PYTHON FLASH CARDS

Welcome to *Python Flash Cards*! These cards will reinforce your learning as you begin to work with Python and think like a programmer.

The cards cover core programming and Python concepts, as well as basic Python syntax. This booklet will guide you on how to use the cards and help you install Python if you haven't already. Additional resources are available at *https://nostarch.com/pythonflashcards/*.

WHY USE PYTHON?

Python is one of the most popular programming languages in the world, and for good reason. Python has simple syntax that encourages you to write efficient, readable code, so you can concentrate on how to solve problems. Python is also concise, making it easier to write programs and to maintain them.

Python has a rich ecosystem of stable libraries intended for a wide variety of uses, such as data visualization, web development, application development, and more. For a given problem, there's a good chance someone has already written a Python library you can use. The Python community is also very welcoming to beginners, and you'll often find someone willing to help you out if you get stuck.

WHAT'S IN THE CARDS?

The cards are split into two categories:

- Concept and vocabulary cards
- Syntax cards

CONCEPT AND VOCABULARY CARDS

Programmers use a lot of terminology, and knowing these words will help you think and communicate more clearly. Most of these cards present concepts without code so you can focus on the big ideas without getting into the syntax.

A solid grounding in concepts will help you learn Python more efficiently, and these concepts generally apply to other languages as well. Even if you're already starting to learn some more advanced syntax, reviewing basic concepts can be immensely helpful.

SYNTAX CARDS

In order to write code that works, you need to learn the syntax of a language. These cards show syntax examples in different forms depending on how the information is best conveyed: as code from a Python terminal session, snippets from a full program, or tables.

HOW TO USE THE CARDS

You'll probably figure out your own best way to use the cards, but I can offer a couple of tips:

- Pick out the cards on the topics you're currently working on and carry them around with you. It's sometimes easier to review a small set of cards, especially when you aren't able to pull out a computer and practice coding there and then. When you do get back to coding, you'll find it easier to write code efficiently.
- Ask yourself the questions on the front of the card and see how well your answers match the information on the back. If you can consistently answer the questions correctly, you can put that card into your "learned" pile.

INSTALLING PYTHON

Some operating systems come preloaded with Python, but it's often an outdated version. If you haven't already done so, install the latest version of Python before you start using the cards. *Python Flash Cards* will work for any version of Python 3.

PYTHON ON WINDOWS

Download the installer for your system from *https://python.org/downloads/*. Run the installer, and make sure to check the **Add Python 3.*x* to your PATH** box. This will ensure your installation of Python is ready to use.

To start using Python, open a terminal (enter **`terminal`** into the search box on your taskbar) and then enter **`python`** at the terminal prompt. You should see a Python prompt appear that looks like this: `>>>`. Enter the text in bold from the following example, and you should see the output appear below it:

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

To get back to the regular Windows prompt and out of Python, press CTRL-Z and then ENTER.

PYTHON ON MACOS

You can use the package manager Homebrew to install the latest version of Python on macOS.

First, install Homebrew: visit the project's home page at *https://brew.sh/* and follow the directions you see there.

Homebrew might inform you of additional requirements, which you should also install.

With Homebrew installed, install the latest version of Python by opening the terminal and entering this command:

```
$ brew install python
```

The latest version of Python should install. To check exactly which version was installed, enter the following:

```
$ python3 --version
```

You should see output showing the version you installed.

To use Python, open a terminal and enter **python3**. You should see a Python prompt (>>>) appear. To test it, enter the following:

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

Press CONTROL-D to get back to a regular terminal prompt.

PYTHON ON LINUX

Python comes installed on most Linux systems, but you'll probably want to install a more recent version. On APT-based systems such as Ubuntu, you can use the `deadsnakes` package to do this by entering the following commands into the terminal:

```
$ sudo add-apt-repository ppa:deadsnakes/ppa
$ sudo apt-get update
$ sudo apt-get install python3.7
```

To start a Python terminal session and test it, enter the following:

```
$ python3.7
>>> print("Hello, Python world!")
Hello, Python world!
```

Press CTRL-D to get back to a regular terminal prompt.

PYTHON EXAMPLES

When the examples on the cards are based in the terminal, they will appear like this, where the code you enter comes after the >>> prompt:

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

The output will appear just below the code, without a prompt. You probably won't see colored syntax highlighting when you enter code in a terminal, but it's used on the cards to clarify syntax. For longer programs, you'll want to install a text editor, such as Sublime Text or Atom, or a Python IDE (integrated development environment), such as PyCharm. When examples are from longer programs, they'll appear like this:

```
message = "Hello, Python world!"
print(message)
```

The output will usually appear in a separate section below the code, like this:

```
Hello, Python world!
```

NEXT STEPS

Here are some resources to help you learn more advanced Python concepts and start using Python for real-world projects.

- **Python documentation** The official Python documentation (*https://docs.python.org/*) is an excellent resource. The Tutorial and Library Reference links are great places to start your exploration. Search for topics using the search box at the top of the home page.
- **Books** A wide variety of books are available on Python in general and for specific kinds of work. No Starch Press has a good collection (*https://nostarch.com/catalog/python/*).
- **Online resources** Python bloggers are another great resource. Official documentation for major libraries such as Django and matplotlib can be very useful, too.
- **Python community groups and events** The Python community is diverse and strong, and you'll find Python gatherings all over the world. Search for a local Python user group to visit or attend one of the larger PyCon events. Events almost always offer something for both new and experienced Python users.
- **Your own projects** Once you're familiar with the basics, start building your own projects. This lets you practice what you've learned in real-world contexts, pushes you to learn new techniques and concepts, and stretches the limits of what you're capable of. Reach out to others in the community when you get stuck—and have fun!

ABOUT THE PYTHON FLASH CARDS

You can use the cards in this box to study Python concepts and syntax. You'll find nine sets of cards:

1 CONCEPTS AND VOCABULARY

2 SIMPLE DATA TYPES

3 LISTS AND TUPLES

4 DICTIONARIES

5 CONDITIONAL STATEMENTS

6 FUNCTIONS

7 CLASSES

8 TESTING

9 PACKAGES

Publisher: William Pollock
Production Editor: Riley Hoffman
Developmental Editor: Liz Chadwick
Designer: Mimi Heft
Technical Reviewer: Kenneth Love
Copyeditors: Anne Marie Walker and Barton D. Reed
Proofreader: Paula L. Fleming

no starch press

For information on distribution, translations, or bulk sales, please contact No Starch Press, Inc. directly:

No Starch Press, Inc.
245 8th Street, San Francisco, CA 94103
phone: 1.415.863.9900; info@nostarch.com; www.nostarch.com