



# Red Hat Enterprise Linux 9.0 Beta

## Installing and using dynamic programming languages

A guide to installing and using dynamic programming languages in Red Hat Enterprise Linux 9



# Red Hat Enterprise Linux 9.0 Beta Installing and using dynamic programming languages

---

A guide to installing and using dynamic programming languages in Red Hat Enterprise Linux 9

## Legal Notice

Copyright © 2021 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This document describes the basics of installing and using dynamic programming languages, such as Python and PHP on Red Hat Enterprise Linux 9.

---

## Table of Contents

<b>RHEL BETA RELEASE</b> .....	<b>3</b>
<b>MAKING OPEN SOURCE MORE INCLUSIVE</b> .....	<b>4</b>
<b>PROVIDING FEEDBACK ON RED HAT DOCUMENTATION</b> .....	<b>5</b>
<b>CHAPTER 1. INTRODUCTION TO PYTHON</b> .....	<b>6</b>
1.1. PYTHON VERSIONS	6
1.2. MAJOR DIFFERENCES IN THE PYTHON ECOSYSTEM SINCE RHEL 8	6
<b>CHAPTER 2. INSTALLING AND USING PYTHON</b> .....	<b>7</b>
2.1. INSTALLING PYTHON 3	7
2.2. INSTALLING ADDITIONAL PYTHON 3 PACKAGES	7
2.3. INSTALLING ADDITIONAL PYTHON 3 TOOLS FOR DEVELOPERS	7
2.4. USING PYTHON	8
<b>CHAPTER 3. USING THE PHP SCRIPTING LANGUAGE</b> .....	<b>9</b>
3.1. INSTALLING THE PHP SCRIPTING LANGUAGE	9
3.2. USING THE PHP SCRIPTING LANGUAGE WITH A WEB SERVER	9
3.2.1. Using PHP with the Apache HTTP Server	9
3.2.2. Using PHP with the nginx web server	11
3.3. RUNNING A PHP SCRIPT USING THE COMMAND-LINE INTERFACE	12
3.4. ADDITIONAL RESOURCES	13



## RHEL BETA RELEASE

Red Hat provides Red Hat Enterprise Linux Beta access to all subscribed Red Hat accounts. The purpose of Beta access is to:

- Provide an opportunity to customers to test major features and capabilities prior to the general availability release and provide feedback or report issues.
- Provide Beta product documentation as a preview. Beta product documentation is under development and is subject to substantial change.

Note that Red Hat does not support the usage of RHEL Beta releases in production use cases. For more information, see [What does Beta mean in Red Hat Enterprise Linux and can I upgrade a RHEL Beta installation to a General Availability \(GA\) release?](#).

## MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).



## PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better. To do so:

- For simple comments on specific passages:
  1. Make sure you are viewing the documentation in the *Multi-page HTML* format. In addition, ensure you see the **Feedback** button in the upper right corner of the document.
  2. Use your mouse cursor to highlight the part of text that you want to comment on.
  3. Click the **Add Feedback** pop-up that appears below the highlighted text.
  4. Follow the displayed instructions.
- For submitting more complex feedback, create a Bugzilla ticket:
  1. Go to the [Bugzilla](#) website.
  2. As the Component, use **Documentation**.
  3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
  4. Click **Submit Bug**.

# CHAPTER 1. INTRODUCTION TO PYTHON

Python is a high-level programming language that supports multiple programming paradigms, such as object-oriented, imperative, functional, and procedural paradigms. Python has dynamic semantics and can be used for general-purpose programming.

With Red Hat Enterprise Linux, many packages that are installed on the system, such as packages providing system tools, tools for data analysis, or web applications, are written in Python. To use these packages, you must have the **python\*** packages installed.

## 1.1. PYTHON VERSIONS

**Python 3.9** is the default **Python** implementation in RHEL 9. **Python 3.9** is distributed in a non-modular **python3** RPM package in the BaseOS repository and usually installed by default. **Python 3.9** will be supported for the whole life cycle of RHEL 9.

In the future, additional versions of **Python 3** will be distributed as RPM packages with a shorter life cycle through the AppStream repository. These versions will be installable in parallel with Python 3.9.

**Python 2** is not distributed with RHEL 9.

## 1.2. MAJOR DIFFERENCES IN THE PYTHON ECOSYSTEM SINCE RHEL 8

This section summarizes major changes in the Python ecosystem in RHEL 9 compared to RHEL 8.

### The unversioned **python** command

The unversioned form of the **python** command (**/usr/bin/python**) is available in the **python-unversioned-command** package. On some systems, this package is not installed by default. To install the unversioned form of the **python** command manually, use the **yum install /usr/bin/python** command.

