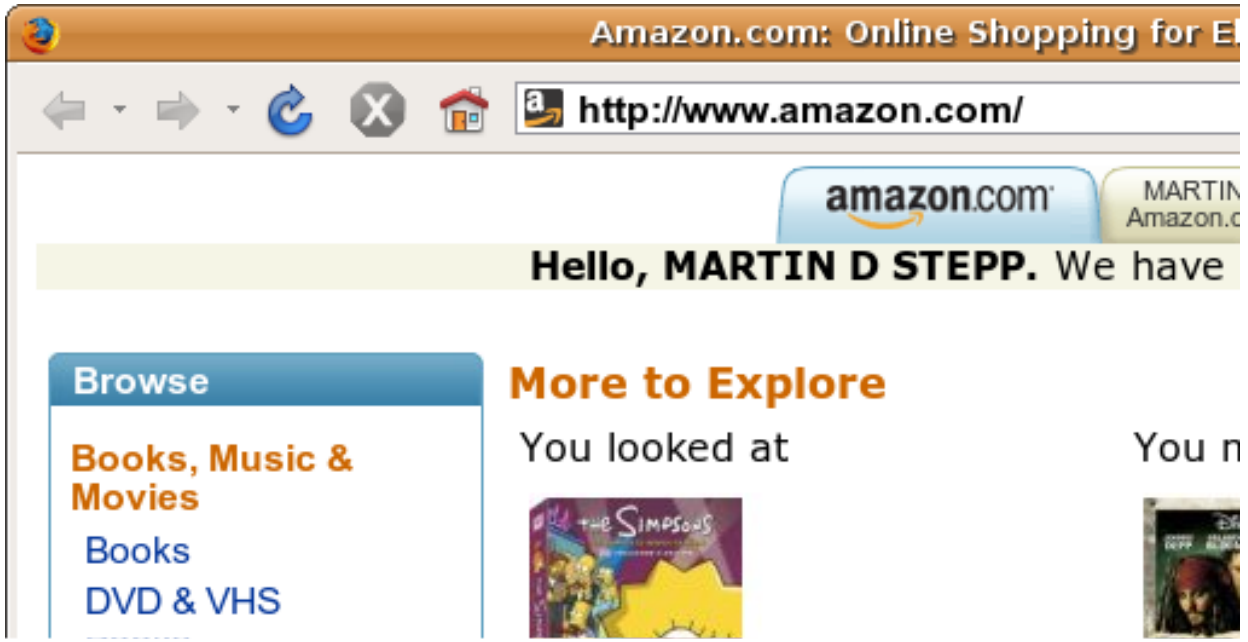
CSC 337

Cookies

Marty Step, Allison Oburne

Stateful client/server interaction with Cookies

- *Sites like amazon.com seem to "know who I am." How do they do this? How does a client uniquely identify itself to a server, and how does the server provide specific content to each client?*



- HTTP is a **stateless** protocol; it simply allows a browser to request a single document from a web server

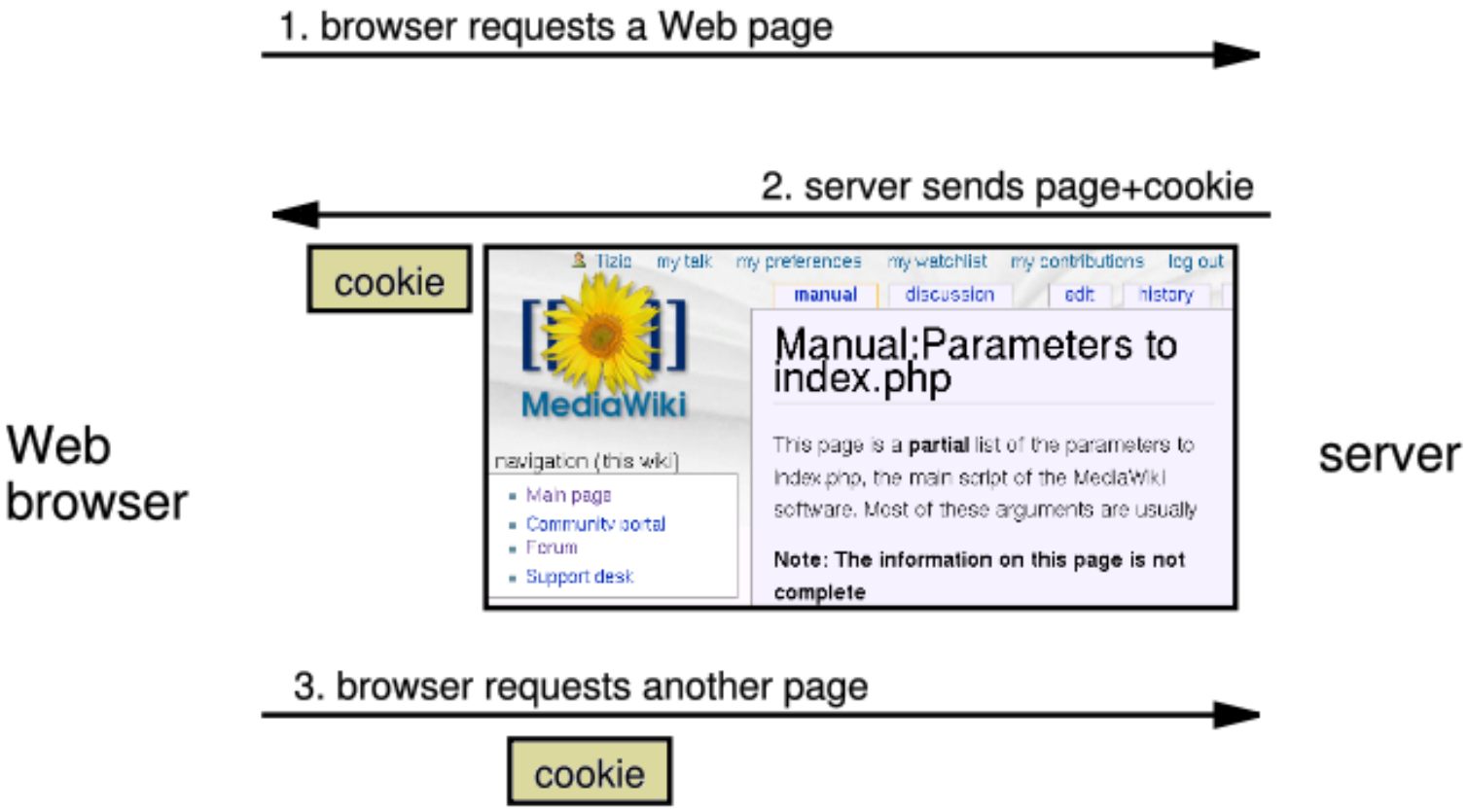
- Pieces of data called **cookies** are one way to work around this problem, which are used as the basis of higher-level **sessions** between clients and servers

What is a cookie?

- **Cookie**: a small amount of information sent by a server to a browser, and then sent back by the browser on future page requests
- Cookies have many uses:
  - authentication
  - user tracking
  - maintaining user preferences, shopping carts, etc.
- A cookie's data consists of a single name/value pair, sent in the header of the client's HTTP GET or POST request

How cookies are sent

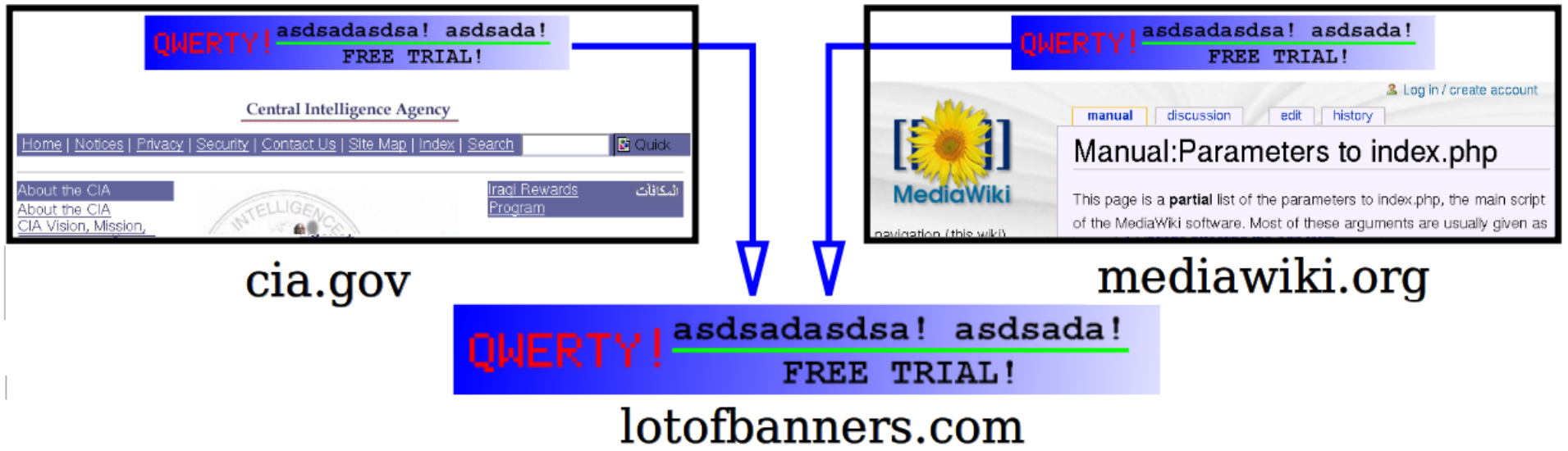
- When the browser requests a page, the server may send back a cookie(s) with it
- If your server has previously sent any cookies to the browser, the browser will send them back on subsequent requests
- Alternate model: client-side JavaScript code can set/get cookies



Myths about cookies

- Myths:
  - Cookies are like worms/viruses and can erase data from the user's hard disk.
  - Cookies are a form of spyware and can steal your personal information.
  - Cookies generate popups and spam.
  - Cookies are only used for advertising.
- Facts:
  - Cookies are only data, not program code.
  - Cookies cannot erase or read information from the user's computer.
  - Cookies are usually anonymous (do not contain personal information).
  - Cookies CAN be used to track your viewing habits on a particular site.

A "tracking cookie"



- An advertising company can put a cookie on your machine when you visit one site, and see it when you visit another site that also uses that advertising company
- Therefore they can tell that the same person (you) visited both sites
- Can be prevented by setting your browser not to accept "third-party cookies"

Where are the cookies on my computer?

- IE: *HomeDirectory*\Cookies
  - e.g. C:\Documents and Settings\jsmith\Cookies
  - each is stored as a `.txt` file similar to the site's domain name
- Chrome:

  C:\Users*username*\AppData\Local\Google\Chrome\User Data\Default
- Firefox: *HomeDirectory*\.mozilla\firefox*???*.default\cookies.txt
  - view cookies in Firefox preferences: Privacy, Show Cookies...
- See cookie settings in Chrome

  1. On your computer, open Chrome.
  2. At the top right, click More ⋮ > **Settings**.
  3. At the bottom, click **Advanced**.
  4. Under "Privacy and security," click **Content settings**.
  5. Click **Cookies**.

How long does a cookie exist?

- **Session cookie**: the default type; a temporary cookie that is stored only in the browser's memory
  - when the browser is closed, temporary cookies will be erased
  - can not be used for tracking long-term information
  - safer, because no programs other than the browser can access them
- **Persistent cookie**: one that is stored in a file on the browser's computer
  - can track long-term information
  - potentially less secure, because users (or programs they run) can open cookie files, see/change the cookie values, etc.

Setting a cookie in PHP

```php
setcookie("name", "value");
```

```php
setcookie("username", "allllison");
setcookie("age", 19);
```

- `setcookie` causes your script to send a cookie to the user's browser
- `setcookie` must be called before any output statements (HTML blocks, `print`, or `echo`)
- We can set multiple cookies (20-50) per user, each up to 3-4K bytes
- By default, the cookie expires when browser is closed (a "session cookie")

Retrieving information from a cookie

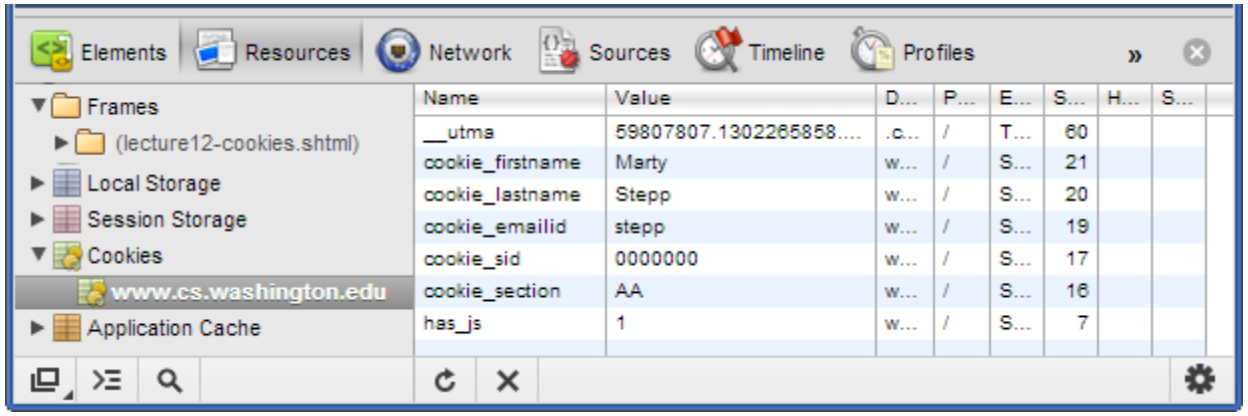
```
$variable = $_COOKIE["name"];    # retrieve
```

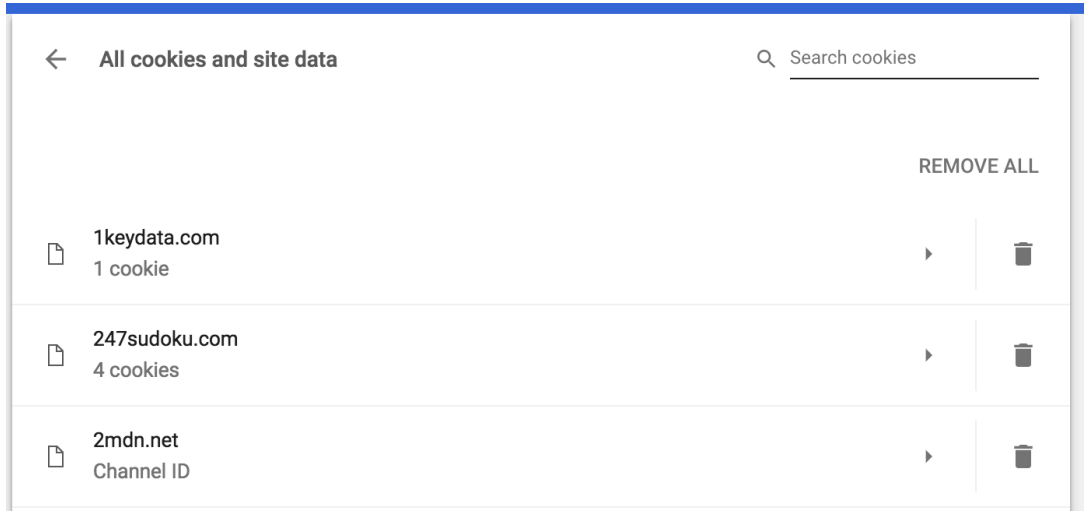
```php
if (isset($_COOKIE["username"])) {
  $username = $_COOKIE["username"];
  print("Welcome back, $username.\n");
} else {
  print("Never heard of you.\n");
}
print("All cookies received:\n");
print_r($_COOKIE);
```

- Any cookies sent by client are stored in `$_COOKIES` associative array
- Use `isset` function to see whether a given cookie name exists

What cookies are stored on your computer?

- **Chrome:** F12 → Resources → Cookies; **Firefox:** F12 → Cookies





Expiration / persistent cookies

```
setcookie("name", "value", expiration);                    PHP
```

```
$expireTime = time() + 60*60*24*7;    # 1 week from now
setcookie("CouponNumber", "389752", $expireTime);
setcookie("CouponValue", "100.00", $expireTime);           PHP
```

- To set a persistent cookie, pass a third parameter for when it should expire
- Indicated as an integer representing a number of seconds, often relative to current timestamp
- If no expiration passed, cookie is a session cookie; expires when browser is closed
- `time` function returns the current time in seconds
  - `date` function can convert a time in seconds to a readable date

Deleting a cookie

```php
setcookie("name", FALSE);
```

```php
setcookie("CouponNumber", FALSE);
```

- Setting the cookie to `FALSE` erases it
- You can also set the cookie but with an expiration that is before the present time:

```php
setcookie("count", 42, time() - 1);
```

- Remember that the cookie will also be deleted automatically when it expires, or can be deleted manually by the user by clearing their browser cookies

Clearing cookies in your browser

- **Chrome:** Wrench → History → Clear all browsing data...
- **Firefox:** Firefox menu → Options → Privacy → Show Cookies... → Remove (All) Cookies