



# **Python**

**here, there, and  
everywhere**

# Brief History of Python

- Invented in the Netherlands, early 90s by Guido van Rossum
- Named after Monty Python
- Open sourced from the beginning
- Considered a scripting language, but is much more
- Scalable, object oriented and functional from the beginning
- Used by Google from the beginning
- Increasingly popular

# Python's Benevolent Dictator For Life

“Python is an experiment in how much freedom programmers need. Too much freedom and nobody can read another's code; too little and expressiveness is endangered.”

- Guido van Rossum



# Why Python?

- The most popular language programming
- Using in many fields, especially in data and AI

## Projections of future traffic for major programming languages

Future traffic is predicted with an STL model, along with an 80% prediction interval.

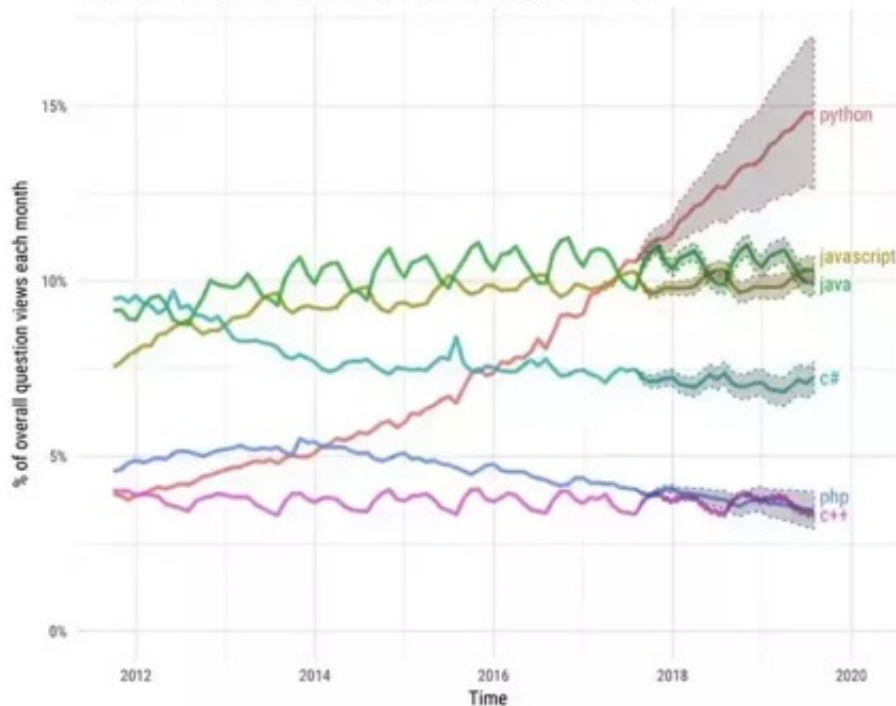
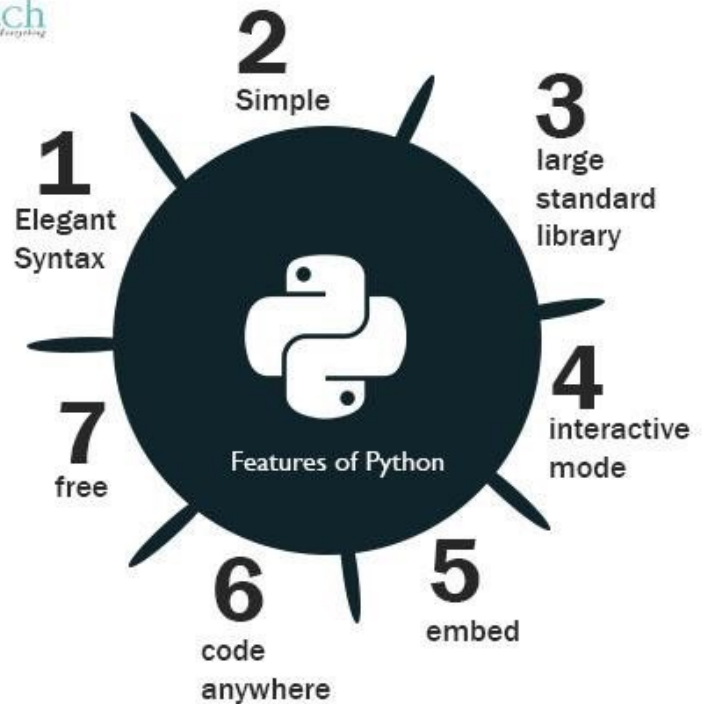


