**Perceiving Python programming paradigms**

Python supports imperative, functional, procedural, and object-oriented programming; here are tips on choosing the right one for a specific use case.

* [**LOG IN**](https://opensource.com/user/login?current=/article/19/10/python-programming-paradigms)
* [RSS](https://opensource.com/feed)

## Main navigation

* [**ARTICLES**](https://opensource.com/)
* [**RESOURCES**](https://opensource.com/resources)
* [**DOWNLOADS**](https://opensource.com/downloads)
* [**FREQUENTLY ASKED QUESTIONS**](https://opensource.com/faq)

Top of Form

****

****

Bottom of Form

# Perceiving Python programming paradigms

Python supports imperative, functional, procedural, and object-oriented programming; here are tips on choosing the right one for a specific use case.

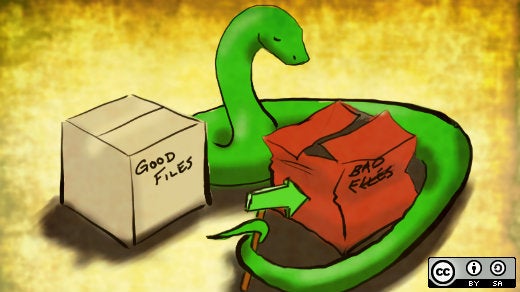
By [Jigyasa Grover](https://opensource.com/users/jigyasa-grover)

October 18, 2019 | [2 Comments](https://opensource.com/article/19/10/python-programming-paradigms#comments)| 11 min read

Top of Form

159 readers like this.

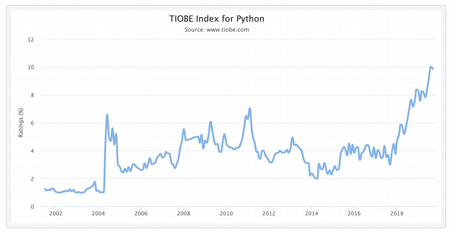
Bottom of Form



***Image by:***

*Jason van Gumster. CC BY-SA 4.0*

Early each year, TIOBE announces its Programming Language of The Year. When its latest annual [TIOBE index](https://www.tiobe.com/tiobe-index/programming-languages-definition/) report came out, I was not at all surprised to see [Python again winning the title](https://www.tiobe.com/tiobe-index/python/), which was based on capturing the most search engine ranking points (especially on Google, Bing, Yahoo, Wikipedia, Amazon, YouTube, and Baidu) in 2018.



Adding weight to TIOBE's findings, earlier this year, nearly 90,000 developers took Stack Overflow's annual [Developer Survey](https://insights.stackoverflow.com/survey/2019), which is the largest and most comprehensive survey of people who code around the world. The main takeaway from this year's results was:

"Python, the fastest-growing major programming language, has risen in the ranks of programming languages in our survey yet again, edging out Java this year and standing as the second most loved language (behind Rust)."

Ever since I started programming and exploring different languages, I have seen admiration for Python soaring high. Since 2003, it has consistently been among the top 10 most popular programming languages. As TIOBE's report stated:

"It is the most frequently taught first language at universities nowadays, it is number one in the statistical domain, number one in AI programming, number one in scripting and number one in writing system tests. Besides this, Python is also leading in web programming and scientific computing (just to name some other domains). In summary, Python is everywhere."

There are several reasons for Python's rapid rise, bloom, and dominance in multiple domains, including web development, scientific computing, testing, data science, machine learning, and more. The reasons include its readable and maintainable code; extensive support for third-party integrations and libraries; modular, dynamic, and portable structure; flexible programming; learning ease and support; user-friendly data structures; productivity and speed; and, most important, community support. The diverse application of Python is a result of its combined features, which give it an edge over other languages.

[**SKIP TO CONTENT**](https://opensource.com/article/19/10/python-programming-paradigms)

**More Python Resources**

* [**What is an IDE?**](https://www.redhat.com/en/topics/middleware/what-is-ide?intcmp=7016000000127cYAAQ)
* [**Cheat sheet: Python 3.7 for beginners**](https://opensource.com/downloads/cheat-sheet-python-37-beginners?intcmp=7016000000127cYAAQ)
* [**Top Python GUI frameworks**](https://opensource.com/resources/python/gui-frameworks?intcmp=7016000000127cYAAQ)
* [**Download: 7 essential PyPI libraries**](https://opensource.com/downloads/7-essential-pypi-libraries?intcmp=7016000000127cYAAQ)
* [**Red Hat Developers**](https://developers.redhat.com/?intcmp=7016000000127cYAAQ)
* [**Latest Python content**](https://opensource.com/tags/python?intcmp=7016000000127cYAAQ)

But in my opinion, the comparative simplicity of its syntax and the staggering flexibility it provides developers coming from many other languages win the cake. Very few languages can match Python's ability to conform to a developer's coding style rather than forcing him or her to code in a particular way. Python lets more advanced developers use the style they feel is best suited to solve a particular problem.

While working with Python, you are like a snake charmer. This allows you to take advantage of Python's promise to offer a non-conforming environment for developers to code in the style best suited for a particular situation and to make the code more readable, testable, and coherent.

## Python programming paradigms

Python supports four main [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm): imperative, functional, procedural, and object-oriented. Whether you agree that they are valid or even useful, Python strives to make all four available and working. Before we dive in to see which programming paradigm is most suitable for specific use cases, it is a good time to do a quick review of them.

