



# **CherryPy: Dynamic websites with Python**

**<http://www.cherrypy.org>**

Rémi Delon

[remi@cherrypy.org](mailto:remi@cherrypy.org)



## What is CherryPy ?

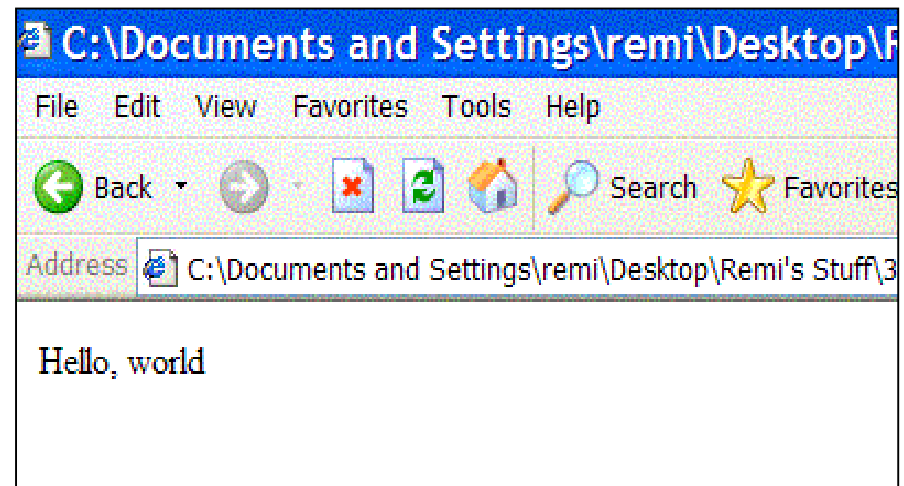
- Pure Python (no C) tool for developing dynamic websites
- Works with python 2.1, 2.2, 2.3, Jython
- Runs everywhere python runs (Linux, Windows, ...)
- Works like a compiler:
  - Source files (.cpy) -> Single executable (.py)
  - Source files use :  
python + a few extra keywords + templating language



## Sample source file

Root.cpy

```
CherryClass Root:
view:
    def index(self):
        return """
            <html><body>
            Hello, world
            </html></body>
        """
```



```
[remi@boa] cherrypy.py Root.cpy
[remi@boa] python RootServer.py
[....]
Serving HTTP on socket port: 8000
```

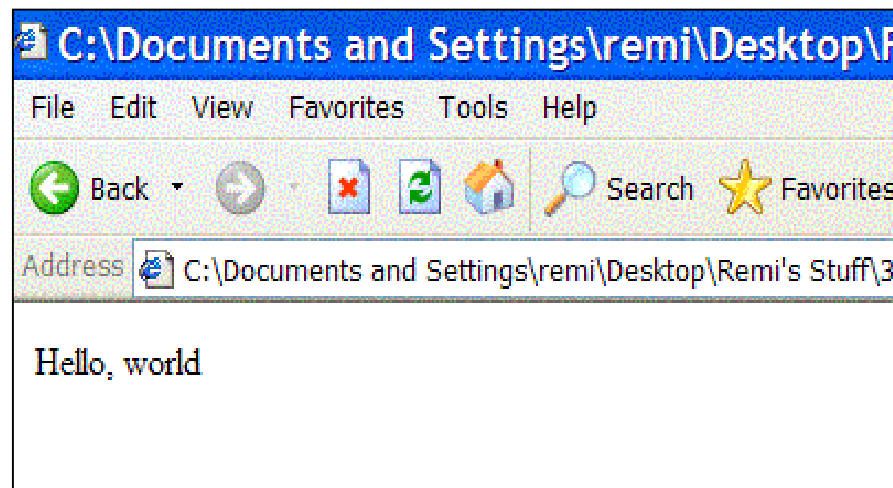
Browser -> <http://localhost:8000>



CherryClass Root:

mask:

```
def index(self):  
    <html><body>  
        Hello, world  
    </html></body>
```

