Lecture 10 –ICT1233

PHP Cookies

DEPARTMENT OF ICT

FACULTY OF TECHNOLOGY

Objectives

After the successful completion of this lecture, students should be able to,

- Identify what are cookies in php
- Use cookies in web development

HTTP Cookies

- A cookie is a small file on your client's hard drive which contains data which you have asked to store

- Introduces state into HTTP

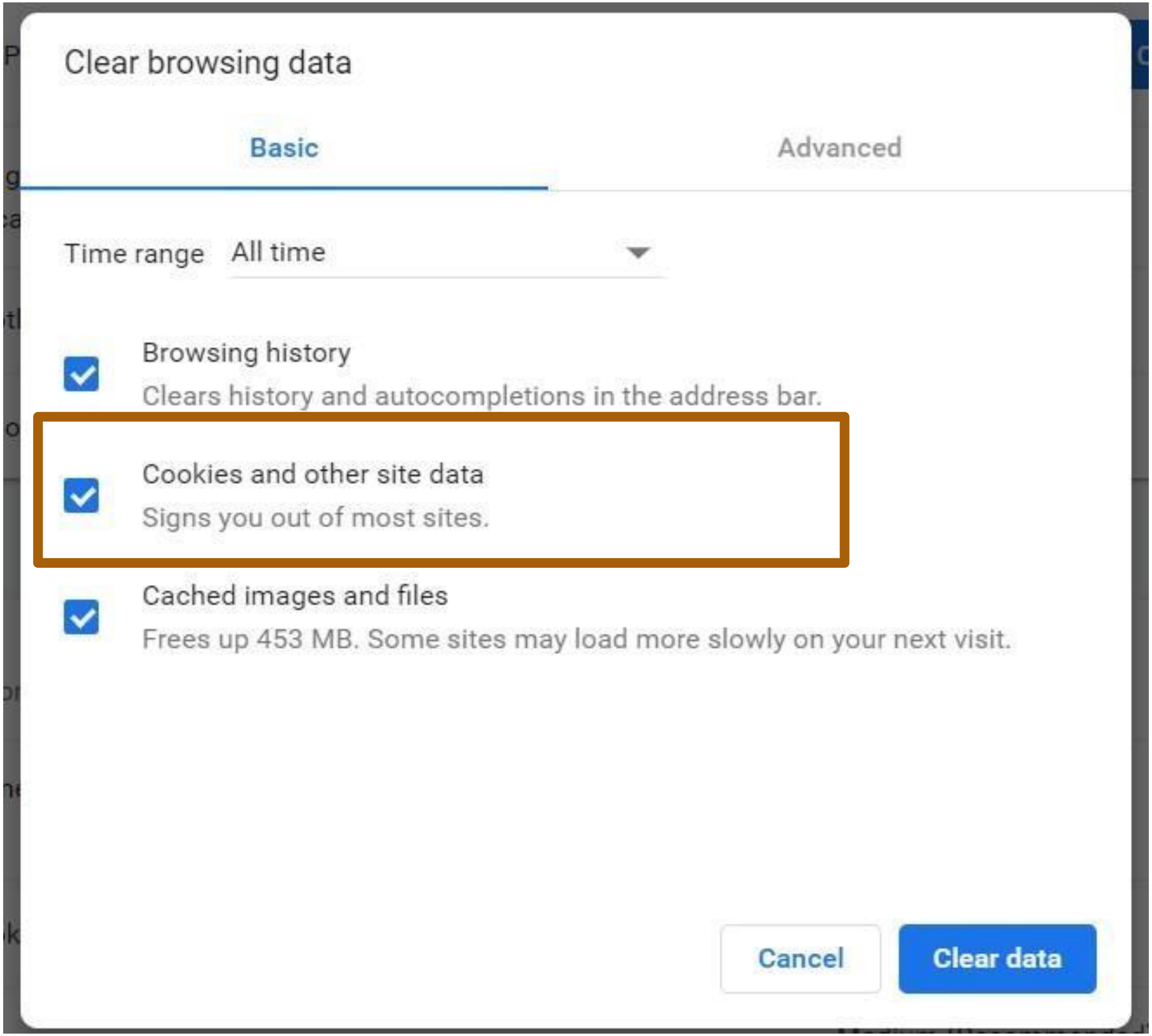
- **Cookies** are **transferred** between the **server** and the **client** according to **http**

- PHP supports http cookies

- A cookie is often used to identify a user

Cookies

- Each time the same computer requests a page with a browser, it will send the cookie too
- With PHP, you can both create and retrieve cookie values
- Specific variable information is stored on the users computer using cookies
- Cookies can also be removed by the user at any time, so don't use them to store anything too important



Clear browsing data

Time range: All time

☑ Browsing history
Clears history and autocompletions in the address bar.

