



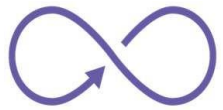
Basic Programming

Lesson 01

Python Introduction



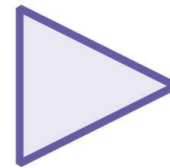
High-level Programming Language



Dynamic



Readability

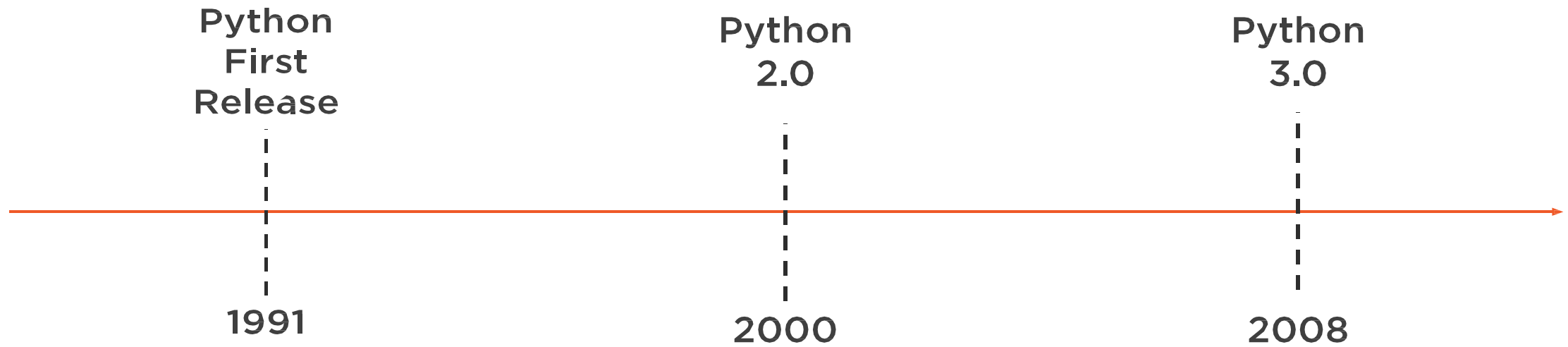


Interpreted

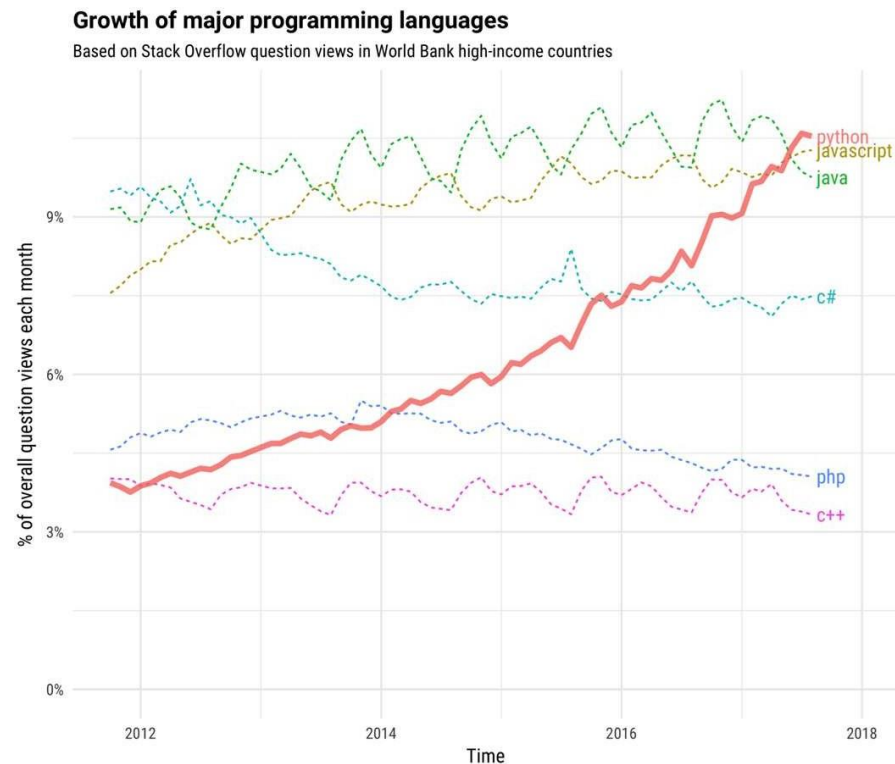


Multi-paradigm

History



Growing in Popularity



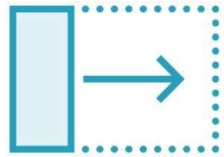
Worldwide, Sept 2017 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Java	22.4 %	-0.8 %
2		Python	17.0 %	+4.0 %
3		PHP	8.7 %	-1.0 %
4		C#	8.1 %	-0.4 %
5		Javascript	8.0 %	+0.6 %
6		C++	6.8 %	-0.2 %
7		C	6.1 %	-1.1 %
8	↑	R	3.7 %	+0.6 %
9	↓	Objective-C	3.5 %	-1.4 %
10		Swift	2.9 %	-0.1 %

Stack Overflow (<http://bit.ly/2wkMATk>)

PYPL (<http://bit.ly/2eJ2rnC>)

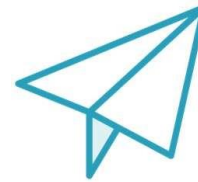
What makes Python different?



