



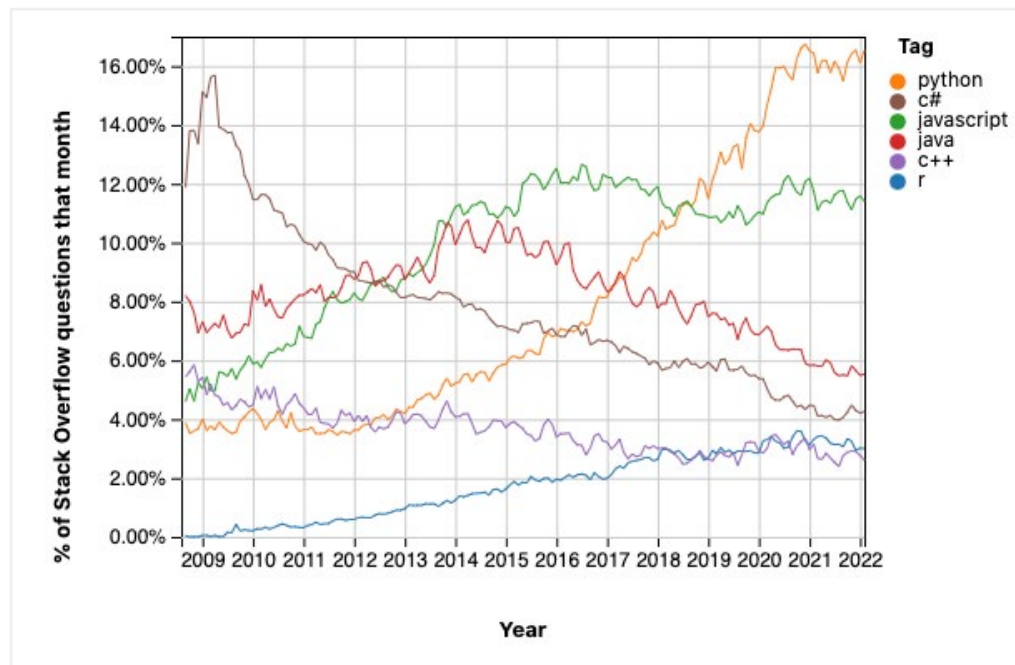
An Introduction to Python

A Guide for Marketers, Developers,
and Data Analysts



Introduction

Python has quickly become the most-searched programming language on Stack Overflow and the #1 language on the [TIOBE index](#), surpassing C and Java in 2022.



Source: [Stack Overflow](#)

With its versatility, simple syntax and formatting, and applications in data, development, and education, it is becoming the preferred programming language of data experts, engineers, and operations professionals in all industries.

This guide will explore Python programming, uses, and best practices, and provide resources to help you leverage Python in your day-to-day role. Let's get started.



Build your optimized website on the HubSpot CMS.

Why do the work when HubSpot can do it for you? Use HubSpot's CMS to build your website without being a programming expert.

HubSpot's customizable, marketer and developer-friendly CMS offers:

- ✓ Flexible website themes
- ✓ Built-in integrations with the HubSpot CRM and 1,000+ other tools
- ✓ SEO recommendations
- ✓ Adaptive testing
- ✓ Drag-and-drop editor
- ✓ Serverless functions
- ✓ And more!

[Get started](#)

