Python development - Intro

May 14, 2020

1 Python development - Intro

1.1 Agenda

- Intro: class structure and dynamics
- Tools: environment (Linux), editor(s), git
- Class Intro: basics, Python philosoply

1.2 Syllabus

1.2.1 Discussed topics

- Intro Web
- Intro HTML
- Intro CSS
- Python intro and env setup
- Python syntax
- Python data model
- Lists and tuples
- Comprehensions
- Dictionaries
- Dictionaries and Sets
- Functions
- Closures
- Functions recap
- Decorators
- Pythonic Objects
- Inheritance
- Mixins

- Generators
- Django intro
- Django Models
- Django Views
- Django Templates

1.2.2 Classes

4 hours / week

Tue & Thu @ 17.00

1.2.3 Assignments and evaluation

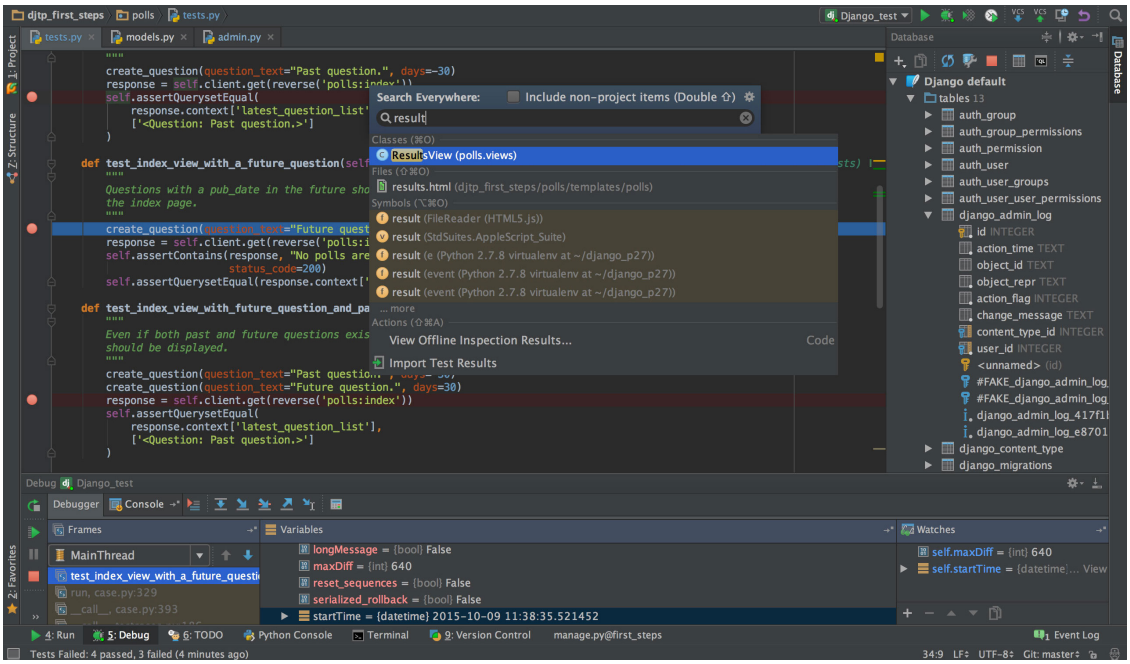
- regular assignments after most of the classes
- **1** group project
- end of the class final exam

