

# Server-Side Web Programming: CGI (Part 3)

Copyright © 2022 by  
Robert M. Dondero, Ph.D.  
Princeton University

## Objectives

- We will cover:
  - Stateful web programming
  - Redirection in web programming

# Agenda

- **Fundamental example**
- Stateful web programming
- Stateful web programming with cookies
- Redirection in web programming

## Fundamental Example

- **Penny app**

- Website for a very small bookstore
- Browser sends author name prefix
- App sends list of books written by authors whose names have that prefix
  - Author, book title, book price
- We'll see *many* versions

4

### Fundamental Example

[see slide]

Named after the Pennypack Creek

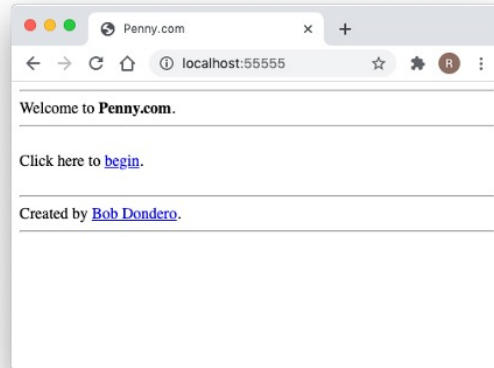
## Fundamental Example

- See **PennyCgi** app

```
$ python runserver.py 5555  
Serving HTTP on 0.0.0.0 port 5555  
(http://0.0.0.0:5555/) ...
```

# Fundamental Example

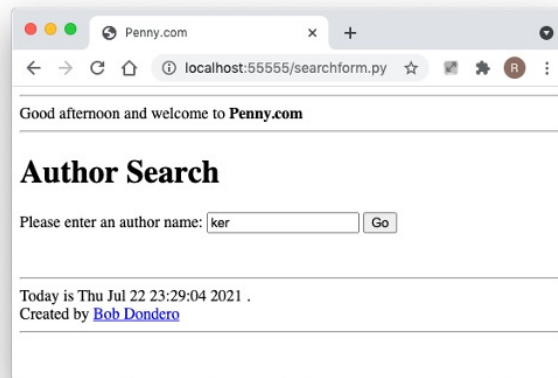
- See **PennyCgi** app (cont.)



The  
index  
page

# Fundamental Example

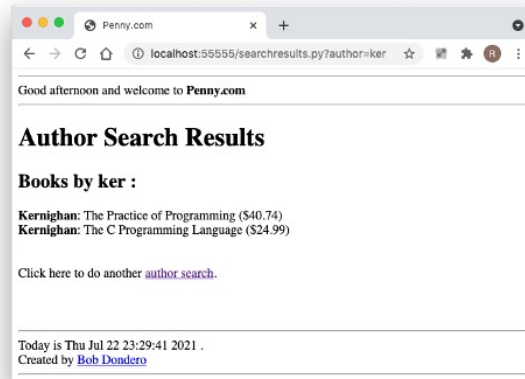
- See **PennyCgi** app (cont.)



The  
searchform  
page

# Fundamental Example

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

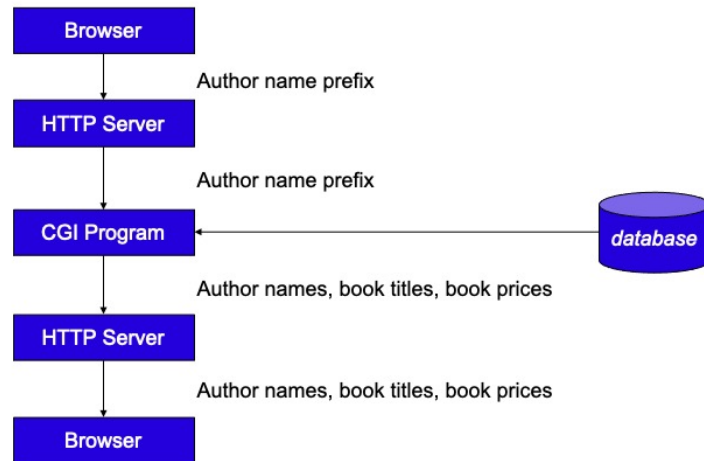


The  
searchresults  
page



# Fundamental Example

See **PennyCgi** app (cont.)



