

Server-Side Web Programming: Python (Part 1)

Copyright © 2025 by
Robert M. Dondero, Ph.D
Princeton University

Objectives

- We will cover:
 - Fast CGI programming
 - Python WSGI programming

Agenda

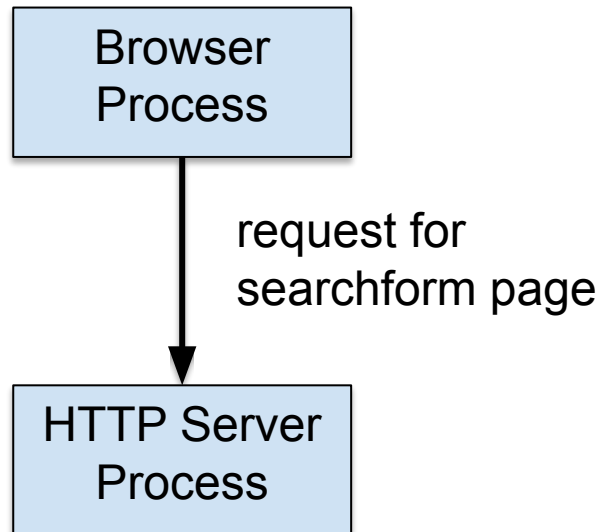
- **Fast CGI programming**
- Python WSGI programming

Fast CGI Programming

- Consider Option 1...
 - Web app **forks a child process** to handle each HTTP request

Fast CGI Programming

Option 1



Fast CGI Programming

Browser
Process

Option 1
(cont.)

HTTP Server
Process

request for
searchform page

searchform
Process

Fast CGI Programming

Browser
Process

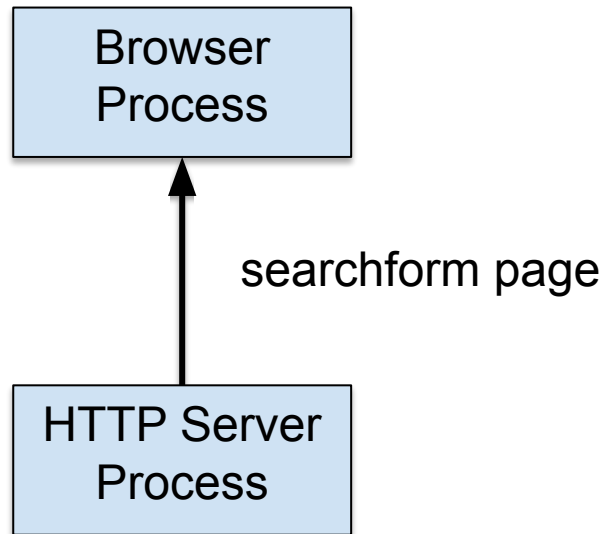
Option 1
(cont.)

HTTP Server
Process

searchform page

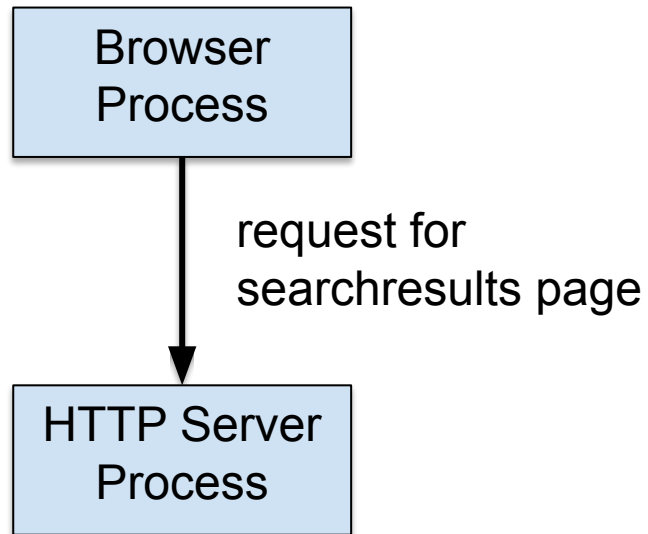
~~searchform
Process~~

Fast CGI Programming



Option 1
(cont.)

Fast CGI Programming



Option 1
(cont.)

Fast CGI Programming

Browser
Process

Option 1
(cont.)

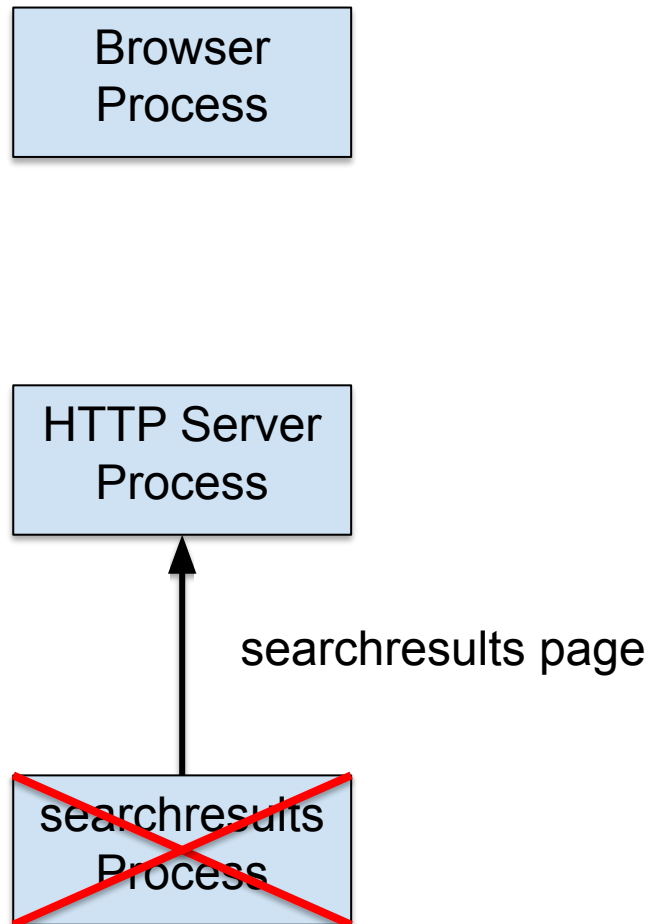
HTTP Server
Process

request for
searchresults page

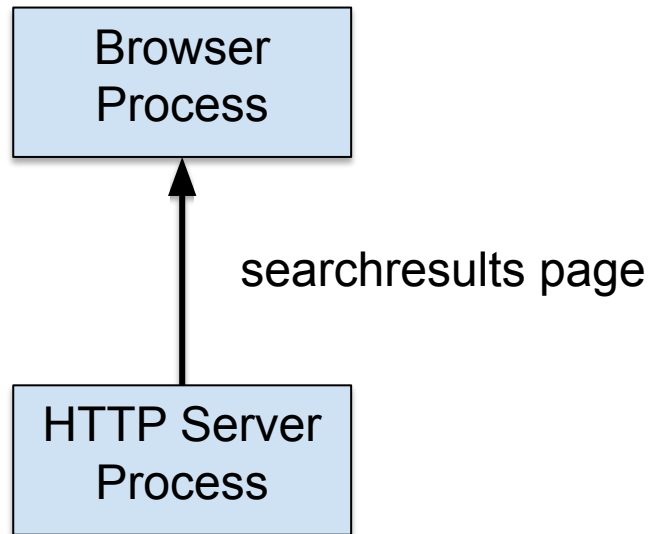
searchresults
Process

Fast CGI Programming

Option 1
(cont.)