Extensible
Design



Community
Involved
Design



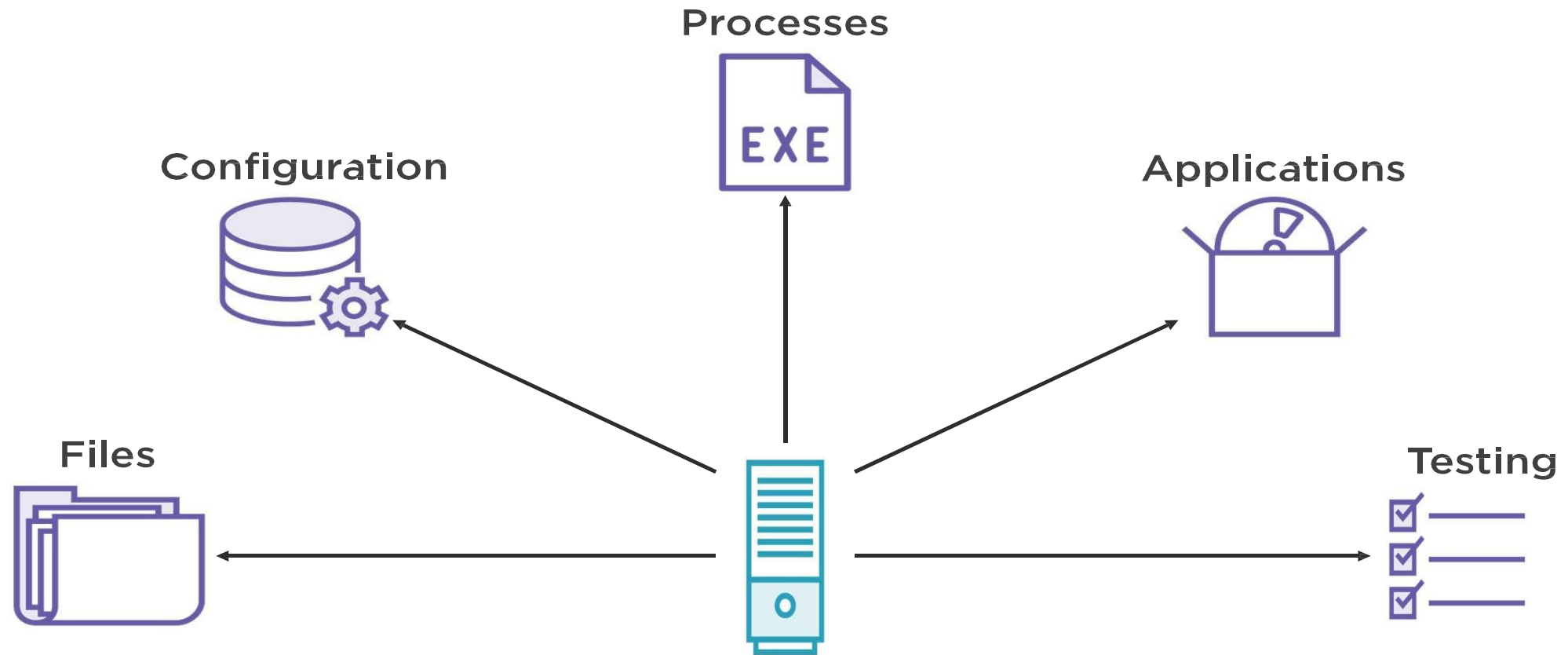
Emphasizing
Fun



Culture

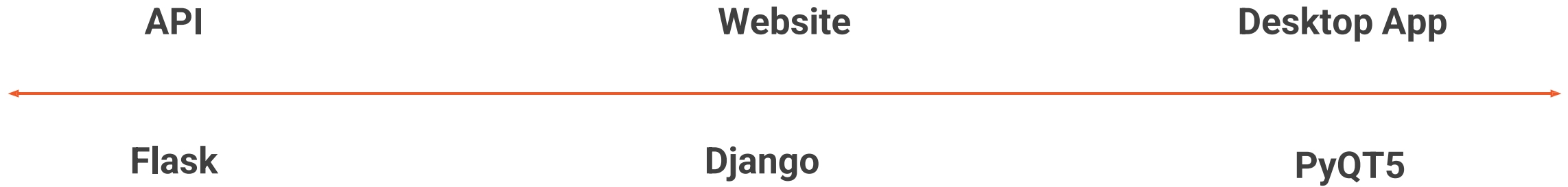
When and Where is Python being used?