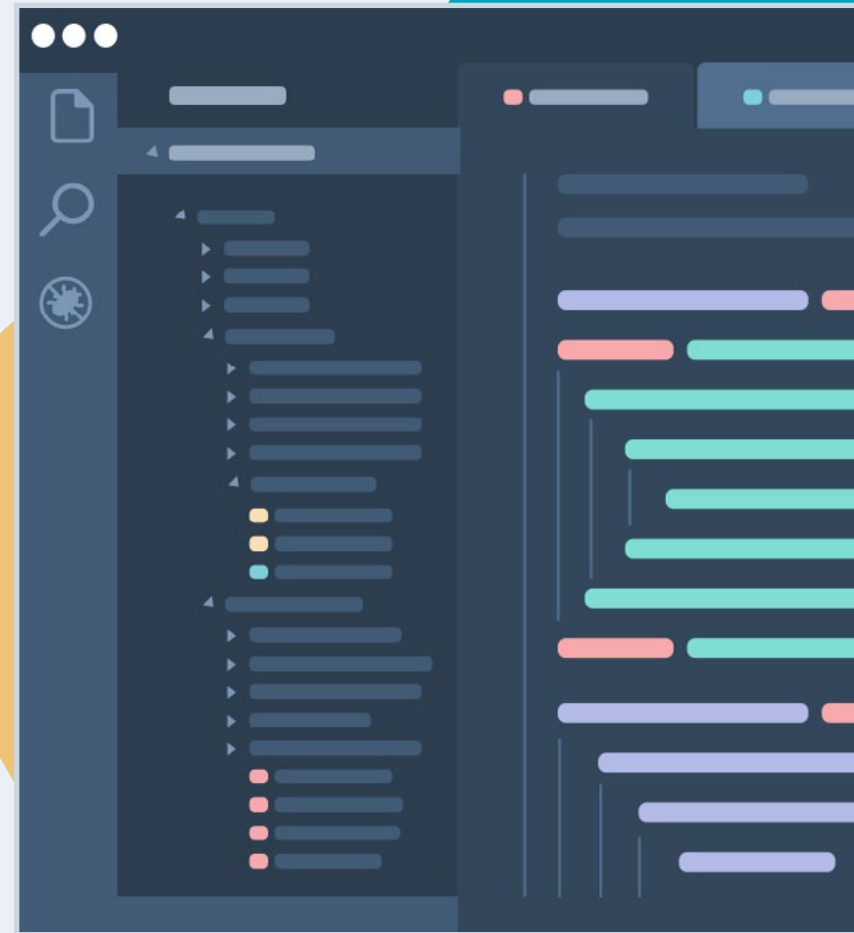


Table of Contents

CHAPTER ONE

What is Python?	4
-----------------------	---

CHAPTER TWO

Methods & Functions	12
---------------------------	----

CHAPTER THREE

Coding Standards and Programming Best Practices	20
---	----

CHAPTER FOUR

Advanced Python	22
-----------------------	----

CHAPTER FIVE

Resources for Future Learning	24
-------------------------------------	----

CLOSING	26
---------------	----

CHAPTER ONE

What is Python?

Python is one of the fastest growing, most multipurpose programming languages available. Originally developed in the 1990s by Dutch programmer Guido van Rossum and expanded by community members over the years, Python has gained popularity due to its collaborative nature, powerful data capabilities, and simplicity.

FUN FACT

Python was named after a Monty Python sketch.

Source: [Python Institute](#)

"Python works for everything. It's a one-size-fits-all, industry agnostic, flexible language that allows teams to capture, enrich, and store data, automate processes, and utilize machine learning, among other things. Organizations depend on Python to function."



Python – a free, open source object-oriented programming (OOP) language

According to the [TIOBE Index](#), which ranks programming language popularity, Python is the top programming language as of February 2022. So, why has Python grown as the top choice for developers, engineers, and data scientists, as well as an increasing number of non-programmers?



Python is:

- Open source, with a strong community of developers ([Pythonistas](#) and Pythoneers) and extensive library of code ([PyPI](#))
- Object-oriented programming (OOP)
- An interpreted programming language
- Free
- Flexible
- Intuitive and simple
- Collaborative
- Trusted
- Easy to learn, reads like English
- Low cost to maintain
- Interpreted, and easier to debug than compiled languages

Python was built on the principles that **code should be simple over complex**, and anyone should be able to understand it. It reads like English, and is easier to debug than compiled languages. Because of this and the vast capabilities that this programming language holds, Python has become a fan favorite in the developer community.



What is Python most commonly used for?

Python is mainly used for data science, data visualization, machine learning, and website and app development. Top platforms like [YouTube](#), [Pinterest](#), [Google](#), [Industrial Light and Magic](#) (special effects company originally founded by George Lucas), and [Instagram](#) are powered by [Python plus Django](#) (a Python-based web framework).

Python can be used to develop websites and apps, to clean, analyze, and model data, to build games and services, to automate processes, and to create machine learning algorithms. It's also an easy language to learn, making it a top choice for teaching software engineering.

Many data experts and engineers use Python when they've maxed out the functionality of Excel or Google Sheets' pivot tables, or it's simply too much data for those programs to execute. Python has proven a more adaptable solution than industry-specific competitors like MATLAB or Perl due to its ease of use, transferability between applications, and lower costs. The platform is free and upkeep is less expensive and specialized than more complex tools.

"The flexibility of being able to program custom code when you have large amounts of data is invaluable. **Python combines a low technical cost of entry while providing a near limitless set of data processing capabilities.**"



[Joe Camilo Ph.D.](#),
[Engineering Director at](#)
[CoVar](#), a machine learning
company.