Fast CGI Programming



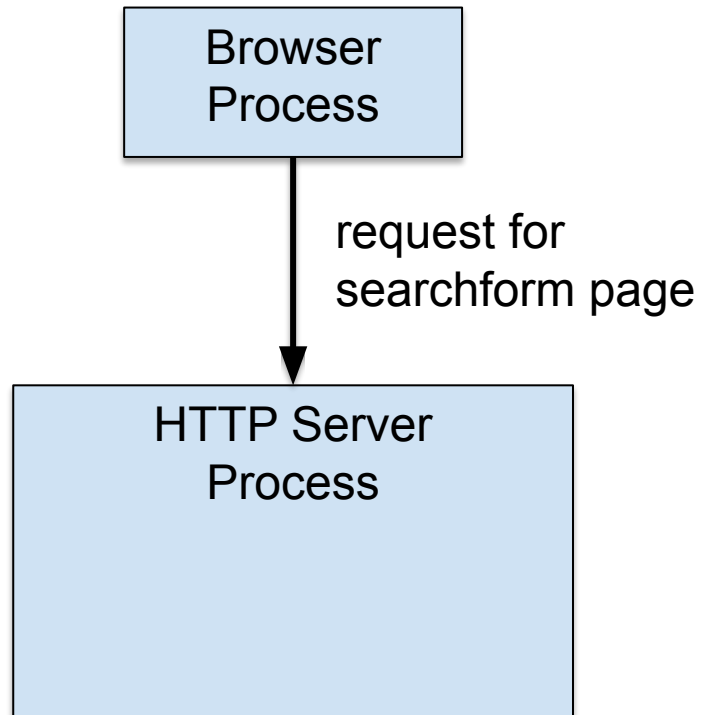
Option 1
(cont.)

Fast CGI Programming

- Consider Option 2...
 - Web app **spawns a child thread** to handle each HTTP request

Fast CGI Programming

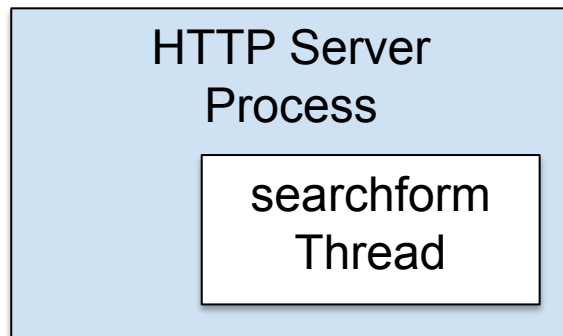
Option 2



Fast CGI Programming

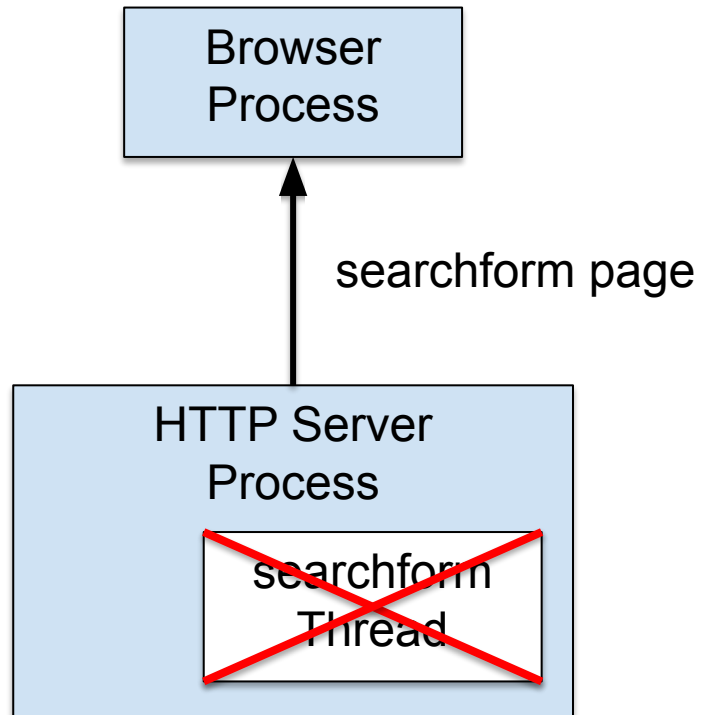
Browser
Process

Option 2
(cont.)



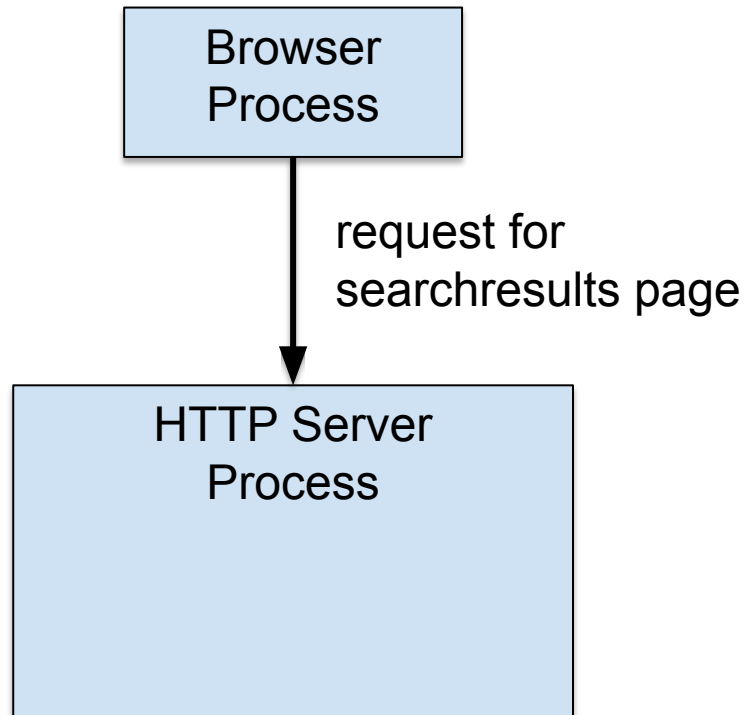
Fast CGI Programming

Option 2 (cont.)



Fast CGI Programming

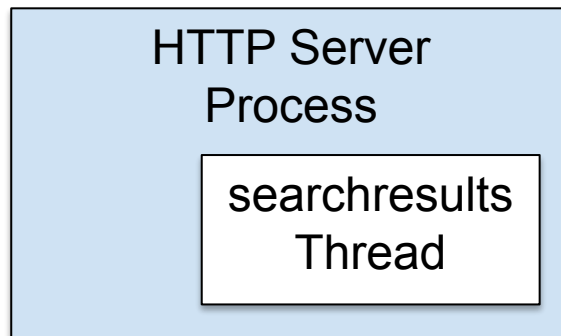
Option 2 (cont.)