System Administration





Application Development





Big Data

Kilobytes

Megabytes

Gigabytes

Terabytes

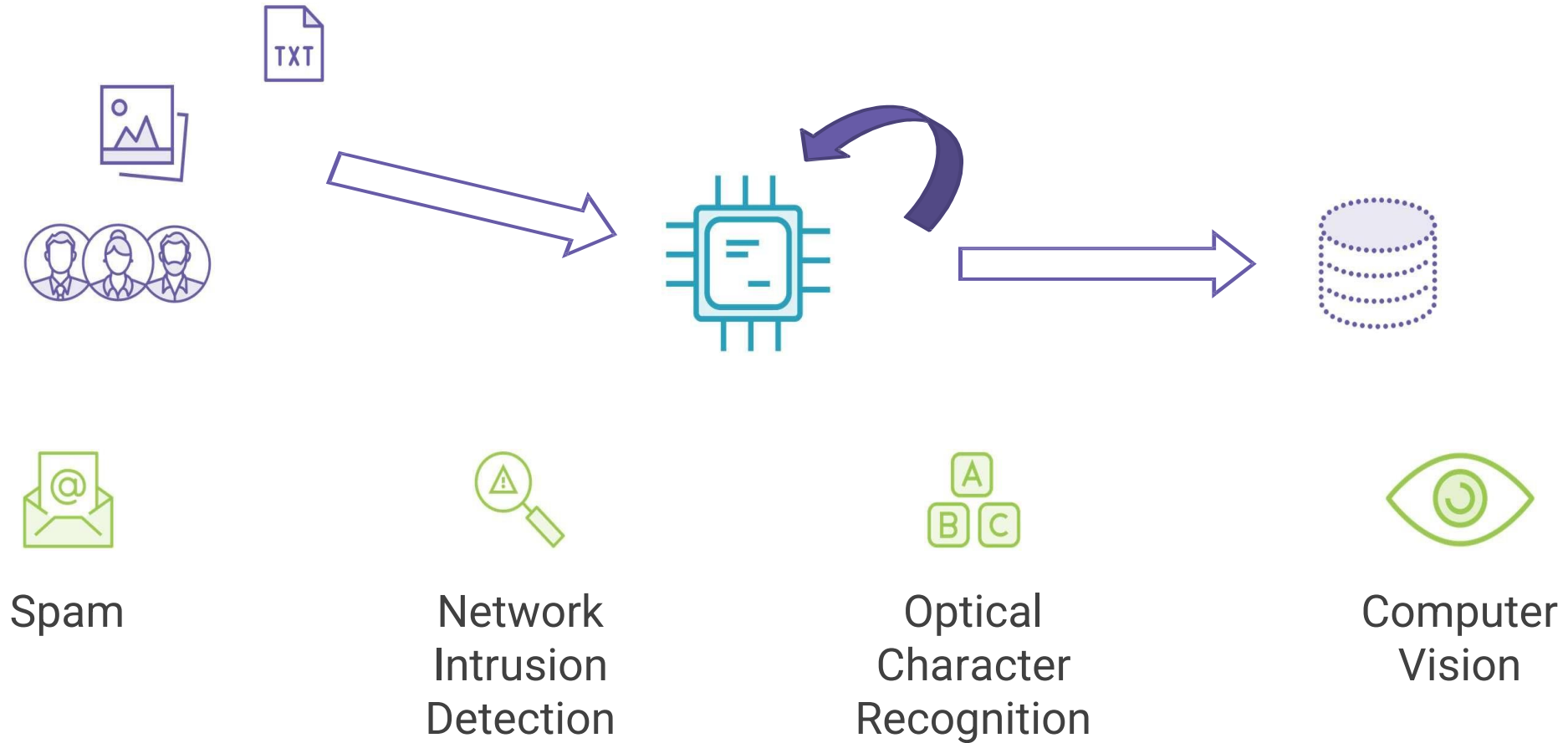
Petabytes

Exabytes

25000000000000000000 bytes/day

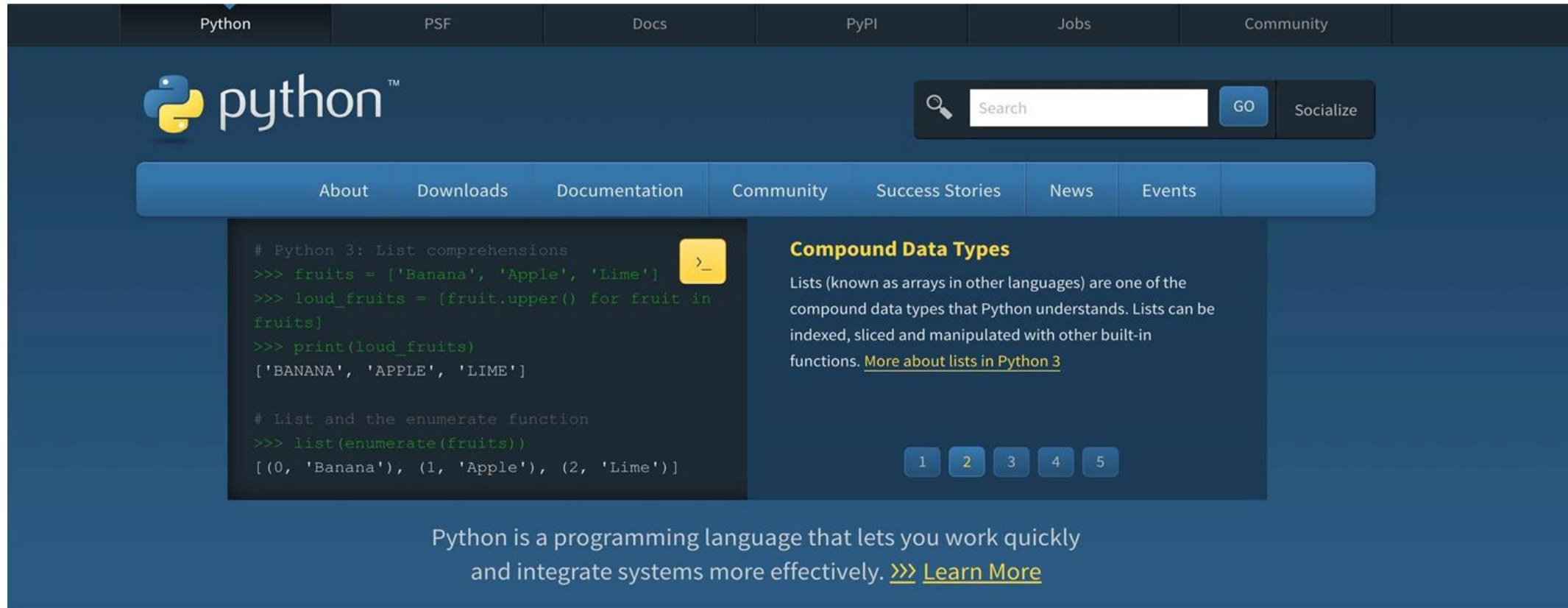
New Systems Are Needed!

Artificial Intelligence/Machine Learning



Documentation

https://python.org/



The screenshot shows the Python.org homepage with a dark blue header and navigation bar. The main content area features a code snippet on the left, a section on 'Compound Data Types' on the right, and a large blue banner at the bottom with the text 'Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)'.