In RHEL 9, the unversioned form of the **python** command points to the default **Python 3.9** version and it is an equivalent to the **python3** and **python3.9** commands.

The **python** command is intended for interactive sessions. In production, Red Hat recommends using **python3** or **python3.9** explicitly.

You can uninstall the unversioned **python** command by using the **yum remove /usr/bin/python** command.

If you need a different python command, you can create custom symlinks in **/usr/local/bin** or **~/local/bin** or a Python virtual environment.

Several other unversioned commands are available, such as **/usr/bin/pip** in the **python3-pip** package. In RHEL 9, all unversioned commands point to the default **Python 3.9** version.

### Architecture-specific Python wheels

Architecture-specific Python **wheels** built on RHEL 9 newly adhere to the upstream architecture naming, which allows customers to build their Python **wheels** on RHEL 9 and install them on non-RHEL systems. Python **wheels** built on previous releases of RHEL are forward compatible and can be installed on RHEL 9. Note that this affects only **wheels** containing Python extensions, which are built for each architecture, not Python **wheels** with pure Python code, which is not architecture-specific.

## CHAPTER 2. INSTALLING AND USING PYTHON

In RHEL 9, **Python 3.9** is the default **Python** implementation. The unversioned **python** command points to the default **Python 3.9** version.

### 2.1. INSTALLING PYTHON 3

The default Python implementation is usually installed by default. To install it manually, use the following procedure.

#### Procedure

- To install Python, use:

```
# yum install python3
```

#### Verification steps

- To verify the Python version installed on your system, use the following command:

```
$ python3 --version
```

### 2.2. INSTALLING ADDITIONAL PYTHON 3 PACKAGES

Packages prefixed with **python3** contain modules for the default **Python 3.9** version.

#### Procedure

- To install the **Requests** module for Python, use:

```
# yum install python3-requests
```

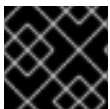
- To install the **pip** package installer from Python, use:

```
# yum install python3-pip
```

### 2.3. INSTALLING ADDITIONAL PYTHON 3 TOOLS FOR DEVELOPERS

Additional Python tools for developers are distributed through the CodeReady Linux Builder repository.

This repository contains, for example, the **python3-pytest**, **python3-Cython** packages and many others.



#### IMPORTANT

The CodeReady Linux Builder repository and its content is unsupported by Red Hat.

To install packages from the repository, use the following the procedure.

#### Procedure

1. Enable the CodeReady Linux Builder repository:

```
# subscription-manager repos --enable codeready-builder-for-rhel-9-x86_64-rpms
```

2. Install the **python3-pytest** package:

```
# yum install python3-pytest
```

### Additional resources

- [How to enable and make use of content within CodeReady Linux Builder](#)

## 2.4. USING PYTHON

The following procedure contains examples of running the Python interpreter or Python-related commands.

### Prerequisites

- Ensure that Python is installed.

### Procedure

- To run the Python interpreter or related commands, use, for example:

```
$ python3
$ python3 -m pip --help
$ python3 -m pip install package
```

## CHAPTER 3. USING THE PHP SCRIPTING LANGUAGE

Hypertext Preprocessor (PHP) is a general-purpose scripting language mainly used for server-side scripting, which enables you to run the PHP code using a web server.

### 3.1. INSTALLING THE PHP SCRIPTING LANGUAGE

This section describes how to install PHP.

#### Procedure

- To install PHP, use:

```
# yum install php
```

### 3.2. USING THE PHP SCRIPTING LANGUAGE WITH A WEB SERVER

#### 3.2.1. Using PHP with the Apache HTTP Server

In Red Hat Enterprise Linux 9, the **Apache HTTP Server** enables you to run PHP as a FastCGI process server. FastCGI Process Manager (FPM) is an alternative PHP FastCGI daemon that allows a website to manage high loads. PHP uses FastCGI Process Manager by default in RHEL 9.

This section describes how to run the PHP code using the FastCGI process server.

#### Prerequisites

- The PHP scripting language is installed on your system.