Fast CGI Programming

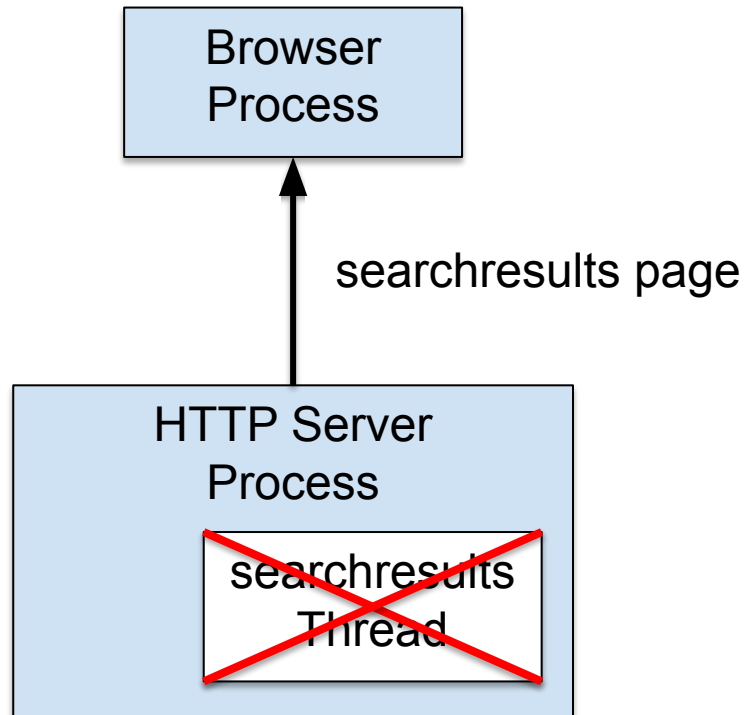
Browser
Process

Option 2
(cont.)



Fast CGI Programming

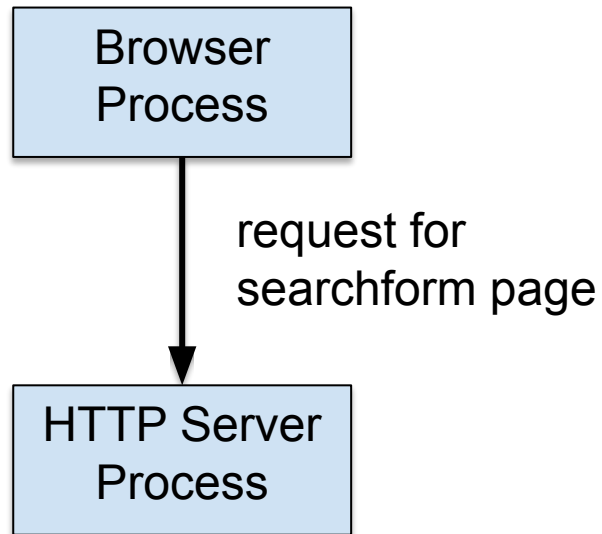
Option 2 (cont.)



Fast CGI Programming

- Consider Option 3...
- **Fast CGI programming**
 - Like Option 1
 - HTTP server forks processes
 - Unlike Option 1
 - Server typically forks one process for each **application**, not for each **request**
 - Application process spawns one thread for each request