Python PSF Docs PyPI Jobs Community

python™

Search GO Socialize

About Downloads Documentation Community Success Stories News Events

```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']

# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```

Compound Data Types

Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions. [More about lists in Python 3](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use

Download

Python source code and installers are available for download for all versions! Not sure which version to

Docs

Documentation for Python's standard library, along with tutorials and guides, are available online

Jobs

Looking for work or have a Python related position that you're trying to hire for? Our **relaunched**

Install Python

python.org/downloads/windows/

Python Releases for Windows

- [Latest Python 3 Release - Python 3.10.0](#)
- [Latest Python 2 Release - Python 2.7.18](#)

Stable Releases

- [Python 3.10.0 - Oct. 4, 2021](#)

Note that Python 3.10.0 cannot be used on Windows 7 or earlier.

- Download [Windows embeddable package \(32-bit\)](#)
- Download [Windows embeddable package \(64-bit\)](#)
- Download [Windows help file](#)
- Download [Windows installer \(32-bit\)](#)
- Download [Windows installer \(64-bit\)](#)

- [Python 3.7.12 - Sept. 4, 2021](#)

Note that Python 3.7.12 cannot be used on Windows XP or earlier.

- No files for this release.

- [Python 3.6.15 - Sept. 4, 2021](#)

Note that Python 3.6.15 cannot be used on Windows XP or earlier.

python.org/downloads/mac-osx/

Python Releases for macOS

- [Latest Python 3 Release - Python 3.10.0](#)
- [Latest Python 2 Release - Python 2.7.18](#)

Stable Releases

- [Python 3.10.0 - Oct. 4, 2021](#)

- Download [macOS 64-bit universal2 installer](#)

- [Python 3.7.12 - Sept. 4, 2021](#)

- No files for this release.

- [Python 3.6.15 - Sept. 4, 2021](#)

- No files for this release.

- [Python 3.9.7 - Aug. 30, 2021](#)

