JOHN S. CODE

PYTHON PROGRAMMING

FOR BEGINNERS

A HANDS-ON EASY GUIDE FOR BEGINNERS TO LEARN PYTHON PROGRAMMING FAST, CODING LANGUAGE, DATA ANALYSIS WITH TOOLS AND TRICKS

PYTHON PROGRAMMING FOR BEGINNERS:

A HANDS-ON EASY GUIDE FOR BEGINNERS TO LEARN PYTHON PROGRAMMING FAST, CODING LANGUAGE, DATA ANALYSIS WITH TOOLS AND TRICKS.

JOHN S. CODE

© Copyright 2019 - All rights reserved.

The content contained within this book may not be reproduced, duplicated or transmitted without direct written permission from the author or the publisher.

Under no circumstances will any blame or legal responsibility be held against the publisher, or author, for any damages, reparation, or monetary loss due to the information contained within this book. Either directly or indirectly.

Legal Notice:

This book is copyright protected. This book is only for personal use. You cannot amend, distribute, sell, use, quote or paraphrase any part, or the content within this book, without the consent of the author or publisher.

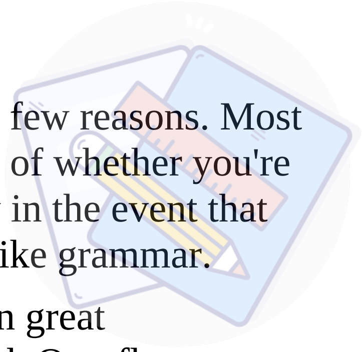
Disclaimer Notice:

Please note the information contained within this document is for educational and entertainment purposes only. All effort has been executed to present accurate, up to date, and reliable, complete information. No warranties of any kind are declared or implied. Readers acknowledge that the author is not engaging in the rendering of legal, financial, medical or professional advice. The content within this book has been derived from various sources. Please consult a licensed professional before attempting any techniques outlined in this book.

By reading this document, the reader agrees that under no circumstances is the author responsible for any losses, direct or indirect, which are incurred as a result of the use of information contained within this document, including, but not limited to, — errors, omissions, or inaccuracies.

Table of Contents

Introduction



Python is an awesome decision on machine learning for a few reasons. Most importantly, it's a basic dialect at first glance. Regardless of whether you're not acquainted with Python, getting up to speed is snappy in the event that you at any point have utilized some other dialect with C-like grammar.

Second, Python has an incredible network which results in great documentation and inviting and extensive answers in Stack Overflow (central!).

Third, coming from the colossal network, there are a lot of valuable libraries for Python (both as "batteries included" an outsider), which take care of essentially any issue that you can have (counting machine learning).

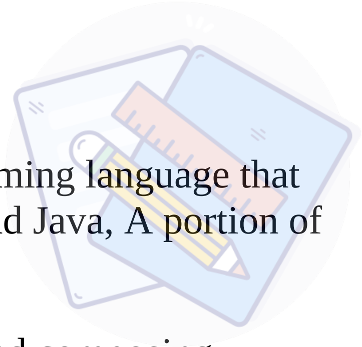
History of Python

Python was invented in the later years of the 1980s. Guido van Rossum, the founder, started using the language in December 1989. He is Python's only known creator and his integral role in the growth and development of the language has earned him the nickname "Benevolent Dictator for Life". It was created to be the successor to the language known as ABC.

The next version that was released was Python 2.0, in October of the year 2000 and had significant upgrades and new highlights, including a cycle-distinguishing junk jockey and back up support for Unicode. It was most fortunate, that this particular version, made vast improvement procedures to the language turned out to be more straightforward and network sponsored.

Python 3.0, which initially started its existence as Py3K. This version was rolled out in December of 2008 after a rigorous testing period. This particular version of Python was hard to roll back to previous compatible versions which are the most unfortunate. Yet, a significant number of its real highlights have been rolled back to versions 2.6 or 2.7 (Python), and rollouts of Python 3 which utilizes the two to three utilities, that helps to automate the interpretation of the Python script.

Python 2.7's expiry date was originally supposed to be back in 2015, but for unidentifiable reasons, it was put off until the year 2020. It was known that there was a major concern about data being unable to roll back but roll FORWARD into the new version, Python 3. In 2017, Google declared that there would be work done on Python 2.7 to enhance execution under simultaneously running tasks.



