



**FREE eBook**

# LEARNING cherry.py

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#cherry.py**

# Table of Contents

About.....	1
<b>Chapter 1: Getting started with cherrypy.....</b>	<b>2</b>
Remarks.....	2
Examples.....	2
Installation instructions.....	2
<b>Preconditions.....</b>	<b>2</b>
<b>Install.....</b>	<b>2</b>
Hello world in CherryPy.....	3
File upload with CherryPy.....	3
<b>server.py.....</b>	<b>3</b>
<b>webpage.html.....</b>	<b>4</b>
<b>cli.py.....</b>	<b>4</b>
<b>upload.txt.....</b>	<b>5</b>
<b>Upload from browser.....</b>	<b>5</b>
<b>Upload from command line.....</b>	<b>5</b>
<b>Credits.....</b>	<b>6</b>

---

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [cherrypy](#)

It is an unofficial and free cherrypy ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official cherrypy.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with cherrypy

## Remarks

This section provides an overview of what cherrypy is, and why a developer might want to use it.

It should also mention any large subjects within cherrypy, and link out to the related topics. Since the Documentation for cherrypy is new, you may need to create initial versions of those related topics.

## Examples

### Installation instructions

---

## Preconditions

- These instructions suppose you have any type of Linux, Unix, Mac with bash or Git-bash Windows.
- Windows: Download and install [Git-bash for Windows](#), then execute 'bash' from command line.
- Other shells than bash are fine too, just replace the `activate` command below with `activate.csh` or Google: "[virtualenv activate your-shell-name](#)".

Before you start, check that Python, virtualenv and pip are installed:

- `$ python --version`
- `$ virtualenv --version`
- `$ pip --version`

---

## Install

Create a directory with your web/app, create environment and install CherryPy package.

- `$ mkdir /develop/myapp/`
- `$ cd /develop/myapp/`
- `$ virtualenv venv`
- `$ source venv/bin/activate`
  - **On Windows in Git-bash:** `$ source venv/Scripts/activate`
- `(venv) $ pip install cherrypy`
- `(venv) $ python`

```
Python 3.5.2 ...
>>> import cherrypy
>>> cherrypy
<module 'cherrypy' from '... venv/site-packages/cherrypy/__init__.py'>
```

Congratulations! Now you are ready for your first CherryPy application.

## Hello world in CherryPy

If you have a virtualenv and CherryPy is already installed in it, create a file `hello.py`:

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-

import cherrypy

class HelloWorld(object):
    @cherrypy.expose
    def index(self):
        return 'Hello World!'

    @cherrypy.expose
    def greet(self, name):
        return 'Hello {}'.format(name)

cherrypy.quickstart(HelloWorld())
```

Then execute the file: `$ hello.py` or `$ python hello.py`. You should see output similar to this:

```
user@computer /develop/myapp $ python hello.py
[06/Nov/2016:05:58:44] ENGINE Listening for SIGTERM.
[06/Nov/2016:05:58:44] ENGINE Bus STARTING
[06/Nov/2016:05:58:44] ENGINE Set handler for console events.
CherryPy Checker:
The Application mounted at '' has an empty config.

[06/Nov/2016:05:58:44] ENGINE Started monitor thread '_TimeoutMonitor'.
[06/Nov/2016:05:58:44] ENGINE Started monitor thread 'Autoreloader'.
[06/Nov/2016:05:58:45] ENGINE Serving on http://127.0.0.1:8080
[06/Nov/2016:05:58:45] ENGINE Bus STARTED
```

- To see 'Hello World!' point your browser to <http://localhost:8080/>
- To see a greeting, go to <http://localhost:8080/greet?name=John>

## File upload with CherryPy

This example consists of three parts:

- `server.py` - CherryPy application that can receive and save a file.
- `webpage.html` - Example how to upload a file to `server.py` from a webpage.
- `cli.py` - Example how to upload a file to `server.py` from a command line tool.
- **Bonus** - `upload.txt` - file that you will upload.

---

## server.py

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

```

# -*- coding: UTF-8 -*-

import os
import cherrypy

config = {
    'global' : {
        'server.socket_host' : '127.0.0.1',
        'server.socket_port' : 8080
    }
}

class App:

    @cherrypy.expose
    def upload(self, ufile):
        # Either save the file to the directory where server.py is
        # or save the file to a given path:
        # upload_path = '/path/to/project/data/'
        upload_path = os.path.dirname(__file__)

        # Save the file to a predefined filename
        # or use the filename sent by the client:
        # upload_filename = ufile.filename
        upload_filename = 'saved.txt'

        upload_file = os.path.normpath(
            os.path.join(upload_path, upload_filename))
        size = 0
        with open(upload_file, 'wb') as out:
            while True:
                data = ufile.file.read(8192)
                if not data:
                    break
                out.write(data)
                size += len(data)
        out = ''
        File received.
        Filename: {}
        Length: {}
        Mime-type: {}
        ''' .format(ufile.filename, size, ufile.content_type, data)
        return out

if __name__ == '__main__':
    cherrypy.quickstart(App(), '/', config)

```

---

## webpage.html

```

<form method="post" action="http://127.0.0.1:8080/upload" enctype="multipart/form-data">
    <input type="file" name="ufile" />
    <input type="submit" />
</form>

```

# cli.py

This example requires [Python requests](#) package, however file can be sent to server in plain Python.

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-

import requests

url = 'http://127.0.0.1:8080/upload'
files = {'ufile': open('file.txt', 'rb')}

r = requests.post(url, files=files)

print(r)
print(r.text)
```

---

## upload.txt

```
Hello! This file was uploaded to CherryPy.
```

---

## Upload from browser

- Run `$ server.py`
- Open `webpage.html` in your web browser.
- After you select file from your drive and submit it, it will be saved as `saved.txt`.

---

## Upload from command line

- Open one console and run `$ server.py`
- Open another console and run `$ cli.py`
  - Note: Test file `upload.txt` should be in the same directory with `cli.py`
- File `upload.txt` should be uploaded and saved as `saved.txt`.

Read [Getting started with cherrypy online](https://riptutorial.com/cherrypy/topic/7696/getting-started-with-cherrypy): <https://riptutorial.com/cherrypy/topic/7696/getting-started-with-cherrypy>

---

# Credits

S. No	Chapters	Contributors
1	Getting started with cherrypy	<a href="#">Community</a> , <a href="#">dwich</a>