

Ch01-Introduction

September 10, 2025

1 Introduction

- HTML version of textbook: http://openbookproject.net/thinkcs/python/english3e/way_of_the_program.html
- PDF version of textbook: <http://www.greenteapress.com/thinkpython/thinkCSpy/thinkCSpy.pdf>

1.1 Topics:

- Python and its features
- Ways to run Python code
- The First Program and its anatomy
- Computer program and its building blocks
- Errors and Debugging

1.2 Python Programming Language

- the single most important skill for a computer scientist is problem solving
- Python is a tool that helps computer scientists and programmers solve problems by writing code
- One of the most popular programming languages used in various fields: Data Science and Machine Learning, Security, Web Apps, etc.
- Popularity of Python has been increasing over the years: <https://www.tiobe.com/tiobe-index/>

1.3 Python Features

- high-level general purpose programming language such as PHP, Perl, Java, JavaScript, C++, etc.
 - as opposed to low level machine language such as assembly
- interpreted language; needs Python interpreter to execute Python code
- platform independent/portable; python programs can be run in many platforms including Raspberry Pi
- open source, can be freely downloaded and use: <http://python.org>
- installed using Python package manager such as Anaconda or Miniconda: <https://www.anaconda.com/download/>
- two versions Python 2.x and Python 3.x - Notebooks and the text use version 3.x which is the new standard

1.4 The Zen of Python

- Pythoneer Tim Peters succinctly channels the Benevolent dictator for life (BDFL)'s guiding principles for Python's design into 20 aphorisms, only 19 of which have been written down
- Python Enhancement Proposal (PEP) 20 : <https://www.python.org/dev/peps/pep-0020/> describes the Zen
- the following statement is hidden as Easter Egg which states the Zen

```
[3]: 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!
```

1.5 Learning and Writing Python Code

- one must write code to learn it
- Python provides interactive mode to practice and use Python for quick calculations such as a calculator
- most use code editors to write long program also called scripts
- use pythontutor.com to visualize code execution

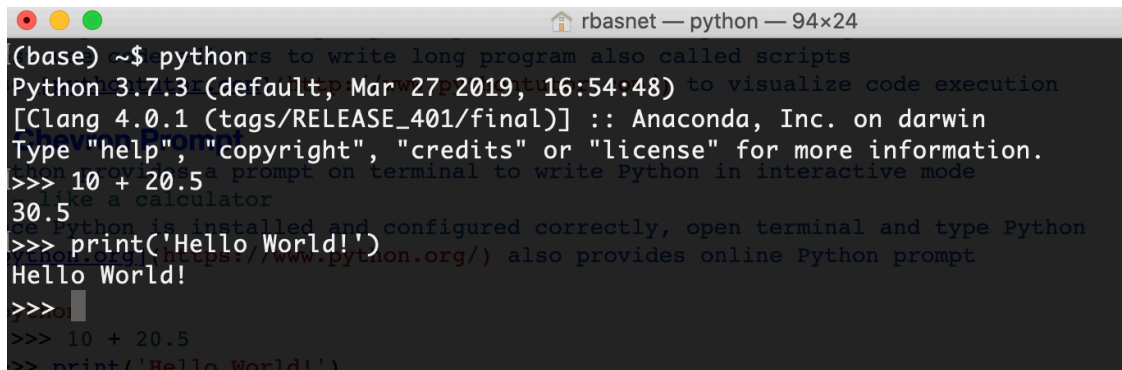
1.5.1 Python Development Environment

- it's recommended to install a good set of programming tools such as Editor, version control software, etc. on your system and be familiar with them to become an effective problem solver
- complete instructions on setting up development environment for Python on various Operating systems can be found here: <https://github.com/rambasnet/DevEnvSetup>

1.5.2 Chevron Prompt

- Python provides a prompt on terminal to write Python in interactive mode
 - like a calculator
- Once Python is installed and configured correctly, open terminal and type Python
- python.org also provides online Python prompt

```
>>> 10 + 20.5
30.5
>>> print('Hello World!')
Hello World!
```



```
(base) ~$ python
Python 3.7.3 (default, Mar 27 2019, 16:54:48)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> help
>>> 10 + 20.5
30.5
>>> print('Hello World!')
Hello World!
>>>
```

1.5.3 Jupyter Notebook

- interactive notebook that can have live code, execution results and HTML, texts and multimedia contents!
- great way to learn, experiment, and take notes while coding
- Jupyter supports many programming languages such as C, C++, Java, JavaScript, etc.; Python is default!

1.5.4 Python Script

- complete program written using a text editor or an Integrated Development Environment (IDE) such as PyScripter, PyCharm, and Visual Studio (VS) Code, Notepad++, etc.

