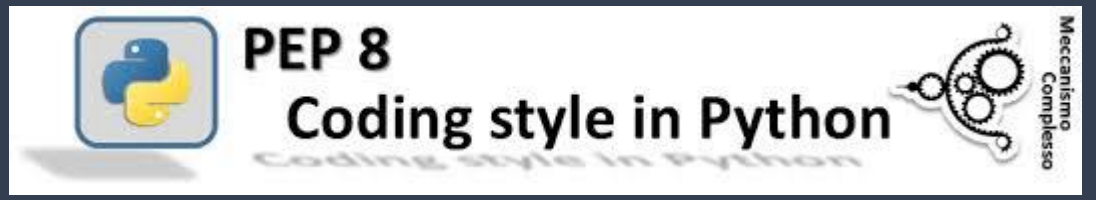


From previous episode:

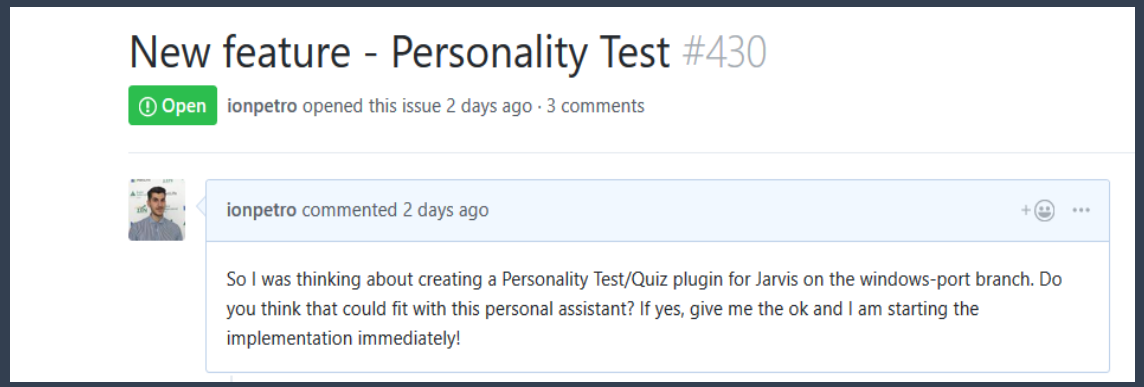
☑ BUILD

```
Command Prompt - Jarvis
  File "<frozen importlib._bootstrap>", line 265, in _load_module_shim
  File "<frozen importlib._bootstrap>", line 696, in _load
  File "<frozen importlib._bootstrap>", line 677, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 728, in exec_module
  File "<frozen importlib._bootstrap>", line 219, in _call_with_frames_removed
  File "C:\Users\ionpe\Documents\GitHub\Jarvis\jarviscli\plugins\imgur.py", line 6, in <module>
    import readline
ModuleNotFoundError: No module named 'readline'
Jarvis' sound is by default disabled.
In order to let Jarvis talk out loud type: enable sound
Type 'help' for a list of available actions.
71 plugins loaded 17 plugins disabled. More information: status

~> Hi, what can I do for you?
help

Documented commands (type help <topic>):
========================================
bmi         coin          enable     health  os        roll        tell
bye         cricket       equations  help    pinpoint  run         tempconv
calc        currencyconv  evaluate   ip      play      say         timer
calculate   curve         exit       limit   plot      shutdown    todo
cancel      curvesketch   factor     lyrics  q         solve       translate
check       dictionary    file       movie   quit      speedtest   umbrella
chuck       directions    gmail      music   quote     status      update
clear       disable       goodbye    near    reboot    stopwatch   weather
clock       display       hackathon  news    remind    systeminfo  wiki

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

☑ READ CODE



☑ OPENED ISSUE



New feature - Personality Test #430

Open ionpetro opened this issue 2 days ago · 3 comments

ionpetro commented 2 days ago

So I was thinking about creating a Personality Test/Quiz plugin for Jarvis on the windows-port branch. Do you think that could fit with this personal assistant? If yes, give me the ok and I am starting the implementation immediately!

☑ WROTE PLUGIN

```python
from plugin import plugin
from six.moves import input

@plugin('bmi')
def bmi(jarvis, s):
    """
    Calculates Body Mass Index.
    It is available for metric and imperial system
    -- Type bmi, press enter and then follow the instructions
    """

    jarvis.say("Type m for metric and i for imperial system")
    syst = input()
    jarvis.say("Please insert your height")
    height = input()
    jarvis.say("Please insert your weight")
    weight = input()

    height = int(height)

    weight = int(weight)

    """Calculate bmi"""
    if syst == "m":
        height = height/100
        bmi = weight/height**2
    else:
        bmi = weight/height**2 * 703

    """ Find body state """
    if bmi < 18.5:
        state = "Underweight"
    elif bmi < 24.9:
        state = "Healthy"
    elif bmi < 30:
        state = "Overweight"
    else:
        state = "Obese"
    print(str(bmi), " ", state)
```

FEEDBACK

pnhofmann commented 2 days ago

Hi!

> Do you think that could fit with this personal assistant?

Of course!

pnhofmann commented 7 hours ago • edited ▾ Collaborator +😀 ···

> What do you think I can do about testing?

Something I definitely should do more often (and also plan to write more tests for Jarvis) ;). Always a good idea especially with a interpreted language with no compile-guarantees ;).

WHAT HAVE I DONE SO FAR?

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
```

