



# **COMP 4021**

# **Internet Computing**

## **Cookies**

Dr. Kenneth LEUNG

Slides created by Dr. David ROSSITER

# Stateful vs Stateless Protocol

- The result of a request depends on the “state” of the communicating parties, e.g., the meaning of “next page” depends on what the “current page” is
- Stateful protocols are more efficient but stateless protocols are more scalable (i.e., suitable for internet) **WHY?**

Get a page given exact URL	Stateless	
Get the current page	Stateful	“current” depends on where you are
Get the “next” page	Stateful	“next” depends on where you are

# HTTP is Stateless

- An exact URL is required to get a page
- The returned page is the same whether or not you send the URL today or tomorrow or via different webpages (of course, page content could have been updated)
- Bing's navigation bar for the result pages

Prev 1 2 3 4 5 6 7 8 Next

- Current page showing results from 31 to 40
- Prev URL: `http://www.bing.com/search?q=4021&go=&qs=n&pq=4021&sc=0-0&sp=-1&sk=&first=21&FORM=PQRE`
- Next URL: `http://www.bing.com/search?q=4021&go=&qs=n&pq=4021&sc=0-0&sp=-1&sk=&first=41&FORM=PORE`

# What Are Cookies?



- Basically, cookies are just text messages that allow the browser/server to remember the “states” of the interaction
  - E.g., Whom am I talking to (user name), visit history, etc.
- A web site can store one or more cookies
- If you visit the web site again then JavaScript code in the web page can read the cookies that were stored earlier
- (Doesn't have to be a web site – also works for files loaded from the local hard disk)

# Things Stored in Cookies

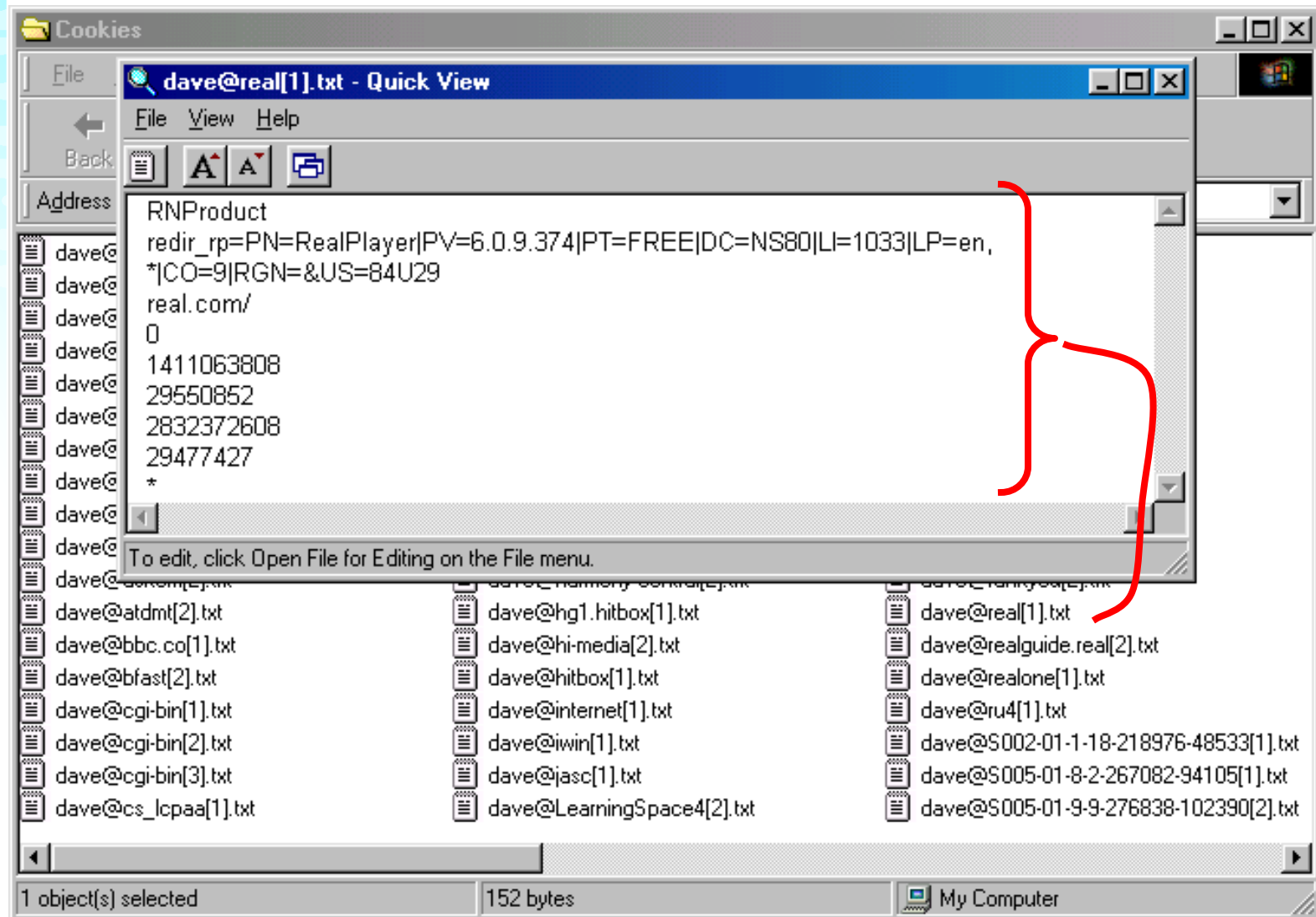
- You can store anything you want in a cookie
- Examples of things commonly stored:
  - The date/ time you last visited the web site
  - How many times you have been to the web site
  - What things you clicked on (e.g. books, etc)
  - The highest score so far – for a game

