Code Academy

PYTHON

PYTHON PROGRAMMING LANGUAGE

Core Programming Concepts

Code Academy

Common to ALL OOP programming languages

- **Data structures**: ways of storing a collection of related data, for example, an ordered list or sequence of items.

- **Flow control:** ways of controlling and changing the flow of a program.

  - **Conditionals** or "`if`" statements that allow us to make decisions and execute code based on logical conditions, and

  - **Loops** which allow us to repeat code or perform an operation multiple times

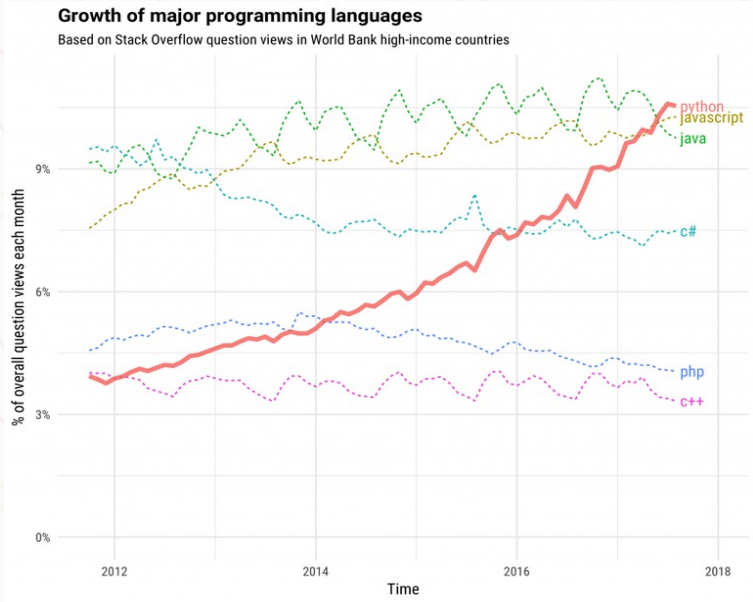
- **Variables**: symbolic names for or references to values or pieces of information. You can use variables to store all kinds of data

- **Functions:** blocks of organized and reusable code that can be used to perform a single related action.

Why Python Language?

Python is a great first language,

- Python is an **open source** programming language,
- Python is powerful, flexible, and intuitive.
- Python is very easy to get up and running quickly
- It's also a lot easier to learn than Java, C++, …
- Python is **general purpose**: it can be used for artificial intelligence, machine learning, natural language processing, web development, data analysis and visualization, game programming, GUIs, and many other purposes.



Python allows you to spend more time thinking about programming logic and algorithms, without having to worry about learning complex code syntax.

Python is known for its simple syntax

Code Academy

What is Python?

- Python is a **high level** programming language
  - provides abstraction from the details of the computer.
  - does most of the work in communicating with the computer,
  - the code is intuitive and easy to understand.

- Python is an **object-oriented programming** language
  - organized around “**objects**” rather than “actions”.

- Python is **dynamically typed**, and **garbage-collected**

- Python is an **interpreted** programming language

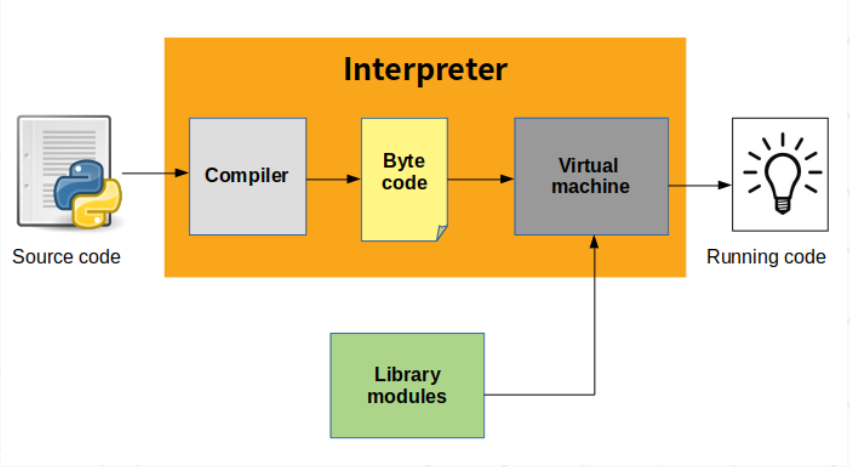
Don't make the mistake of typing out large chunks of code and not testing it at all.

Code Academy

How Python runs?

