(http://csmatters.org) 2 - 2

0b10 - 0b10

Using Python and PyCharm

Unit 2. Developing Programming Tools

Revision Date: Sep 08, 2019
Duration: 1 50-minute session

Lesson Summary

Summary

PyCharm, an IDE for Python, will be introduced. Keywords, file, and variable naming conventions will be addressed.

Outcomes

- Students will find and launch PyCharm.
- Students will configure PyCharm's appearance.
- Students will create a Python project in PyCharm.
- Students will name and save projects according to the requirements of their instructor.
- Students will create a Python file (.py file).
- Students will add line numbers to the file.
- Students will add comments to the file.
- Students will write, debug, and run a simple Python program.

Overview

1. Getting Started (10 min)
2. Guided Activity (20 min)
    1. Introduction to PyCharm
3. Summative Assessment (20 min)
    1. Programming Exercise
4. Homework Assignment

Learning Objectives

CSP Objectives

- *EU CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.*
  - LO CRD-2.B - Explain how a program or code segment functions.
  - LO CRD-2.C - Identify input(s) to a program.
  - LO CRD-2.J - Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program.
- *EU AAP-1 - To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.*
  - LO AAP-1.A - Represent a value with a variable.
  - LO AAP-1.B - Determine the value of a variable as a result of an assignment.
- *EU CSN-2 - Parallel and distributed computing leverage multiple computers to more quickly solve complex problems or process large data sets.*
  - LO CSN-2.A - For sequential, parallel, and distributed computing: a. Compare problem solutions. b. Determine the efficiency of solutions.

Math Common Core Practice:

- MP1: Make sense of problems and persevere in solving them.

Common Core ELA:

- RST 12.3 - Precisely follow a complex multistep procedure
- RST 12.4 - Determine the meaning of symbols, key terms, and other domain-specific words and phrases
- WHST 12.2 - Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes
- WHST 12.6 - Use technology, including the Internet, to produce, publish, and update writing products

Key Concepts

Students will learn what an IDE (Integrated Development Environment) is and why it is good to use one when programming.

Students will be able to find, configure and use the PyCharm IDE to write, save, run, debug and retrieve their Python modules according to the requirements of their instructor.

Essential Questions

- How can computing and the use of computational tools foster creative expression?
- How are vastly different kinds of data, physical phenomena, and mathematical concepts represented on a computer?
- How do computer programs implement algorithms?
- How do people develop and test computer programs?

Teacher Resources

Student computer usage for this lesson is: **required**

For the Students

- Python 3.4.1:
  - https://www.python.org/downloads/ (https://www.python.org/downloads/)
- Free community edition of PyCharm:
  - http://www.jetbrains.com/pycharm/ (http://www.jetbrains.com/pycharm/)
- Non-Programmer's Tutorial for Python 3/Who Goes There?:
  - http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3/Who_Goes_There%3F (http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3/Who_Goes_There%3F)
- An Informal Introduction to Python:
  - https://docs.python.org/3.4/tutorial/introduction.html#using-python-as-a-calculator (https://docs.python.org/3.4/tutorial/introduction.html#using-python-as-a-calculator)

Lesson Plan

Getting Started (10 min)

- **Journal:** What environments have you already used to write Python programs?
- Share answers in class.
- After students have shared their answers, introduce the topic of Integrated Development Environments (IDEs). Explain that an IDE provides a way to manage files, write code, get help with code syntax and usage, and the ability to test, run, and debug programs in an environment they are able to configure to their own liking.
- As defined by Wikipedia: "An **integrated development environment** (**IDE**) or **interactive development environment** is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, automation tools, and a debugger. Most modern IDEs offer intelligent code completion features." (From: http://en.wikipedia.org/wiki/Integrated_development_environment (http://en.wikipedia.org/wiki/Integrated_development_environment))
- Have students go to Python Central at http://www.pythoncentral.io/the-best-python-ides-you-can-use-for-development/ (http://www.pythoncentral.io/the-best-python-ides-you-can-use-for-development/) for a list of the top IDEs for Python. PyCharm is listed as one of the best. Explain to students that they will be learning to use the PyCharm IDE.
- Share tweets from the homework about the 7 aspects of programming quickly standing up in place and sharing.
  1. Catch a shooting star (variables)
  2. Dictionary (data types)
  3. Russian Dolls (things within things, instances)
  4. Sausage (processes)
  5. The dog, the cat, and the fish (causation, event change)

6. Pizza (abstraction) include why abstraction is "like making pizza," and what other kinds of activities might fall into that category.