Industries that Use Python

[Engineering](#)

[Finance](#)

[Marketing](#)

[Business](#)

[Ecommerce](#)

[Math](#)

[Science + Research](#)

[Education](#)

[Gaming](#)

Python for Digital Marketing

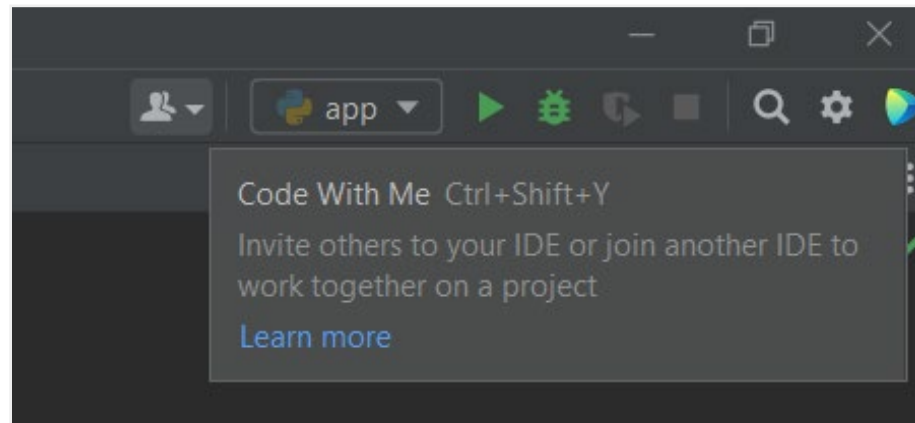
In digital marketing, Python is a powerful data processing tool that picks up where pivot tables and business intelligence tools leave off. Since it's open source, you can connect any API to Python using web hooks. With Python, you can create programs that [automate email responses, data and reporting, data cleanups](#), and other manual tasks. You can pull new trends and patterns from large data sets and even [connect Python with your HubSpot account](#).



Setting Up and Installing Python

To get started using Python, [download the latest version of Python](#) to your computer right from the Python website, or use a [web-based code editor like Codepen](#). It's important to have the latest version to avoid bugs and have your programs run as smoothly as possible. Python is an interpreted language, so you write your Python code files in a text editor, then execute them using an interpreter.

Most people who use Python use an integrated development environment (IDE), which includes both the text editor and the interpreter, displayed in a graphical user interface (GUI), like [PyCharm](#), [VS Code](#), or [Sublime](#), and these vary by industry or preferred programming language. Most are free and they include features like checking code for errors, color coding, customizable UI, plugins, improved code quality, and other collaboration and developer tools.

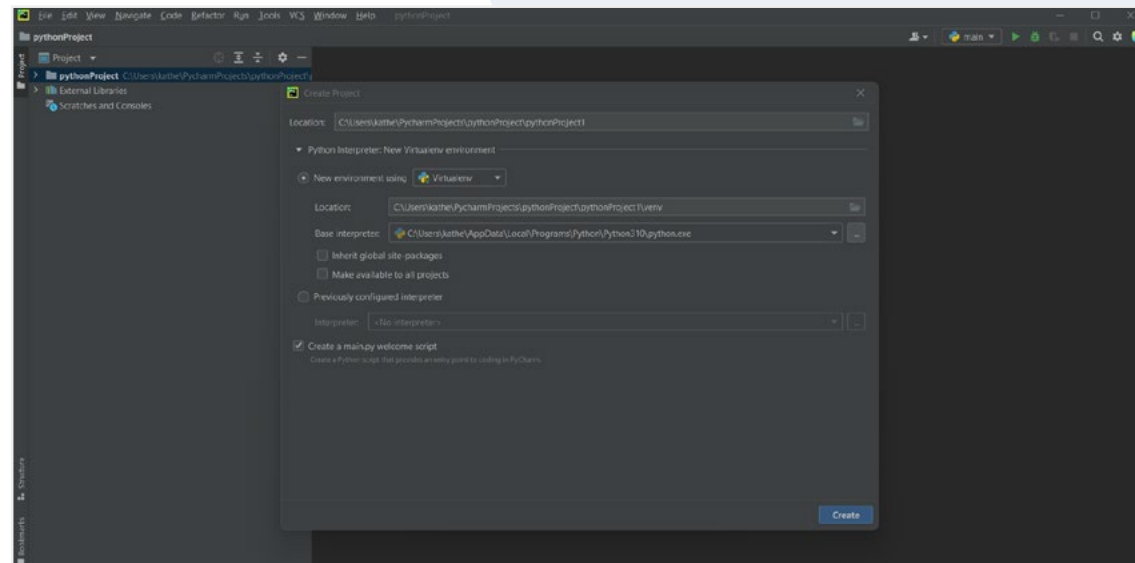
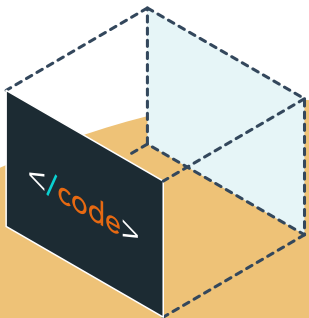


