Server-Side Web Programming: CGI (Part 1)

Objectives

- We will cover…
    - Common Gateway Interface (CGI) programming
    - The HTTP GET method
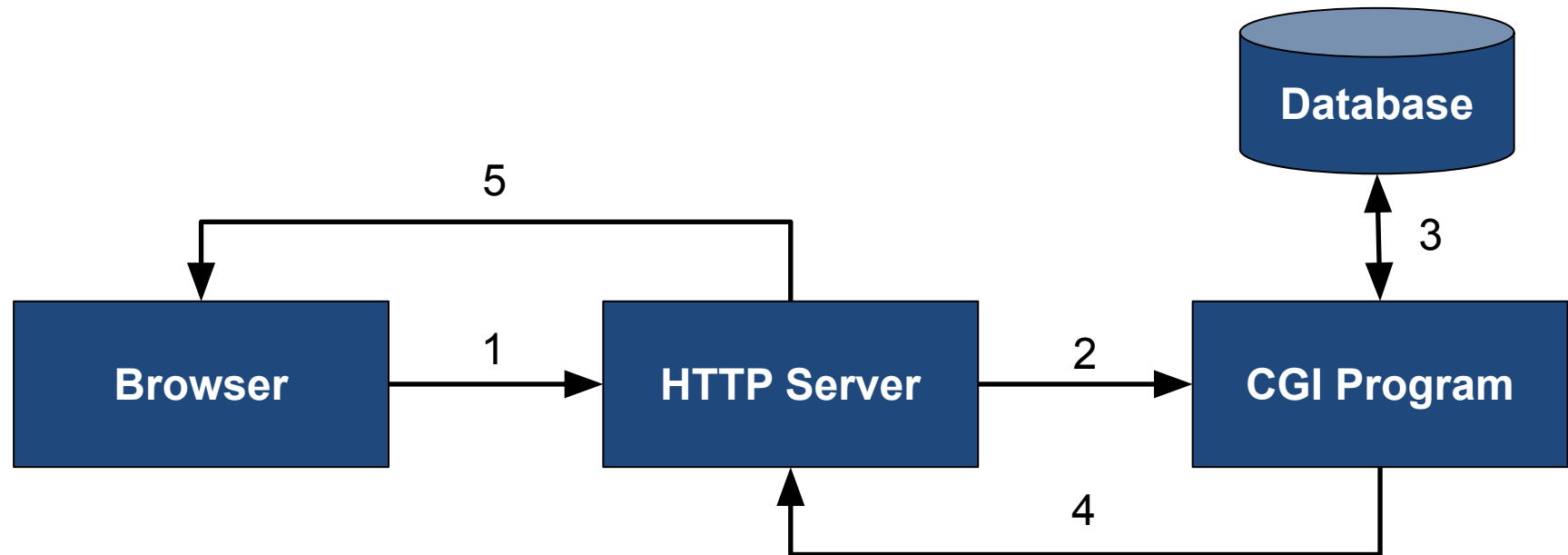   - The HTTP POST method

Agenda

- **CGI programming**
- URL encoding
- CGI with arguments
- CGI using the HTTP POST method

CGI

- **Problem**:
  - Often HTTP server must generate HTML docs **dynamically**

CGI

- One **solution**:
- *Common Gateway Interface (CGI)* protocol

CGI

- **Question**: How does user command browser to communicate with HTTP server using CGI?

- **Answer 1**: User enters URL at top of browser

CGI

- Answer 1 elaboration

protocol://*host*:*port*/cgi-bin/*somepgm.py*

The default protocol is http
There is no default host
The default port is 80 for http, 443 for https

CGI

- **Question**: How does user command browser to communicate with HTTP server using CGI?

- **Answer 1**: User enters URL at top of browser

- **Answer 2**: User clicks on page link

CGI

- Answer 2 elaboration

```
<a href="protocol://host:port/cgi-bin/somepgm.py">
   ...
</a>
```

The default protocol is the protocol used to deliver this page
The default host is the host that delivered **this** page
The default port is the port from which **this** page was delivered

CGI

- **Question**: How does user command browser to communicate with HTTP server using CGI?

