

PYTHON CHEAT SHEET

WHAT IS PYTHON?

Python is one of the top three most popular programming languages, it's often used to build websites and software, automate tasks, and conduct data analysis.

Python is a general purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today.

 The name Python comes from Monty Python. When Guido van Rossum was creating Python, he was also reading the scripts from BBC's Monty Python's Flying Circus. He thought the name Python was appropriately short and slightly mysterious.



HOW TO INSTALL PYTHON?

Python comes preinstalled on most Linux distributions and is available as a package on all others. Unlike most Unix systems and services, Windows does not include a system supported installation of Python. [Learn more about different versions and installation types >](#)

CODING IN PYTHON

Hello World

```
print('Hello, world!')
```

Hello World with variable

```
msg = "hello world" print(msg)
```

Variables Assignments

```
>>> x=8
>>> X
8
```

User Input

```
print('What is your name?')
myName = input()
print('It is nice to meet you, {}'.format(myName))
```

Comparison Operators

Operator	Meaning
<code>==</code>	Equal to
<code>!=</code>	Not equal to
<code><</code>	Less than
<code>></code>	Greater than
<code><=</code>	Less than or equal to
<code>>=</code>	Greater than or equal to

Boolean Operators

```
print(2 < 8)
>>> True
print(2 > 8)
>>> False
```

Python Membership Operators

```
in
x in y
```

Returns True if a sequence with the specified value is present in the object

```
not in
x not in y
```

Returns True if a sequence with the specified value is not present in the object

WHAT IS PIP?

Pip is the standard package manager for Python. It allows you to install and manage additional packages that are not part of the Python standard library. Package management is highly important that pip has been included with the Python installer from versions 3.4 for Python 3 and 2.7.9 for Python 2, and is used by many Python projects.

MOST DOWNLOADED PIP PACKAGES?

1. [setuptools](#) - a fully-featured, actively-maintained, and stable library designed to facilitate packaging Python projects.
2. [botocore](#) - a low-level interface to a growing number of Amazon Web Services. The botocore package is the foundation for the [AWS CLI](#) as well as [boto3](#).
3. [urllib3](#) - a powerful, user-friendly HTTP client for Python. Much of the Python ecosystem already uses urllib3 and you should too.
4. [boto3](#) - the Amazon Web Services (AWS) Software Development Kit (SDK) for Python, which allows Python developers to write software that makes use of services like Amazon S3 and Amazon EC2.
5. [six](#) - a Python 2 and 3 compatibility library. It provides utility functions for smoothing over the differences between the Python versions with the goal of writing Python code that is compatible on both Python versions.
6. [s3transfer](#) - a Python library for managing Amazon S3 transfers.
7. [python-dateutil](#) - provides powerful extensions to the standard datetime module, available in Python.
8. [requests](#) - a simple, yet elegant, HTTP library.
9. [idna](#) - supports the Internationalised Domain Names in Applications (IDNA) protocol as specified in [RFC 5891](#).
10. [pyyaml](#) - a YAML parser and emitter for Python.

PYTHON & JFROG

JFrog provides an end-to-end solution for your Pypi packages. Easily manage your pypi packages using JFrog Artifactory with [local](#), [remote](#) and [virtual repositories](#).

You can also scan your Pypi packages with [JFrog Xray](#), to identify and open the different layers of your Python packages and their dependencies and analysis on each one to ensure that your npm application is safe to use.