Image: Stack Overflow



# Why Python?

```
#include <iostream>
int main() {
    std::cout << "Hello, world! ";
    return 0;
}
```

C++

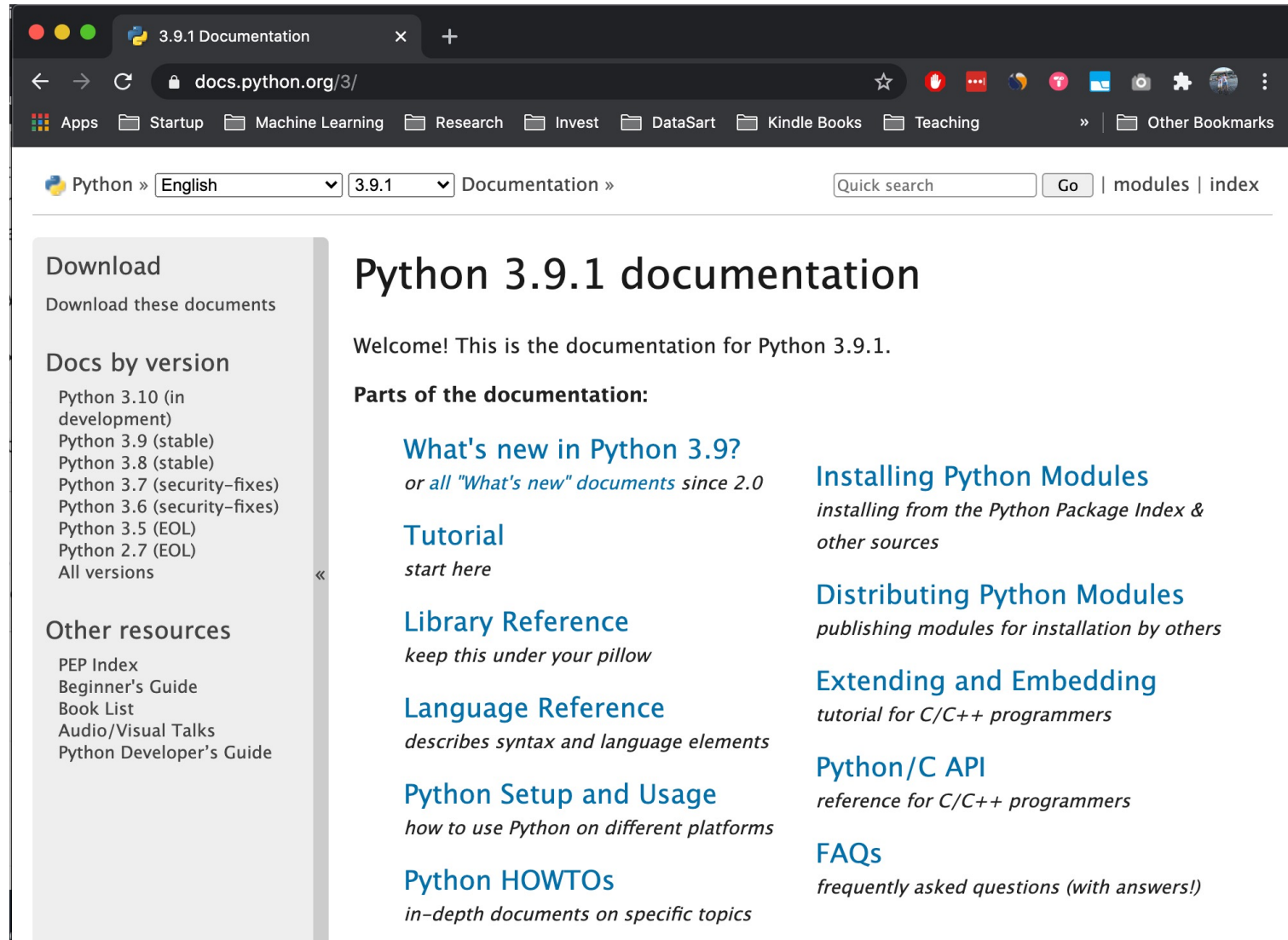
```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```

Java

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

Python

# http://docs.python.org/



The image is a screenshot of a web browser displaying the Python 3.9.1 documentation page. The browser's address bar shows 'docs.python.org/3/'. The page has a dark-themed header with navigation links like 'Apps', 'Startup', 'Machine Learning', 'Research', 'Invest', 'DataSart', 'Kindle Books', and 'Teaching'. Below the header, there's a navigation bar with 'Python »', a language dropdown set to 'English', a version dropdown set to '3.9.1', and a 'Documentation »' link. A search bar labeled 'Quick search' and a 'Go' button are also present, along with links to 'modules' and 'index'.

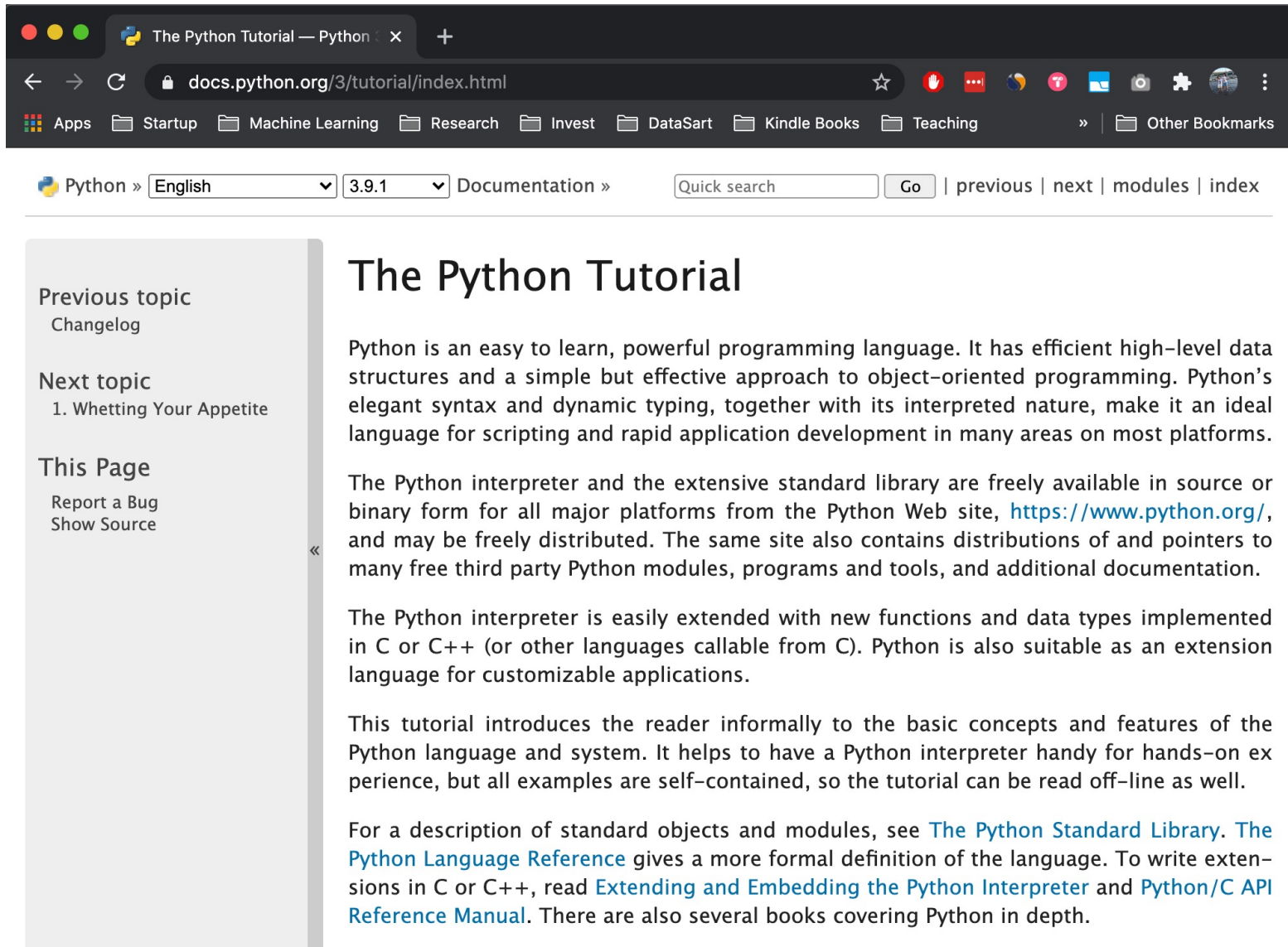
The main content area is titled 'Python 3.9.1 documentation' and includes a welcome message: 'Welcome! This is the documentation for Python 3.9.1.' Below this, a section titled 'Parts of the documentation:' lists various resources with blue links and descriptive subtitles:

- [What's new in Python 3.9?](#)  
*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*
- [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!)*

A left sidebar contains additional navigation options:

- Download**  
Download these documents
- Docs by version**
  - [Python 3.10 \(in development\)](#)
  - [Python 3.9 \(stable\)](#)
  - [Python 3.8 \(stable\)](#)
  - [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](#)

# The Python tutorial is good!



The screenshot shows a web browser window with the address bar displaying `docs.python.org/3/tutorial/index.html`. The browser's bookmark bar includes links to 'Apps', 'Startup', 'Machine Learning', 'Research', 'Invest', 'DataSart', 'Kindle Books', 'Teaching', and 'Other Bookmarks'. Below the browser window, the Python documentation header is visible, featuring a language dropdown set to 'English', a version dropdown set to '3.9.1', and a 'Documentation »' link. A search bar labeled 'Quick search' and a 'Go' button are also present, along with navigation links for 'previous', 'next', 'modules', and 'index'.