- **Answer 1**: User enters URL at top of browser

- **Answer 2**: User clicks on page link

- **Answer 3**: User submits a form

CGI

- Answer 3 elaboration

```
<form action="protocol://host:port/cgi-bin/somepgm.py" method="get">
    <input type="submit">
</form>
```

The default protocol is the protocol used to deliver this page
The default host is the host that delivered **this** page
The default port is the port from which **this** page was delivered
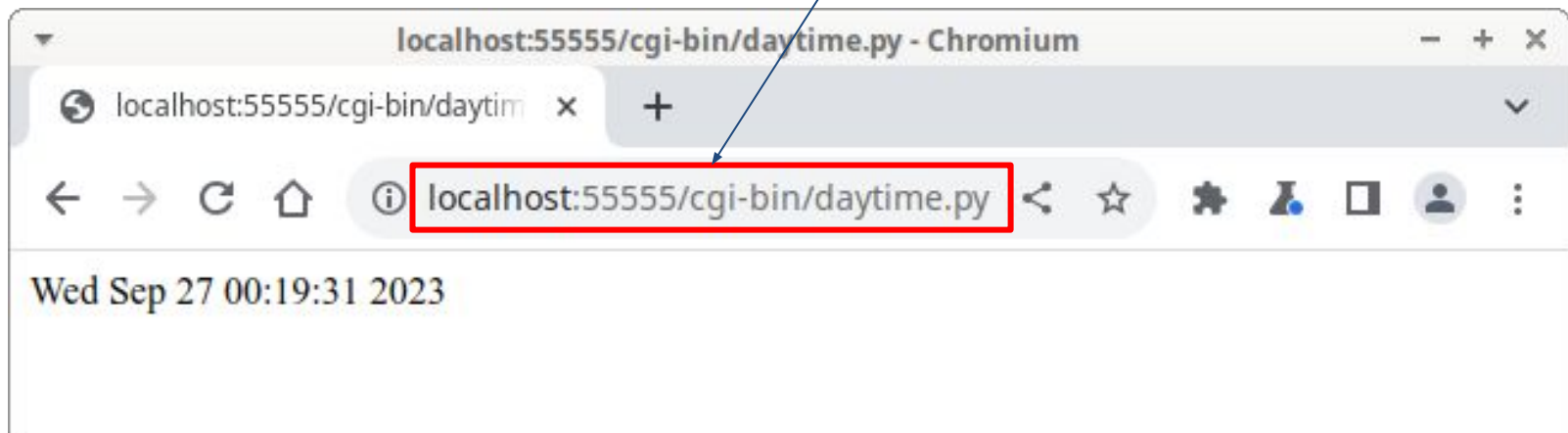
CGI

- See **<u>DayTime</u>** app

```
$ python runserver.py 55555
Serving HTTP on 0.0.0.0 port 55555 (http://0.0.0.0:55555/) ...
```

CGI

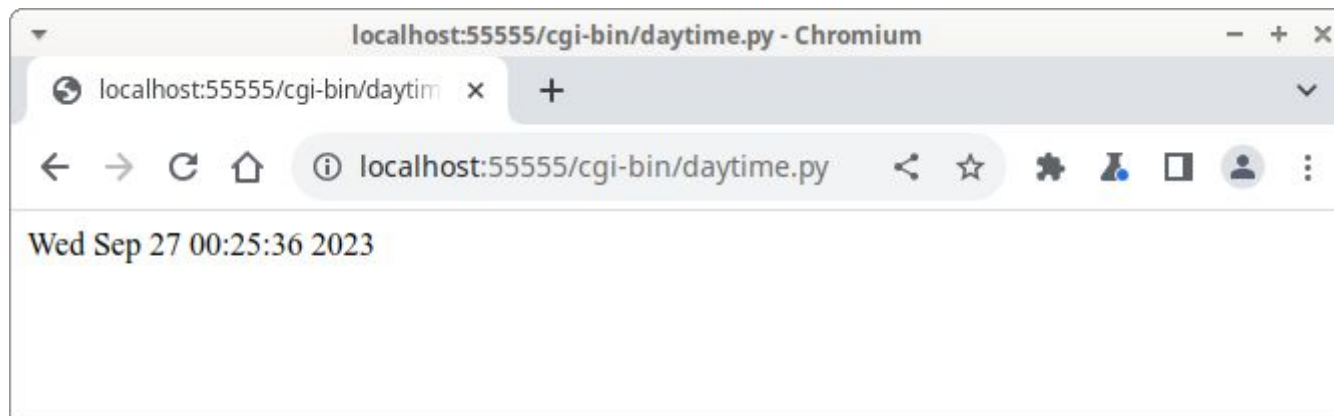- See **DayTime** app (cont).