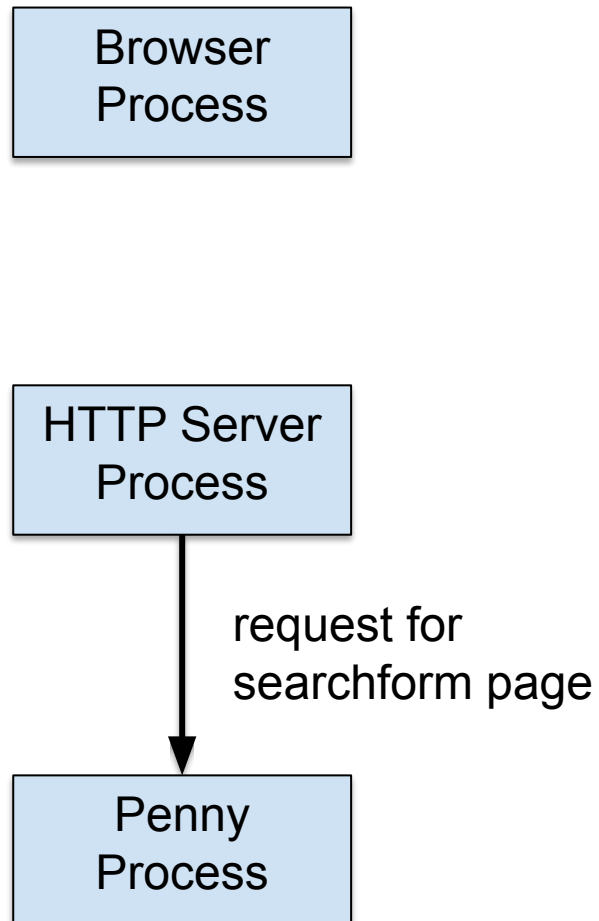
Fast CGI Programming



Fast CGI
Example

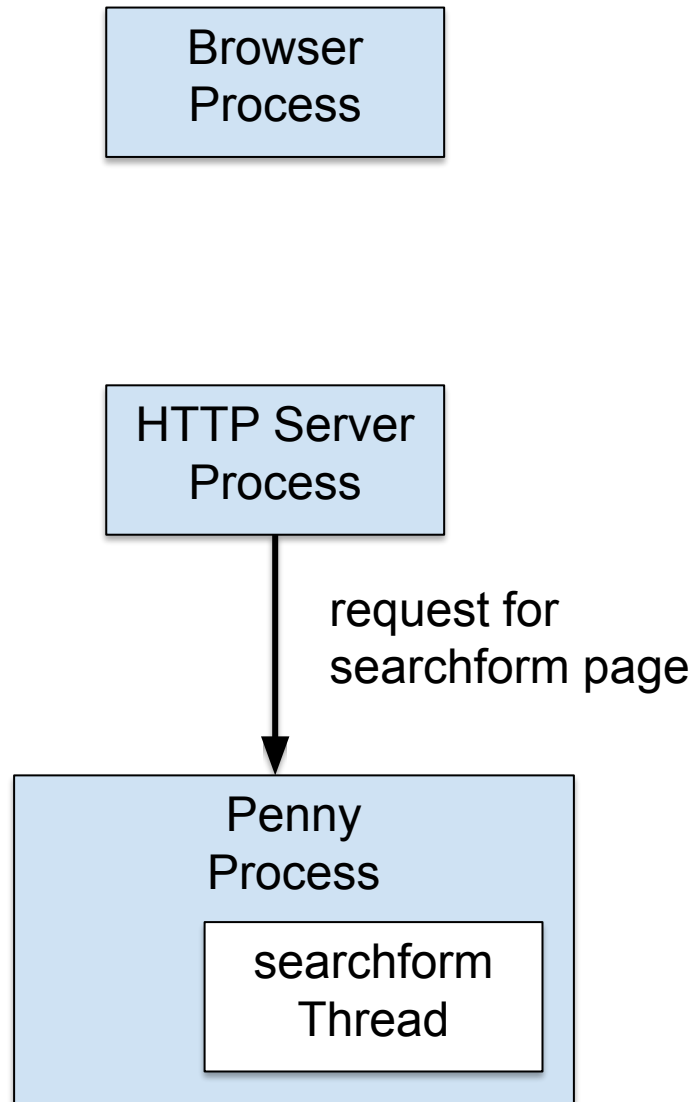
Fast CGI Programming

Fast CGI Example



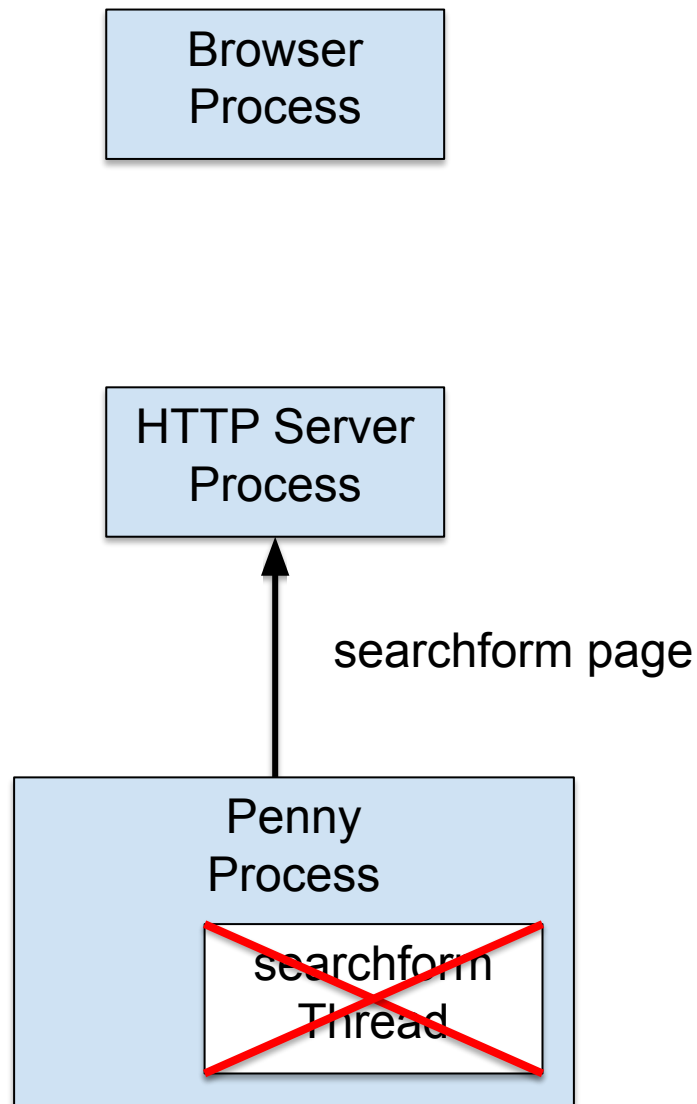
Fast CGI Programming

Fast CGI Example



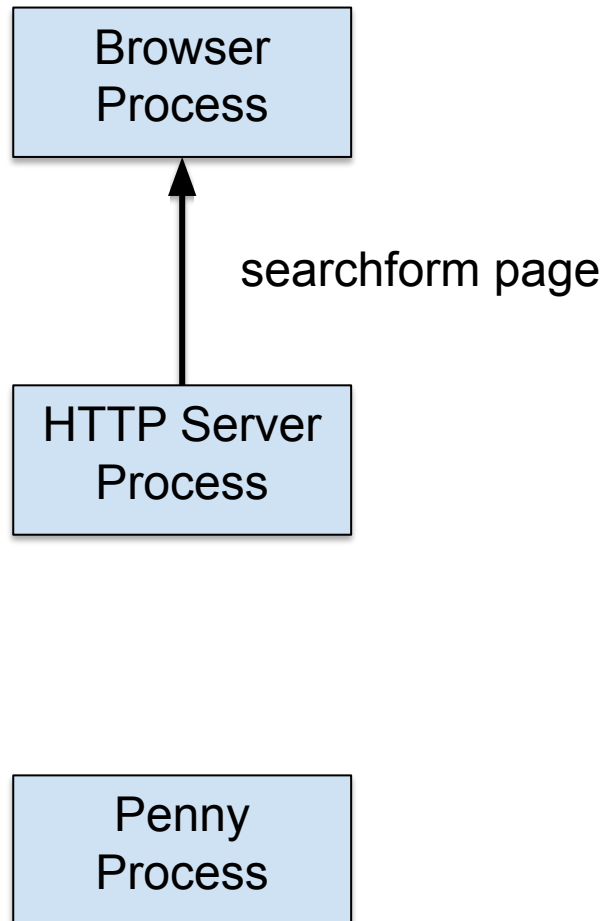
Fast CGI Programming

Fast CGI Example



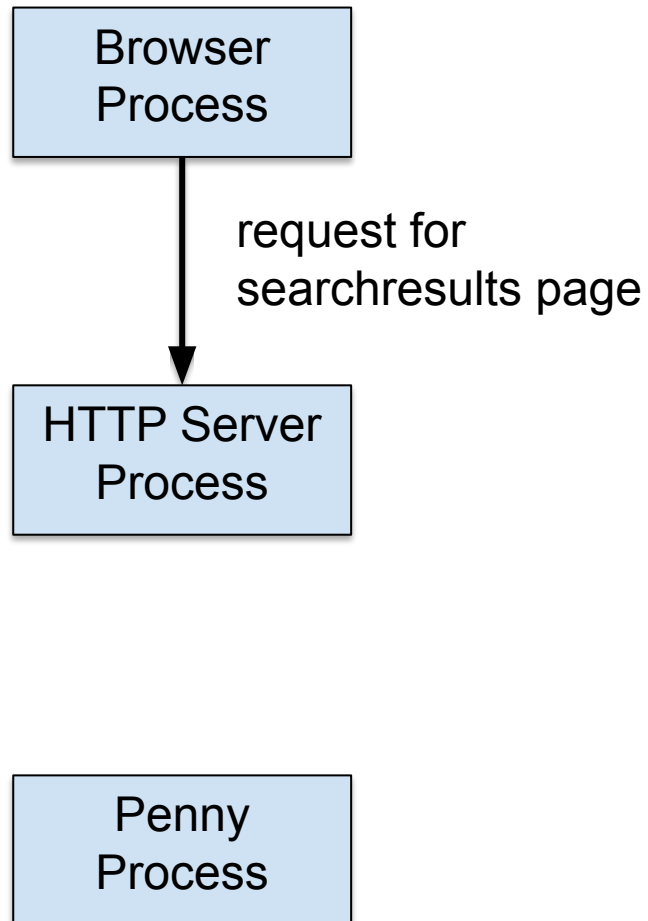
Fast CGI Programming

Fast CGI Example