- Download [macOS 64-bit Intel installer](#)

- Download [macOS 64-bit universal2 installer](#)

- [Python 3.8.12 - Aug. 30, 2021](#)

- No files for this release.

- [Python 3.9.6 - June 28, 2021](#)

- Download [macOS 64-bit Intel installer](#)

- Download [macOS 64-bit universal2 installer](#)

https://docs.python.org/3/

Python » English » 3.10.0 » 3.10.0 Documentation » | modules | index

Download

Download these documents

Docs by version

Python 3.11 (in development)
Python 3.10 (stable)
Python 3.9 (stable)
Python 3.8 (security-fixes)
Python 3.7 (security-fixes)
Python 3.6 (security-fixes)
Python 3.5 (EOL)
Python 2.7 (EOL)
All versions

Other resources

PEP Index
Beginner's Guide
Book List
Audio/Visual Talks
Python Developer's Guide

Python 3.10.0 documentation

Welcome! This is the official documentation for Python 3.10.0.

Parts of the documentation:

What's new in Python 3.10?

or all "What's new" documents since 2.0

Tutorial

start here

Library Reference

keep this under your pillow

Language Reference

describes syntax and language elements

Python Setup and Usage

how to use Python on different platforms

Python HOWTOs

in-depth documents on specific topics

Indices and tables:

Global Module Index

quick access to all modules

General Index

all functions, classes, terms

Glossary

the most important terms explained

Installing Python Modules

installing from the Python Package Index & other sources

Distributing Python Modules

publishing modules for installation by others

Extending and Embedding

tutorial for C/C++ programmers

Python/C API

reference for C/C++ programmers

FAQs

frequently asked questions (with answers!)

Search page

search this documentation

Complete Table of Contents

lists all sections and subsections

Python Release Notes

This is the stable release of Python 3.10.0

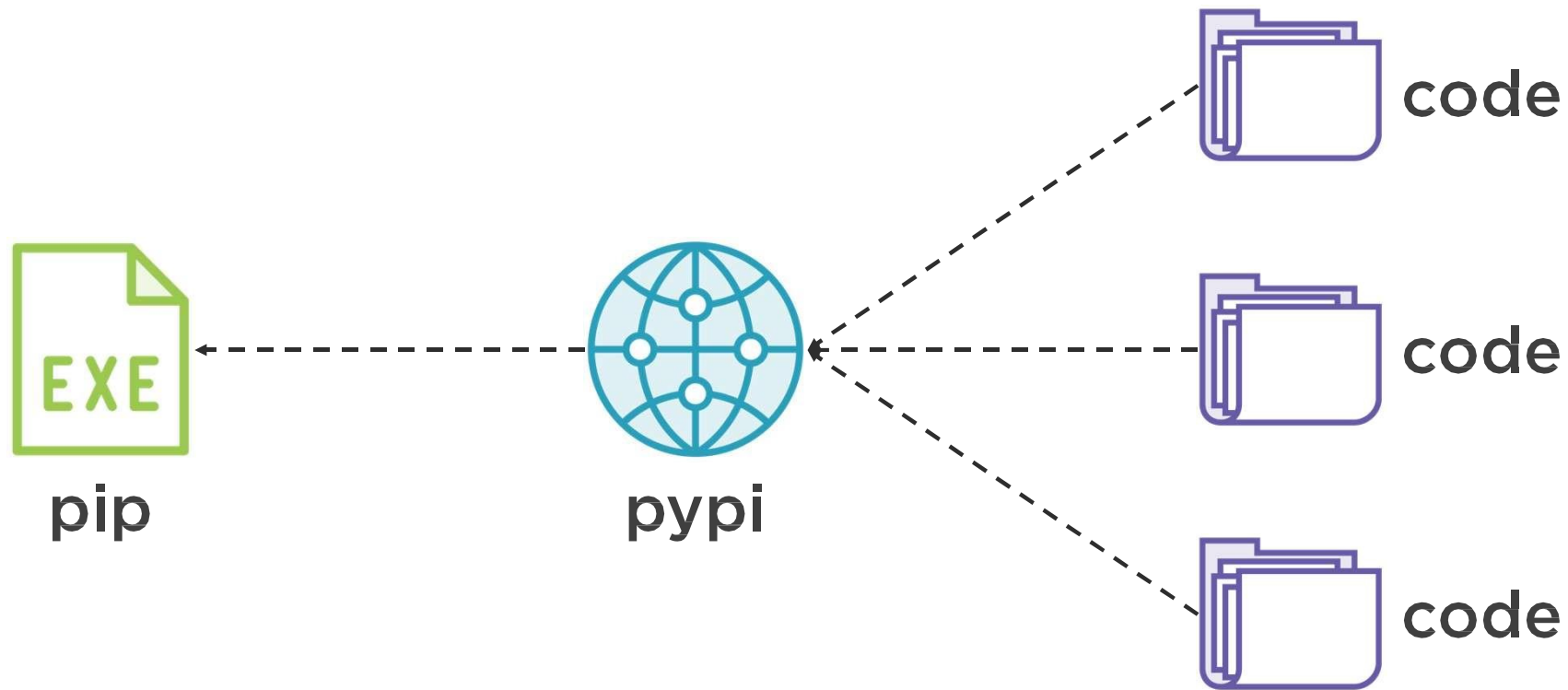
Python 3.10.0 is the newest major release of the Python programming language, and it contains many new features and optimizations.