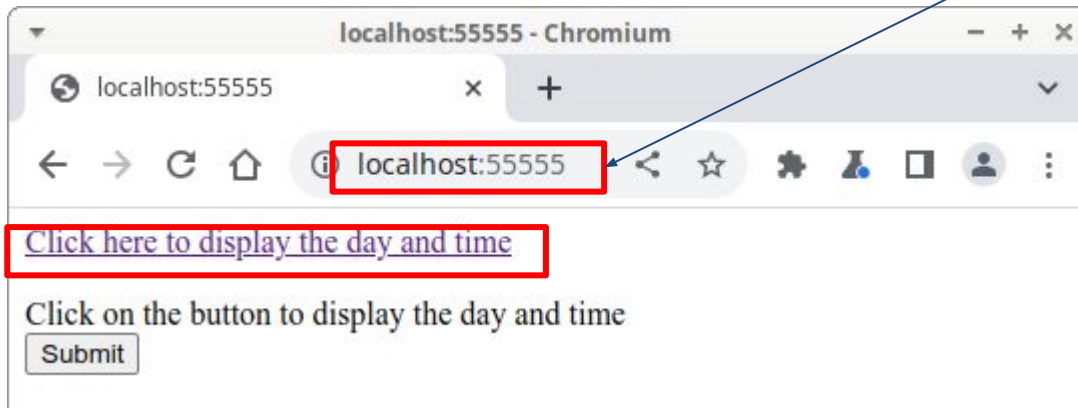Approach 1:

Default URL prefix is `http://`

localhost:55555/cgi-bin/daytime.py - Chromium

localhost:55555/cgi-bin/daytim × +

localhost:55555/cgi-bin/daytime.py

Wed Sep 27 00:19:31 2023

CGI

- See **<u>DayTime</u>** app (cont.)

Approach 2:

Default page is index.html

localhost:55555 - Chromium

localhost:55555

Click here to display the day and time

Click on the button to display the day and time
Submit

localhost:55555/cgi-bin/daytime.py - Chromium

localhost:55555/cgi-bin/daytime.py

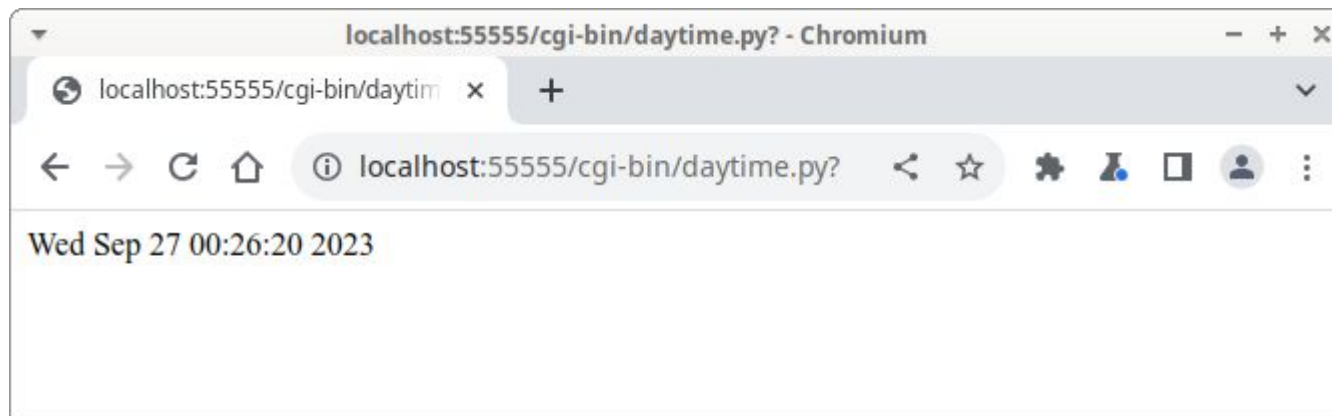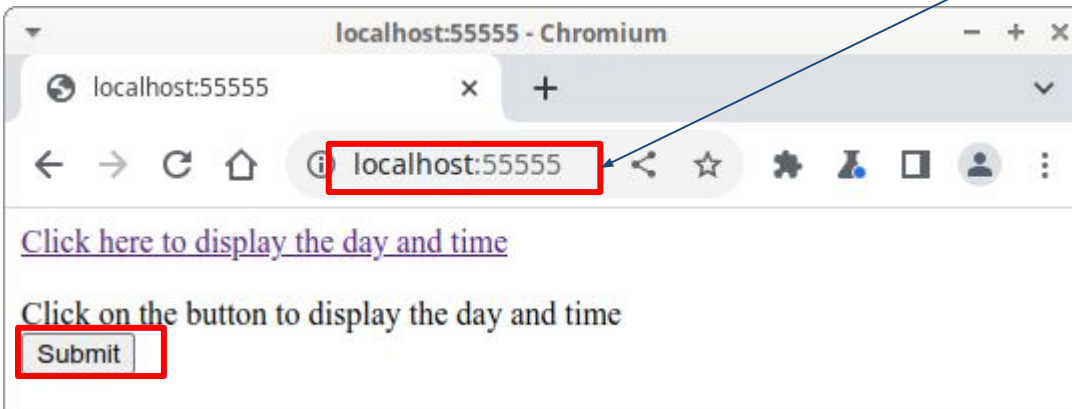Wed Sep 27 00:25:36 2023

