





<!DOCTYPE html>

<html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Semantic Elements</title>

</head>

<body>

<header>

<h1>Python</h1>

<h2>Table of Contents</h2>

<nav>

<ol>

<li><a href="#intro">Introduction</a></li>

<li><a href="#history">History</a></li>

<li><a href="#features">Features</a></li>

<li><a href="#logo">Logo</a></li>

<li><a href="#ref">References</a></li>

</ol>

</nav>

</header>

<main>

<article>

<header id="intro">

<h2>Introduction</h2>

</header>

<p>Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects</p>

<p>Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting.

Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.</p>

<aside>

<h4>API</h4>

<p>API stands for Application Programming Interface. It is a set of routines, protocols, and tools for building software and applications.

<mark>This content uses the aside element.</mark> This is only for semantic meaning and so doesn't carry any instructions for presentation. It will appear like a normal paragraph in your document. If you wish to include this in a sidebar, you will have to provide the appropriate markup and styling for it.</p>

</aside>

</article>

<section>

<header id="history">

<h2>History</h2>

</header>

<p>Python was conceived in the late 1980s[35] by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL),[36] capable of exception handling and interfacing with the Amoeba operating system.[8] Its implementation began in December 1989.[37] Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's Benevolent Dictator For Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker.[38] He now shares his leadership as a member of a five-person steering council.[39][40][41] In January 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and Van Rossum to a five-member "Steering Council" to lead the project.</p>

</section>

<!-- Our features section with smaller headings-->

<section>

<header id="features">

<h2>Features</h2>

<h4>Keeping up with times</h4>

</header>

<p>Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one—and preferably only one—obvious way to do it" design philosophy.[57] Alex Martelli, a Fellow at the Python Software Foundation and Python book author, writes that "To describe something as 'clever' is not considered a compliment in the Python culture."[58]

Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity.[59] When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name—a tribute to the British comedy group Monty Python[60]—and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.[61][62]</p>

<h4>New APIs</h4>

<p>APIs work much the same way, except instead of your web browser asking for a webpage, your program asks for data. This data is usually returned in JSON format (for more on this, checkout our tutorial on working with JSON data). In order to get the data, we make a request to a webserver</p>

<h4>Error Handling</h4>

<p>In Python, exceptions can be handled using a try statement. The critical operation which can raise an exception is placed inside the try clause. The code that handles the exceptions is written in the except clause. </p>

</section>

<section>

<header id="logo">

<h2>Logo</h2>

</header>

<figure>

<img src="https://seeklogo.com/images/P/python-logo-C50EED1930-seeklogo.com.png" alt="Logo">

<figcaption>Python Logo</figcaption>

</figure>

</section>

<section>

<header id="ref">

<h2>References</h2>

</header>

<p>Contents of this sample have been adapted from the <a href="https://en.wikipedia.org/wiki/Python\_(programming\_language)" target="\_blank">Python on Wikipedia</a>

</p>

</section>

</main>

<footer>

<p>Copyrights information...</p>

<details>

<summary>Authors</summary>

<p>Author1</p>

<p>Author2</p>

</details>

<details>

<summary>Contact</summary>

<nav>

<ul style="margin: 0; padding: 0; list-style-type: none;">

<li style="display: inline;"><a href="mailto:jaswanth.jr@gmail.com" style="text-decoration: none; padding: .2em 1em;">email</a></li>

</ul>

</nav>

</details>

</footer>

</body>

</html>