*- How the Python code is **actually** executed by the Python interpreter?*

- Compiler compiles (translates) your **source code** stored with **.py** into a **byte code** stored with **.pyc**



- **Byte code** is a lower level, platform independent, efficient and intermediate representation of your source code.

- The **byte code** is fed into **PVM (Python Virtual Machine)**.

- The *PVM* is a big loop that iterates through your byte code instructions, one by one, to carry out their operations

- The Python interpreter has a number of built-in functions. They are loaded automatically as the interpreter starts and are always available.
Example: **`print()`**, and **`input()`** for I/O

Programming Errors

➢ Invalid Syntax *aka* (Compile-time error)

- Violation of *syntax rule* - how the elements of a programming language must be written

- The *interpreter* will find any invalid syntax in Python during this first stage of program execution, also known as the *parsing* stage.

- When the interpreter encounters invalid syntax in Python code, it will raise a **`SyntaxError`** exception and provide a **traceback** with some helpful information to help you debug the error.

```
In [71]: cost = 5$
  File "<ipython-input-71-805d380451a5>", line 1
    cost = 5$
           ^
SyntaxError: invalid syntax
```

- If your code is **syntactically** correct, then you may get other exceptions raised that are not a **`SyntaxError`**.

Programming Errors

➢ Exception errors

- Occur whenever syntactically correct Python code results in an error
- Python details what type of exception error was encountered

```
In [110]: 1/0
Traceback (most recent call last):

  File "<ipython-input-110-9e1622b385b6>", line 1, in <module>
    1/0

ZeroDivisionError: division by zero
```

```
In [72]: sqrt(-4)
Traceback (most recent call last):

  File "<ipython-input-72-ee0bfbfc0490>", line 1, in <module>
    sqrt(-4)

ValueError: math domain error
```

Programming Errors

- **Logic errors *aka* Semantic errors**
  - Occur during the analysis or design of the algorithm.
  - Code with Logic errors can be run, but it does not operate as intended.
  - Logic errors are the most difficult to fix.

```
 9 x = float(input('Enter a number: '))
10 y = float(input('Enter a number: '))
11
12 z = x+y/2
13 print ('The average of the two numbers you have entered is:',z)
14
```

```
In [38]: runfile('/Users/hamzazidoum/Documents/2101/2101_S2021/Code/
syntaxErrors.py', wdir='/Users/hamzazidoum/Documents/2101/2101_S2021/
Code')

Enter a number: 4

Enter a number: 6
The average of the two numbers you have entered is: 7.0

In [39]:
```

Getting Python

- **Python** is a programming language in which we write computer programs. These programs would be stored in *text* files that have the ending `.py`, for example `hello.py`.

- **Spyder** is an interactive development environment (IDE) for the Python language with editing, interactive testing, debugging and introspection features

- **Anaconda** is one of several Python distributions. install the Anaconda Python distribution using installation instructions. It provides the Python interpreter itself and all packages we need: *https://docs.continuum.io/anaconda/install/*