Fast CGI Programming

Fast CGI Example



Fast CGI Programming

Browser
Process

Fast CGI
Example

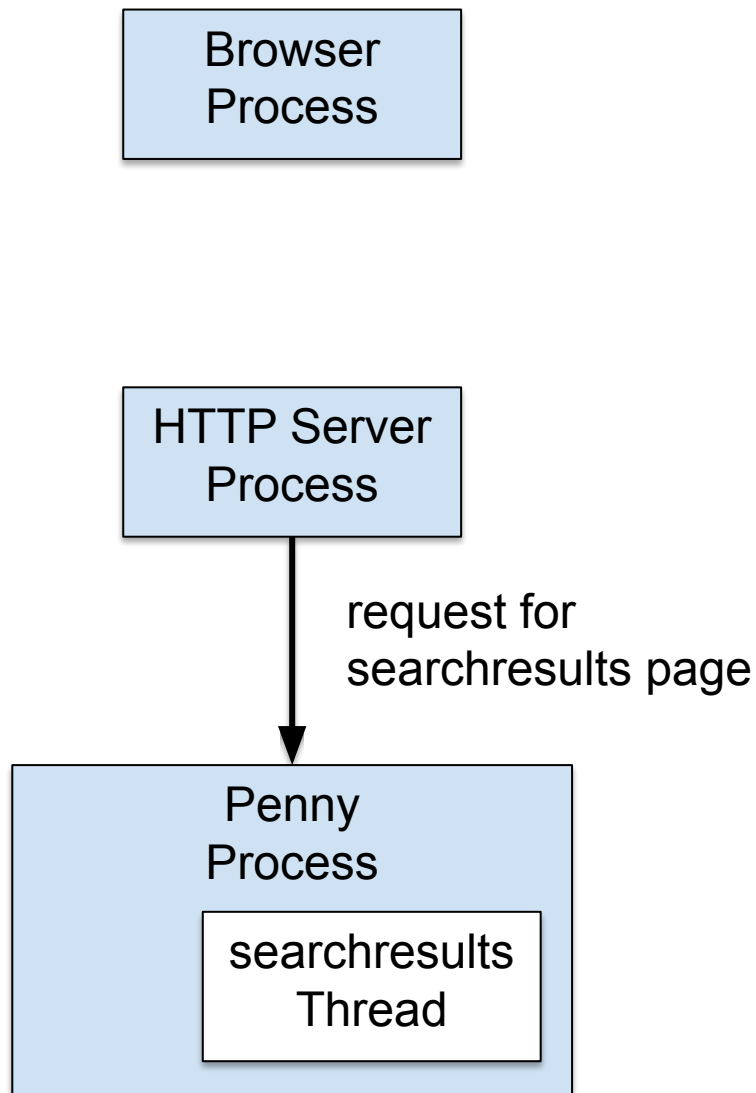
HTTP Server
Process

request for
searchresults page

Penny
Process

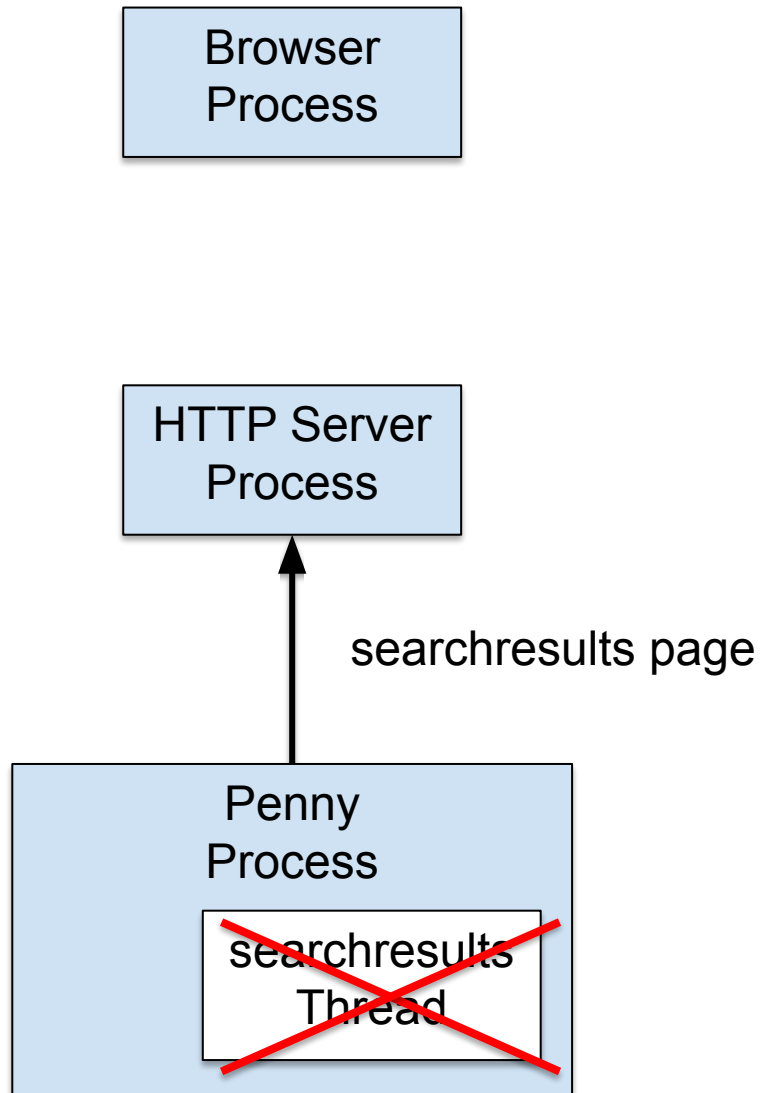
Fast CGI Programming

Fast CGI Example



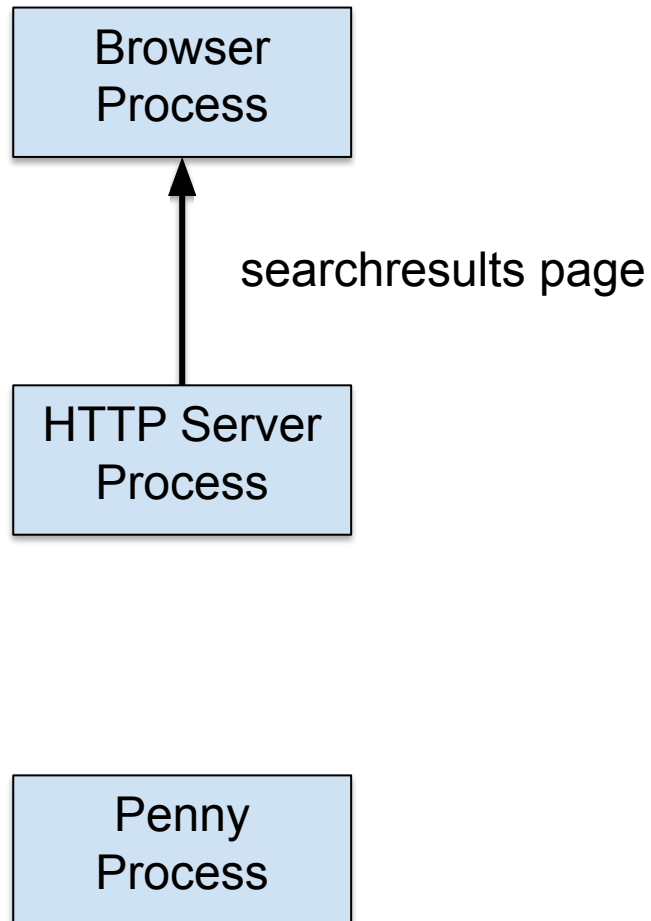
Fast CGI Programming

Fast CGI Example



Fast CGI Programming

Fast CGI Example