## Fundamental Example

- See **PennyCgi** app (cont.)
  - runserver.py
  - **penny.sql**, penny.sqlite
  - **index.html**
  - **book.py**, database.py
  - **common.py**
  - **searchform.py**, searchresults.py

10

### Fundamental Example

[see slide]

Code notes: penny.sql

Used to create the penny.sqlite database  
sqlite3 penny.sqlite  
.read penny.sql

Code notes: index.html

The default page  
Displays a page link to searchform.py

Code notes: book.py

An ordinary class; unrelated to CGI programming

Code notes: database.py

Handles interaction with the database  
Note use of prepared statement

Code notes: common.py

Defines functions that return HTML fragments

Code notes: searchform.py

- Dynamically generates the searchform page

- Searchform page contains a form

- Form sends a HTTP request for searchresults.py

Code notes: searchresults.py

- Accepts data from FieldStorage

- Accesses the database

- Dynamically generates the searchresults page

- Writes the searchresults page to stdout

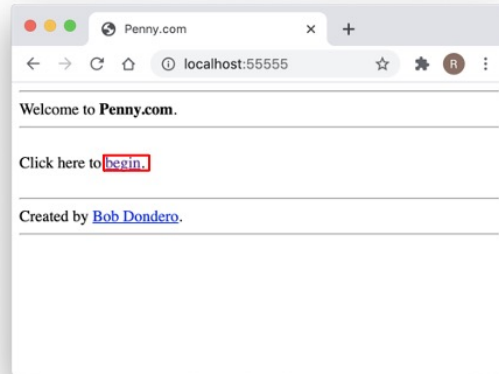
- searchresults page contains convenient page link back to searchform page

## Agenda

- Fundamental example
- **Stateful web programming**
- Stateful web programming with cookies
- Redirection in web programming

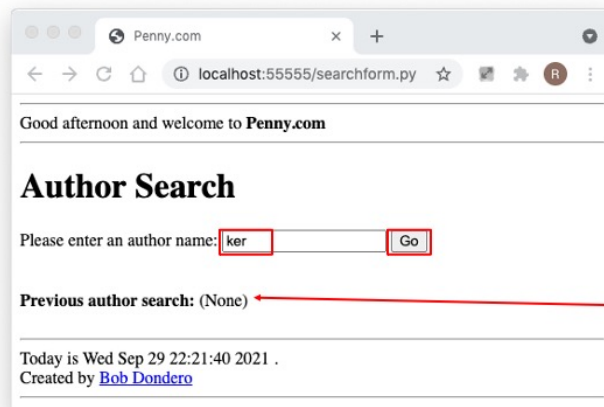
# Stateful Web Programming

- See **PennyCgiState** app



# Stateful Web Programming

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

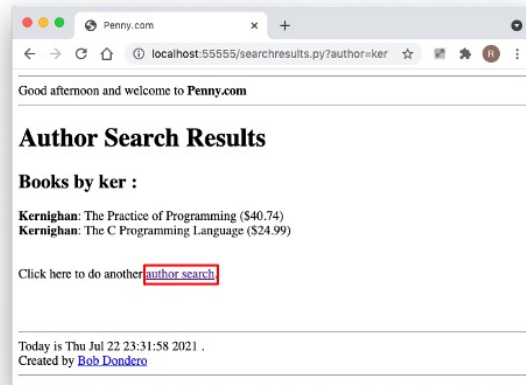


The screenshot shows a web browser window with the title "Penny.com". The address bar displays "localhost:55555/searchform.py". The page content includes a welcome message "Good afternoon and welcome to Penny.com", a section titled "Author Search", a form with the label "Please enter an author name:" containing a text input field with the value "ker" and a "Go" button, and a line indicating "Previous author search: (None)". At the bottom, it shows the date and time "Today is Wed Sep 29 22:21:40 2021 ." and the creator "Created by [Bob Dondoro](#)".

