Client Side Technologies

JavaScript Cookies

ITI – Assiut Branch
Eng. Hany Saad

Cookies

❑What're cookies?

- Cookies are small text strings that you can store on the computers of people that visit your Web site.
- Cookies were originally invented by Netscape to give 'memory' to web servers and browsers.
- Normally, cookies are simple variables set by the server in the browser and returned to the server every time the browser accesses a page on the same server.
- Cookie is a method of storing information locally in the browser and sending it to the server whenever the appropriate pages are requested by the user.
- A cookie is not a script, it is a mechanism of the HTTP server accessible by both the client and the server.
- A cookie may be written and accessed by a script but the cookies themselves are simply passive text strings.

Cookies (Cont.)

- **Why do we need cookies?**
  - The HTTP protocol, is responsible for:

    → arranging the transfer of web pages to your browser and
    → browser requests for pages to servers.

  - HTTP protocol is a state-less, which means that once the server has sent a page to a browser requesting it, it doesn't remember a thing about it
  - HTTP server has no information in a request to tie it to any other request.
  - The data in a response is based only on the information the client sends in the request.
  - So if you come to the same web page a second, third, hundredth or millionth time, the server once again considers it the very first time you ever came there.
  - Cookies are a method for a server to ask the client to store arbitrary data for use in future connections "to keep state information".

Benefits of Cookies

❑ **Benefits of Cookies:**

- Shopping carts
  - → By storing data as you move from one page (or frame) to another.
- Authentication
  - → No longer need to enter password
  - → Greeting people by name.
- Maintaining state
  - → Adventure games that use cookies to keep track of pertinent character data and the current state of the game.
- Saving time for returning visitors
  - → The user does not have to re-enter information
- Notifying visitor
  - → What has changed since their last visit.

Cookies Limitations

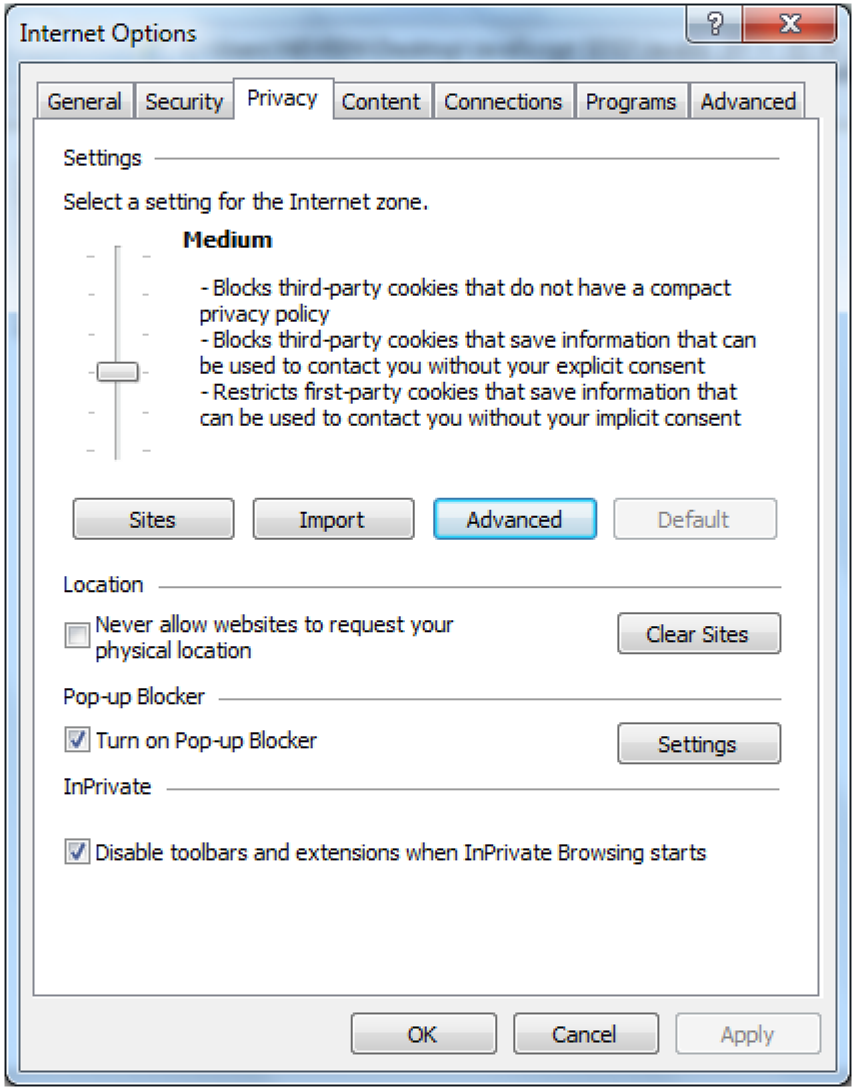
❑Cookies Limitations:

- All browsers are required to be able to store at least 300 cookies, after which automatic deletion based on expiry date and usage.
- Each individual domain name can store 20 cookies.
- each cookie having a maximum size of (4k).

Security Limitations & Facts

- **Cookies security limitations & Facts:**
  - Highly unreliable, from a programming perspective
    - → It's like having your data stored on a hard drive that sometimes will be missing, corrupt, or missing the data you expected.
  - Cookies just identify the computer being used not the individual using the computer.
  - A server can set, or deposit, a cookie only if a user visits that particular site.
    - → i.e. one domain cannot deposit a cookie for another, and cross-domain posting is not possible.
  - only the originating domain can ever use the contents of your cookie "Same-origin policy".
  - Any user can choose which cookies to accept by adjusting the settings in the browser.

Controlling Cookies in the Browser


Internet Options

General | Security | Privacy | Content | Connections | Programs | Advanced

Settings

Select a setting for the Internet zone.

Medium

- Blocks third-party cookies that do not have a compact privacy policy
- Blocks third-party cookies that save information that can be used to contact you without your explicit consent
- Restricts first-party cookies that save information that can be used to contact you without your implicit consent

Sites | Import | Advanced | Default

Location

[ ] Never allow websites to request your physical location — Clear Sites

Pop-up Blocker

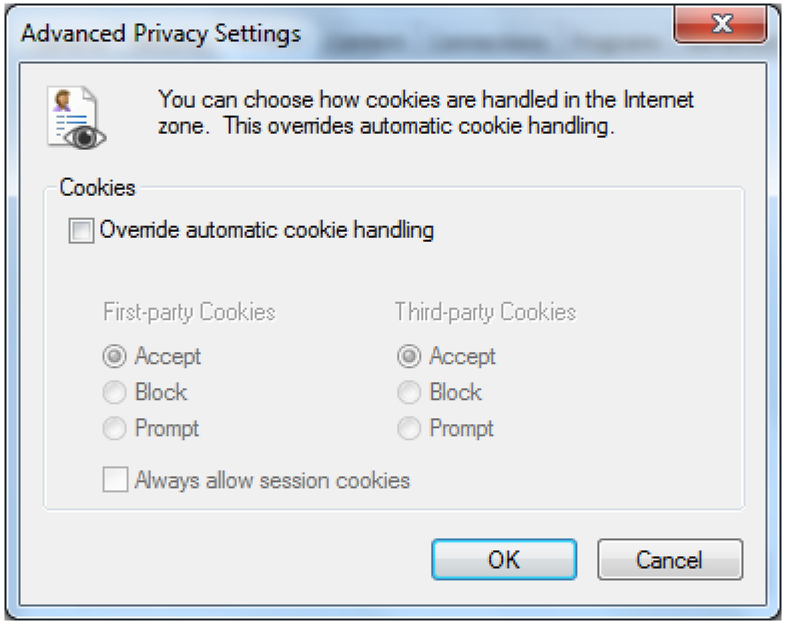
[✓] Turn on Pop-up Blocker — Settings

InPrivate

[✓] Disable toolbars and extensions when InPrivate Browsing starts

OK | Cancel | Apply

MSIE


Advanced Privacy Settings

You can choose how cookies are handled in the Internet zone. This overrides automatic cookie handling.