#### Procedure

1. Install the **httpd** package:

```
# yum install httpd
```

2. Start the **Apache HTTP Server**:

```
# systemctl start httpd
```

Or, if the **Apache HTTP Server** is already running on your system, restart the **httpd** service after installing PHP:

```
# systemctl restart httpd
```

3. Start the **php-fpm** service:

```
# systemctl start php-fpm
```

4. Optional: Enable both services to start at boot time:

```
# systemctl enable php-fpm httpd
```

- To obtain information about your PHP settings, create the **index.php** file with the following content in the **/var/www/html/** directory:

```
echo '<?php phpinfo(); ?>' > /var/www/html/index.php
```

- To run the **index.php** file, point the browser to:

```
http://<hostname>/
```

- Optional: Adjust configuration if you have specific requirements:

- /etc/httpd/conf/httpd.conf** - generic **httpd** configuration
- /etc/httpd/conf.d/php.conf** - PHP-specific configuration for **httpd**
- /usr/lib/systemd/system/httpd.service.d/php-fpm.conf** - by default, the **php-fpm** service is started with **httpd**
- /etc/php-fpm.conf** - FPM main configuration
- /etc/php-fpm.d/www.conf** - default **www** pool configuration

### Example 3.1. Running a "Hello, World!" PHP script using the Apache HTTP Server

- Create a **hello** directory for your project in the **/var/www/html/** directory:

```
# mkdir hello
```

- Create a **hello.php** file in the **/var/www/html/hello/** directory with the following content:

```
# <!DOCTYPE html>
<html>
<head>
<title>Hello, World! Page</title>
</head>
<body>
<?php
    echo 'Hello, World!';
?>
</body>
</html>
```

- Start the **Apache HTTP Server**:

```
# systemctl start httpd
```

- To run the **hello.php** file, point the browser to:

```
http://<hostname>/hello/hello.php
```

As a result, a web page with the "Hello, World!" text is displayed.

- [Setting up the Apache HTTP web server](#)

### 3.2.2. Using PHP with the nginx web server

This section describes how to run PHP code through the **nginx** web server.

#### Prerequisites

- The PHP scripting language is installed on your system.

#### Procedure

1. Install the **nginx** package:

```
# yum install nginx
```

2. Start the **nginx** server:

```
# systemctl start nginx
```

Or, if the **nginx** server is already running on your system, restart the **nginx** service after installing PHP:

```
# systemctl restart nginx
```

3. Start the **php-fpm** service:

```
# systemctl start php-fpm
```

4. Optional: Enable both services to start at boot time:

```
# systemctl enable php-fpm nginx
```

5. To obtain information about your PHP settings, create the **index.php** file with the following content in the **/usr/share/nginx/html/** directory:

```
echo '<?php phpinfo(); ?>' > /usr/share/nginx/html/index.php
```

6. To run the **index.php** file, point the browser to:

```
http://<hostname>/
```

7. Optional: Adjust configuration if you have specific requirements:

- **/etc/nginx/nginx.conf** - **nginx** main configuration
- **/etc/nginx/conf.d/php-fpm.conf** - FPM configuration for **nginx**
- **/etc/php-fpm.conf** - FPM main configuration
- **/etc/php-fpm.d/www.conf** - default **www** pool configuration

### Example 3.2. Running a "Hello, World!" PHP script using the nginx server

1. Create a **hello** directory for your project in the `/usr/share/nginx/html/` directory:

```
# mkdir hello
```

2. Create a **hello.php** file in the `/usr/share/nginx/html/hello/` directory with the following content:

```
# <!DOCTYPE html>
<html>
<head>
<title>Hello, World! Page</title>
</head>
<body>
<?php
    echo 'Hello, World!';
?>
</body>
</html>
```

3. Start the **nginx** server:

```
# systemctl start nginx
```

4. To run the **hello.php** file, point the browser to:

```
http://<hostname>/hello/hello.php
```

As a result, a web page with the "Hello, World!" text is displayed.

#### Additional resources

- [Setting up and configuring NGINX](#)

## 3.3. RUNNING A PHP SCRIPT USING THE COMMAND-LINE INTERFACE

A PHP script is usually run using a web server, but also can be run using the command-line interface.

#### Prerequisites

- The PHP scripting language is installed on your system.

#### Procedure

1. In a text editor, create a **filename.php** file  
Replace *filename* with the name of your file.
2. Execute the created **filename.php** file from the command line:

```
# php filename.php
```



**Example 3.3. Running a "Hello, World!" PHP script using the command-line interface**

1. Create a **hello.php** file with the following content using a text editor:

```
<?php
    echo 'Hello, World!';
?>
```

2. Execute the **hello.php** file from the command line:

```
# php hello.php
```

As a result, "Hello, World!" is printed.

### 3.4. ADDITIONAL RESOURCES

- **httpd(8)** – The manual page for the **httpd** service containing the complete list of its command-line options.
- **httpd.conf(5)** – The manual page for **httpd** configuration, describing the structure and location of the **httpd** configuration files.
- **nginx(8)** – The manual page for the **nginx** web server containing the complete list of its command-line options and list of signals.
- **php-fpm(8)** – The manual page for PHP FPM describing the complete list of its command-line options and configuration files.