14

CGI

- See **<u>DayTime</u>** app (cont.)

Approach 3:

Default page is index.html

CGI

- See **<u>DayTime</u>** app (cont.)
  - **runserver.py**
  - **index.html**
  - **cgi-bin/daytime.py**

Agenda

- CGI programming
- **URL encoding**
- CGI with arguments
- CGI using the HTTP POST method

URL Encoding

- ## *URL encoding*

  - A way of encoding a string such that it may appear within a URL

  - Each space is encoded as +

  - Each special character (`"`, `'`, `=`, `&`, etc.) is encoded as:

    - `%nn` where `nn` are hex digits as per UTF-8, or

    - `%nn%nn`, or

    - `%nn%nn%nn`, or

    - `%nn%nn%nn%nn`

URL Encoding

- Python support for URL encoding
  - `urllib.parse.quote_plus()`
    - string → URL encoded string
  - `urllib.parse.unquote_plus()`
    - URL encoded string → string

URL Encoding

- See **urlencode.py**

```
$ python urlencode.py one
Original string: one
Encoded string:  one
Decoded string:  one
$ python urlencode.py 't wo'
Original string: t wo
Encoded string:  t+wo
Decoded string:  t wo
$ python urlencode.py 'th" ree'
Original string: th" ree
Encoded string:  th%22+ree
Decoded string:  th" ree
$ python urlencode.py '~!@#$%^&*()_+'
Original string: ~!@#$%^&*()_+
Encoded string:
~%21%40%23%24%25%5E%26%2A%28%29_%2B
Decoded string:  ~!@#$%^&*()_+
$
```

Agenda

- CGI programming
- URL encoding
- **CGI with arguments**
- CGI using the HTTP POST method

CGI with Arguments

- **Question**: How does browser pass arguments to a CGI program?
- **Answer 1**: User enters URL at top of browser with name=value pairs appended

CGI with Arguments

- Answer 1 elaboration

```
protocol://host:port/cgi-bin/somepgm.py?
    name1=value1&name2=value2
```

The default protocol is http
There is no default host
The default port is 80 for http, 443 for https

CGI with Arguments

- **Question**: How does user command browser to communicate with HTTP server using CGI with the **GET** method?

- Answer 1: User enters URL at top of browser

- **Answer 2**: User clicks on page link that has name=value pairs appended

CGI with Arguments

- Answer 2 elaboration

```
<a href="protocol://host:port/cgi-bin/somepgm.py?
    name1=value1&name2=value2">
    ...
</a>
```

The default protocol is the protocol used to deliver **this** page
The default host is the host that delivered **this** page
The default port is the port from which **this** page was delivered

CGI with Arguments

- **Question**: How does user command browser to communicate with HTTP server using CGI with the **GET** method?