Note

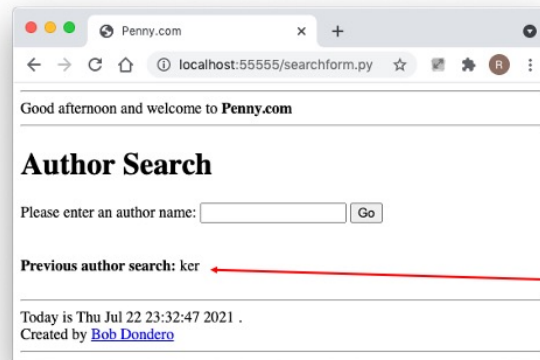
# Stateful Web Programming

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



# Stateful Web Programming

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



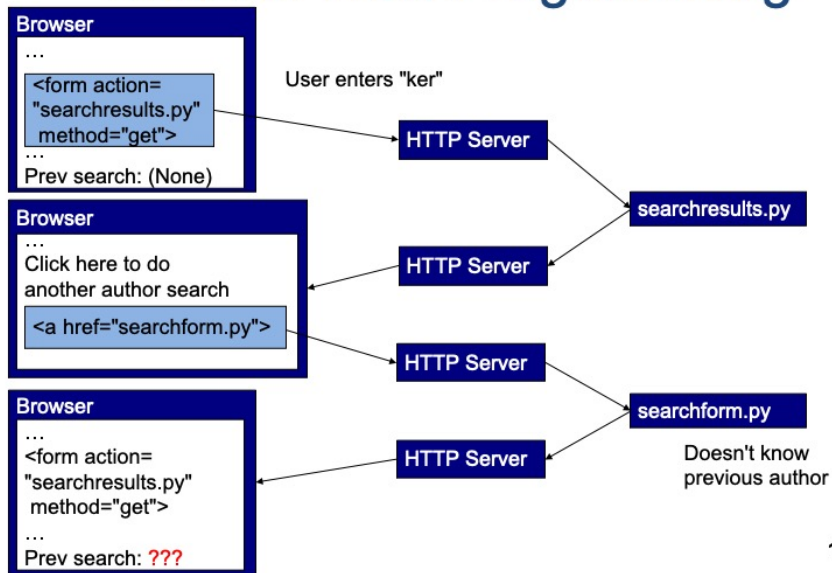
Note



## Stateful Web Programming

- See **PennyCgiState** app (cont.)
  - Displays name of previously-searched-for author in searchform page
  - But how???
  - Application must “remember” previous author name when generating searchform page

# Stateful Web Programming



## Stateful Web Programming

- Generalizing...
- **Problem:**
  - HTTP is a *stateless* protocol
  - Neither browser nor HTTP server remembers previous interactions
  - CGI program exits; so it can't remember either!

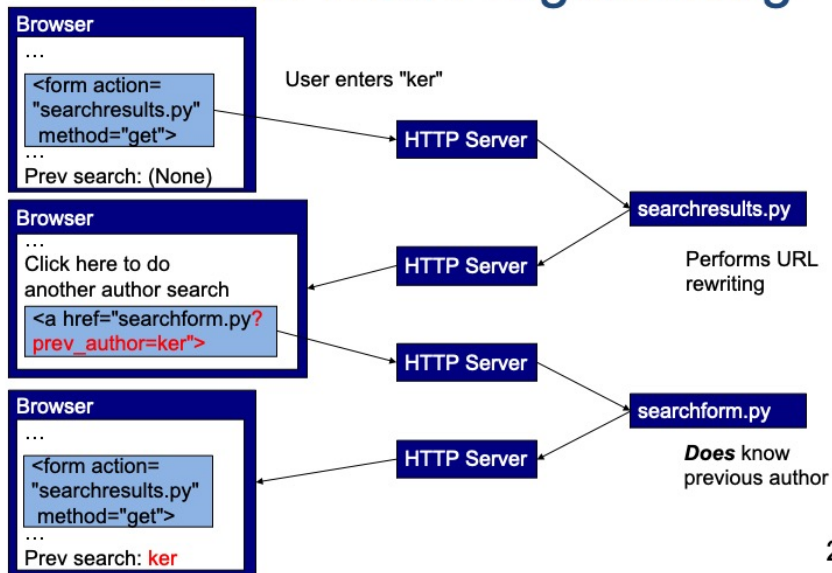
# Stateful Web Programming

- Examples of the need for state
  - Penny app
    - Want to remember previous author
  - Search engine
    - Want to remember previous searches
  - Commercial app
    - Want to remember “shopping cart”
  - Secure app
    - Want to remember that user logged in

# Stateful Web Programming

- **Solution 1: *URL rewriting***
  - Append state data to end of URL

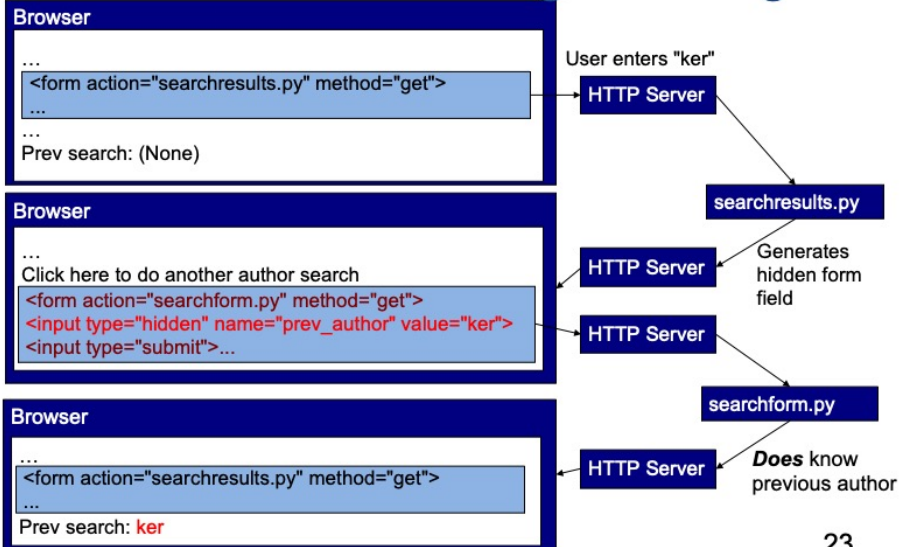
# Stateful Web Programming



# Stateful Web Programming

- **Solution 2: *Hidden form fields***
  - Place state data into form
    - In an `input` element of type `hidden`
    - Browser does not display