☑ Cookies and other site data
Signs you out of most sites.

☑ Cached images and files
Frees up 453 MB. Some sites may load more slowly on your next visit.

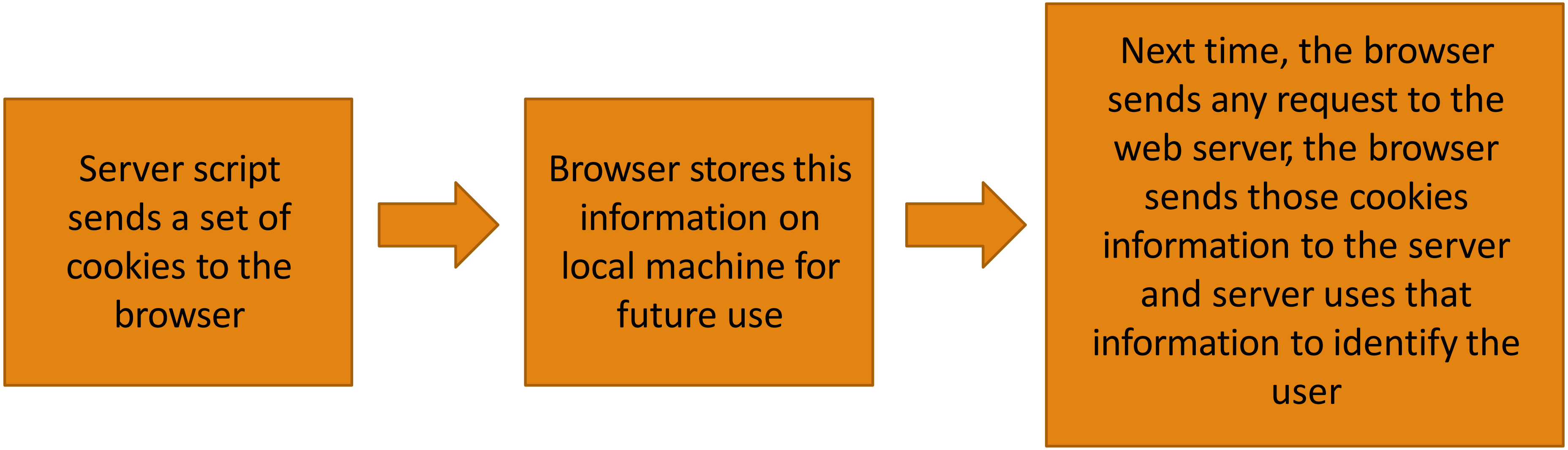
Cookies Structure

- Contains data as name-value pairs
- Comes with expiration dates
  - Expired cookie contents are no longer accessed by the browser
- Can be managed by server-side scripts(PHP)
  - Relevant cookies are automatically submitted from client to server with HTTP request
  - PHP stores information in $_COOKIE super global array

Cookies

- Cookies are automatically sent to the web server (and received/parsed by PHP) each time a user visits
  - PHP will read the value into the $_COOKIE super global array

How the Cookies Work..??



Server script sends a set of cookies to the browser

Browser stores this information on local machine for future use

Next time, the browser sends any request to the web server, the browser sends those cookies information to the server and server uses that information to identify the user

Types of Cookies

session cookie

- Also called a transient cookie
- A cookie that is erased when you close the Web browser
- The session cookie is stored in temporary memory and is not retained after the browser is closed
- Session cookies do not collect information from your computer They typically will store information in the form of a session identification

Types of Cookies

persistent cookie

- Also called a permanent cookie, or a stored cookie
- A cookie that is stored on your hard drive until it expires (persistent cookies are set with expiration dates) or until you delete the cookie
- Persistent cookies are used to collect identifying information about the user, such as Web surfing behavior or user preferences for a specific Web site

Uses of Cookies

Session management:

Cookies are widely used to manage user sessions

For example, when you use an online shopping cart, you keep adding items in the cart and finally when you checkout, all of those items are added to the list of items you have purchased. This can be achieved using cookies.