- **Answer 1**: User enters URL at top of browser

- **Answer 2**: User clicks on page link

- **Answer 3**: User submits form that has input elements of type text

CGI with Arguments

- Answer 3 elaboration

User enters value1 and
value2 in input elements

```
<form action="protocol://host:port/cgi-bin/somepgm.py" method="get">
    <input type="text" name="name1" value="init1">
    <input type="text" name="name2" value="init2">
    <input type="submit">
</form>
```

Sends HTTP request as if generated by clicking on page link with href
protocol://*host*:*port*/cgi-bin/*somepgm.py?name1=value1&name2=value2*
The default protocol is the protocol used to deliver **this** page
The default host is the host that delivered **this** page
The default port is the port from which **this** page was delivered

27

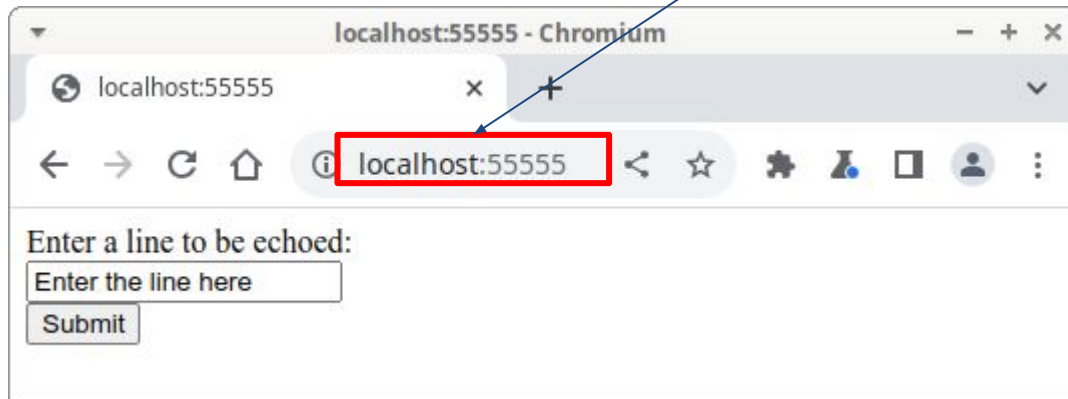# CGI with Arguments

- See **<u>EchoGet</u>** app

```
$ python runserver.py 55555
Serving HTTP on 0.0.0.0 port 55555 (http://0.0.0.0:55555/) ...
```
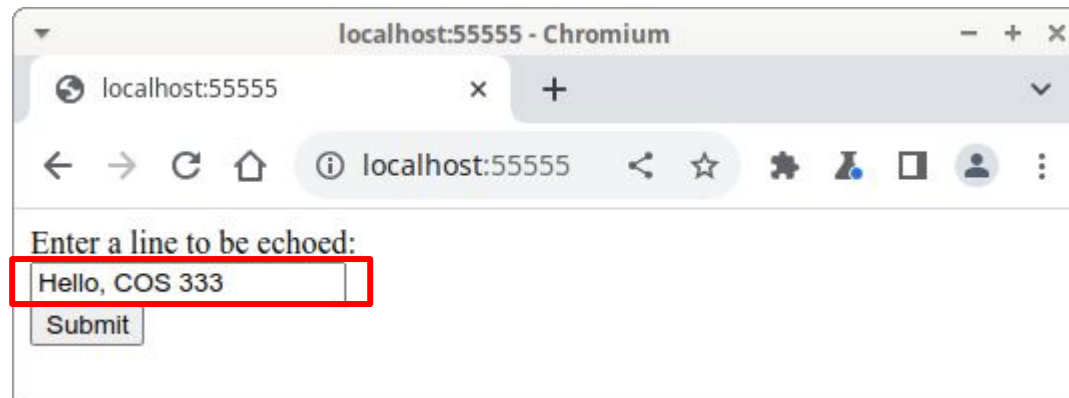
CGI with Arguments

- See **<u>EchoGet</u>** app (cont.)