Collaborative Coding in PyCharm, the Python IDE

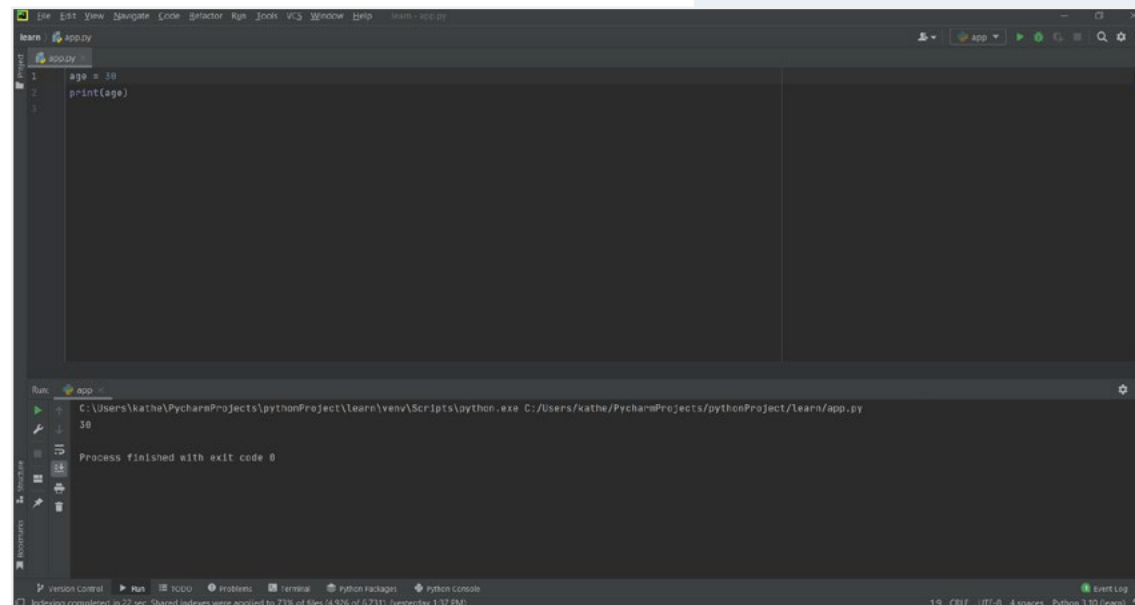
Once you've downloaded PyCharm or another text editor or have Codepen open, you're ready to start writing Python code. [Python has their own helpful guide to getting started](#) that includes resources and additional documentation based on your level of programming experience.

To create a Python program, name your Python file using the command line. Your IDE will prompt the file storage location and create a new virtual environment when you go to create a new project file, saving with the extension .py. Using a command line, here's how to create your Python project:

```
C:\Users\Your Name>python newproject.py
```



Creating a new Python project in PyCharm



PyCharm, a popular Python code editor.

Syntax and Format in Python

Python was designed to read like English and operates using white space formatting or spaces or tabs that denote objects and classes. Python programs execute from top to bottom, line by line, making it efficient to debug since you can troubleshoot an issue and stop the code in the middle, rather than having to stop the entire program like in a compiled language like C++.

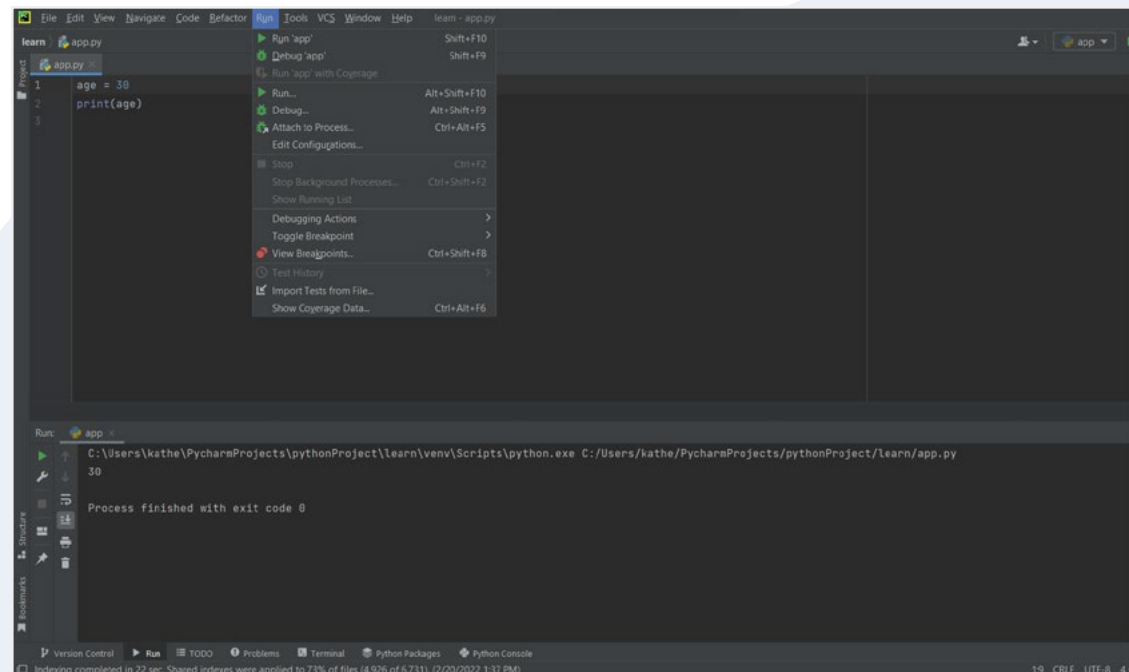
To write in Python, use the code editor in the top screen to write your code. Python code includes declaring variables, creating lists, processing user inputted data, and using functions to manipulate data. This can be coded manually, or implemented using imported packages that others developed. When you are ready to run your code, use the “Run” function and the output will appear in the terminal window below. If there are any errors, Python will identify the error using the line where the error has occurred and provide insights as to why the error is happening.