**Previous topic**  
Changelog

**Next topic**  
1. Whetting Your Appetite

**This Page**  
Report a Bug  
Show Source

## The Python Tutorial

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

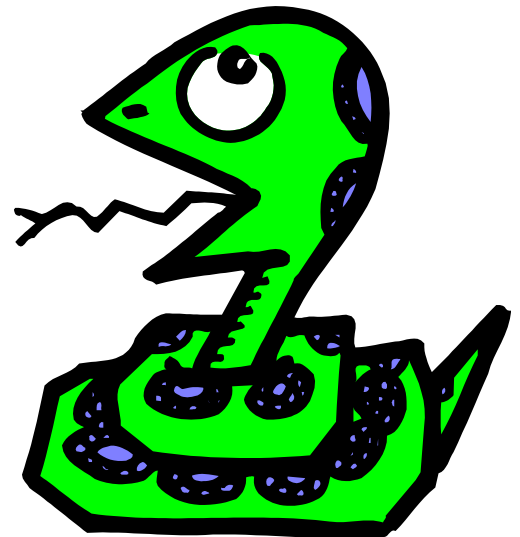
The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <https://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well.

For a description of standard objects and modules, see [The Python Standard Library](#). [The Python Language Reference](#) gives a more formal definition of the language. To write extensions in C or C++, read [Extending and Embedding the Python Interpreter](#) and [Python/C API Reference Manual](#). There are also several books covering Python in depth.

# Running Python





# Where should you run Python

- It's best to install it on your own computer
  - You'll have more control, can run Jupyter notebooks and learn more about it
- You can also run it on Jupyterhub unix system
- You can also use remote notebook servers
  - At PhenikaaCS, Google Colab, ...
- We'll give some details here

# Installing Python 3

- Using Anaconda (Miniconda): A distribution of the Python (good for AI & data science).
- [How to install](#)
- HW: Install Anaconda (Miniconda) on your computer.
- Running it on your own computer makes it easier to install packages, IDEs, and use notebooks
- And will give you more experience

# IDE or not?

- Python's an interpreted language so it comes with a [read-eval-print-loop](#) environment
- I'll admit to mostly using vim to edit code in one window and the Python REPL in another
- But you may prefer a [Python IDE](#)
  - Python comes with a simple one, [IDLE](#)
  - [PyCharm](#) is very popular and good
  - Jupyter notebooks is cool
- Here's a [guide](#) to Python editors and IDEs

# Running Interactively on UNIX

On Unix...