Cookies

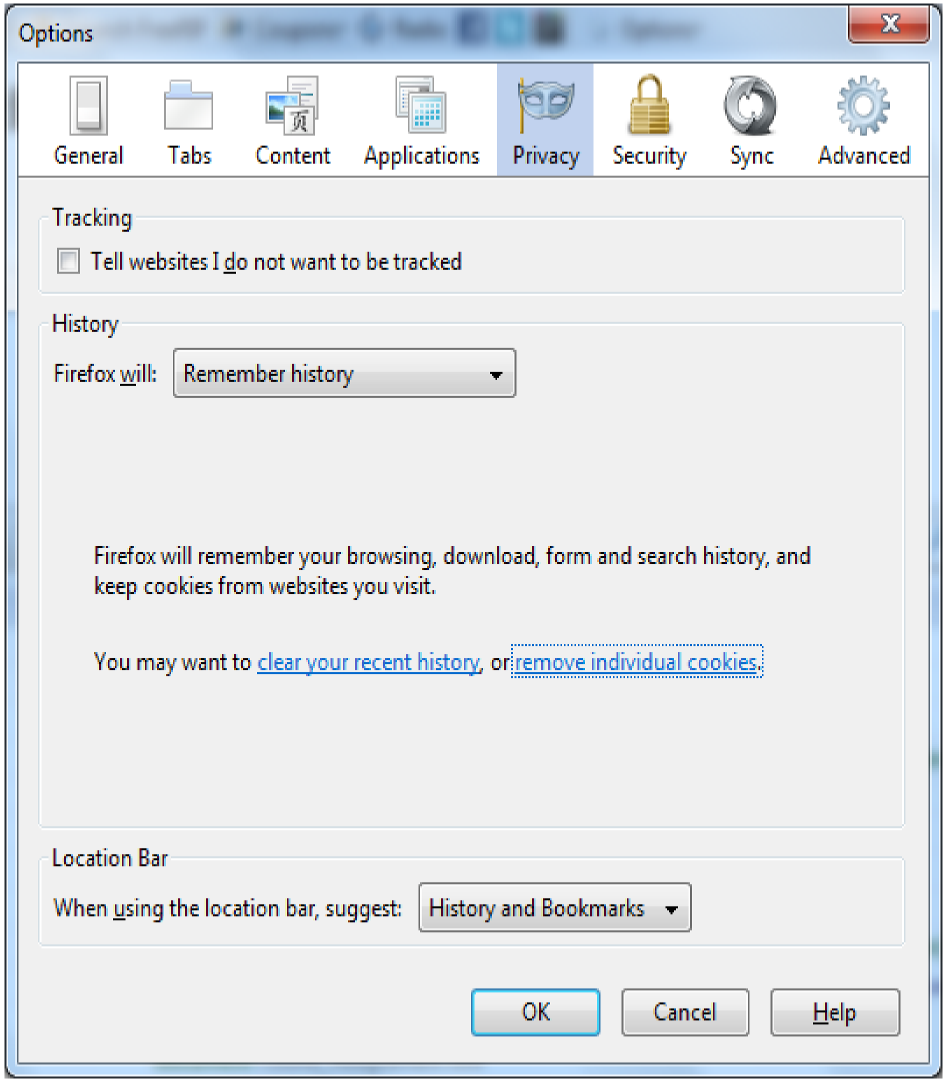
[ ] Override automatic cookie handling

First-party Cookies	Third-party Cookies
● Accept	● Accept
○ Block	○ Block
○ Prompt	○ Prompt