Once you’ve opened a Python file, basic Python syntax looks like this, and is entered into the code editor:

```
print("Testing this code!")
```

Then, when you run the program, the output, which appears in the terminal window, will be:

```
Testing this code!
```



The “Run” dropdown menu in PyCharm.

PRO TIP

Using keyboard shortcuts is a game-changer in Python.

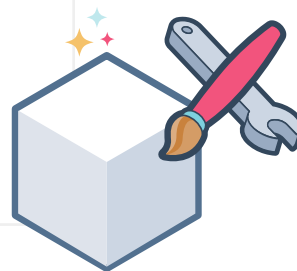
Python Syntax + Formatting Guidelines

- Python is case sensitive
- Built-in Python functions and special keywords will appear in color
- Some of the primary data types available in Python are:
 - Integers (whole numbers like 1, 25, or 160)
 - Floats (numbers with floating points, or decimals, like 10.16, or 8.12)
 - Boolean (True or False)
- Other data types include strings, lists, tuples, dictionaries, and sets
- To define an object, you don't need to use curly brackets { }, instead use tabs and white space to identify the object that a function will apply to
- Always surround text with quotes

To settle the tabs versus spaces debate from Silicon Valley... "It's all spaces," says Camilo.



Source: [@FunFuse on YouTube](#)



CHAPTER TWO

Methods & Functions

Python is an object-oriented programming (OOP) language, meaning the focus of the code is on the objects or datasets, rather than on the functions themselves. Here's a crash course in computer science to set the foundation for Python programming.

PYTHON GLOSSARY		
TERM	DEFINITION	EXAMPLE
Argument	The value that follows the function name, in parentheses. Brings information into a function.	<pre>def test_function(fname): print(fname + " Test")</pre>
Attribute	A value that corresponds to a given attribute, associated using a dot/period.	<pre>Object.attribute</pre>
Class	Combines data and functionality, creates a new type of object. Methods are added to classes to modify them.	<pre>class ClassName: <example-1> . . . <example-N></pre>

PYTHON GLOSSARY

TERM	DEFINITION	EXAMPLE
Command Line	A text-based interface where you enter code. The basis for a command line interface (CLI) where you can execute code.	<pre>\$ python -c "print('Python Test')"</pre> <p>Python Test</p>
Dictionary	A data type that's a group of objects in the form of key-value pairs that map to one another, or a list of values associated with a matching key.	<pre>d = { <key>: <value>, <key>: <value>, . . . <key>: <value> }</pre>
Function	A group of statements that produce a response when called. They can include parameters or return data.	<pre>def test_function(): print("Hello world")</pre>
List	Saves multiple items as a single variable, built-in Python function. Ordered.	<pre>testlist = ["dog", "cat", "lizard"] print(testlist)</pre>

PYTHON GLOSSARY

TERM	DEFINITION	EXAMPLE
Method	A function applied to a class. Includes string methods, list methods, dictionary methods, tuple methods, set methods, and file methods.	<pre>txt = "hello world" x = txt.capitalize() print (x)</pre>
Module	A file with a group of functions, variables, and class definitions that can be imported into other applications and files.	<pre>import testmodule testmodule.greeting("Mario")</pre>
Package	A directory of Python modules, must contain <code>__init__.py</code> file to initialize the package.	<pre>import matplotlib.pyplot as plt plt.plot([1,2,3,4]) plt.ylabel('some numbers') plt.show()</pre>
Script	A file that contains commands that execute as a program.	<pre>print("Hello world")</pre>

PYTHON GLOSSARY

TERM	DEFINITION	EXAMPLE
Set	A built-in data type that stores multiple items in an unordered list in a single variable, defined by curly brackets { }.	<pre>testset = {"dog", "cat", "lizard"} print(testset)</pre>
String	A sequence of characters, textual data. Enclosed in quotes.	<pre>"Hello world"</pre>
Variable	Temporary data storage in a computer's memory. If using multiple words in a variable name, separate them using underscores.	<pre>age = 20</pre>



Object-Oriented Programming Basics

The four main elements of OOPs are encapsulation, abstraction, inheritance, and polymorphism. Object-oriented programming was designed to simplify the code development process, minimizing the need to copy and paste code.