If they did not assign the homework, review Seven things you should know if you're starting out programming

Have students go to the article *Seven things you should know if you're starting out programming* at

http://www.theguardian.com/info/developer-blog/2011/oct/07/programming-developer-journalist (http://www.theguardian.com/info/developer-blog/2011/oct/07/programming-developer-journalist)

Activity (20 min)

Introduction to PyCharm

- Find and launch PyCharm. If there is not a shortcut on your desktop, go to Programs >> JetBrains >> JetBrains PyCharm Community Edition 3.4.1.
- The first screen will provide options for Create New Project, Open Directory, Check out from Version Control, Configure, and Docs and How-Tos.
- Select Configure >> Settings >> Appearance. Explore Screens under UI Options. IntelliJ is the default appearance
    - Darcula and Windows are some other options.
    - Students should explore other options available.
    - Students should play with the appearance and select a new one (with your approval) if they have a preference.
- Click the arrow to go back to the Quick Start menu.
- Click Create New Project.
- Give the project a name.
    - Click the ellipses (the … at the end of the location field) to navigate to the location you need to save your work.

Teacher note: This may be a good time to have students configure the default Working Directory as shown in the Python and PyCharm Installation and Configuration Guide file in the lesson resources folder.

- When you click "OK," your project will be created.
- Click File >> New >> Python File to create your .py file. Name your first .py file **rate_time.**
    - The code for this module will be copied from the 'Non-Programmer's Tutorial for Python 3/Who Goes There?' tutorial shortly.
- Click OK to create the file.
    - The user name of the author of the file will appear at the top of the code window with the default comment method.
- Take students through some configuration modifications, such as adding line numbers, showing the console, and changing the default for comment style, all covered in the Installation and Configuration Guide.
    - If you are projecting your screen, you can use Presentation Mode to make viewing easier for students.
- Allow students some time to explore the environment.
- Visit Non-Programmer's Tutorial for Python 3/Who Goes There?

    - http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3/Who_Goes_There%3F (http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3/Who_Goes_There%3F)
- Prior to copying the code for rate_time.py and area.py, take students through the sections in the Non-Programmer's Tutorial for Python 3/Who Goes There? on Input and Variables.
- Explain and model variable types and their behaviors, and concatenation.
    - For ease of use, consider using the comma at this time to avoid confusion about concatenation.
- Create, save, run and debug programs rate_time.py and area.py.
- To run a program for the first time, go to **Run** and select the file you want to run.
    - After that, the Run command will appear in the upper right corner of the PyCharm IDE.
- As you add new modules to the project you will need to go to Run and select the new file.
    - After that, the file names will appear in a drop-down menu next to the green Run arrow.
- If it has not already been configured, go to Tools >> Run Python Console to show the console at the bottom of the PyCharm environment. Students can use this to test their code as they have in other environments.
- In the PyCharm console, test code from An Informal Introduction to Python 3.1 Using Python as a Calculator. (https://docs.python.org/3.4/tutorial/introduction.html#using-python-as-a-calculator (https://docs.python.org/3.4/tutorial/introduction.html#using-python-as-a-calculator)).