Default protocol is HTTP
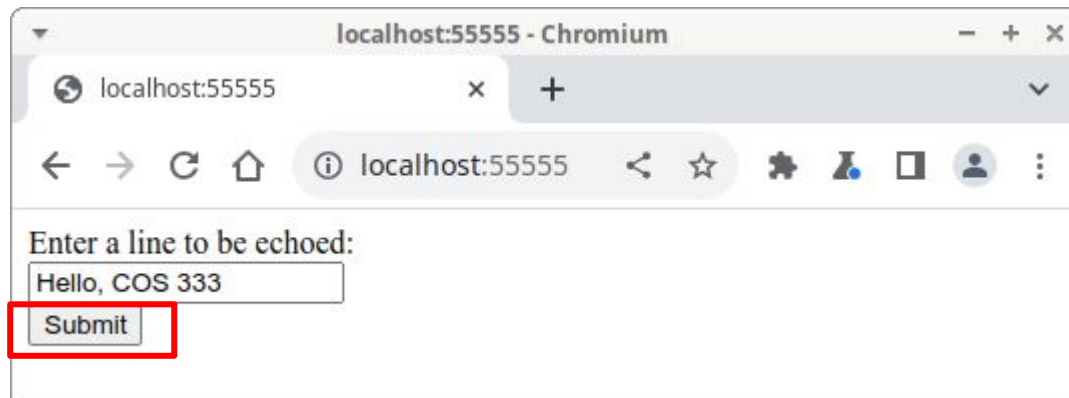Default page is index.html

CGI with Arguments

- See **<u>EchoGet</u>** app (cont.)
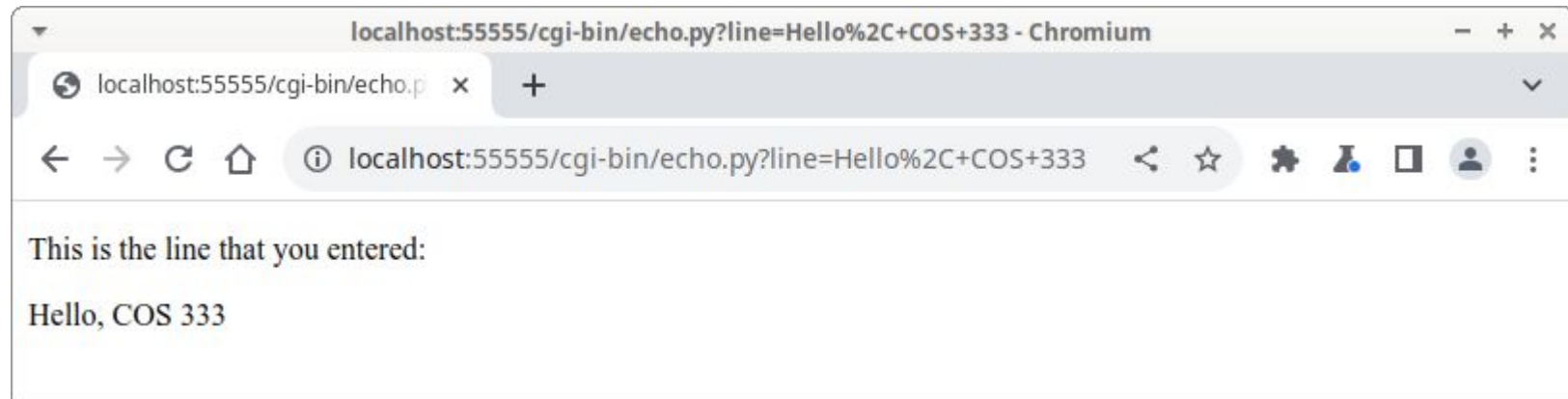
CGI with Arguments

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

Browser sends HTTP request as if generated by clicking on a page link with href
`http://localhost:55555/cgi-bin/echo.py?`
`line=Hello%2C+COS+333`

CGI with Arguments

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

CGI with Arguments

Browser

User enters "Hello, COS 333" and clicks form "submit" button

Names and values are URL encoded

Socket

GET /cgi-bin/echo.py?line=Hello%2C+COS+333 HTTP/1.1
Host: localhost
<Blank line>

HTTP Server

Environment variable

fork/exec   Pipe (usually)

QUERY_STRING: line=Hello%2C+COS+333

CGI with Arguments

```
python echo.py
```

wait

Pipe
(usually)

```
HTTP Server
```

Socket

```
Browser
```

Content-Type: text/html; charset=utf-8

<!DOCTYPE html>
<html>
<body>
<p>Hello, COS 333</p>
</body>
</html>

Uses
QUERY_STRING
Writes to stdout

HTTP/1.1 200 OK
…
Content-Type: text/html;charset=utf-8

<!DOCTYPE html>
<html>
<body>
<p>Hello, COS 333</p>
</body>
</html>

There are
some additional
headers

34

CGI with Arguments

- See **EchoGet** app (cont.)
  - runserver.py
  - **index.html**
  - **cgi-bin/parseargs.py**
  - **cgi-bin/echo.py**

Agenda

- CGI programming
- URL encoding
- CGI with arguments
- **CGI using the HTTP POST method**

CGI Using POST

- **So far:** CGI with the GET method

- **Question**: How does user command browser to communicate with HTTP server using CGI with the **POST** method?