```
% python
```

```
>>> 3+3
```

```
6
```

- Python prompts with '>>>'.
- To exit Python:
  - In Unix, press CONTROL-D
  - In Windows, press CONTROL-Z + <Enter>
  - run `exit()` or `quit()`

# Running Python Programs on UNIX

- Call python program via the python interpreter

```
% python fact.py
```

- Make a python file directly executable by
  - Adding the appropriate path to your python interpreter as the first line of your file

```
#!/usr/bin/python
```

- Making the file executable

```
% chmod a+x fact.py
```

- Invoking file from Unix command line

```
% fact.py
```

# Example 'script': fact.py

```
#!/usr/bin/python
```

```
def fact(x):
```

```
    """Returns the factorial of its argument, assumed to be a posint"""
```

```
    if x == 0:
```

```
        return 1
```

```
    return x * fact(x - 1)
```

```
print ('N fact(N)')
```

```
print("-----")
```

```
for n in range(10):
```

```
    print(n, fact(n))
```

# Write your first Python Program



Library

Data  
structure

Expression  
& Control  
flow

Variable

```
import sys
import random

def say_hello(user):
    # some greeting in different languages
    prefix_dict = {
        1: "Hello ",
        2: "Xin Chao ",
        3: "ni hao "
    }
    key = random.randint(1,3)
    prefix = prefix_dict[key]
    print(prefix + user)

if __name__ == "__main__":
    user = sys.argv[1]
    say_hello(user)
```



# Notebooks

**On your own computer**

# Using a notebook server

# Using Google Colab

# Resources

- Official Website: <https://www.python.org/>
- Python tutorials for beginners:  
<https://thepythonguru.com/>
- Many guides for beginners:  
<https://wiki.python.org/moin/BeginnersGuide>
- Good book: Learn Python 3 the Hard Way