Encapsulation

Encapsulation is when a group of similar variables and functions are combined into one object, by defining them in a group like a class. Encapsulation helps protect and secure data and prevents code from being altered by others. To [make a given element private in Python](#), add an underscore before it in the code, like this:

`_private,`

Or, to perform name mangling, add a double underscore before an attribute, like this:

`__private.`

Encapsulation makes code more efficient by eliminating the need for lengthy parameters attached to functions. And it allows you to reuse objects in different places and programs.

```
1
2 let baseSalary = 30_000;
3 let overtime = 10;
4 let rate = 20;
5
6 function getWage(baseSalary, overtime, rate) {
7   return baseSalary + (overtime * rate);
8 }
9
10 let employee = {
11   baseSalary: 30_000,
12   overtime: 10,
13   rate: 20,
14   getWage: function() {
15     return this.baseSalary + (this.overtime * this.rate);
16   }
17 };
18 employee.getWage();
```

An example of functional programming (above) versus OOP (below) from [Programming with Mosh](#).

Abstraction

Abstraction is when you are able to block off certain functions and methods from the rest of the code. Abstraction is helpful in coding because it:

- Reduces the number of functions and methods needed
- Simplifies the code
- Minimizes the effects of changes

Inheritance

Inheritance is the principle of object-oriented programming that represents the streamlined, simple nature of the language. It helps to reduce redundancy by applying a set of properties and methods to multiple objects. Rather than having to repeat those properties and methods every time, they can inherit the information, reducing the overall amount of code.

Polymorphism

Polymorphism gives OOPs flexibility – rather than applying a method in the same way to each element or object, methods are applied to individual objects and can be run in different ways based on the type of object.

This replaces more complex code called a [switch-case statement](#) in other programming languages. In Python, you can write one line of code with a method added on to achieve the same result.

OOPs vs Functional Programming Languages

OOPS	FUNCTIONAL PROGRAMMING
<ul style="list-style-type: none">• Newer programming method• Related variables and functions are combined into objects• Functions in OOP languages are methods• Less parameters on functions, making them more flexible	<ul style="list-style-type: none">• Older programming method• Functions operate on data, variables and functions are separate• Bulkier• Harder to debug• Functions depend on each other, leading to broken code when one is changed• Lots of parameters on functions



Using Functions and Methods in Python

Functions are elements of code that produce a result and can pass parameters (data), which are noted in parentheses following the function. An example of a function is:

```
def test_function():  
    print("Hello world")
```

A method is a modification applied to an object that can operate or return data within a given class. In other words, it's a function applied to an object. An example of a method, `.upper`, which converts the string into all uppercase letters, is:

```
message = 'intro to python'  
print(message.upper())
```

The output for the `.upper` method in the above example is "INTRO TO PYTHON".

The difference between methods and functions in Python is that methods are associated with a class and dependent on that class, but functions are independent. Python has a number of [built-in functions](#), including common go-tos like `print()`, `range()`, and `list()`.

Basic Python Functions

Arithmetic operators: Includes functions like add, subtract, multiple, divide, percentage, and to the power of as built-in functions.

Comparison operators: Greater than or equal to, greater than, less than or equal to, less than, equal to, not equal to.

Logical operators: And, or, and not. Combines arithmetic and comparison operators into logical statements.

If statements: If, then statements.

Leveraging PyPI

The Python Package Index

[PyPI](#) is the centralized database of Python packages maintained by the community – it's a place to upload or access packages for different use cases, industries, and data needs. With more than 350,000 projects, you're bound to find something that helps with your next project.

Common PyPI Packages and Their Uses

NumPy

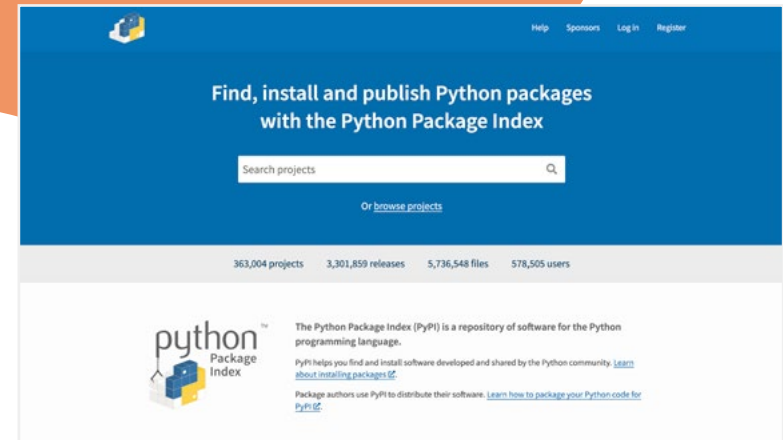
A scientific computing package that includes mathematical functions, random number generators, linear algebra routines, and more.

pandas

A data manipulation package that helps with data alignment, merging and joining data sets, aggregating data, time series-functionality, and more for academic and commercial purposes.