Fast CGI Programming

- Fast CGI assessment (simplified):
 - (pro) Faster than Option 1 (CGI)
 - (pro) More flexible than Option 2
 - (pro) More secure than Option 2

Agenda

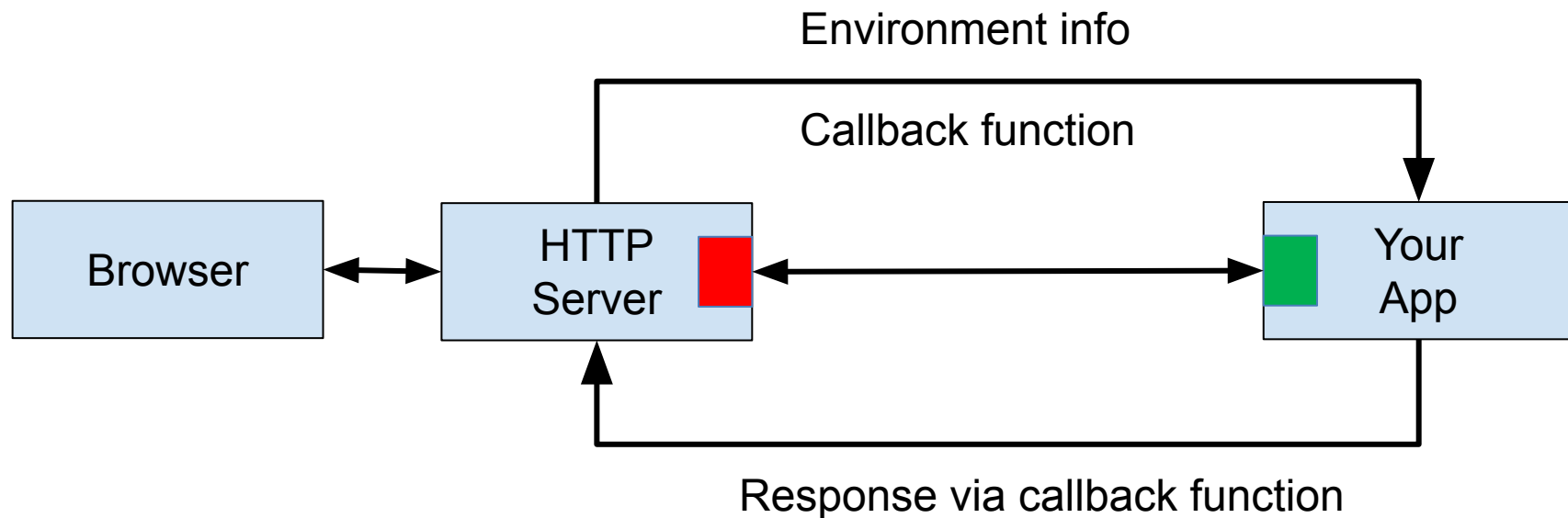
- Fast CGI programming
- **Python WSGI programming**

Python WSGI Programming

- **Question:**
 - How to do fast CGI programming in Python?
- **Answer:**
 - *Web Server Gateway Interface (WSGI)*