1.2.4 Tools and necessary materials

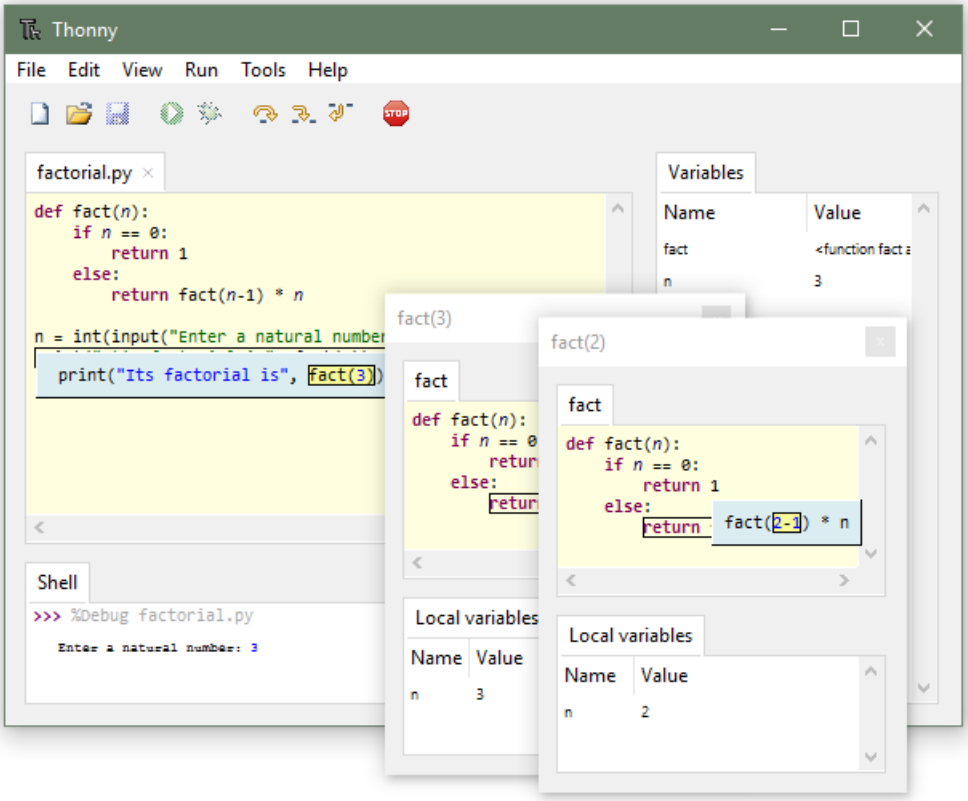
- https://platforma.scoalainformala.ro/
- laptop with Linux Mint installed How to install Linux Mint
- Github account
- curiosity and desire to learn ;)

1.3 Code editors

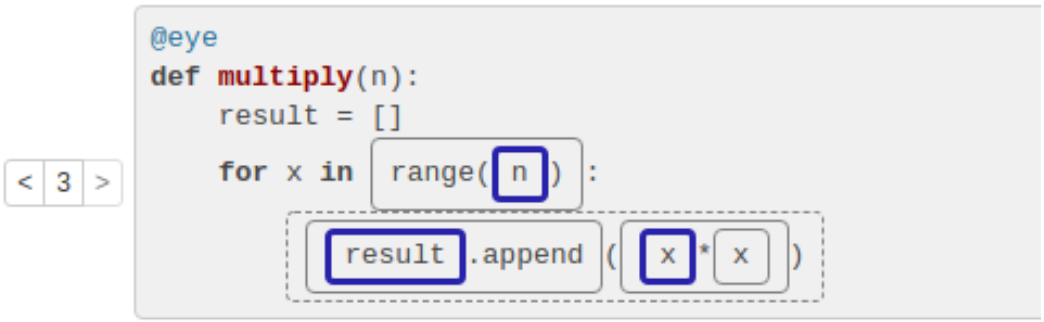
1.3.1 PyCharm

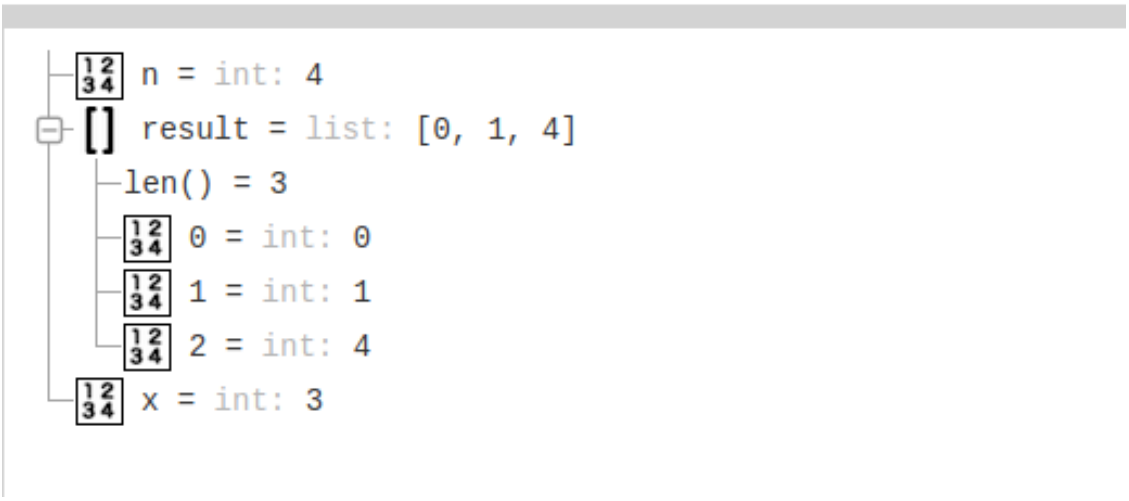


1.3.2 Thonny - Python IDE for beginners



1.3.3 birdseye





1.4 Class Intro - How does the web work?

Open discussion and where does Python fit in

1.5 Python philosoply

[4]:
```
import this
```

```
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
```

```
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```