[ ] Always allow session cookies

OK | Cancel

Controlling Cookies in the Browser



Tracking

☐ Tell websites I do not want to be tracked

History

Firefox will: Remember history

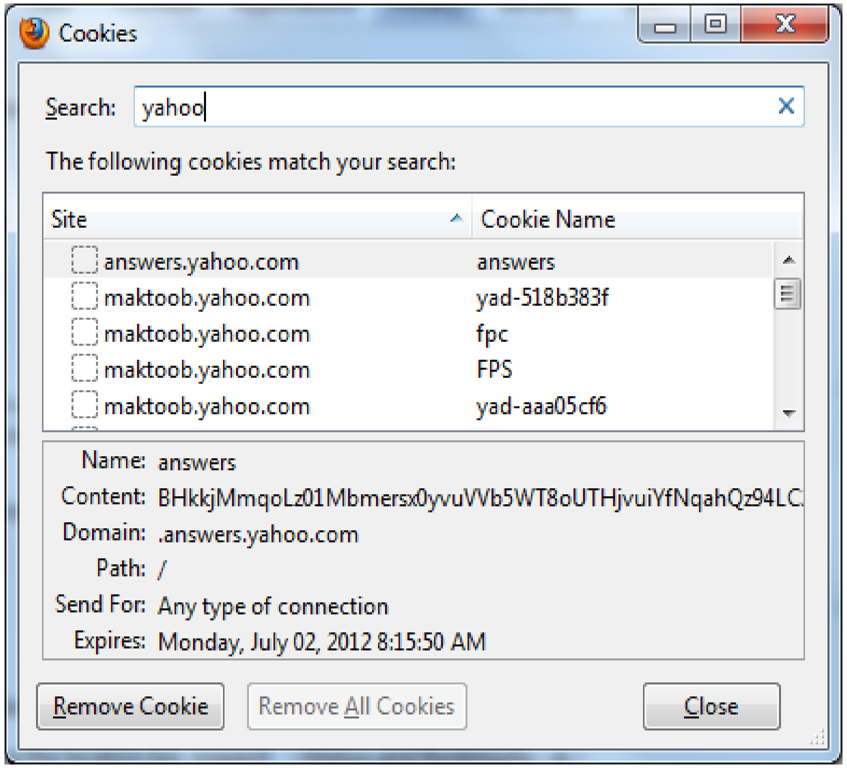
Firefox will remember your browsing, download, form and search history, and keep cookies from websites you visit.

You may want to clear your recent history, or remove individual cookies.

Location Bar

When using the location bar, suggest: History and Bookmarks

Firefox



Search: yahoo

The following cookies match your search:

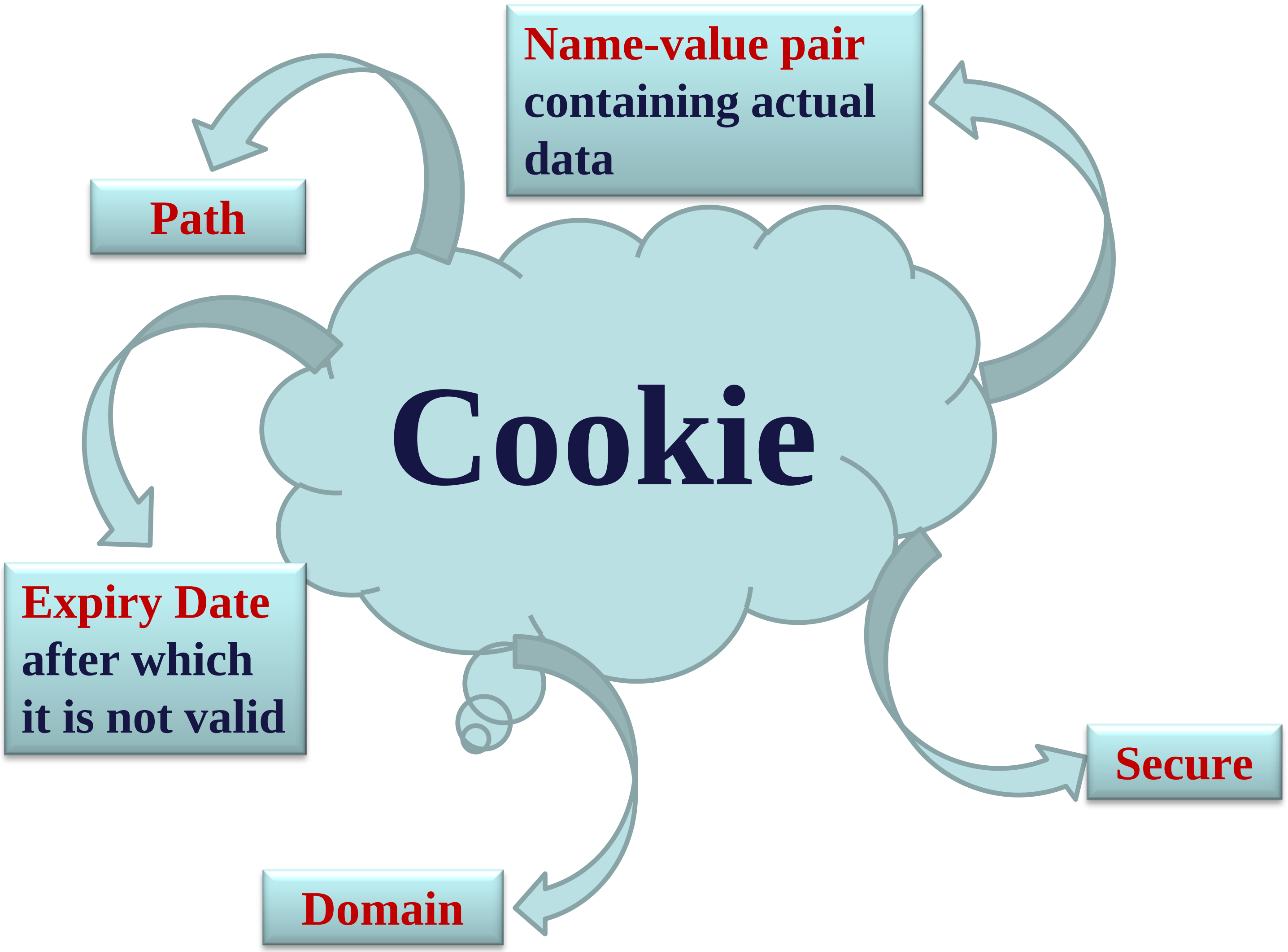
Site	Cookie Name
answers.yahoo.com	answers
maktoob.yahoo.com	yad-518b383f
maktoob.yahoo.com	fpc
maktoob.yahoo.com	FPS
maktoob.yahoo.com	yad-aaa05cf6

Name:	answers
Content:	BHkkjMmqoLz01Mbmersx0yvuVVb5WT8oUTHjvuiYfNqahQz94LC
Domain:	.answers.yahoo.com
Path:	/
Send For:	Any type of connection
Expires:	Monday, July 02, 2012 8:15:50 AM

Cookies false claims

- **Cookies false claims:**
  - Cookies are like worms and viruses in that they can erase data from the user's hard disks
  - Cookies generate popups
  - Cookies are used for spamming
  - Cookies are only used for advertising

Cookies parts


Name-value pair containing actual data
Path
Cookie
Expiry Date after which it is not valid
Secure
Domain

Cookies parts (Cont.)

- Name-value pair: This sets both cookies name and its value.
- Domain: This optional value specifies a domain within which the cookie applies. Only websites in this domain will be able to retrieve the cookie. Usually this is left blank, meaning that only the domain that set the cookie can retrieve it.
- Path: This optional value specifies a path within the site to which the cookies applies. Only documents in this path will be able to retrieve the cookies. Usually this is left blank, meaning that only the path that set the cookies can retrieve it.
- Secure: This optional flag indicates that the browser should use SSL when sending the cookies to the server. This flag is rarely use.

Cookies parts (Cont.)