# Stateful Web Programming





## Agenda

- Fundamental example
- Stateful web programming
- **Stateful web programming with cookies**
- Redirection in web programming

## Stateful Web Pgmming: Cookies

- **Solution 3: *Cookies***
- HTTP server:
  - Places state in a cookie
  - Passes cookie to browser as header
- And then...

## Stateful Web Pgmming: Cookies

- . Browser:
  - Retains cookie in its memory or client file system
  - When sending subsequent HTTP request **to same app** (same host & directory)...
  - Automatically passes cookie to HTTP server as header

## Stateful Web Pgmming: Cookies

- Cookie attributes:

- Name
- Content
- Host & path
- Expiration date
- ...

27

### Stateful Web Programming

Cookie attributes:

Name

Content

Host & path

The host that sent the cookie

The path/directory where the cookie is active

Default is the directory of page that created the cookie

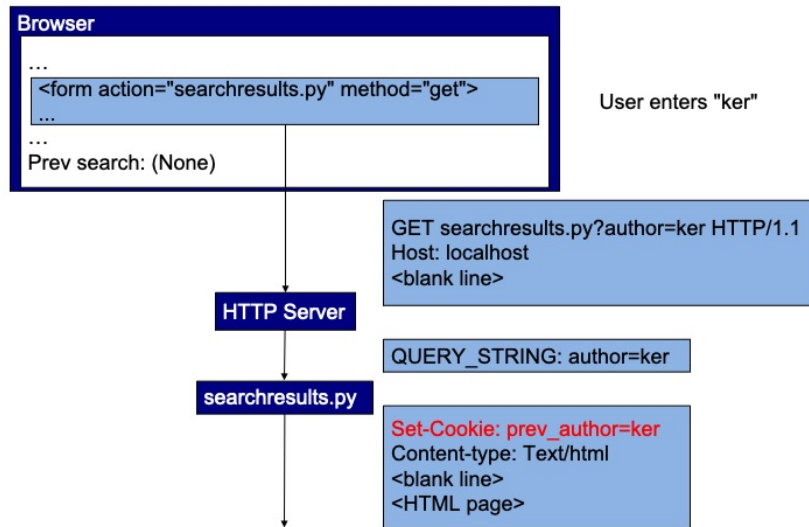
Host & path define the web app that sent the cookie & so should receive the cookie

Expiration date

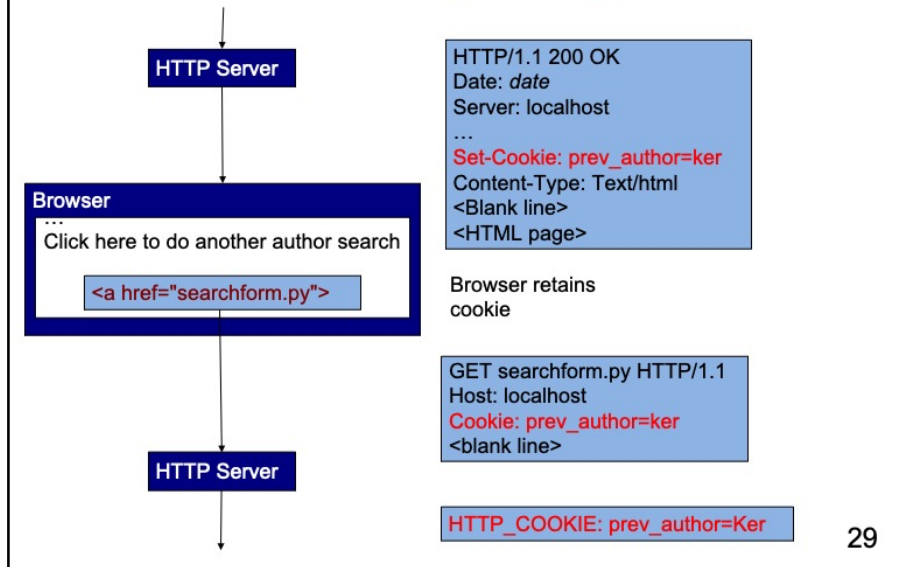
Default is this browser session

Etc.

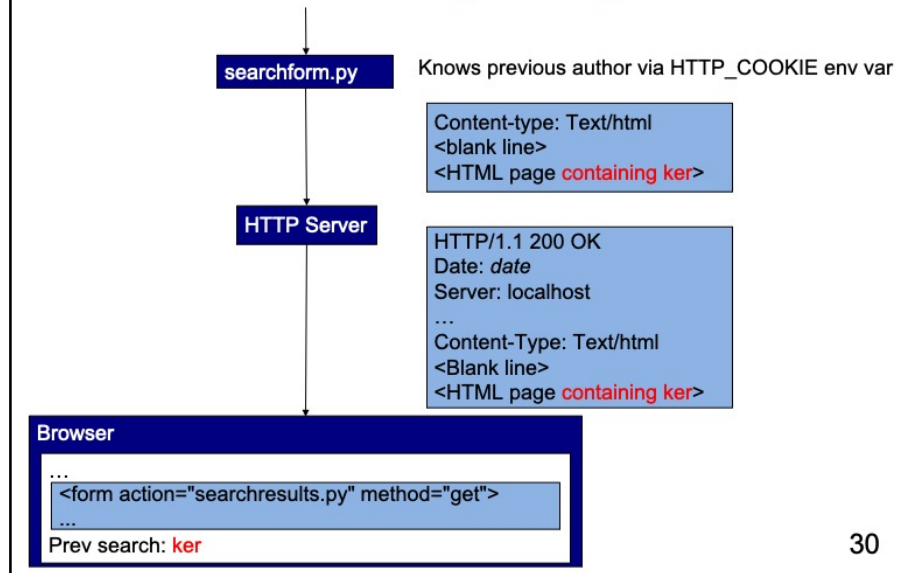
# Stateful Web Pgmming: Cookies



## Stateful Web Pgmming: Cookies



# Stateful Web Pgmming: Cookies



## Stateful Web Pgmming: Cookies

- See **PennyCgiState** app
  - runserver.py
  - penny.sql, penny.sqlite,
  - index.html
  - book.py, database.py
  - **searchresults.py**
  - **searchform.py**

31

### Stateful Web Programming

[see slide]

Code notes: searchresults.py

Asks browser to store prev\_author cookie

Code notes: searchform.py

Uses prev\_author cookie sent by browser



## Stateful Web Pgmming: Cookies

- Viewing cookies in Chrome:
  - From the menu at the upper right...
  - Settings →
  - Advanced →
  - Security and Privacy →
  - Cookies and other site data →
  - See all cookies and site data →
  - localhost

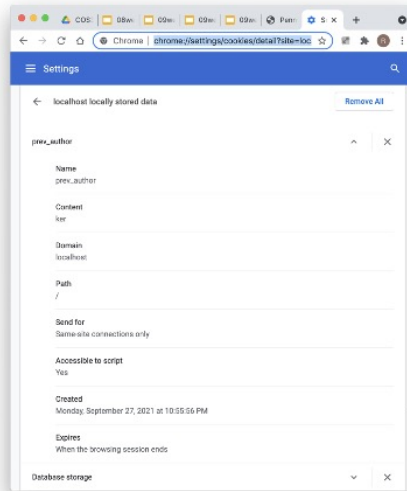
32

### Stateful Web Programming

There's nothing magic about cookies  
You can look at them

[see slide]

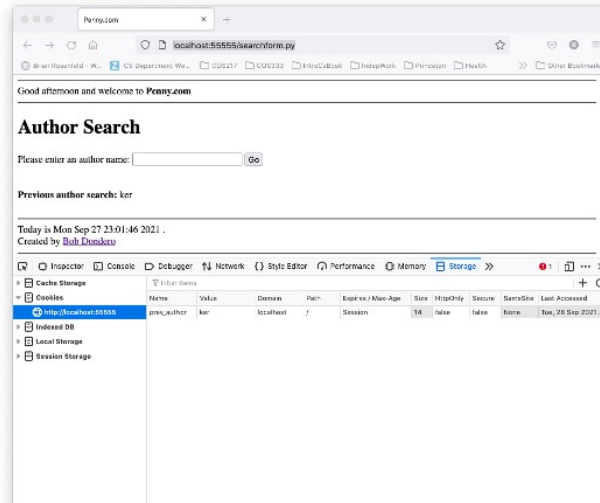
# Stateful Web Pgmming: Cookies



## Stateful Web Pgmming: Cookies

- Viewing cookies in Firefox:
  - From the menu at the upper right...
  - More tools →
  - Web Developer Tools →
  - Storage →
  - Cookies →
  - `http://localhost:5555`

# Stateful Web Pgmming: Cookies



## Stateful Web Pgmming Summary

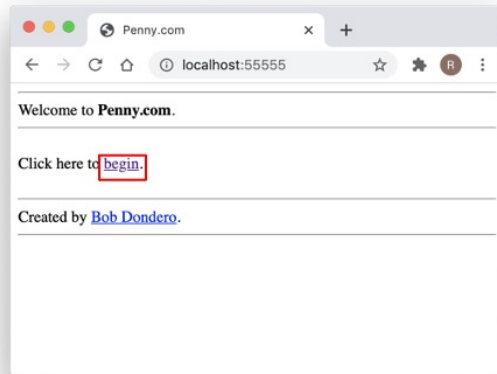
- **Question:** How to do stateful web programming?
- **Answer 1:** URL rewriting
- **Answer 2:** Hidden form fields
- **Answer 3:** Cookies
  - See Appendix for limitations and dangers

## Agenda

- Fundamental example
- Stateful web programming
- Stateful web programming with cookies
- **Redirection in web programming**

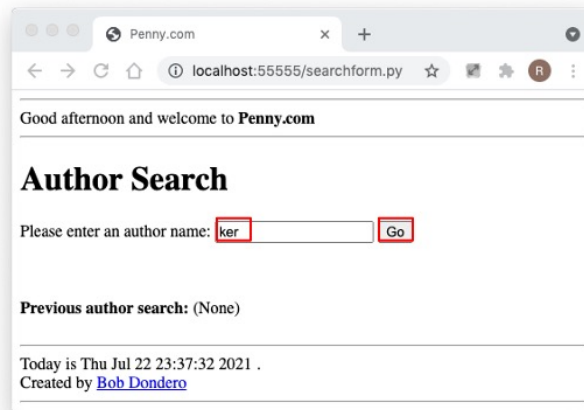
# Redirection

- See **PennyCgiRedirect** app



# Redirection

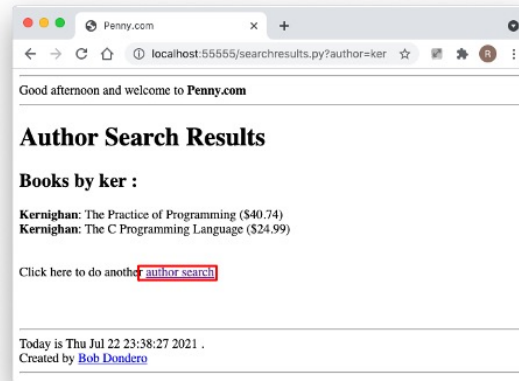
- See **PennyCgiRedirect** app





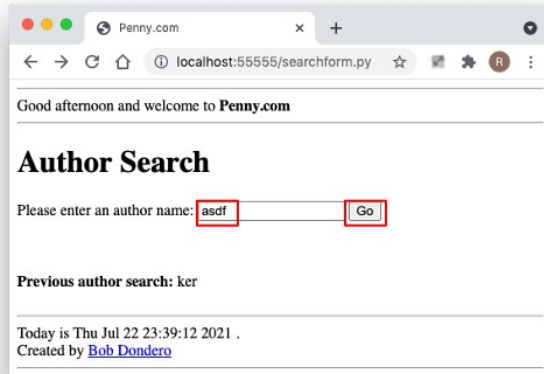
# Redirection

- See **PennyCgiRedirect** app



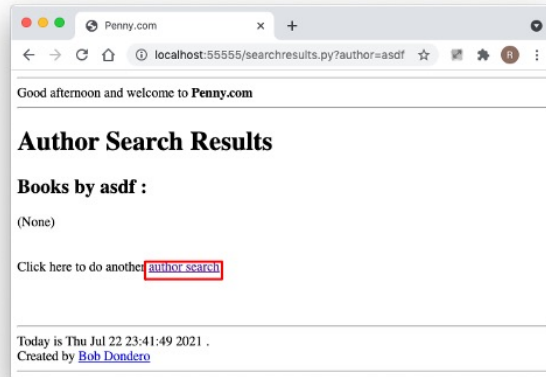
# Redirection

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



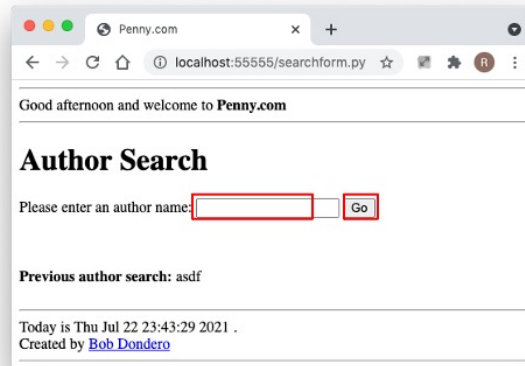
# Redirection

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



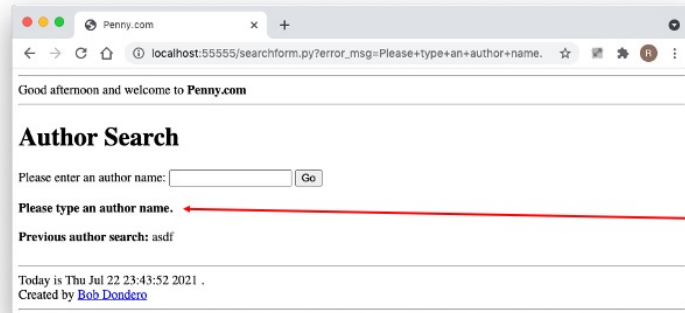
# Redirection

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



# Redirection

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



Note

## Redirection

- See **PennyCgiRedirect** app (cont.)
  - Distinguishes between:
    - Name of author who has no books in database
      - Displays “(none)”
    - Missing author name
      - Causes **redirection** back to searchform page
      - Displays error message, prompting user to enter a name

## Redirection

- Generalizing...
- **Problem:** Often web app must *redirect* to a page other than requested one
  - Original page
  - Login page
  - Error page
  - ...

# Redirection

- **Solution:** *HTTP redirection*

- Server sends headers to browser:

```
Status: 307 Temporary Redirect  
Location: newURL  
<blank line>
```

- Browser accepts headers, requests newURL



## Redirection

- See **PennyCgiRedirect** app
  - runserver.py
  - penny.sql, penny.sqlite
  - index.html
  - book.py, database.py
  - common.py
  - **searchresults.py**
  - **searchform.py**

48

### Redirection

[see slide]

In PennyCgiRedirect, there are ways of implementing the desired functionality without using redirection

In other applications (e.g., those that do authentication), redirection is unavoidable

## Aside: HTTP Status Codes

HTTP Status Code	Description
200 OK	Standard response for successful HTTP requests.
307 Temporary Redirect	The request should be repeated with another URL; however, future requests should still use the original URL.
404 Not Found	The requested resource could not be found but may be available in the future. Subsequent requests by the client are permissible.
...	...

Descriptions from:  
[https://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_status\\_codes](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes)

## Summary

- We have covered:
  - Stateful web programming
  - Redirection in web programming

50

### **Summary**

In summary of this lecture...

[see slide]

## Summary

- We have covered:
  - CGI programming
  - CGI using the HTTP GET method
  - CGI using the HTTP POST method
  - GET vs. POST
  - Stateful web programming
  - Redirection in web programming
- See also:
  - **Appendix 1: Cookie Dangers**

51

### Summary

In summary of the sequence of lectures on CGI Web Programming...

[see slide]

## Appendix 1: Cookie Dangers

## Cookie Dangers

- **Problem:**

- Cookie size is limited to 4K

- **Solution:**

- Cookie content stored on server-side (in database), indexed by a unique key
  - Cookie contains key only
  - See *sessions* as implemented in frameworks and PHP

## Cookie Dangers

- **Problem:**
  - User may disable browser cookies
- **Solution:**
  - Ask the user to enable them! Or...
  - Add more logic

54

### Cookie Dangers

#### Problem:

User may disable browser cookies

#### Solution:

Ask the user to enable them! Or...

Add more logic

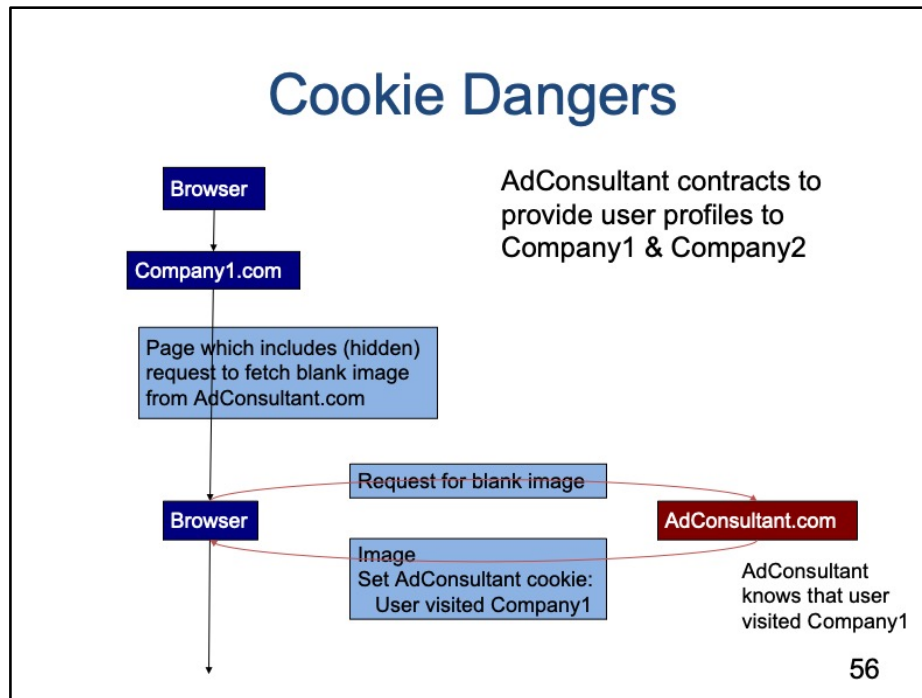
Fall back to URL rewriting or hidden form fields if necessary

## Cookie Dangers

- **Problem:**

- *Third-party cookies* can invade privacy
- (See [https://en.wikipedia.org/wiki/Cookie\\_stuffing](https://en.wikipedia.org/wiki/Cookie_stuffing))



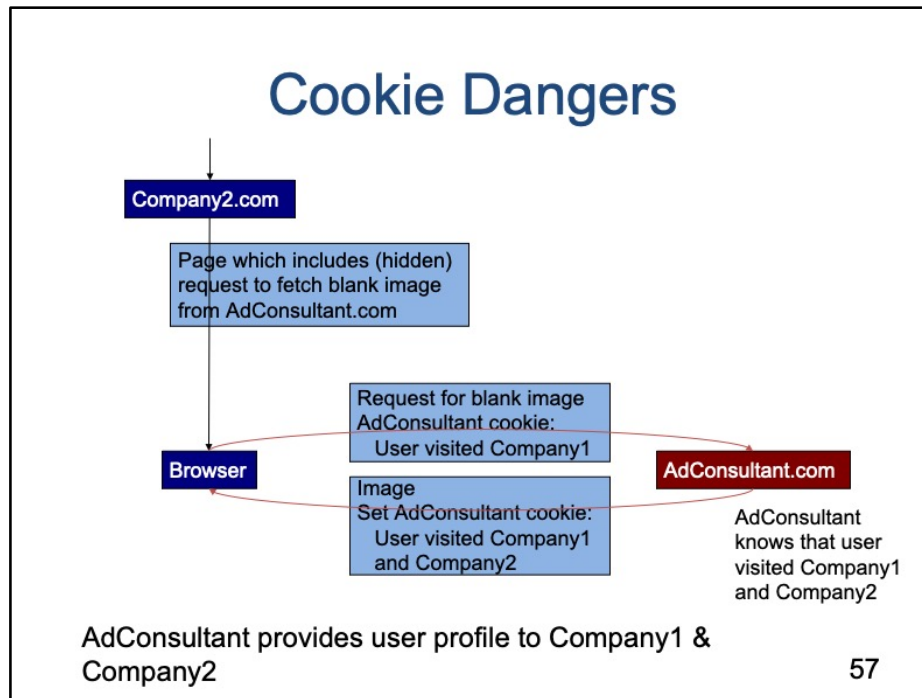


## Cookie Dangers

### Sequence

AdConsultant contracts with Company1 and Company2 and ...  
 If you pay me some money, then I'll provide you with a (partial) browsing history of people who visit your website

UserX browses to Company1.com  
 Browser sends request to Company1.com  
 HTTP server sends to browser results page which includes a hidden (one-pixel) image element, requesting fetch of blank image from AdConsultant.com  
 Browser sends request to AdConsultant.com  
 AdConsultant.com now knows that UserX visited Company1.com  
 Creates cookie "UserX visited Company1.com"  
 Sends (one-pixel) image element to browser, along with cookie



## Cookie Dangers

### Sequence (cont.)

UserX browses to Company2.com  
 Browser sends request to Company2.com  
 HTTP server sends to browser results page which includes a hidden (one-pixel) image element, requesting fetch of blank image from AdConsultant.com  
 Browser sends request to AdConsultant.com  
     Request includes cookie: "UserX visited Company1.com"  
 AdConsultant.com now knows that UserX visited Company1.com previously, and Company2.com now  
     Creates cookie "UserX visited Company1.com; UserX visited Company2.com"  
     Sends (one-pixel) image element to browser, along with cookie

### Eventually

AdConsultant provides profile of UserX to Company1 & Company2  
 UserX visited Company1.com and Company2.com, and ...

Invasion of privacy???

## Cookie Dangers

- **Solution:** Tell browser to refuse third-party cookies
  - In Chrome:
    - Settings → Advanced → Privacy & security → Content settings → Cookies → Block third-party cookies
  - In Firefox:
    - Preferences → Privacy & Security icon → Firefox will → Accept third-party cookies and site data → Never

58

### Cookie Dangers

[see slide]

But beware that refusing third-party cookies disables CAS authentication  
I've never been able to keep that setting