- **Answer**: User submits form

CGI Using POST

- Answer elaboration

User enters value1 and
value2 in input elements

```
<form action="protocol://host:port/cgi-bin/somepgm.py" method="post">
    <input type="text" name="name1" value="init1">
    <input type="text" name="name2" value="init2">
    <input type="submit">
</form>
```

The default protocol is the protocol used to deliver **this** page
The default host is the host that delivered **this** page
The default port is the port from which **this** page was delivered
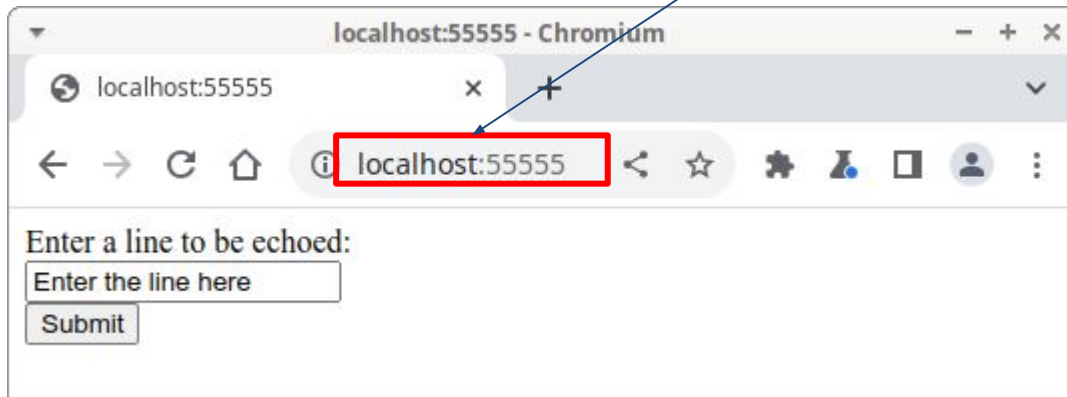
38

CGI Using POST

- See **EchoPost** app

```
$ python runserver.py 55555
Serving HTTP on 0.0.0.0 port 55555 (http://0.0.0.0:55555/) ...
```
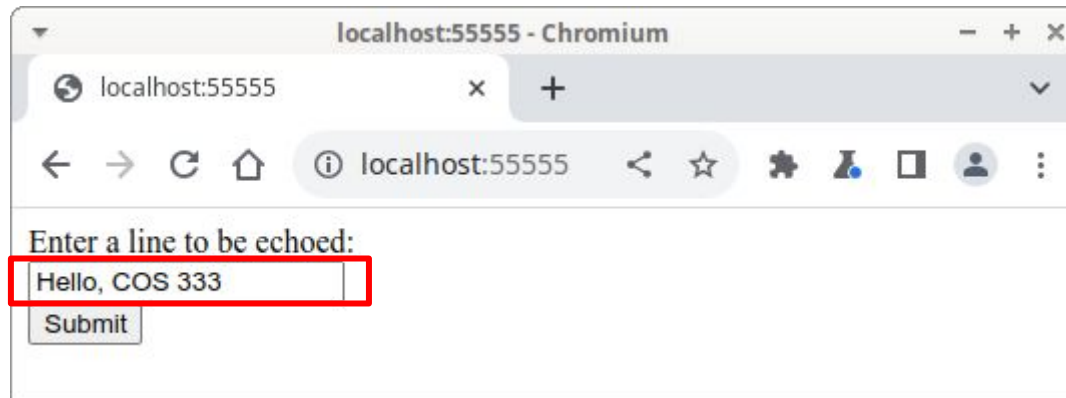
CGI with Arguments