# Example Cookie Data - IE

- For IE, each web site is given its own cookies file
- All cookie data for that web site is stored in that file



# Example Cookie Data - IE



# Cookie Limitations

- A browser can hold up to 300 cookies
- There is a maximum of 20 cookies per web site
  - If you already have 20 cookies and you add a new cookie the browser finds the oldest cookie for that web site and throws it away
- A cookie has a byte size limitation of 4096 bytes (=4Kb)
  - If you add a new cookie and this makes the size larger than 4kB, then the browser finds the oldest cookie for that web site and throws it away
- All of these limitations may be different from version to version and from browser to browser



# Using JavaScript for Cookies

- JavaScript controls cookies through the `document.cookie` property
- You can read cookies by reading it
- You can make a cookie by changing it

# Reading the Cookies

- `document.cookie` is a string containing all of the cookies associated with the current web site
- To see all the cookies do: `alert(document.cookie)`
- If `document.cookie` has three cookies it will have this structure:

`cookieName1=cookieValue1;cookieName2=cookieValue2;  
cookieName3=cookieValue3`

- For example: `name=dave;score=8900;total_time=46`
- So you need some JavaScript string handling to extract the individual data out of the string

# Saving a Cookie

- Because `document.cookie` is just a string, changing it is straightforward
- For example, the following JavaScript statement sets two cookies:  
`document.cookie = "name=peter;score=260";`

# Example JavaScript

- Put the date/time in a cookie, and shows it in the web page

```
<head>  <script>
    var today = new Date();
    now = today.getHours() + ":" + today.getMinutes() + ":"
        + today.getSeconds();

    document.cookie = "last_visit=" + now;
    document.writeln(document.cookie );
</script> </head>
```

# Cookie Expiry Time

- It is a good idea to also set a time for the cookie to 'die' (expire)
- Here a cookie is made with a specific time to expire:

```
document.cookie = "name=peter;  
    expires=Tue, 17-Mar-08 00:00:01 GMT";
```

# Deleting a Cookie

- The way to delete a cookie is to set the date/time of the cookie to a date/time that has already finished (i.e. 1970, or one second ago, or one hour ago)
- The browser will then automatically remove the cookie
- For example:  
`document.cookie = "name=peter;  
expires=Thu, 01-Jan-70 00:00:01 GMT";`

# Altering a Cookie

- What if you have already made a cookie, but now you want to change it?
- Cookies can be altered by simply
  - reading the `document.cookie` string
  - changing the string as appropriate
  - copying the string back to `document.cookie`

# Functions for Handling Cookies

- You can define functions to help with handling cookies
- The source code for a set of routines for handling cookies is shown on the next few slides
  - `setCookie()`
  - `getCookie()`
  - `deleteCookie()`