Major new features of the 3.10 series, compared to 3.9


Among the new major new features and changes so far:

- [PEP 623](#) -- Deprecate and prepare for the removal of the wstr member in PyUnicodeObject.
- [PEP 604](#) -- Allow writing union types as X | Y
- [PEP 612](#) -- Parameter Specification Variables
- [PEP 626](#) -- Precise line numbers for debugging and other tools.
- [PEP 618](#) -- Add Optional Length-Checking To zip.
- [bpo-12782](#): Parenthesized context managers are now officially allowed.
- [PEP 632](#) -- Deprecate distutils module.
- [PEP 613](#) -- Explicit Type Aliases
- [PEP 634](#) -- Structural Pattern Matching: Specification
- [PEP 635](#) -- Structural Pattern Matching: Motivation and Rationale
- [PEP 636](#) -- Structural Pattern Matching: Tutorial
- [PEP 644](#) -- Require OpenSSL 1.1.1 or newer
- [PEP 624](#) -- Remove Py_UNICODE encoder APIs
- [PEP 597](#) -- Add optional EncodingWarning

Working with the Code of others




https://pypi.org/



Help Sponsors Log in Register

Find, install and publish Python packages with the Python Package Index




Or [browse projects](#)

332,205 projects 2,932,717 releases 4,977,415 files 542,372 users



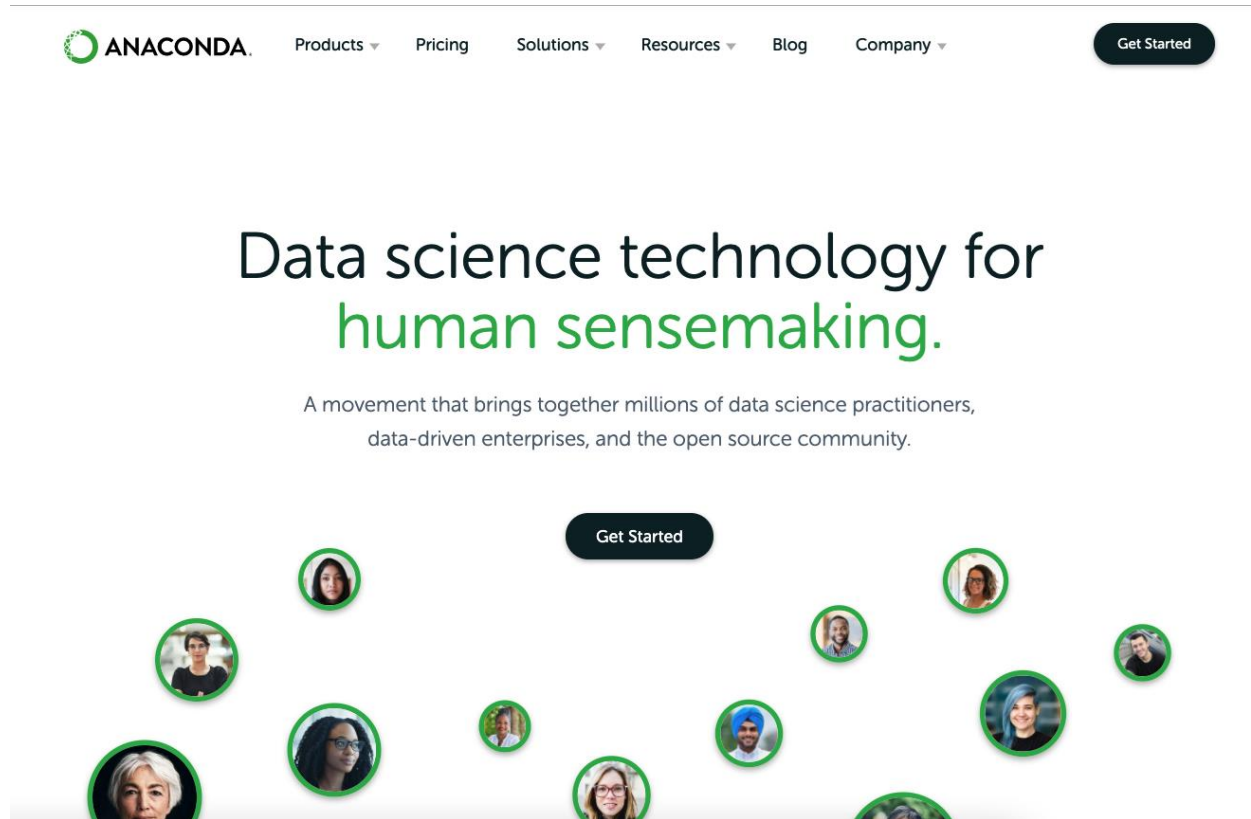
The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. [Learn about installing packages](#) .

Package authors use PyPI to distribute their software. [Learn how to package your Python code for PyPI](#) .

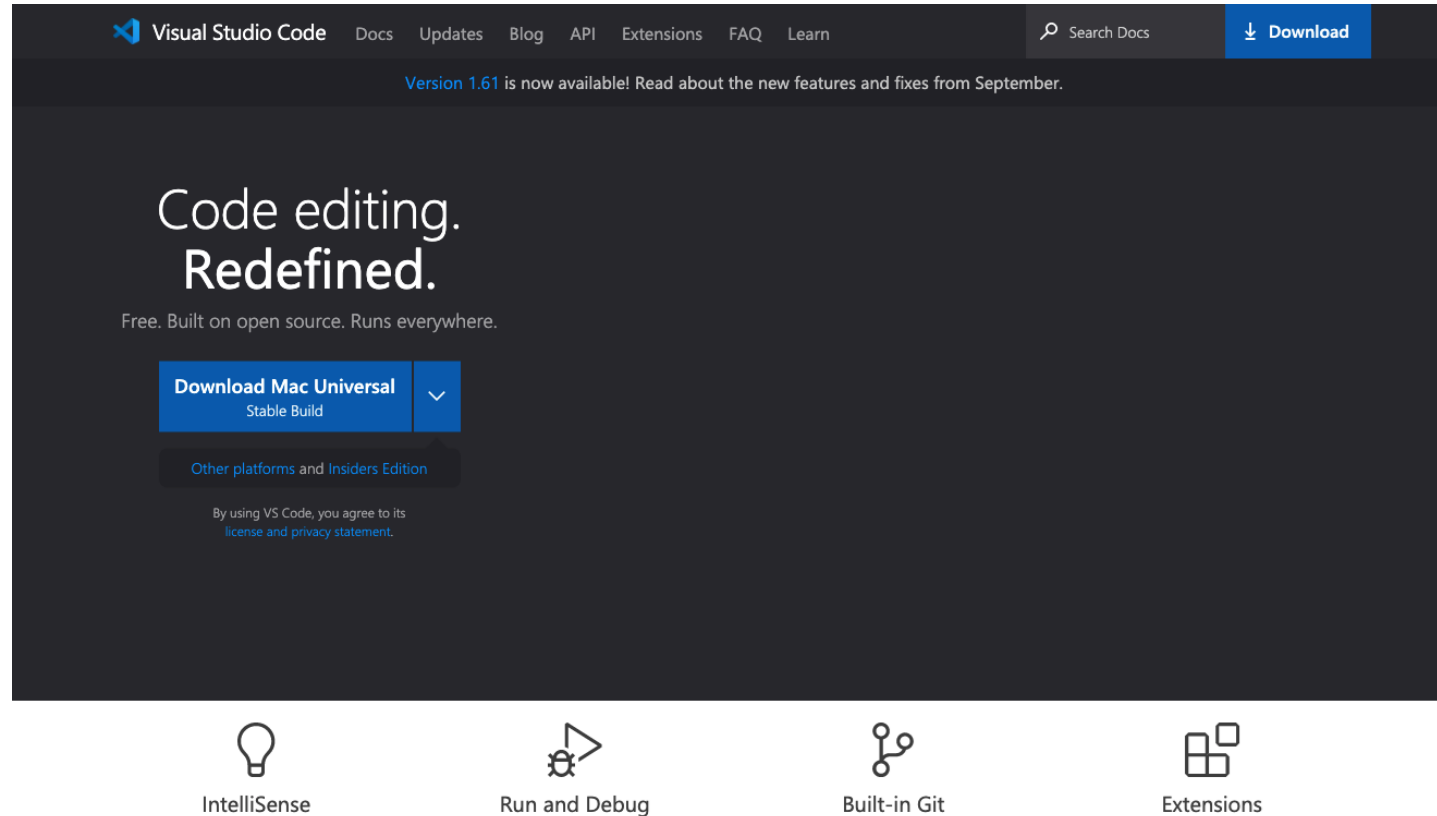
Install the Python environment

Install Anaconda



- Python and Pip will automatically be installed
- Download: <https://www.anaconda.com/products/individual>

Install Visual Studio Code and Extensions



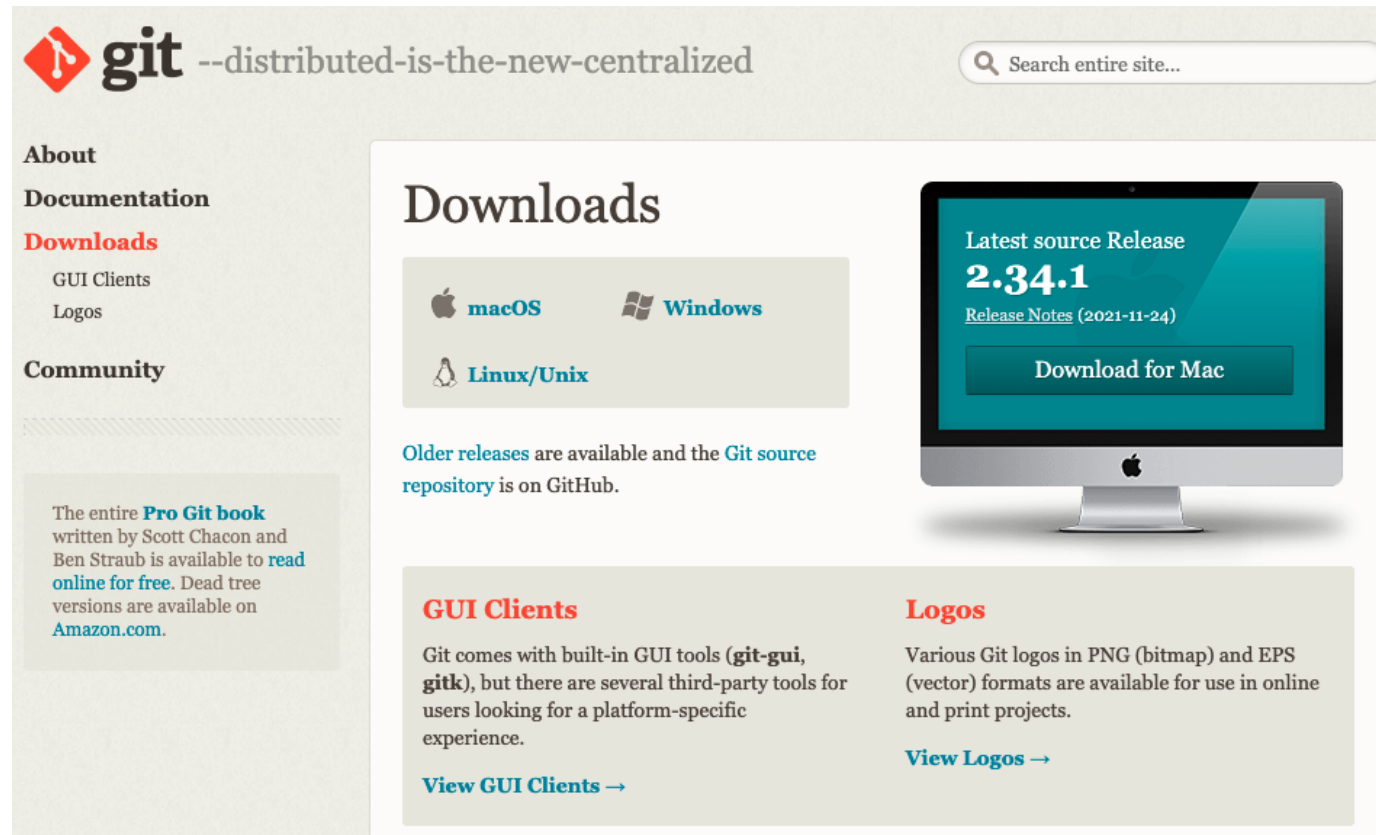
- A code editor redefined and optimized for building and debugging
- Download: <https://code.visualstudio.com/download>

Install Visual Studio Code and Extensions



- Install: Python, GitLens, Prettier, Todo Tree