Basic features of Python

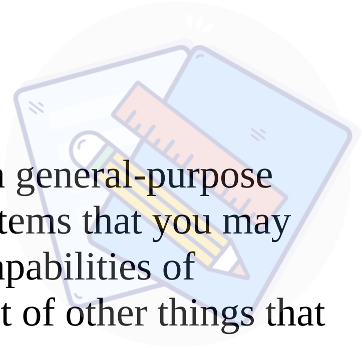
Python is an unmistakable and extremely robust programming language that is object-oriented based almost identical to Ruby, Perl, and Java, A portion of Python's remarkable highlights:

- Python uses a rich structure, influencing, and composing projects that can be analyzed simpler.
- It accompanies a huge standard library that backs tons of simple programming commands, for example, extremely seamless web server connections, processing and handling files, and the ability to search through text with commonly used expressions and commands.
- Python's easy to use interactive interface makes it simple to test shorter pieces of coding. It also comes with IDLE which is a "development environment".

The Python programming language is one of many different types of coding languages out there for you. Some are going to be suited the best to help out with websites. There are those that help with gaming or with specific projects that you want to handle. But when it comes to finding a great general-purpose language, one that is able to handle a lot of different tasks all at once, then the Python coding language is the one for you.

There are a lot of different benefits to working with the Python language. You will find that Python is easy enough for a beginner to learn how to work with. It has a lot of power behind it, and there is a community of programmers and developers who are going to work with this language to help you find the answers that you are looking for. These are just some of the benefits that we get to enjoy with the Python language, and part of the reason why we will want to get started with this language as soon as possible!

The Python programming language is a great general-purpose language that is able to take care of all your computing and programming needs. It is also freely available and can make solving some of the bigger computer programs that you have as easy as writing out some of the thoughts that you have about that solution. You are able to write out the code once, and then, it is able to run on almost any kind of program that you would like without you needing to change up the program at all.



How is Python used?

Python is one of the best programming languages that is a general-purpose and is able to be used on any of the modern operating systems that you may have on your system. You will find that Python has the capabilities of processing images, numbers, text, scientific data, and a lot of other things that you would like to save and use on your computer.

Python may seem like a simple coding language to work with, but it has a lot of the power and more that you are looking for when it is time to start with programming. In fact, many major businesses, including YouTube, Google, and more, already use this coding language to help them get started on more complex tasks.

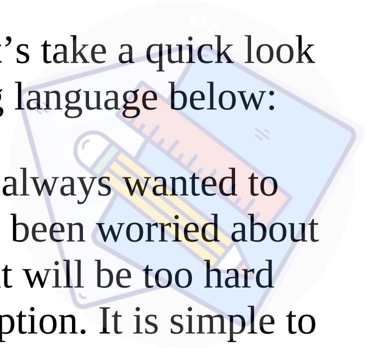
Python is also known as a type of interpreted language. This means that it is not going to be converted into code that is readable by the computer before the program is run. Instead, this is only going to happen at runtime. Python and other programming languages have changed the meaning of this kind of coding and have ensured that it is an accepted and widely used coding method for many of the projects that you would like to handle.

There are a lot of different tasks that the Python language is able to help you complete. Some of the different options that you are able to work with include:

1. Programming any of the CGI that you need on your web applications.
2. Learning how to build up your own RSS reader
3. Working with a variety of files.
4. Creating a calendar with the help of HTML
5. Being able to read from and write in MySQL
6. Being able to read from and write to PostgreSQL

The Benefits of Working with Python

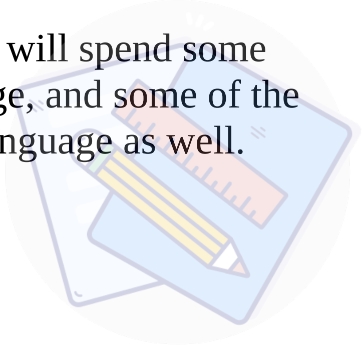
When it comes to working with the Python language, you will find that there are a lot of benefits with this kind of coding language. It is able to help you to complete almost any kind of coding process that you would like and can still



have some of the ease of use that you are looking for. Let's take a quick look at some of the benefits that come with this kind of coding language below:

- Beginners can learn it quickly. If you have always wanted to work with a coding language, but you have been worried about how much work it is going to take, or that it will be too hard for you to handle, then Python is the best option. It is simple to use and has been designed with the beginner in mind.
- It has a lot of power to enjoy. Even though Python is easy enough for a beginner to learn how to use, that doesn't mean that you are going to be limited to the power that you are able to get with some of your codings. You will find that the Python language has the power and more that you need to get so many projects done.
- It can work with other coding languages. When we get to work on data science and machine learning, you will find that this is really important. There are some projects where you will need to combine Python with another language, and it is easier to do than you may think!
- It is perfect for simple projects all the way up to more complex options like machine learning and data analysis. This will help you to complete any project that you would like.
- There are a lot of extensions and libraries that come with the Python language, which makes it the best option for you to choose for all your projects. There are a lot of libraries that you are able to add to Python to make sure that it has the capabilities that you need.
- There is a large community that comes with Python. This community can answer your questions, show you some of the different codes that you can work with, and more. As a beginner, it is always a great idea to work with some of these community members to ensure that you are learning as much as possible about Python.

When it comes to handling many of the codes and more that you would like in your business or on other projects, nothing is going to be better than working with the Python language. In this guidebook, we will spend some time exploring the different aspects of the Python language, and some of the different things that you are able to do with this coding language as well.



Chapter 1 Mathematical Concepts

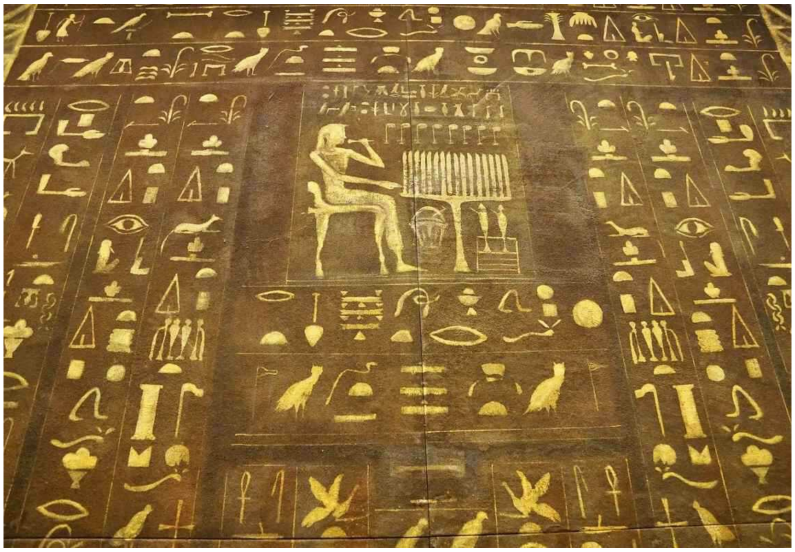
As we have stated before, computers are physical manifestations of several mathematical concepts. Mathematics are the scientific language of solving problems. Over the centuries, mathematicians have theoretically solved many complex issues. Mathematics includes concepts like algebra and geometry.

Number Systems

Mathematics is a game of number manipulation which makes number systems at the center stage of mathematical concepts. There are several different types of number systems. Before we take a look at the number systems, we have to understand the concept of coding.

Coding

A way to represent values using symbols is called coding. Coding is as old as humans. Before the number systems we use today, there were other systems to represent values and messages. An example of coding from ancient times is the Egyptian hieroglyphs.



Number systems are also examples of coding because values are represented using special symbols.

There are different types of number systems, and we are going to discuss a few relevant ones.

Binary System

A binary system has only two symbols, 1 and 0 which are referred to as bits. All the numbers are represented by combining these two symbols. Binary systems are ideal for electronic devices because they also have only two states, on or off. In fact, all electronic devices are based on the binary number system. The number system is positional which means the position of symbols determines the final value. Since there are two symbols in this system, the system has a base of 2.

The sole purpose of input and output systems is to convert data to and from binary system to a form that makes better sense to the user. The first bit from the left side is called Most Significant Bit (MSB) while the first bit from the right is called the Least Significant Bit (LSB).

Here is the binary equivalent code of “this is a message”:

01110100 01101000 01101001 01110011 00100000 01101001 01110011 00100000 01100001 00100000 01101101 01100101 01110011 01110011 01100001 01100111 01100101

Decimal System

The decimal system has ten symbols, the numbers 0 through 9. This is also a positional number system where the position of symbols changes the value it represents. All the numbers in this system are created with different combinations of the initial ten symbols. This system has a base 10.