Python WSGI Programming

The Python WSGI Architecture

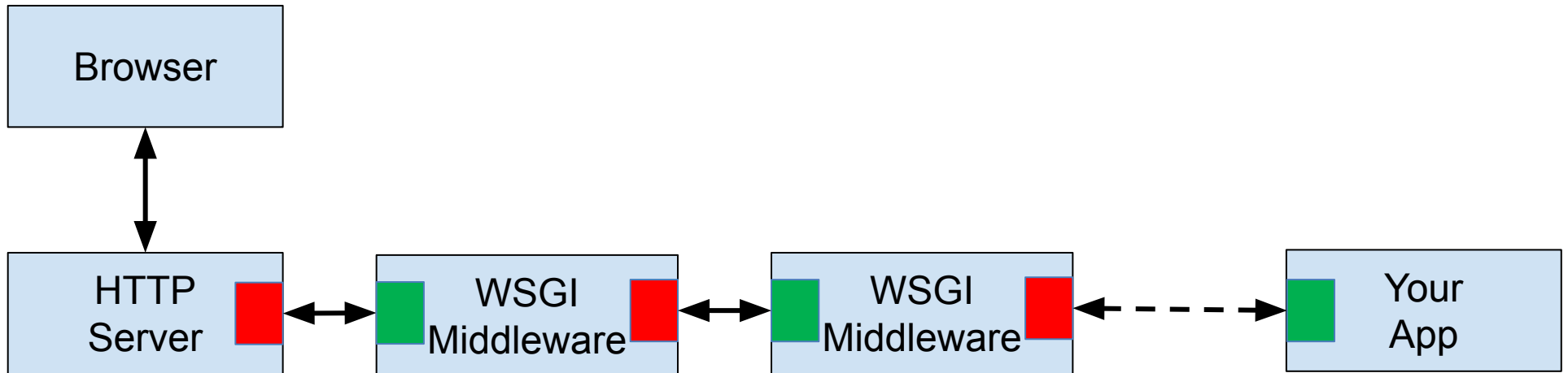


WSGI server-side
Defines messages your app
Can send to HTTP server

WSGI client-side
Defines messages HTTP server
can send to your app

Aside: WSGI Middleware

WSGI allows *middleware*



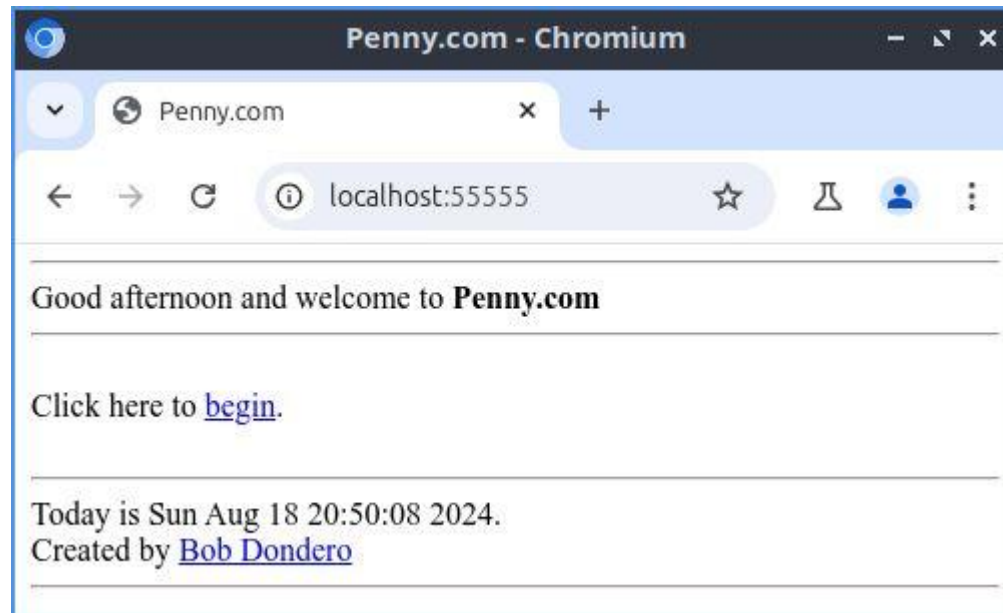
Python WSGI Programming

- See **PennyWsgi** app

```
$ python runserver.py 55555  
Listening on port 55555
```

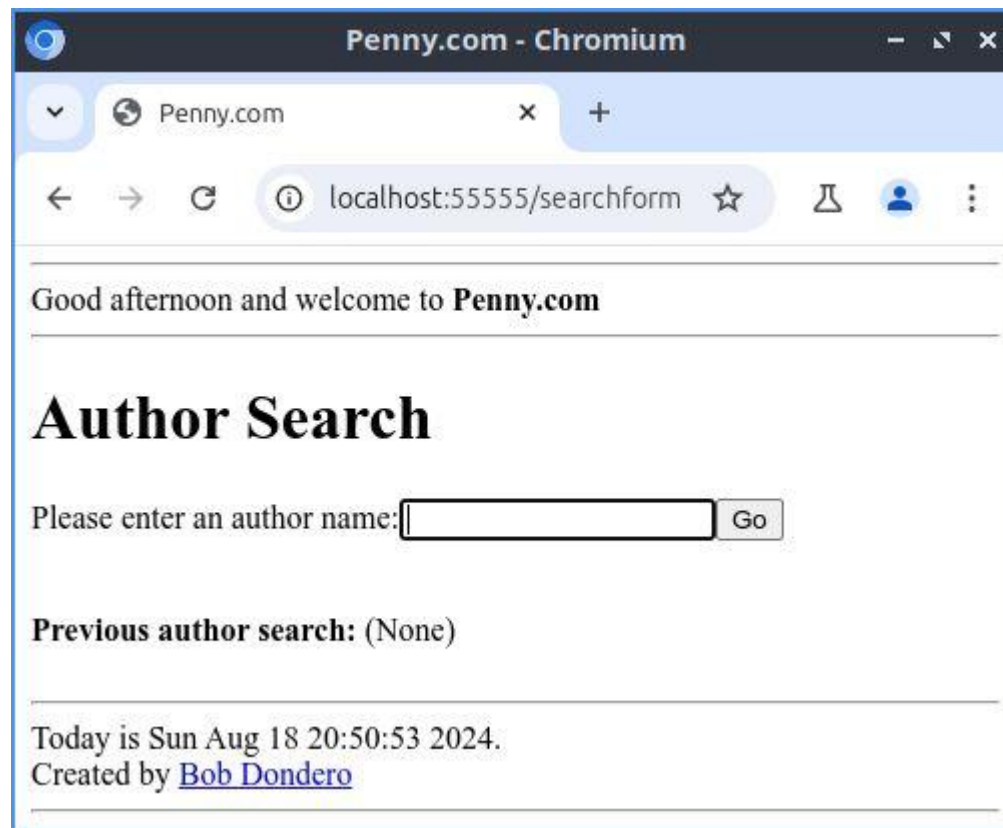
Python WSGI Programming

- See **PennyWsgi** app (cont.)



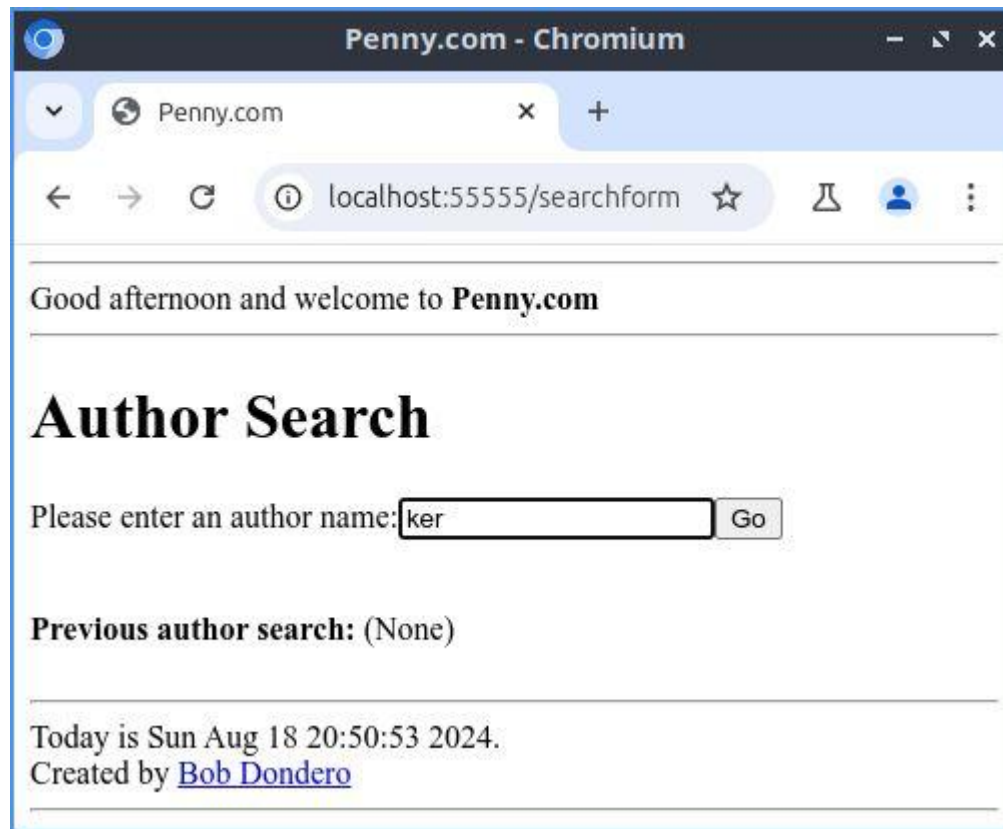
Python WSGI Programming

- See **PennyWsgi** app (cont.)