# Cookie Handling - setCookie

```
function setCookie(name, value, expires, path, domain, secure) {  
    var curCookie = name + "=" + escape(value) +  
        ((expires) ? "; expires=" + expires.toGMTString() : "") +  
        ((path) ? "; path=" + path : "") +  
        ((domain) ? "; domain=" + domain : "") +  
        ((secure) ? "; secure" : "");  
    document.cookie = curCookie; }  
}
```

- This code creates a cookie (using parameters passed to it)
- The expiry time needs to be given to it in milliseconds
- You can see that there are other possible parameters for a cookie not discussed here – path, domain, and secure

# Cookie Handling - getCookie

```
function getCookie(name) {  
    var dc = document.cookie;  
    var prefix = name + "=";  
    var begin = dc.indexOf("; " + prefix);  
    if (begin == -1) {  
        begin = dc.indexOf(prefix);  
        if (begin != 0) return null;  
    } else begin += 2;  
    var end = document.cookie.indexOf(";", begin);  
    if (end == -1) end = dc.length;  
    return unescape(dc.substring(begin + prefix.length, end)); }  
}
```

Use string functions  
to extract data from  
the cookie string

# Cookie Handling - deleteCookie

```
function deleteCookie(name, path, domain) {  
    if (getCookie(name)) {  
        document.cookie = name + "=" +  
            ((path) ? "; path=" + path : "") +  
            ((domain) ? "; domain=" + domain : "") +  
            "; expires=Thu, 01-Jan-70 00:00:01 GMT";    }    }
```

- **path** - path of the cookie (must be same as path used to create cookie)
- **domain** - domain of the cookie (must be same as domain used to create cookie)

# Using the Functions

- Two examples follow
- Example 1 – A web counter
  - Each time you visit the page, it adds one to a counter stored in a cookie
- Example 2 – A name tracker
  - The name of the user is stored in a cookie and is shown every time the page is visited

# Example 1 - Web Counter

- Use a cookie to count how many times someone has visited a particular web page
- The following script displays the number of times the user has visited, assuming just one person uses the browser)
- Reload the page to see the counter increment

By the way, you have been here 4 times.

# Example 1 - Web Counter

```
var now = new Date();    // create an instance of the Date object
now.setTime(now.getTime() + 365*24*60*60*1000); // expires in 365 days
                                     // getTime() and setTime() work in msec
var visits = getCookie("counter");
if (!visits) {
    visits = 1;           // if the cookie wasn't found, this is the first visit
    document.write("By the way, this is your first time here.");
} else {
    visits = parseInt(visits) + 1; // increment the counter
    document.write("By the way, you have been here " + visits + " times."); }
setCookie("counter", visits, now); // set the new cookie
```

## Example 2 - Name Tracker

- The following script asks the user for his/ her name, and "remembers" the input
- It then welcomes the user each time he/ she accesses the page, without asking again for the name

Welcome to this page, Dave.

## Example 2 - Name Tracker

```
var now = new Date();    // create an instance of the Date object

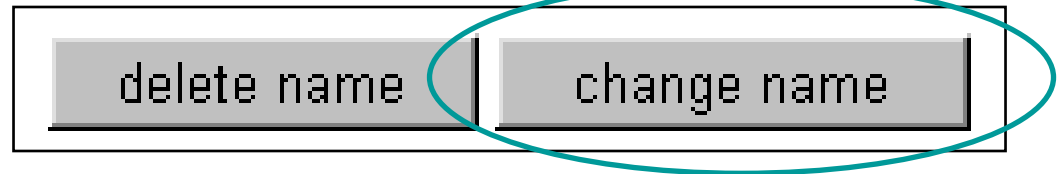
now.setTime(now.getTime() + 365 * 24 * 60 * 60 * 1000);
var username = getCookie("username");
// if the cookie wasn't found, ask for the name
if (!username) username = prompt("Please enter your name:", "");

setCookie("username", username, now); // set the new cookie
if (username) {
    document.write("Welcome to this page, " + username + ".");
    setCookie("username", username, now);
} else document.write("You refused to enter your name.");
```



## Example 2 - Name Tracker

- Altering a cookie, e.g., allow user to change to a new name



```
function changeName() {  
    var now = new Date();  
    // cookie will expire one year later  
    now.setTime(now.getTime() + 365 * 24 * 60 * 60 * 1000);  
    username = prompt("Please enter your name:", "");  
    setCookie("username", username, now);    }
```