Matplotlib

A 2D data visualization and plotting library for Python that includes plotting commands to create visualizations like bar plots, contour plots, and more.



"The secret to coding is knowing where to look,"



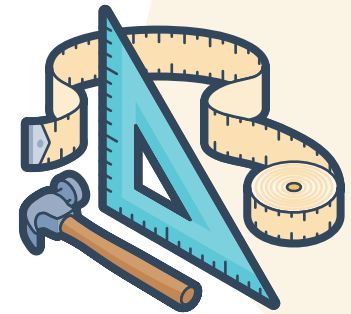
[Emma Cotter](#), Ph. D.,
Environmental research
engineer and data expert.

CHAPTER THREE

Coding Standards and Programming Best Practices

Good quality Python code should follow [PEP 8](#), the most up-to-date style guide for Python that's based on Guido van Rossum's original style guide and tenets of the language.

Here's what distinguishes good Python programming from bad programming, and makes a Pythonista out of a developer. When in doubt – make it readable.



The Principles of Good Python Programming

- Maintain consistency within modules, functions, and projects
- Use spaces for indentation
- Avoid unnecessary whitespace
- When naming files that are public on the API, describe usage rather than implementation
- Use underscores between words
- Don't use lowercase l (el), uppercase O (oh), or uppercase I (eye) as single letter variable names since they appear as 1 and 0 in some fonts
- Use short, lowercase names for modules with or without underscores

Using Black, "The Uncompromising Code Formatter"

When reviewing Python code for compliance, [Black](#), a unique Python tool, is a developer's best friend. The Black program automatically formats code to comply with `pycodestyle` best practices without impacting the integrity of the code.

"I can now do refactorings in about 1% of the keystrokes it would have taken me previously when we had no way for code to format itself," says [Mike Bayer](#), a satisfied Black user.

Poor Python Formatting vs. Good Python Formatting Examples

Python code should be readable, simple, and always use less code when possible. Here are some examples of clunky strings and functions and how they can be simplified.

- 1 Only use one statement per line. Don't combine disjointed statements, like if <complex comparison> and <other complex comparison> in the example below, on the same line.

Bad

```
print('one'); print('two')

if x == 1: print('one')

if <complex comparison> and <other complex comparison>:
    # do something
```

Good

```
print('one')
print('two')

if x == 1:
    print('one')

cond1 = <complex comparison>
cond2 = <other complex comparison>
if cond1 and cond2:
    # do something
```

Source: [From The Hitchhiker's Guide to Python](#)

- 2 Keep code structure as simple as possible. While both statements in the below example achieve the same outcome, the one denoted as #pythonic is more straightforward and considered "better" Python code.

```
1 l = []
2
3 # pythonic
4 if l:
5     # statement
6 # none pythonic
7 if len(l) != 0:
8     # statement
```

Source: [Quora](#)

- 3 Be explicit – concrete and specific, and speak to the behavior of functions in the code. Below, another developer would understand the function in the good example.

Bad

```
def make_complex(*args):
    x, y = args
    return dict(**locals())
```

Good

```
def make_complex(x, y):
    return {'x': x, 'y': y}
```

Source: [From The Hitchhiker's Guide to Python](#)

CHAPTER FOUR

Advanced Python

Python enables developers to connect complex data sets, to automate processes, and to visualize data. Here are more Python elements and functions that enable more advanced programs.

Explore [W3's Python Tutorials](#) for detailed practice examples and additional exercises.



Advanced Python Functions + Features

Dictionary

A collection of key-value pairs that match values in an associative array.

```
d = {  
    <key>: <value>,  
    <key>: <value>,  
    <key>: <value>  
}
```

Set

A built-in Python data type that stores multiple items in an unordered list in a single variable, defined by curly brackets { }. Duplicates are not allowed in sets.

```
testset = ["dog", "cat", "lizard"]  
print(testset)
```

List

A built-in Python function that saves multiple ordered items as a single variable. When indexed, the first item is index [0], then [1], and so on. Lists allow for duplicate values.

```
testlist = ["dog", "cat", "lizard"]  
print(testlist)
```

Command Line + Command Line Arguments

The command line is where you invoke Python and can add options like -c (command), -m (module-name), scripts, and arguments. The interpreter reads the command line within the environment to learn which settings to use.