Uses of Cookies

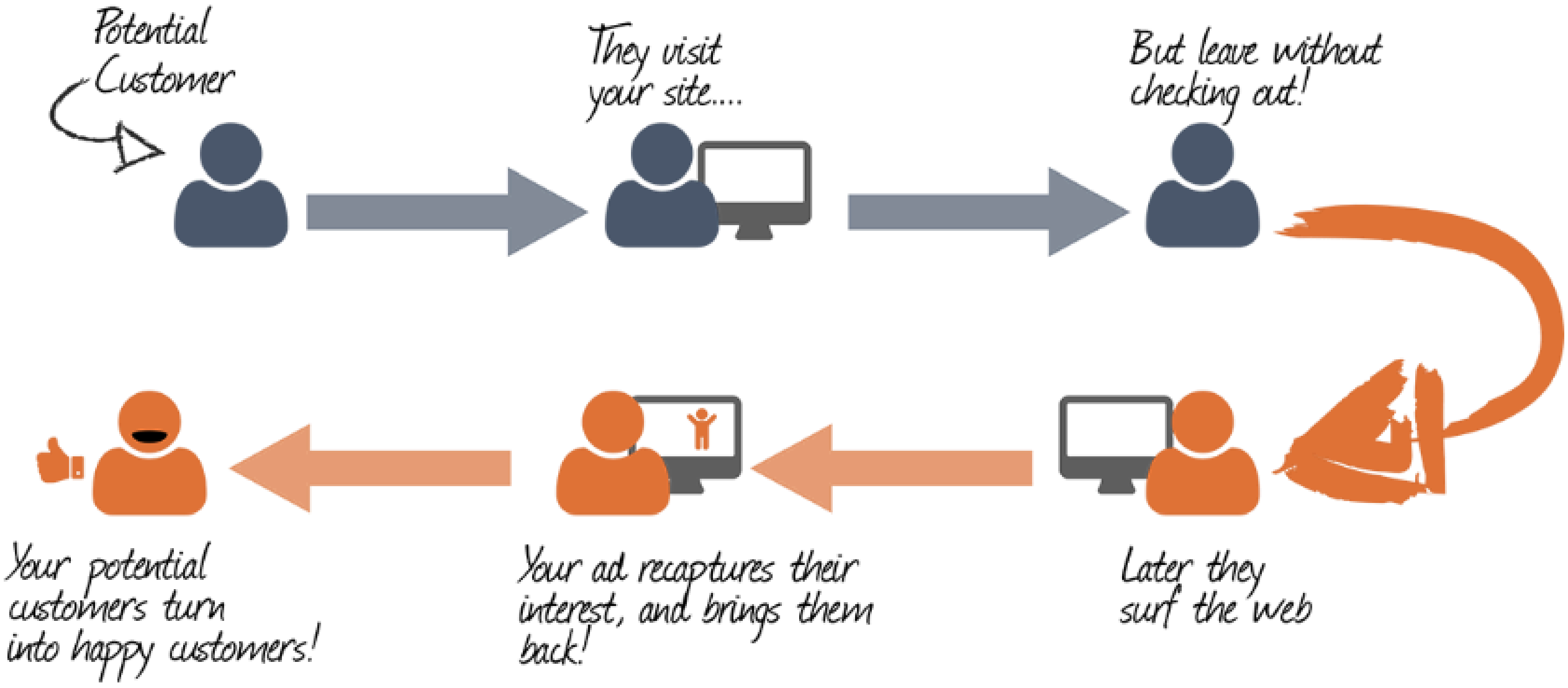
User identification:

Once a user visits a webpage, using cookies, that user can be remembered

And later on, depending upon the search/visit patterns of the user, content which the user likely to be visited are served

*An example for this is '**Retargetting**'. A concept used in online marketing, where depending upon the user's choice of content, advertisements of the relevant product, which the user may buy, are served*

Retargeting


Potential Customer
They visit your site....
But leave without checking out!
Later they surf the web
Your ad recaptures their interest, and brings them back!
Your potential customers turn into happy customers!

Uses of Cookies

Tracking / Analytics:

Cookies are used to track the user

Which, in turn, is used to analyze and serve various kind of data such as,

Each user's amount of visits, and the time of the first visit, the previous visit, and the current visit, technologies (e.g. browser, OS) form where the user visited, how long (s)he stayed on various pages etc.