Install Git



- Git is a free and open source distributed version control system
- Download: <https://git-scm.com/downloads>

Anaconda Commands

To create an environment with Python 3.7:

- `conda create --name myenv python=3.7`

To activate an environment:

- `conda activate myenv`

To deactivate an environment:

- `conda deactivate`

To remove an environment:

- `conda remove --name myenv --all`

To list all environments:

- `conda info --envs`

Reference: [Here](#)

Git Commands

To clone a repository into local:

- `git clone <repository_url>`

To download objects and refs from another repository:

- `git fetch`

To pull source code from a repository:

- `git pull`

To add file contents to the index:

- `git add .`

To commit changes to the repository:

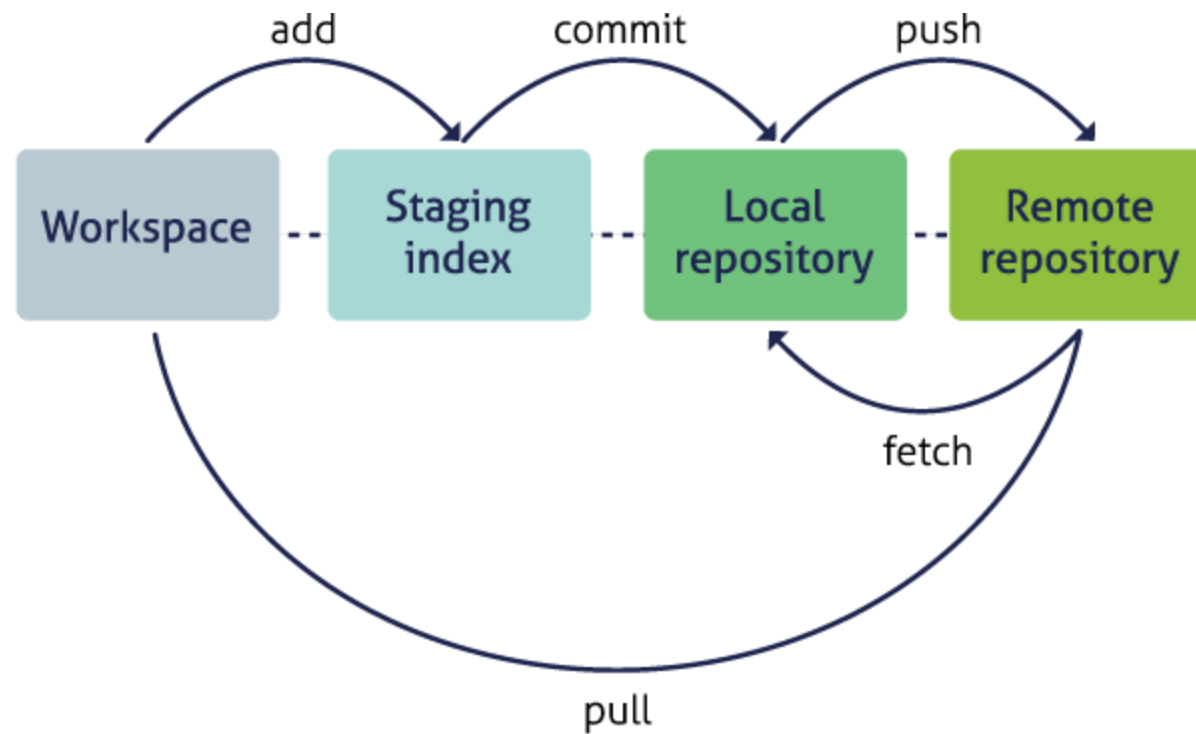
- `git commit -m "This is a message"`

To push changes to a repository:

- `git push`

Reference: https://git-scm.com/docs/git#_git_commands

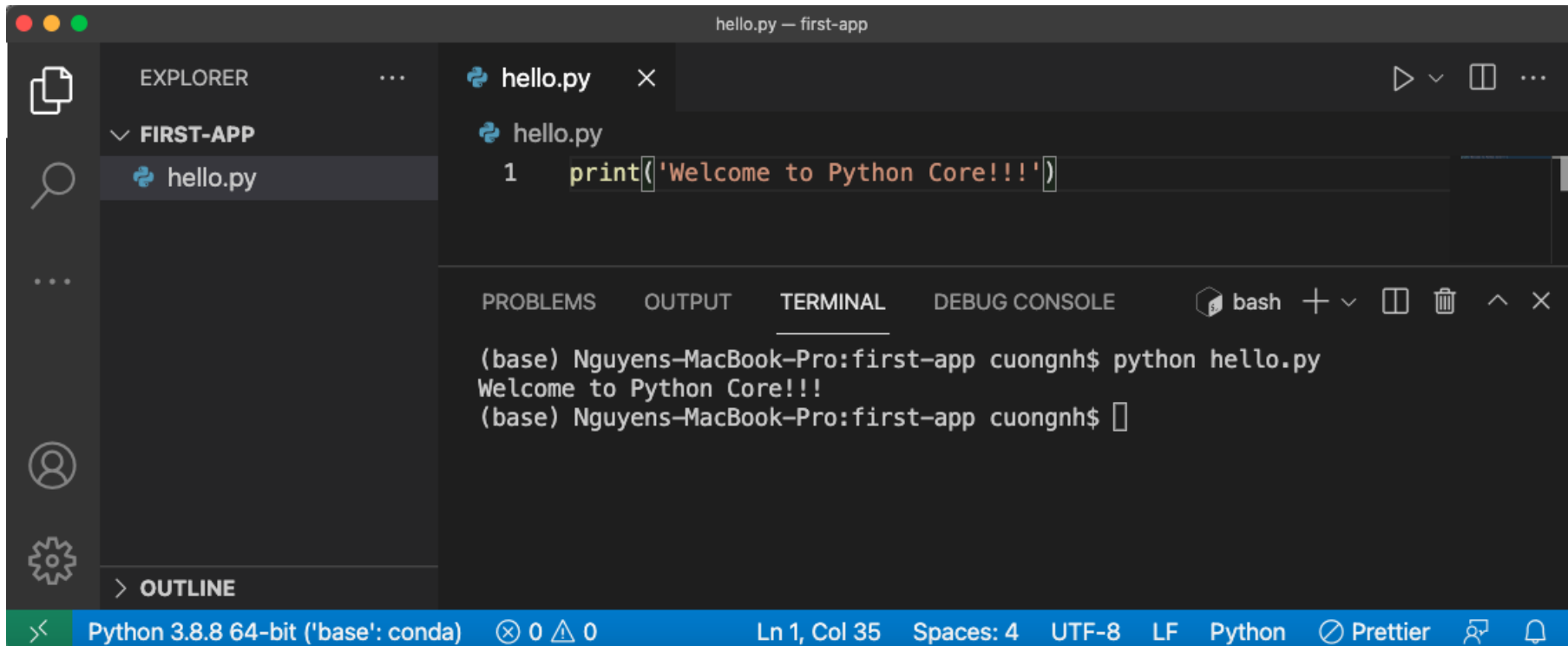
Git Commands



Run first application

Hello World

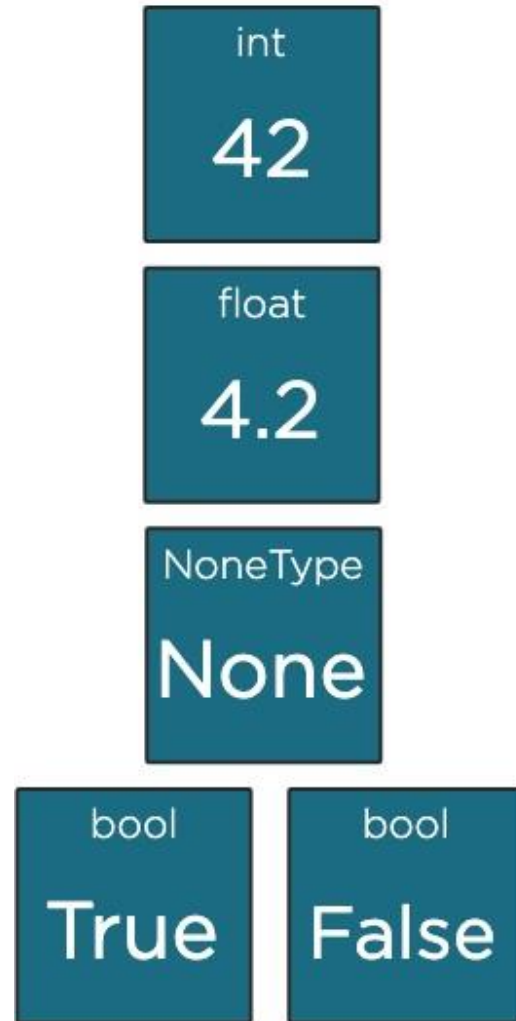
- Create a new file with the name “hello.py”, then open the folder containing the file “hello.py” with VSCode



The screenshot shows the Visual Studio Code (VS Code) interface. The Explorer panel on the left shows a folder named "FIRST-APP" containing a file named "hello.py". The main editor area displays the contents of "hello.py", which is a single line of Python code: `1 print('Welcome to Python Core!!!')`. Below the editor, the TERMINAL panel is open, showing the command `python hello.py` being executed in a bash shell. The output of the command is `Welcome to Python Core!!!`. The status bar at the bottom indicates the Python version is 3.8.8 64-bit, the file encoding is UTF-8, and the line and column numbers are Ln 1, Col 35.

Scalar Types

Scalar Types



arbitrary precision integer

64-bit floating point numbers

the null object

boolean logical values

Int