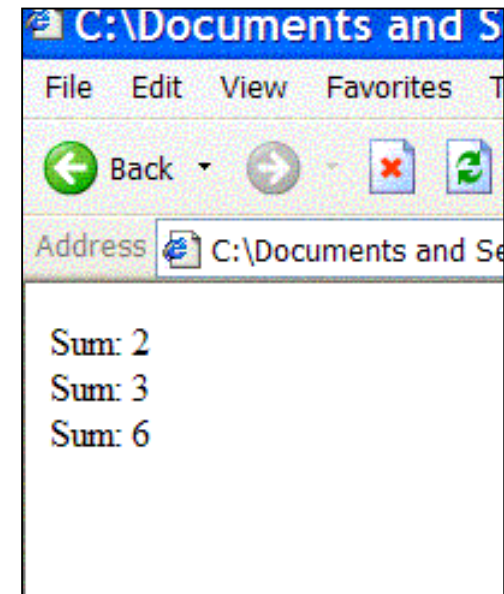




CherryClass Root:

mask:

```
def index(self):  
    <html><body>  
    <py-exec="a=2">  
    <py-for="i,j in [(0,0),(1,0),(2,2)]">  
        Sum: <py-eval="i+j+a"><br>  
    </py-for>  
    <py-code="  
        import smtplib  
        if 0==0:  
            b=2  
    ">  
    </html></body>
```

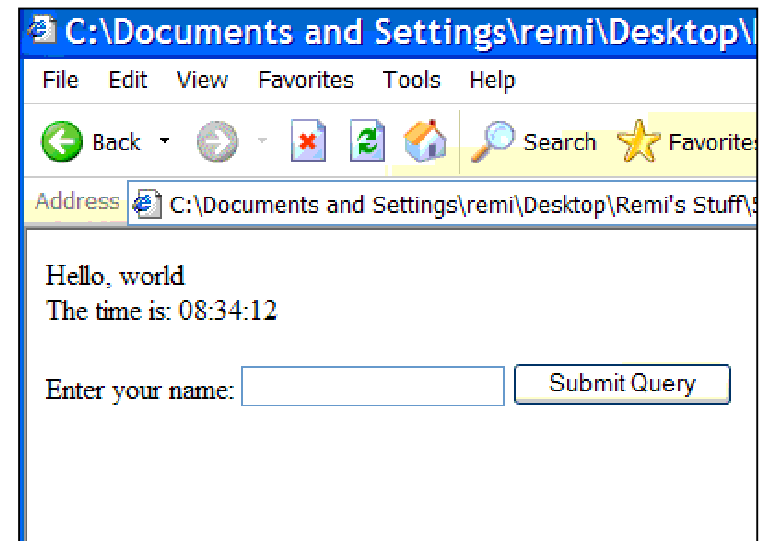




## Sample source file for a dynamic site (also using OOP)

```
import time
CherryClass Design:
mask:
    def header(self):
        <html><body>
    def footer(self):
        </body></html>

CherryClass Root(Design):
function:
    def getTime(self):
        return time.strftime('%H:%M:%S')
mask:
    def index(self, myName="world"):
        <py-eval="self.header()">
        Hello, <py-eval="myName"><br>
        The time is: <py-eval="self.getTime()">
        <form action="index">
        Enter your name: <input type="text" name="myName">
        <input type="submit">
        </form>
        <py-eval="self.footer()">
```





## Key properties of CherryPy

- Everything we love about Python is still there:
  - Clean syntax
  - Powerful constructs. ex: for i,j in [(0,0),(1,2)]
  - OOP
  - Standard library
  - ...
- No unneeded extra code in the source files
- Very fast (thanks to compiler approach)
- Easy to deploy (just one file)

### **+ all the features you would expect from a serious application server:**

- Database connectivity
- Can be run behind another webserver (Apache, ...)
- Load-balancing
- Cacheing, Sessions, Http and cookie-based authentication, Form handling
- XML-RPC
- HTTP server supports SSL, threading, forking, process pooling, ...
- Good documentation

### **+ can be used as an XML/XSL framework if you want**



# Questions ?

<http://www.cherrypy.org>

[remi@cherrypy.org](mailto:remi@cherrypy.org)