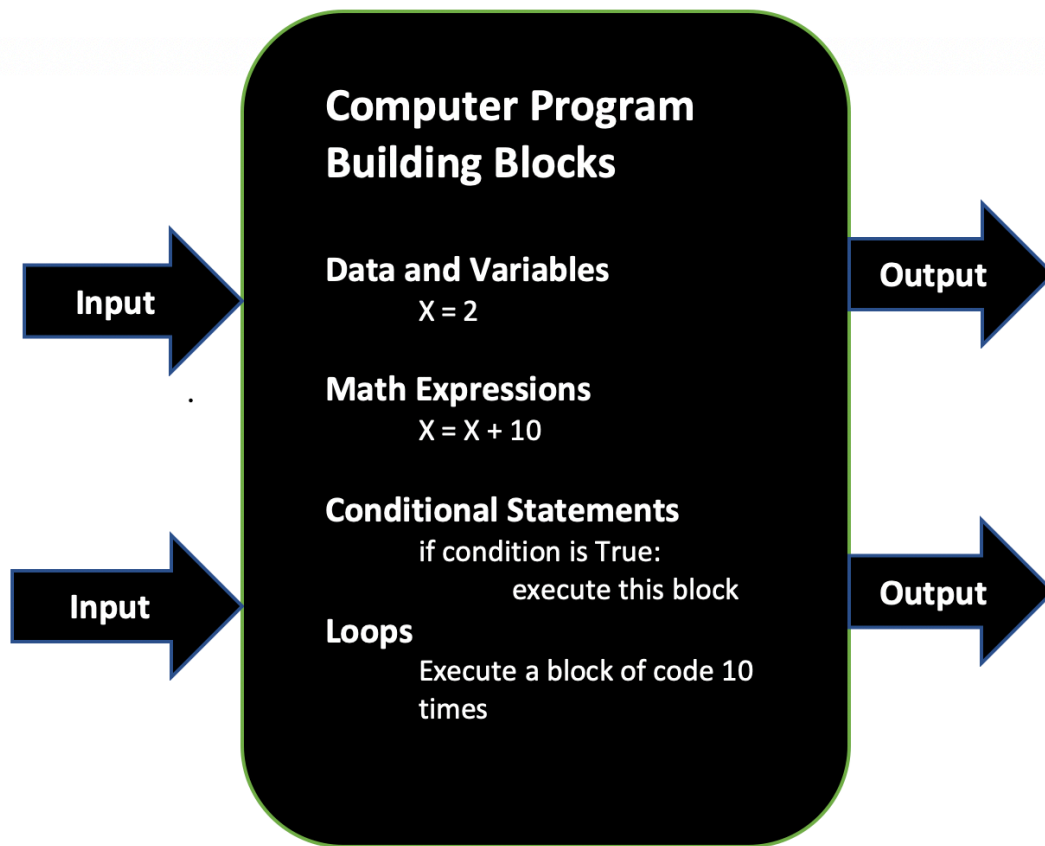
1.5.5 Creating and running Python script

- open your favourite editor
- create a hello.py file
- type `print('Hello World!')` and save the file
- run the script from a terminal

```
python hello.py
```

1.6 Computer Program

- sequence of instructions that specifies how to perform a computation
- some basic/fundamental concepts that make up a computer program:
 - variables, input, output, math, conditional execution and repetition



1.6.1 variables

- memory location where data/value can be stored

1.6.2 input

- get data from keyboard, a file, or some device

1.6.3 output

- display data/answer on screen, or save it to file or to a device

1.6.4 math

- basic mathematical operations such as addition, subtraction, multiplication, etc.

1.6.5 conditional execution

- check for certain conditions and execute appropriate sequence of statements

1.6.6 repetition

- perform some action repeatedly, usually with some variation every time

1.7 Debugging

- finding and getting rid of bugs/errors
- as long as humans write computer codes, there'll be always errors in computer program
- although frustrating at times, it is one of the most intellectually rich, challenging, and interesting part of programming

1.8 Coding Errors

- errors are also called bugs
- 3 types: syntax, run-time and semantic

1.8.1 Syntax errors

- program needs to follow Python syntax or grammar
- Python interpreter points out any syntax error when one tries to run the program
- can't run the program if it has syntax errors

1.8.2 Run-time errors

- also called run-time exceptions
- these errors may appear while program is running under certain scenario
- can be handled to certain extent by adequate testing

1.8.3 Semantic errors

- program runs fine but gives wrong answer
- also called logical errors which occur when the programmers may have misunderstanding of the language quarks or the problem statement or just plain wrong solution
- can be identified and removed by adequate testing

The First Python Program - python programs are usually called scripts - traditionally, Hello World is the first program one writes to learn the language

```
[1]: #-----  
# hello world program  
# by: John Doe  
# Jan 1 2017  
# Copyright: Anyone may freely copy or modify this program  
#-----  
  
print('Hello World!') # Say hello to the beautiful world!
```

Hello World!

1.9 Anatomy of a Python Script

- python script typically consists of instructions (statements) and comments
- instructions can be single line and independent or grouped together as a block
- instructions are for computer to carry out certain computations

1.10 Comments

- Python uses `#` symbol to write a single line comment
- comments are for readers/programmers to convey/explain what the code is doing or supposed to do
- Python interpreter ignores the comments after `#` symbol

1.11 Style Guide for Python Code

- Python creator Guido Van Rossum et al. have published style guide to write beautiful and compliant Python code
- guidelines can be found described on Python Enhancement Proposal (PEP) 8 here: <https://www.python.org/dev/peps/pep-0008/>
- E.g. the guide says that comments must be complete sentence and must be capitalized!

1.12 Exercises

1.12.1 1. Write a hello world script

- write a python script that prints “Hello World!” as an output on the console

1.12.2 2. ASCII Art

- learn about ASCII Art
- print some ASCII arts, texts of your choice
- use ASCII Art text generator: <http://patorjk.com/software/taag/#p=display&f=Graffiti&t=Type%20Some>

1.12.3 2. Write a script that prints various stages of the hangman game

- game description: [https://en.wikipedia.org/wiki/Hangman_\(game\)](https://en.wikipedia.org/wiki/Hangman_(game))
- produce the output seen in Example game section of the Wikipedia page

1.13 Kattis Problems

1.13.1 1. hello problem

- create a free account on Kattis: <https://open.kattis.com/login>
- login and solve the hello problem: <https://open.kattis.com/problems/hello>

1.13.2 2. Solve Velkomin!

- solve the problem - velkomin <https://open.kattis.com/problems/velkomin>

[]: