FRESNO STATE

Discovery. Diversity. Distinction.

Web Programming (CSci 130)

Department of Computer Science

College of Science and Mathematics

California State University Fresno

H. Cecotti

Learning outcomes

- **Goal**
  - HTTP cookie
    - Definition
  - Cookies
    - with Javascript
    - with PHP
  - Web storage with HTML5
    - Local and session

  - By the end of the week, you should be able to use cookies and local storage in webpages

Rationale

- Web page are accessed several times by users
  - Dynamic content, content adapted to the user
  - New connection → New settings … ?
- A mechanism for storing data in the remote browser and therefore tracking or identifying return users
  - → cookie
  - … but we have PHP ?
    - POST/GET

Introduction

- **HTTP cookie**
  - Alias: web cookie, browser cookie, cookie
  - Piece of data (plain text)
    - Sent **from** the website (server)
    - Stored on the user’s computer (client)
      - By the browser
  - Specifications
    - Browsers should support
      - Cookies <= 4096 bytes , 50 cookies / domain, at least 3000 cookies.
  - Cookies can be found on disk and in process memory
- **Goal**
  - To remember information
    - Connection to a website
    - Number of elements, type of elements in a shopping cart
      - → Favorite items
    - History of browsing activity

Cookie

- **Data record**
  - Expires
    - Date the cookie will expire
      - Blank = cookie expires at the end of the session
  - Domain
    - Domain name of the site
      - Example: Localhost
  - Path
    - Path to the directory setting the cookie
  - Secure
    - “secure” → https
  - Name=value
    - Set/retrieved: a pair: key + value

Cookie use

- Shopping cart
  - Once upon a time
    - Cookie = **client**-side storage
      - Designed for CGI (common gateway interface) programming
  - Now
    - Database on a **server**
  - Underlying challenge
    - To define what is transferred between the client and the server
- Examples
  - To tell if 2 requests come from the same browser
  - Keep a user logged-in

Cookie definition

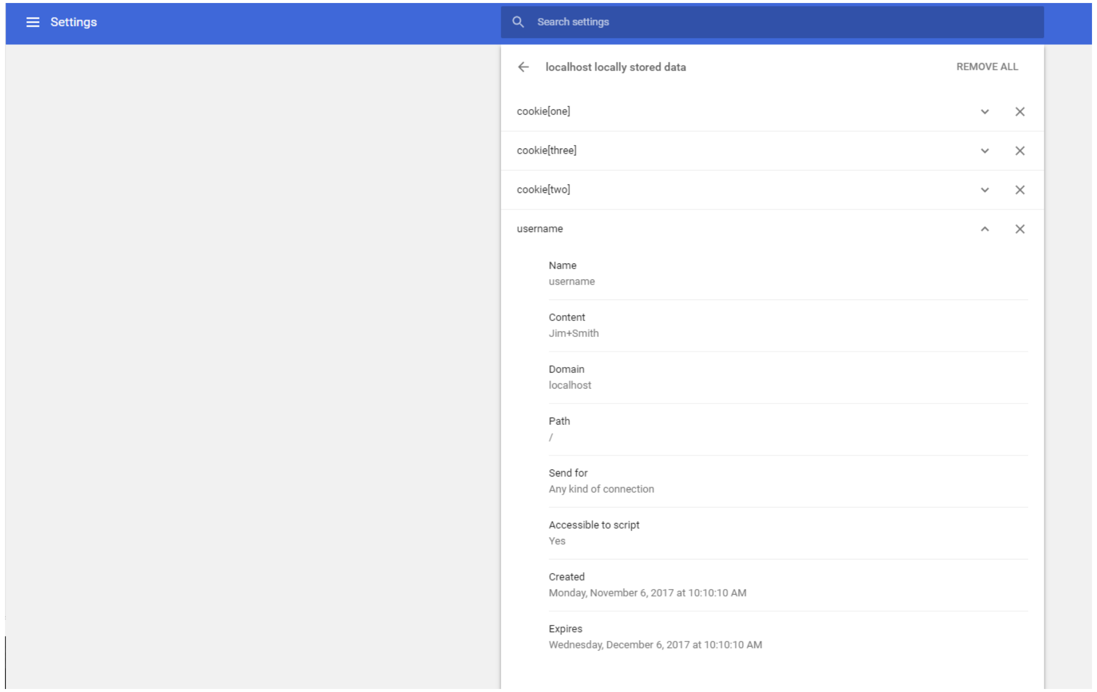
- Cookies are a mechanism for storing data in the remote browser and thus tracking or identifying return users.
  - http://php.net/manual/en/features.cookies.php