- Expiry date: This optional value set the date that the cookie will expire on. The date should be in the format returned by toGMTString() method of the Date object. If the expires value is not given, the cookie will be destroyed the moment the browser is closed.
  - Persistent Cookies →These cookies have an expiry date, are stored on a visitor's hard drive and are read by the visitor's browser each time the visitor visits the Web site that sent the cookie.
  - Session / Non-persistent → A cookie without a valid expiry date will not be stored on your machine but will reside on the Web browser. They expire as soon as the visitor closes the Web browser (ie. until you log off).

Working with cookies

- ❑ How to work with Cookies?
  - Cookies can be created, read and deleted by JavaScript, under these conditions:
    - → The user's navigator must be cookie-enabled. This can be checked using "navigator.cookieEnabled" property .
    - → The cookie(s) that you set or accept are only accessible at pages with a matching domain name, matching path.
    - → The cookies must not have reached or passed their expiry date.
  - When these criteria are met the cookies become available to JavaScript via the document.cookie property.

Creating a Cookie

- Creating a Cookie:
  - The simplest way to create a cookie is to assign a string value to the document.cookie.
  - Syntax:

```
document.cookie = "key1=value1;expires=date";
```

```
<script language=”JavaScript”>
            var myDate = new Date();
            Var d= myDate.getDate() +10;
             myDate .setDate(d);
            document.cookie = “myCookie=” + escape(“This is my Cookie”) +
             “;expires=” + myDate.toGMTString();
 </script>
```

Reading a Cookie

- Reading a Cookie:
  - The document.cookie will keep a list of name=value pairs separated by semicolons, where name is the name of a cookie and value is its string value.
  - We use strings' split() function to break the string into key and values.

```
<script>
        var newCookie = document.cookie;
        var cookieParts = newCookie.split(“=”);
        var cookieName = cookieParts[0];
        var cookieValue = unescape(cookieParts[1]);
        alert(cookieName);
        alert(cookieValue);
</script>
```

Deleting a Cookie

- Deleting a Cookie:
  - If the user logs out or explicitly asks not to save his or her username in a cookie, hence, you need to delete a cookie to remove a username cookie.
  - Simply reassign the cookie, but set the expiration date to a time has already passed

```
<script language=”JavaScript”>
            var newDate = new Date();
             newDate.setTime(newDate.getDate() - 1);
            document.cookie = “myCookie=;expires=“+
            newDate.toGMTString();
</script>
```

Working with multiple cookies

- Reading multiple Cookie:
  - Assign each cookie in turn to the document.cookie object and ensure that each cookie has a different name, and may have a different expiration date and time.

- Retrieving multiple Cookie:
  - more complicated since accessing document.cookie,there will be a series of cookies separated by semicolons;

```
CookieName=firstCookieValue; secondCookieName=secondCookieValue;
```

Creating a Cookie Function Library

- Creating a Cookie Function Library:
  - Working with cookies requires a lot of string and date manipulation, especially when accessing existing cookies when multiple cookies have been set.
  - To address this, you should create a small cookie function library for yourself that can:
    - create
    - access
    - delete cookies

    without needing to rewrite the code to do this every time.

Creating a Cookie Function Library (Cont.)

- Creating a Cookie Function Library:

getCookie(cookieName)

- Retrieves a cookie value based on a cookie name.

setCookie(cookieName,cookieValue,expiryDate)

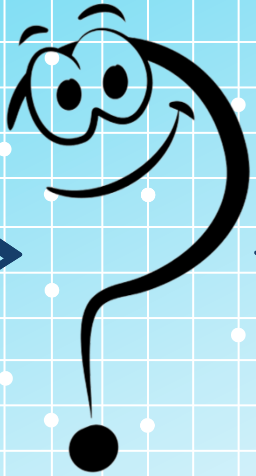
- Sets a cookie based on a cookie name, cookie value, and expiration date.

deleteCookie(cookieName):

- Deletes a cookie based on a cookie name.

Self Study

- ❑ What’s HTML5 Local storage, and session storage?
- ❑ What’s the difference between them and Cookie?

<SCRIPT >  </SCRIPT>

<script>document.writeln("Thank You!")</script>