```
>>> 10
10
>>> 0b10
2
>>> 0o10
8
>>> 0x10
16
>>> int(3.5)
3
>>> int(-3.5)
-3
>>> int("496")
496
```

Float

```
>>> 3.125
3.125
>>> 3e8
300000000.0
>>> 1.616e-35
1.616e-35
>>> float(7)
7.0
>>> float("1.618")
1.618
>>> float("nan")
nan
>>> float("inf")
inf
>>> float("-inf")
-inf
>>> 3.0 + 1
4.0
>>>
```


None

```
>>> None
```

```
>>> a = None
```

```
>>> a is None
```

```
True
```

```
>>>
```

Bool

```
False
>>> bool(42)
True
>>> bool(-1)
True
>>> bool(0.0)
False
>>> bool(0.207)
True
>>> bool(-1.117)
True
>>> bool([])
False
>>> bool([1, 5, 9])
True
>>> bool("")
False
>>> bool("Spam")
True
>>> bool("False")
True
>>> bool("True")
True
>>>
```

Variable

- A variable is created as soon as it is assigned a value for the first time
- No need to specify the data type when declaring the variable, Python will automatically recognize the data type through the data assigned
- Syntax: *variableName = value*
- Examples:
x = 5
a = b = 3.3
name, sex = 'Michael', 1