Unit Testing:

```python
import unittest
from tests import PluginTest
from plugins.bmi import Bmi
from Jarvis import Jarvis


class BmiTest(PluginTest):

    def setUp(self):
        self.test = self.load_plugin(Bmi)


    def check_bmi_calculation(self):
        d = self.test.calc_bmi_m(100, 100)
        self.assertEqual(d, 100)

        d = self.test.calc_bmi_m(200, 400)
        self.assertEqual(d, 300)

if __name__ == '__main__':
    unittest.main()
```

Metrics:

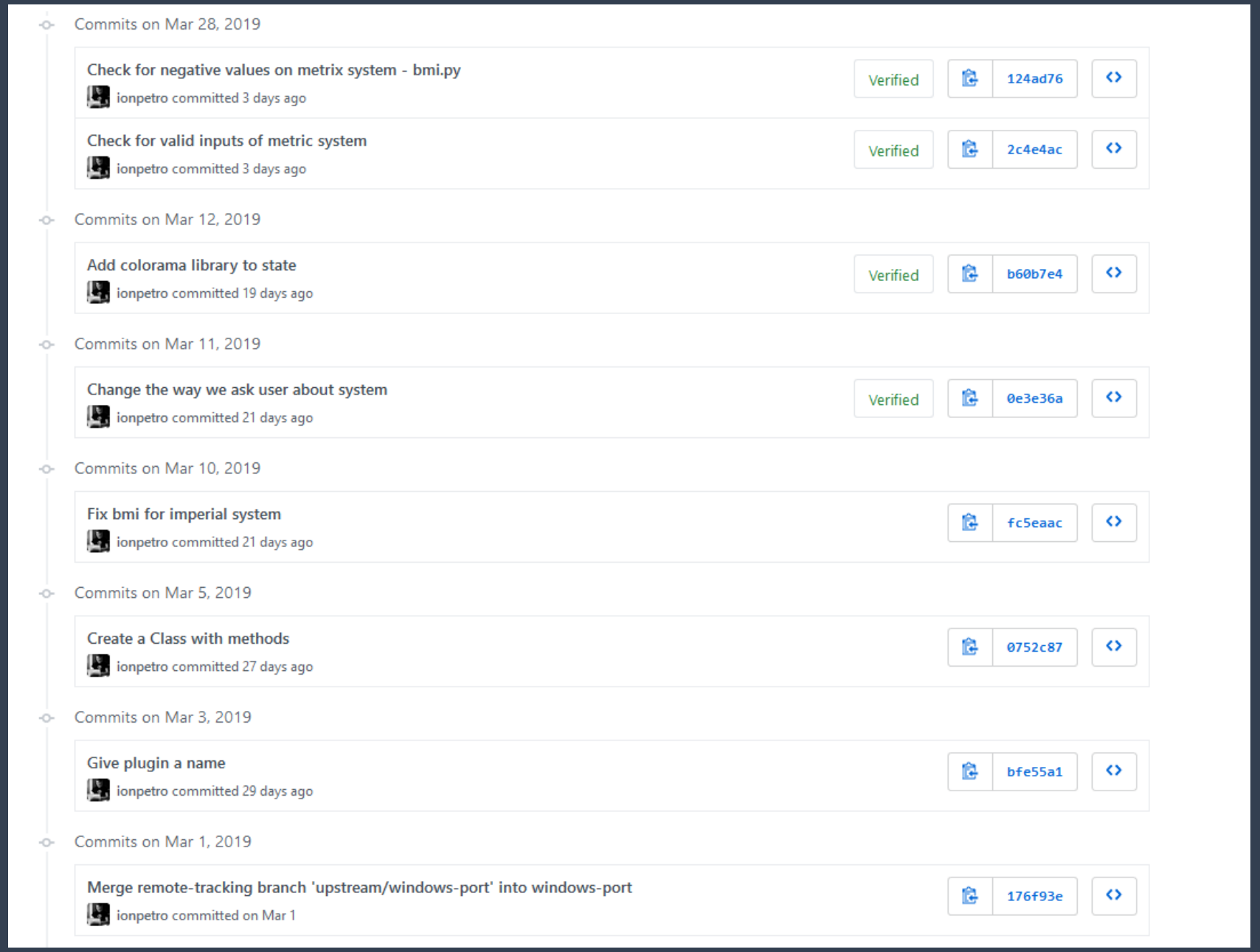
Lines Of Code:
116

Classes:
2

Files:
2

Methods:
8

Committing:


Commits on Mar 28, 2019
Check for negative values on metrix system - bmi.py
ionpetro committed 3 days ago
Verified
124ad76
Check for valid inputs of metric system
ionpetro committed 3 days ago
Verified
2c4e4ac
Commits on Mar 12, 2019
Add colorama library to state
ionpetro committed 19 days ago
Verified
b60b7e4
Commits on Mar 11, 2019
Change the way we ask user about system
ionpetro committed 21 days ago
Verified
0e3e36a
Commits on Mar 10, 2019
Fix bmi for imperial system
ionpetro committed 21 days ago
fc5eaac
Commits on Mar 5, 2019
Create a Class with methods
ionpetro committed 27 days ago
0752c87
Commits on Mar 3, 2019
Give plugin a name
ionpetro committed 29 days ago
bfe55a1
Commits on Mar 1, 2019
Merge remote-tracking branch 'upstream/windows-port' into windows-port
ionpetro committed on Mar 1
176f93e

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
```

Colorama Library:

```
BMI:  2.8
STATE: Severe thinness
```

```
BMI:  24.9
STATE: Healthy
```

```
BMI:  29.0
STATE: Pre-obese
```

```
BMI:  45.9
STATE: Obese
```

Scaling

To do NEXT:

- ✓ Test my Code
- ✓ Write another plugin
- ✓ Pray for Merge

Thank you for your time!