Types of cookies

- **Session**
  - In memory cookie (transient cookie)
  - Only in temporary memory while being on the website
  - close the browser → delete the cookie
  - No expiration date
- **Persistent**
  - Expires at a particular date: Information from the cookie is transferred to the server at each visit
  - Tracking cookies: used for ads
- **Secure**
  - Transmitted through HTTPS only + secure flag to the cookie
- **HttpOnly**
  - Cannot be accessed by client side API (e.g. JS) + httponly flag to the cookie
- **Same site**
  - Introduced by Chrome
  - Rationale: avoid cross site request forgery (session riding)
  - Idea: cookie can be sent only to requests from the same origin as the target domain

Where are they?

- **Chrome**:
  - ➢Settings, privacy and security, content settings, cookies
    - ○ chrome://settings/content/cookies
      - Listed in alphabetical order
  - ➢Your cookies:
    - ○ **localhost**



localhost locally stored data

cookie[one]

cookie[three]

cookie[two]

username

Name
username

Content
Jim+Smith

Domain
localhost

Path
/

Send for
Any kind of connection

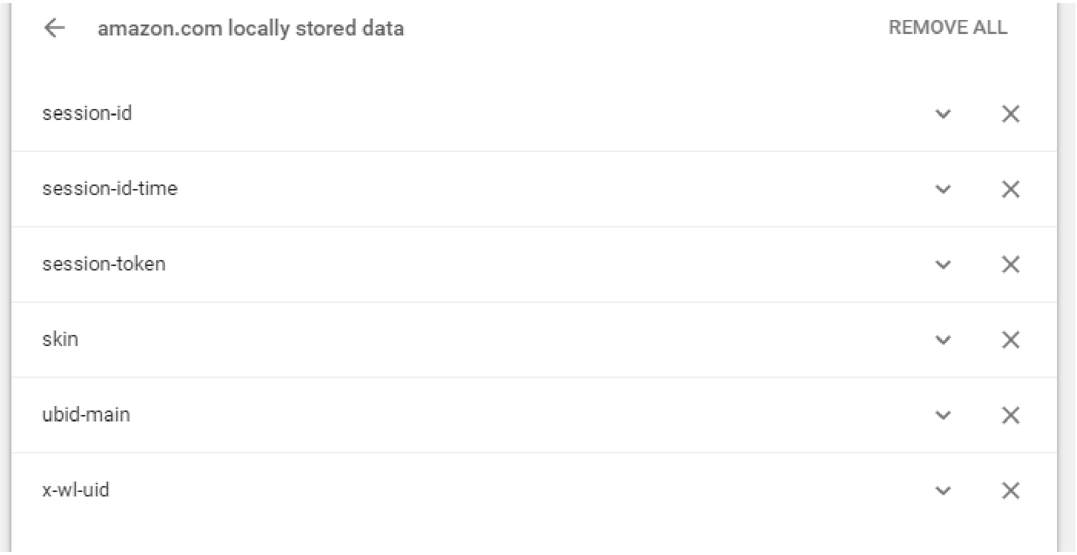
Accessible to script
Yes

Created
Monday, November 6, 2017 at 10:10:10 AM

Expires
Wednesday, December 6, 2017 at 10:10:10 AM

Example

- Amazon



amazon.com locally stored data

session-id

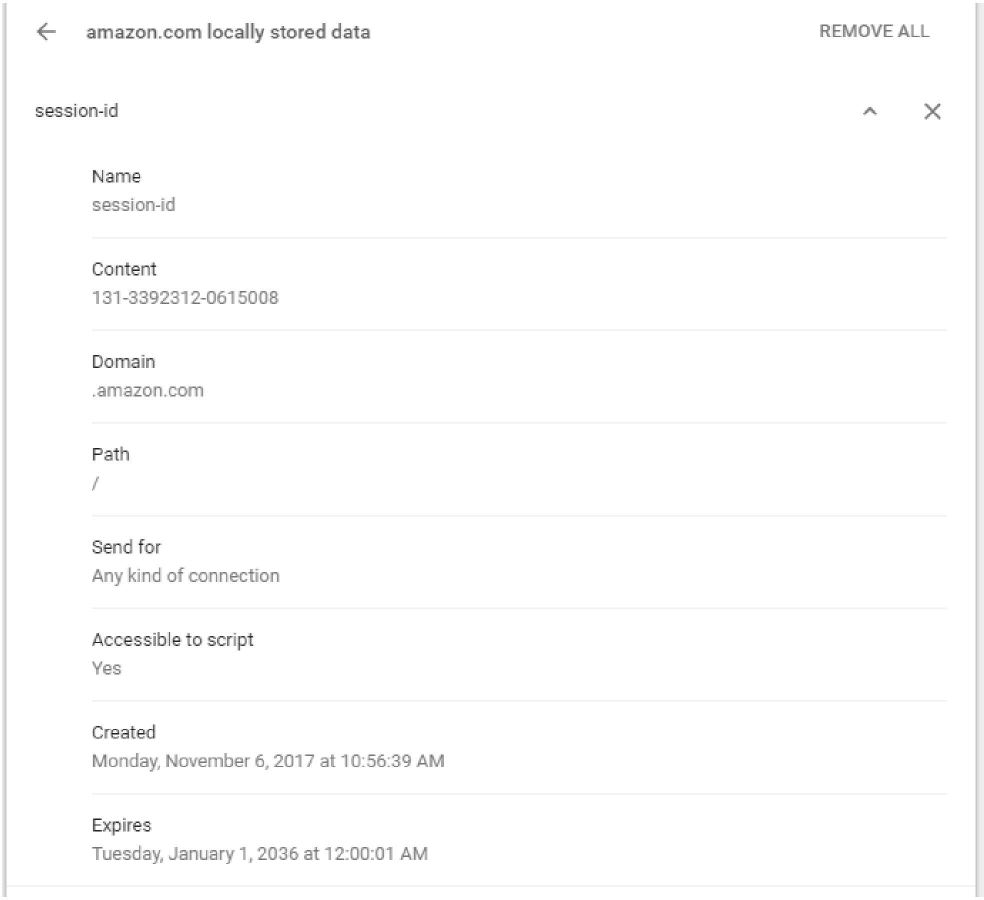
session-id-time

session-token

skin

ubid-main

x-wl-uid



amazon.com locally stored data

session-id

Name
session-id

Content
131-3392312-0615008

Domain
.amazon.com

Path
/

Send for
Any kind of connection

Accessible to script
Yes

Created
Monday, November 6, 2017 at 10:56:39 AM

Expires
Tuesday, January 1, 2036 at 12:00:01 AM

Functions to create/retrieve cookies

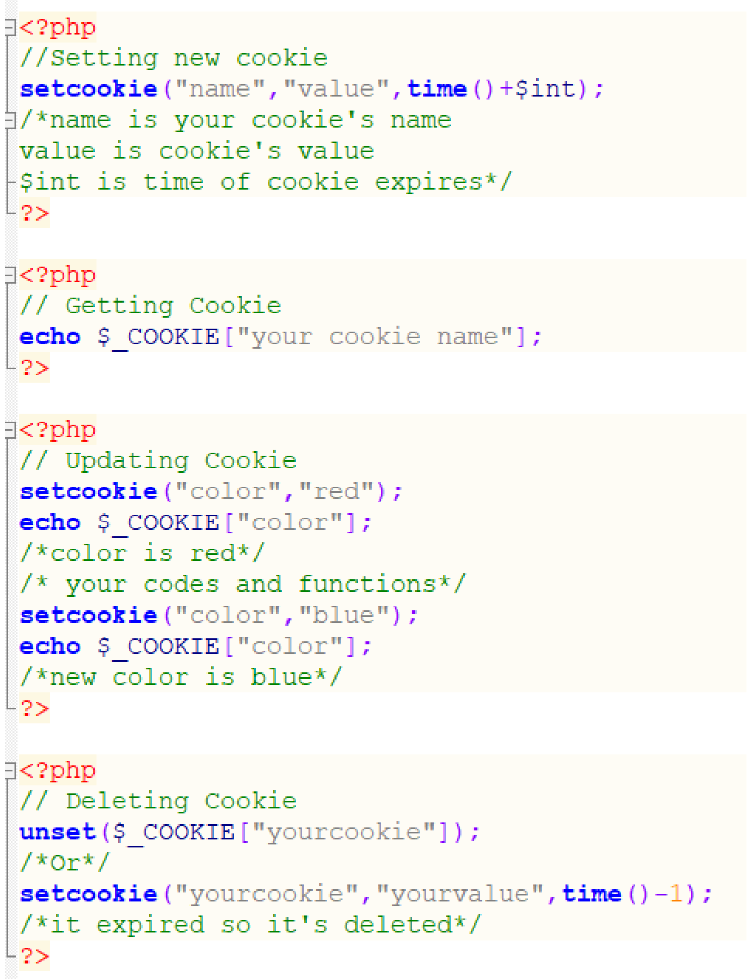
- **JS**
  - example_cookie.html
    - document.cookie
      - String: “name1=value1; expires=date-information;”
      - WRITE pair by pair for the cookies !
- **PHP**
  - example_cookie.php
    - $_COOKIE[$cookie_name]