Important notes about naming your files and variables:

- All names chosen for variables or other abstractions should be meaningful and clear to a reader of the program.
- Our Python coding examples will follow the PEP 8 Style Guide. (http://legacy.python.org/dev/peps/pep-0008/ (http://legacy.python.org/dev/peps/pep-0008/))
    - Teachers are not required to use this standard but should establish their classroom conventions for students to follow.
- Modules written in Python, .py file names, should be in all lower case as is done with the sample files in the Python for Everybody textbook. (https://www.py4e.com/code3/ (https://www.py4e.com/code3/))
- Variable names should begin with a lowercase letter with multiple words separated by an underscore.
    - Teachers may choose to use camelCase or another naming system and should be specific and consistent with students regarding these requirements.
- Python file names and variable names must not begin with a number, and cannot be the same as Python keywords:

```
and       del       from      not       while
as        elif      global    or        with
assert    else      if        pass      yield
break     except    import    print
class     exec      in        raise
continue  finally   is        return
def       for       lambda    try
```

(from http://www.pythonforbeginners.com/basics/keywords-in-python (http://www.pythonforbeginners.com/basics/keywords-in-python))

- For more information concerning variable names and keywords to Python, go to
  - Python for Everybody (http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf), Charles Severance, Chapter 2, section 3

Summative Assessment (20 min)

Students are to:

- Create a new Python module (.py) called name.py.
- Comment in their name, date, and name of project as directed by their instructor at the top of their code window.
- Assign their full name to an appropriately named variable.
- Assign their age to an appropriately named variable.
- Output their name and age as a sentence, with variables and literal values concatenated correctly.
- Debug any errors and run their program with the correct output and without error.

Sample code (will throw an error)

```
# author = 'iam tester'
 # date July 4, 2024
 # name.py
 name = 'Ima Tester'
 age = 15
 print (name + " is " + age + " years old.")
```

The above code will throw an error, because **`age`** is an integer and needs to be expressed as a string OR concatenated using a comma `(,)`.

```
name = 'Ima Tester'
age = 15
print (name,"is",age,"years old.")
```

Important notes about naming your files and variables:

- Our Python coding examples will follow the PEP 8 Style Guide. (http://legacy.python.org/dev/peps/pep-0008/ (http://legacy.python.org/dev/peps/pep-0008/))
  - Teachers are not required to use this standard but should establish their classroom conventions for students to follow.

- Modules written in Python, .py file names, should be in all lower case as is done with the sample files in the Python for Everybody textbook. (https://www.py4e.com/code3/ (https://www.py4e.com/code3/))
- Variable names should begin with a lowercase letter with multiple words separated by an underscore.
    - Teachers may choose to use camelCase or another naming system and should be specific and consistent with students regarding these requirements.
- Python file names and variable names must not begin with a number, and cannot be the same as Python keywords:

```
and       del       from      not       while
as        elif      global    or        with
assert    else      if        pass      yield
break     except    import    print
class     exec      in        raise
continue  finally   is        return
def       for       lambda    try
```

(from http://www.pythonforbeginners.com/basics/keywords-in-python (http://www.pythonforbeginners.com/basics/keywords-in-python))

- For more information concerning variable names and keywords to Python, go to
    - Python for Everybody (http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf), Charles Severance, Chapter 2, section 3

Homework Assignment

1. Use two sources to find definitions of "algorithm", cite the sources
2. Write down the steps involved in a daily task – creating a peanut butter and jelly sandwich; getting to school; brushing teeth; completing homework; etc…)

Options for Differentiated Instruction

Students can work in pairs, side by side, to help each other through each step of the process.

Evidence of Learning

Formative Assessment

Checks for understanding throughout the entire process of learning to use the PyCharm IDE by using active participation in trying each step of the process and having students help their elbow partner when difficulties arise.

Summative Assessment

Assessment tasks:

- Create a new Python program file.
- Add a comment containing their name, date, and name of project as directed by their instructor at the top of their code window.
- Assign their name to an appropriately named variable.
- Assign their age to an appropriately named variable.
- Output their name and age as a sentence with variables and literal values concatenated correctly.
- Debug any errors and run their program with the correct output and without error.

(http://www.umbc.edu/) (http://www.umd.edu/)

(http://www.nsf.gov/)

Authored by: CS Matters in Maryland
Website: csmatters.org (http://csmatters.org)
Email: csmattersinmaryland@gmail.com (mailto:csmattersinmaryland@gmail.com)

This work is licensed under a
Creative Commons Attribution-ShareAlike 3.0 United States License
(http://creativecommons.org/licenses/by-sa/3.0/us/)
by University of Maryland, Baltimore County (http://umbc.edu) and University of Maryland, College Park
(http://umd.edu).