Handling Cookies with PHP

Create Cookies

- PHP provides ***setcookie()*** function to set a cookie
- This function requires up to six arguments and should be called before the <html> tag
- **For each cookie this function has to be called separately**

setcookie(***name, value, expire, path, domain, security***);

Create Cookies

- **Name** – Name of the cookie (required)
- **Value** – Value of the named variable and the content that you actually want to store (required)
- **Expiry** – A future time in seconds
  - If this is not set, then the cookie will automatically expire when the Web Browser is closed
- **Path** –Specifies the server path of the cookie
  - A single forward slash character permits the cookie to be valid for all directories
  - The default value is the current directory that the cookie is being set in

Create Cookies

- **Domain** − Specify the domain name in very large domains and must contain at least two periods to be valid
- All cookies are only valid for the host and the domain which created them

- **Security** – Specifies whether or not the cookie should only be transmitted over a secure HTTPS connection
  - TRUE or 1 indicates that the cookie will only be set if a secure connection exists
  - Default is FALSE or 0

Create Cookies

```
<?php
   setcookie("name", “John Watson", time()+3600, "/","", 0);
   setcookie("age", "36", time()+3600, "/", "", 0);
?>
<html>

   <head>
      <title>Setting Cookies with PHP</title>
   </head>

   <body>
      <?php echo "Set Cookies"?>
   </body>

</html>
```

Exercise

Assign the following values into variables

- Cookie name : name
- Cookie value : Peter
- Expiration time : After an hour
- Domain : 127.0.0.1
- Valid path : Valid for all the directories
- Security : Can be set even without a secure connection

Create a cookie by passing the variables as the arguments

Create Cookies

```
<?php
        $cookiename=“test”;
        $cookievalue=“test value”;
        $cookieExpire=time()+3600; //after an hour
        $cookieDomain=“127.0.0.1”;
        setcookie($cookiename, $cookievalue, $cookieExpire, "/", $cookieDomain,
        0);
?>
<html>

   <body>
      <?php echo "Set Cookies"?>
   </body>

</html>
```

Retrieve Cookie

- When a Web browser accepts a cookie, you can't extract its value until the next HTTP request is made
    - Ex: if you set a cookie called ***name*** with a value of ***Peter*** on page 1, you can't extract that value until the user reaches page 2 (or page 3 or any other page that isn't the page on which the cookie is initially set)

- PHP allows you to access the cookie simply by using it as a variable

Read cookies

- Use **$_COOKIE** super global array to retrieve a cookie value by using the cookie name as the key

```
<?php
        echo $_COOKIE["name"]. "<br />";
 ?>
```

Retrieving All the Cookies

- In order to retrieve a full list of the cookies, iterate through the $_COOKIE array

```
<?php
 echo "<p>There are ".count($_COOKIE)." cookies stored for this domain.</p>";
// count the number of elements in $_cookie array

 for ($i=0; $i < count($_COOKIE); $i++){
              echo "Cookie name: ".key($_COOKIE)." Cookie value: 
".current($_COOKIE).”<br>”;
       next($_COOKIE);
 }
?>
```

isset()

- use **isset()** function to check if a cookie is set or not

```
<?php
        if( isset($_COOKIE["name"]))
           echo "Welcome " . $_COOKIE["name"] . "<br />";

        else
           echo "Sorry... Not recognized" . "<br />";
     ?>
```

Alter Cookie

- To alter a cookie simply use the setcookie() function to pass new parameters to a cookie of the same name

```
//change the value of fname cookie to Shani

/*old values : -
setcookie("test","teststring!",time()+3600,"/examples",localhost",1)
*/

setcookie("test","anothervalue",time()+(3600*24*365),"/examples",
localhost",1)
```

Delete Cookie

- To destroy the cookie, simply use set cookie again, only set the expiration date to be in the past

```
<?php
   setcookie( "name", "", time()- 60, "/","", 0);
   setcookie( "age", "", time()- 60, "/","", 0);
?>
```

Exercise

1. Find out the limitations of cookies in php

2.Explain how the cookies can be used to personalize the user experience

Difference between Cookies & Session

- The main difference between cookies and sessions is that cookies are stored in the user's browser, and sessions are not

- A cookie can keep information in the user's browser until it is deleted
  - *If a person has a logged in once, the login details can be set as a cookie in their browser so they do not have to re-login to the website every time they visit*

- The trouble is that a user can block cookies or delete them at any time
  - *If, for example, your website's shopping cart utilized cookies, and a person had their browser set to block them, then they could not shop at your website*

Difference between Cookies & Session

- The problem with sessions is that when you close your browser you also lose the session
  - *So, if you had a site requiring a login, this couldn't be saved as a session like it could as a cookie, and the user would be forced to re-login every time they visit*

Summary

- PHP sessions pass data from one page to another
- However, this session information is temporary and is usually deleted very quickly after the user has left the website that uses sessions
- PHP Cookie use to store information about the user and their visit on the user's computer

Questions…..