This is also called the Hindu-Arabic number system. Decimals make more sense to humans and are used in daily life. There are two reasons for that.

- Creating large numbers from the base symbols follows a consistent pattern
- Performing arithmetic operations in a decimal system is easier compared to other systems

Hexadecimal System

The hexadecimal number system is the only one that has letters as symbols. It has the 10 symbols of the decimal system plus the six alphabets A, B, C, D, E and F. This is also a positional number system with a base 16.

Hexadecimal system is extensively used to code instructions in assembly language.

Number System Conversion

We can convert the numbers from one system to another. There are various online tools to do that. Python also offers number conversion, but it is better to learn how it is done manually.

Binary to Decimal

Here’s a binary number 01101001, let’s convert it to a decimal number.

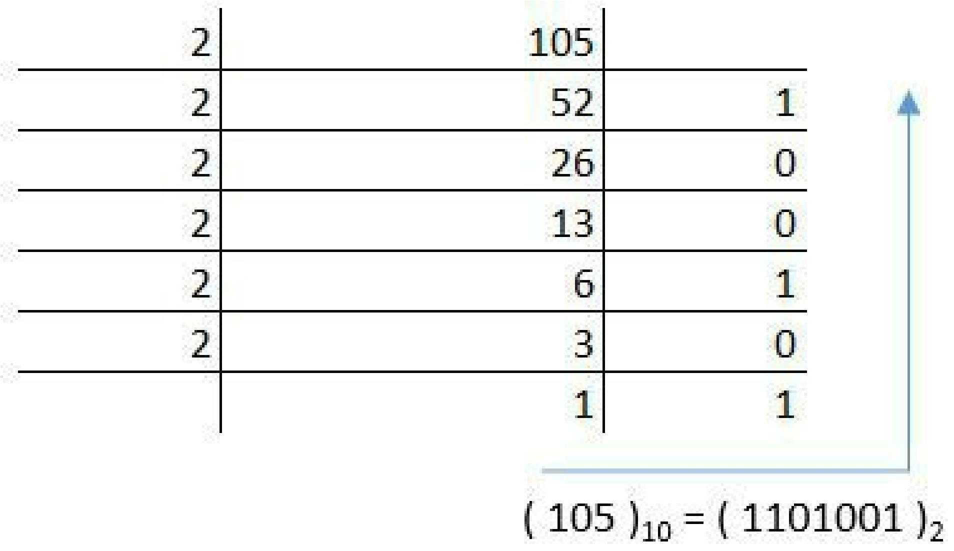
(01101001 $)_2$ = 0 x 2^7 + 1 x 2^6 + 1 x 2^5 + 0 x 2^4 + 1 x 2^3 + 0 x 2^2 + 0 x 2^1 + 1 x 2^0

(01101001 $)_2$ = 0 + 64 + 32 + 0 + 8 + 0 + 0 + 1

(01101001 $)_2$ = (105 $)_{10}$

Decimal to Binary

To convert a decimal number to binary, we have to repeatedly divide the number by two until the quotient becomes one. Recording the remainder generated at each division step gives us the binary equivalent of the decimal number.



2	105	
2	52	1
2	26	0
2	13	0
2	6	1
2	3	0
	1	1

(105 $)_{10}$ = (1101001 $)_2$

An interesting thing to note here is that (01101001 $)_2$ and (1101001 $)_2$ represent the same decimal number (105 $)_{10}$. It means that just like decimal number system, leading zeros can be ignored in the binary number system.

Binary to Hexadecimal

Binary numbers can be converted to hexadecimal equivalents using two methods.

1. Convert the binary number to decimal, then decimal to

hexadecimal number

2. Break binary number in groups of four bits and convert each to its hexadecimal equivalent, keeping the groups' positions in the original binary number intact.

Let's convert (1101001)$_2$ to a hexadecimal number using the second method. The first step is to break the binary number into different groups each of four bits. If the MSB group has less than four bits, make it four by adding leading zeros. Grouping starts from the LSB. So, (1101001)$_2$ will give us (1001)$_2$ and (0110)$_2$. Now, remembering their position in the original binary number, we are going to convert each group to a hexadecimal equivalent.

Here is the table of hexadecimal equivalents of four-bit binary numbers.

Binary	**Hexadecimal**
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A