### Imperative programming paradigm

The [imperative programming paradigm](https://en.wikipedia.org/wiki/Imperative_programming) uses the imperative mood of natural language to express directions. It executes commands in a step-by-step manner, just like a series of verbal commands. Following the "how-to-solve" approach, it makes direct changes to the state of the program; hence it is also called the stateful programming model. Using the imperative programming paradigm, you can quickly write very simple yet elegant code, and it is super-handy for tasks that involve data manipulation. Owing to its comparatively slower and sequential execution strategy, it cannot be used for complex or parallel computations.

# Perceiving Python programming paradigms

Python supports imperative, functional, procedural, and object-oriented programming; here are tips on choosing the right one for a specific use case.

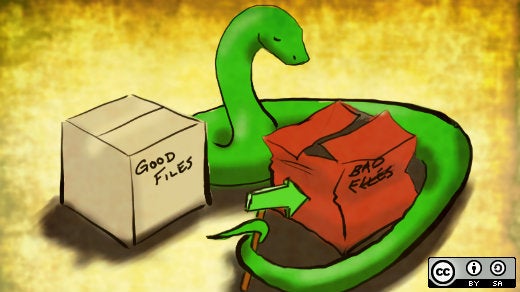
By [Jigyasa Grover](https://opensource.com/users/jigyasa-grover)

October 18, 2019 | [2 Comments](https://opensource.com/article/19/10/python-programming-paradigms#comments)| 11 min read

Top of Form

159 readers like this.

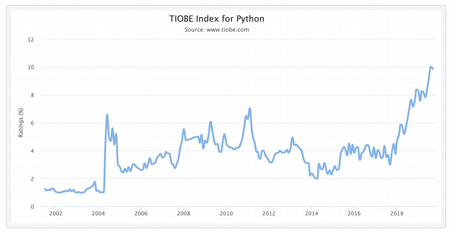
Bottom of Form



***Image by:***

*Jason van Gumster. CC BY-SA 4.0*

Early each year, TIOBE announces its Programming Language of The Year. When its latest annual [TIOBE index](https://www.tiobe.com/tiobe-index/programming-languages-definition/) report came out, I was not at all surprised to see [Python again winning the title](https://www.tiobe.com/tiobe-index/python/), which was based on capturing the most search engine ranking points (especially on Google, Bing, Yahoo, Wikipedia, Amazon, YouTube, and Baidu) in 2018.



Adding weight to TIOBE's findings, earlier this year, nearly 90,000 developers took Stack Overflow's annual [Developer Survey](https://insights.stackoverflow.com/survey/2019), which is the largest and most comprehensive survey of people who code around the world. The main takeaway from this year's results was:

"Python, the fastest-growing major programming language, has risen in the ranks of programming languages in our survey yet again, edging out Java this year and standing as the second most loved language (behind Rust)."

Ever since I started programming and exploring different languages, I have seen admiration for Python soaring high. Since 2003, it has consistently been among the top 10 most popular programming languages. As TIOBE's report stated:

"It is the most frequently taught first language at universities nowadays, it is number one in the statistical domain, number one in AI programming, number one in scripting and number one in writing system tests. Besides this, Python is also leading in web programming and scientific computing (just to name some other domains). In summary, Python is everywhere."

There are several reasons for Python's rapid rise, bloom, and dominance in multiple domains, including web development, scientific computing, testing, data science, machine learning, and more. The reasons include its readable and maintainable code; extensive support for third-party integrations and libraries; modular, dynamic, and portable structure; flexible programming; learning ease and support; user-friendly data structures; productivity and speed; and, most important, community support. The diverse application of Python is a result of its combined features, which give it an edge over other languages.

[**SKIP TO CONTENT**](https://opensource.com/article/19/10/python-programming-paradigms)

**More Python Resources**

* [**What is an IDE?**](https://www.redhat.com/en/topics/middleware/what-is-ide?intcmp=7016000000127cYAAQ)
* [**Cheat sheet: Python 3.7 for beginners**](https://opensource.com/downloads/cheat-sheet-python-37-beginners?intcmp=7016000000127cYAAQ)
* [**Top Python GUI frameworks**](https://opensource.com/resources/python/gui-frameworks?intcmp=7016000000127cYAAQ)
* [**Download: 7 essential PyPI libraries**](https://opensource.com/downloads/7-essential-pypi-libraries?intcmp=7016000000127cYAAQ)
* [**Red Hat Developers**](https://developers.redhat.com/?intcmp=7016000000127cYAAQ)
* [**Latest Python content**](https://opensource.com/tags/python?intcmp=7016000000127cYAAQ)

But in my opinion, the comparative simplicity of its syntax and the staggering flexibility it provides developers coming from many other languages win the cake. Very few languages can match Python's ability to conform to a developer's coding style rather than forcing him or her to code in a particular way. Python lets more advanced developers use the style they feel is best suited to solve a particular problem.

While working with Python, you are like a snake charmer. This allows you to take advantage of Python's promise to offer a non-conforming environment for developers to code in the style best suited for a particular situation and to make the code more readable, testable, and coherent.

## Python programming paradigms

Python supports four main [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm): imperative, functional, procedural, and object-oriented. Whether you agree that they are valid or even useful, Python strives to make all four available and working. Before we dive in to see which programming paradigm is most suitable for specific use cases, it is a good time to do a quick review of them.

### Imperative programming paradigm

The [imperative programming paradigm](https://en.wikipedia.org/wiki/Imperative_programming) uses the imperative mood of natural language to express directions. It executes commands in a step-by-step manner, just like a series of verbal commands. Following the "how-to-solve" approach, it makes direct changes to the state of the program; hence it is also called the stateful programming model. Using the imperative programming paradigm, you can quickly write very simple yet elegant code, and it is super-handy for tasks that involve data manipulation. Owing to its comparatively slower and sequential execution strategy, it cannot be used for complex or parallel computations.