Cookies with JS

- Read a cookie
  - ➢var x=document.cookie;
- Change a cookie
  - ➢document.cookie = "username=Kawhi Leonard; expires=Fri, 18 July 2020 12:00:00 UTC; path=/";
- Delete a cookie
  - ➢document.cookie = "username=; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/;";

Cookies with PHP

- Examples



```php
<?php
//Setting new cookie
setcookie("name","value",time()+$int);
/*name is your cookie's name
value is cookie's value
$int is time of cookie expires*/
?>

<?php
// Getting Cookie
echo $_COOKIE["your cookie name"];
?>

<?php
// Updating Cookie
setcookie("color","red");
echo $_COOKIE["color"];
/*color is red*/
/* your codes and functions*/
setcookie("color","blue");
echo $_COOKIE["color"];
/*new color is blue*/
?>

<?php
// Deleting Cookie
unset($_COOKIE["yourcookie"]);
/*Or*/
setcookie("yourcookie","yourvalue",time()-1);
/*it expired so it's deleted*/
?>
```

Tracking activity

- Web Applications and Services use cookies to authenticate sessions and users
  - **Advantages**
    - No need to re-type, re-search information in a website
    - Predefined personalized content
  - **Disadvantages**
    - Big brother
      - In Europe, law to force websites to tell they re using cookies
    - Steal cookie files
      - Session hijacking / cookie hijacking
        - To gain unauthorized access to information or services in a computer system
      - String with information from the user
      - == steal identification

Tracking activity

- Pass the cookie
  - **Attack**
    - Get the cookie from the victims browser or other processes
      - process dump, or accessing the cookie storage on disk
    - Exfiltration of the necessary authentication cookies
    - Open browser
    - Navigate to the resource to access
      - Domain the cookie is valid for
    - Use the Developer Console
    - Set the cookie
      - document.cookie="key=value"
    - Refresh the page and observe being logged in as the victim.

- Pass the cookie
  - **Detection**
    - Monitor on the client side for applications that perform
      - process dumps on browser processes or others.
    - Monitor for unusual activity on critical web assets
      - cloud provider management consoles
    - Monitor for login anomalies (location, time, unusual access patterns)
    - Leverage features that cloud providers and web apps provide
      - Threat Detection, Access logs
    - Perform authorized adversarial emulation in the company to test detections

Alternative to cookies

- JSON web tokens
  - Access tokens
  - Compact
- Tracking
  - **PHP**
    - GET (URL string)
    - POST (http request body)
  - **IP address**
    - Obtained from the server side

Web storage

- **session**Storage
  - Keep a separate storage area for each given origin
    - available for the duration of the page session
      - the browser is open + page reloads and restores
  - Window.sessionstorage (in JS)
- **local**Storage
  - Same as session storage
    - **but** persists even if the browser is closed and reopened.
  - Window.localstorage (in JS)

Web storage

- HTML5 web storage
    - It stores data locally
- Features
    - Per domain and protocol
    - Information not transfer to the server
    - Space
        - Better than cookies (5 mb)
- See example
    - storage_support.html
- Send local storage info to server
    - Local storage → JS variable → AJAX

Conclusion

- Cookies are used for
  - Session management
    - Logging, shopping cart
  - Personalization
    - Theme of the website (choice of css), general user preference
  - Tracking
    - To record and analyze the behaviors of the visitors of the webpage
- Important to have **personalized** websites
  - Worst case: no personalization
  - Cookie
  - Login system with a profile saved in a file
    - Login + Password
- Examples and links
  - On Canvas

Questions ?

Further reading

- Cookies
  - https://www.w3.org/TR/csp-cookies/
  - https://developer.mozilla.org/en-US/docs/Web/API/Document/cookie
- JSON Web Token
  - https://jwt.io/
    - We will come back to it for user identification