Command-line arguments are additional information within the line of a program after the program.py on the command line.

While Loops

Executes a set of functions or statements *while* a certain condition remains true.

```
i = 1
while i < 5:
    print(i)
    i += 1
```

For Loops

Executes a set of functions or statements for items in lists, tuples, dictionaries, sets, or strings.

```
pets = ["dog", "cat", "lizard"]
for x in pets:
    print(x)
```

List Methods

Object-based functions that are applied to lists.

```
pets = ["dog", "cat", "lizard"]
pets.append("parakeet")
```

The Range Function

A built-in Python function that returns a sequence of numbers. Ranges start at the first number, go up to (but not including) the second number, and use the third number as the increment. Requires integers.

```
x = range(4, 24, 2)
for x in x:
    print(n)
```

Tuples

An ordered list that is unchangeable and stores multiple items in one variable. Allows duplicate values, and is indexed, starting with the first item as [0].

```
thistuple = ("dog", "cat", "lizard", "cat", "lizard")
print(thistuple)
```

CHAPTER FIVE

Resources for Future Learning

To become a better developer, like becoming a better writer (and Python is the closest programming language to English, so it still works), you have to read and keep up with the latest best practices in the community. And write and review code in collaboration with other programmers. Here are some resources for continuing on your professional journey as a developer and honing your Python skills.

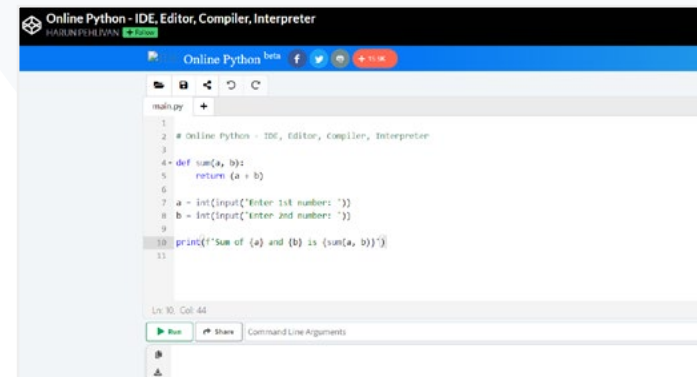
1 Industry Publications

- [The HubSpot Website Blog](#)
- [Science of Computer Programming Journal](#)
- [We Are Developers](#)
- [Towards Data Science](#)
- [Analytics Vidhya](#)
- [Code2040](#)

2

Virtual Code Editors

Browser-based code editors like Codepen provide a virtual environment to practice Python from anywhere.



3

Integrated Development Environments

- [PyCharm](#)
- [VS Code](#)
- [Sublime](#)
- [PyDev](#)

4 Frameworks

Frameworks are platforms for developing software, websites, and applications that simplify the development process and help programmers speed up their processes in a standardized way.

Types of frameworks include full stack frameworks, micro frameworks, and asynchronous frameworks.

Popular frameworks for Python include:

Django

A web app framework “for perfectionists” that can be used for larger, more complex apps and websites.

Masonite

A web development app designed for software and used for SaaS.

CherryPy

A “minimalist Python web framework” that reduces source code size and speeds up development.

Hug

A Python API framework that “drastically simplifies API development over multiple interfaces,” and is the “fastest and most modern way to create APIs on Python3”.

5 Developer Forums + Communities

No matter how long you’ve been coding, every developer will at some point or another run into something they need help with. Chances are, someone else has had the same issue. Rather than wasting hours of your life attempting to troubleshoot solo, turn to the Python developer community and see if a Pythonista or a Pythoneer can lend a helping hand.

- [Python Forum](#)
- [Stack Overflow](#)
- [Reddit: r/python](#)
- [Reddit: r/learnpython](#)
- [Women Who Code: Python](#)
- [Black Girls Code](#)
- [Black Tech Nation](#)



Closing

Python has quickly become one of the most popular, versatile programming languages used by developers. And just as Guido van Rossum imagined when he first developed it, the more people who contribute to the community, the richer and better the language becomes.

We hope this guide helped to introduce you to the vast capabilities that Python has to offer and inspired you to automate processes, explore data, and leverage the PyPI database to rethink the way you work.



Power your website with what it needs to grow better

The HubSpot Website Blog has all the latest tips, best practices, and inspiration for building websites that help businesses grow.

Whether you're a marketer, developer, designer, or strategist, you need to know the most up-to-date information about building and optimizing your website.

The HubSpot Website blog publishes helpful guides and posts, from technical to inspirational, like:

- 10 Best GDPR-Compliant WordPress Themes
- DataOps vs. DevOps: What's the Difference?
- 12 Websites with Stunning Background Images [+ Best Design Practices]

[Subscribe to the HubSpot Website blog](#)