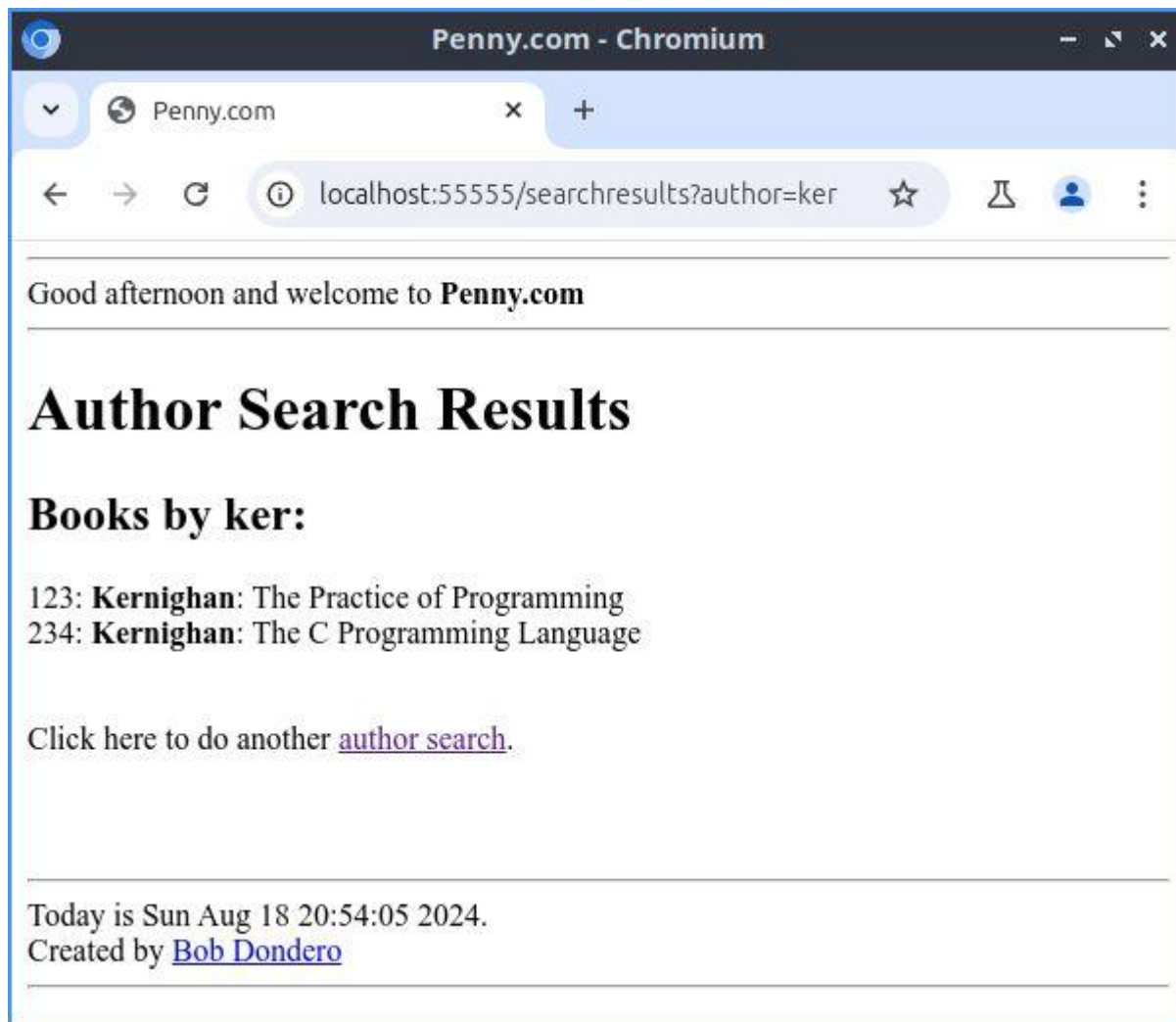
Python WSGI Programming

- See **PennyWsgi** app (cont.)



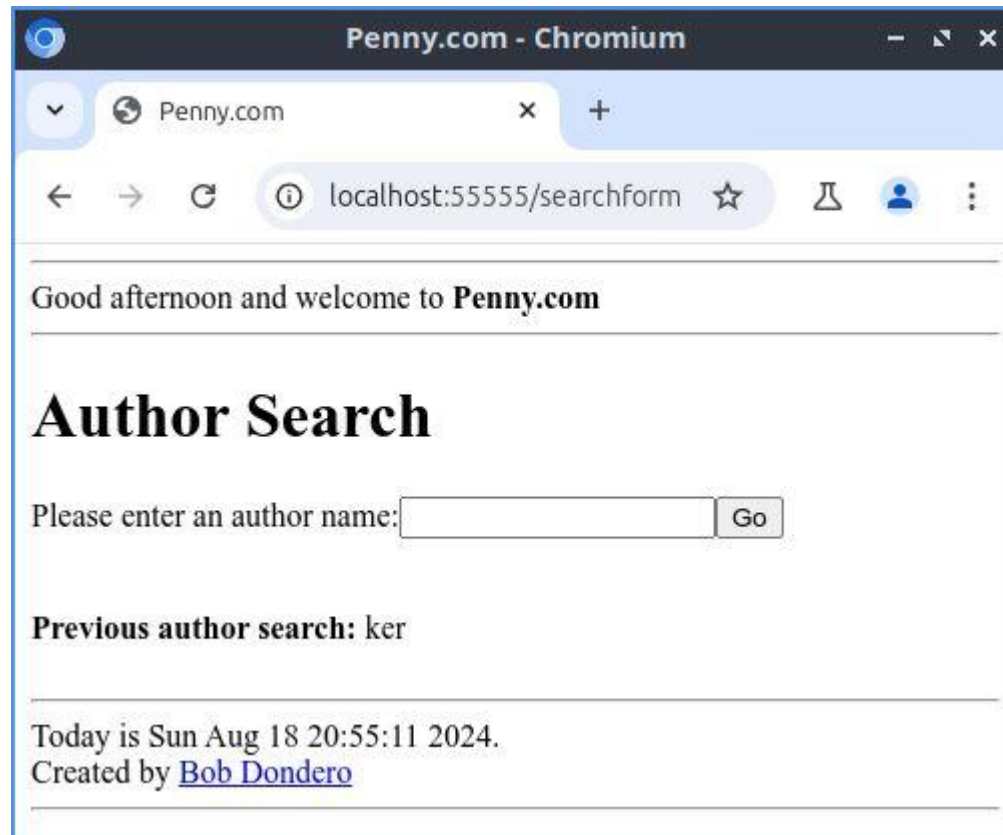
Python WSGI Programming

- See **PennyWsgi** app (cont.)



Python WSGI Programming

- See **PennyWsgi** app (cont.)



Python WSGI Programming

- See **PennyWsgi** app (cont.)
 - **runserver.py**
 - penny.sql
 - penny.sqlite
 - database.py
 - parseargs.py
 - **common.py**
 - **penny.py**

Python WSGI Programming

- WSGI vs. CGI:
 - (con) More complex
 - (pro) Faster!
 - (pro) Separates URLs from **file** names
 - Allows use of “pretty” URLs
- WSGI also separates URLs from **function** names
 - Could improve maintainability

Summary

- We have covered:
 - Fast CGI programming
 - Python WSGI programming