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

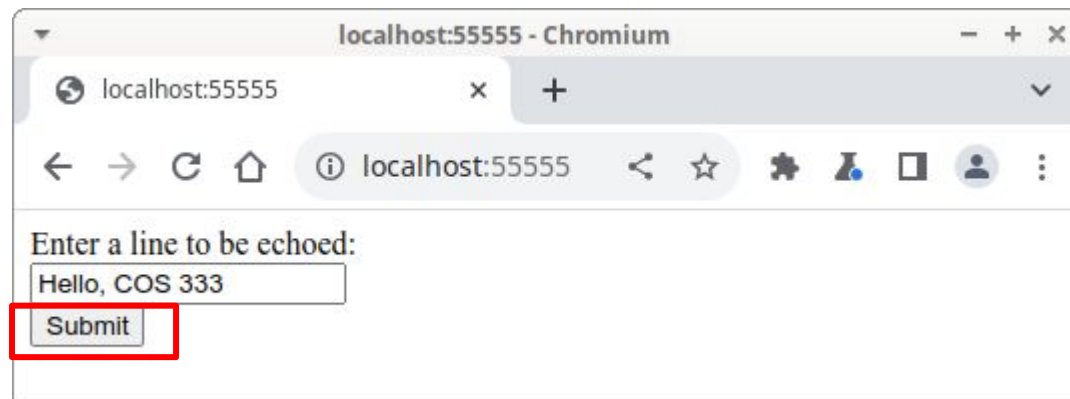Default protocol is HTTP
Default page is index.html

CGI with Arguments

- See **EchoPost** app (cont.)
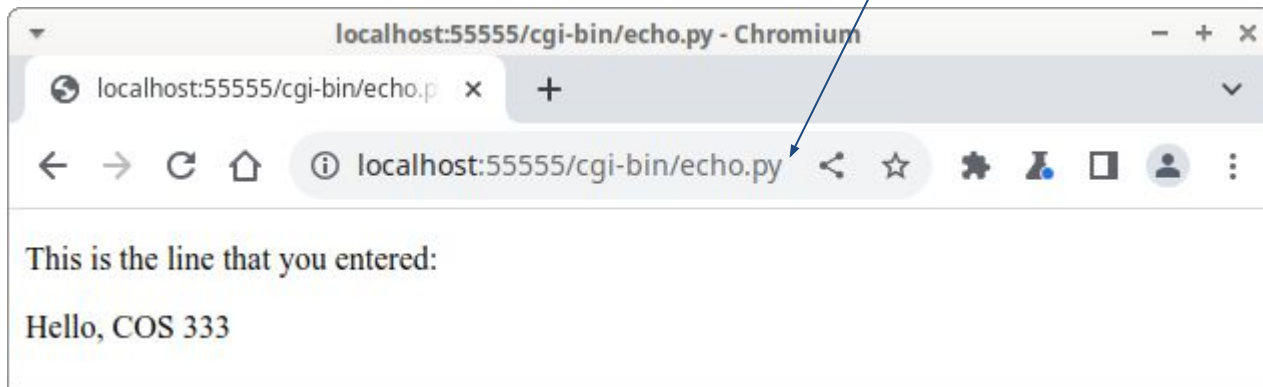
CGI with Arguments

- See **<u>EchoPost</u>** app (cont.)

CGI with Arguments

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

No name=value pairs in URL

CGI Using POST

Browser

Socket

HTTP Server

fork/exec   Pipe
(usually)

User enters "Hello, COS 333"
and clicks form "submit" button

POST /cgi-bin/echo.py HTTP/1.1
Host: localhost
Content-type: application/x-www-form-urlencoded
Content-length: 19

line=Hello%2C+COS+333

Binds stdin of child process to pipe
Writes to pipe: line=Hello%2C+COS+333

CGI Using POST

```
python echo.py
```

Pipe (usually)

```
HTTP Server
```

Socket

```
Browser
```

Content-Type: text/html; charset=utf-8

```
<!DOCTYPE html>
<html>
<body>
<p>Hello, COS 333</p>
</body>
</html>
```

Reads from stdin
Writes to stdout

HTTP/1.1 200 OK
…
Content-Type: text/html; charset=utf-8

```
<!DOCTYPE html>
<html>
<body>
<p>Hello, COS 333</p>
</body>
</html>
```

There are some additional headers

CGI Using POST

- See **EchoPost** app (cont.)
    - runserver.py
    - **index.html**
    - cgi-bin/parseargs.py
    - **cgi-bin/echo.py**

Summary

- We have covered:
  - Common Gateway Interface (CGI) programming
  - The HTTP GET method
  - The HTTP POST method