Lab Exercises

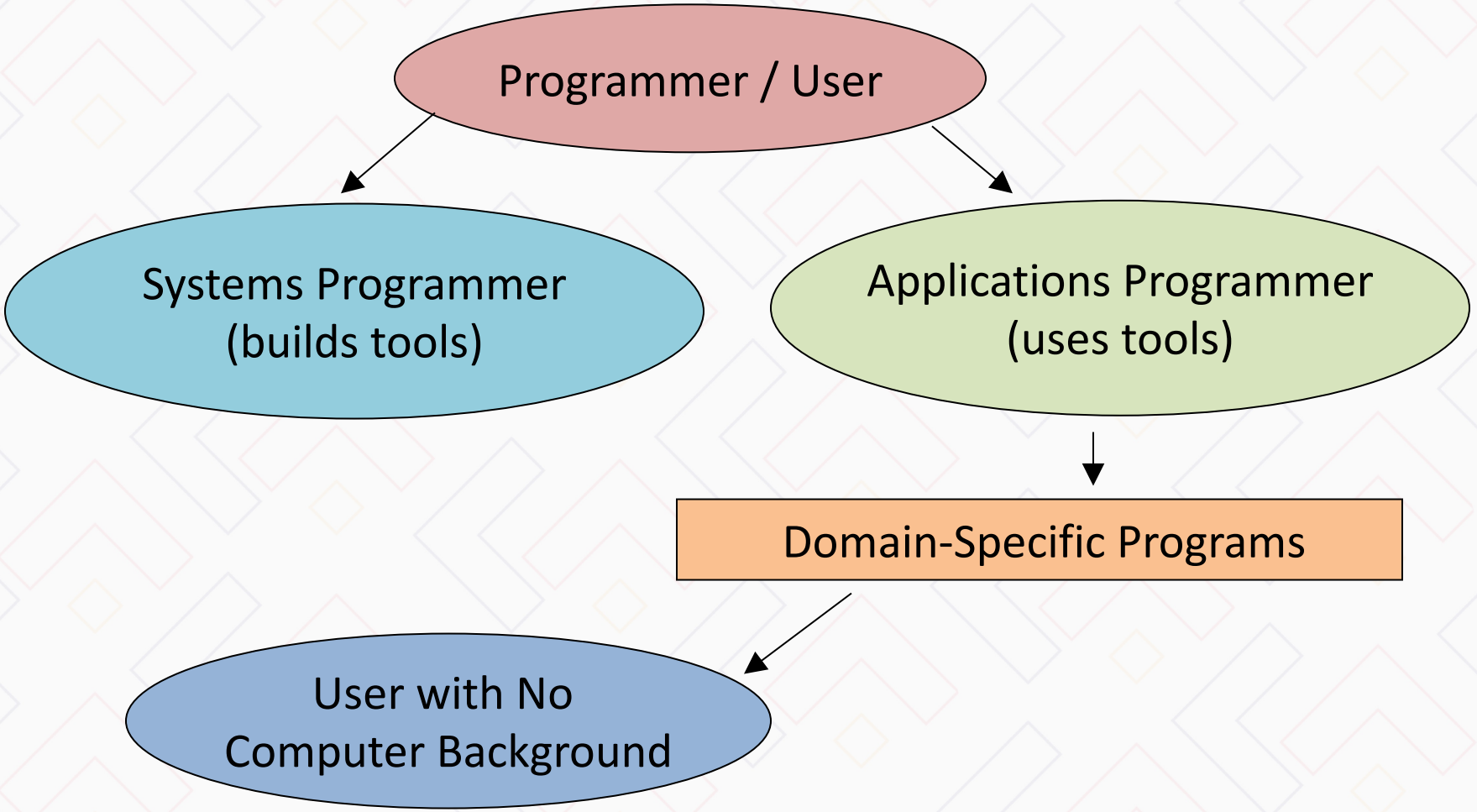
Apply the Development process to solve the lab exercises:

Step 1: Define the problem:	WHAT? Input? Output?
Step 2: Specify the requirements	
Step 3: Design a solution/algorithm:	HOW? Pseudo-code
Step 4: Implement the code	Python Script
Step 5: Test	Verify
Step 6: Repeat Steps 3 – 5	Debug
Step 7: Maintenance	Update

Code Academy

COMPUTING AS A TOOL

Computing as a Tool



12

From Computer Science Illuminated – Nell Dale & John Lewis

Code Academy

Computing as a Discipline

What can be (efficiently) automated?

Four Necessary Skills(Denning)

- **Algorithmic Thinking**
  - To be able to express solutions in terms od step-by-step procedures
- **Representation**
  - To be able to store data in a way that it can be processed efficiently
- **Programming**
  - To be able to convert the algorithm and representation into computer software
- **Design**
  - To be able to write effective and efficient software

From Computer Science Illuminated – Nell Dale & John Lewis

Code Academy

Computing as a Discipline

What do you think?

Is Computer Science a mathematical, scientific, or engineering discipline?

From Computer Science Illuminated – Nell Dale & John Lewis

Systems Areas of Computer Science

- Algorithms and Data Structures
- Programming Languages
- Architecture
- Operating Systems
- Software Methodology and Engineering
- Human-Computer Communication

From Computer Science Illuminated – Nell Dale & John Lewis

Application Areas of Computer Science

- Numerical and Symbolic Computation
- Databases and Information Retrieval
- Artificial Intelligence and Robotics
- Graphics
- Organizational Informatics
- Bioinformatics

From Computer Science Illuminated – Nell Dale & John Lewis

Code Academy

Summary:

- Computers rapidly execute very simple **instructions**
- A **Program** is a sequence of instructions and decisions
- **Programming** is the art (and science) of designing, implementing, and testing computer programs
- **Pseudocode** is an informal description of a sequence of steps for solving a problem.
- An **algorithm** for solving a problem is a sequence of steps that is unambiguous, executable, and terminating.

Summary: Python

- **Python** was designed in a way that makes it easier to learn than other programming languages such as Java, C and C++.
- Python has simpler and cleaner **syntax**.
- It is important to practice with the tool so you can focus on learning Python
- An **editor** is a program for entering and modifying text, such as a Python program.
- The Python **compiler** translates source code into byte code instructions that are executed by the **Virtual machine**.