COMP4021
Internet Computing

Cookies

Gibson Lam and David Rossiter

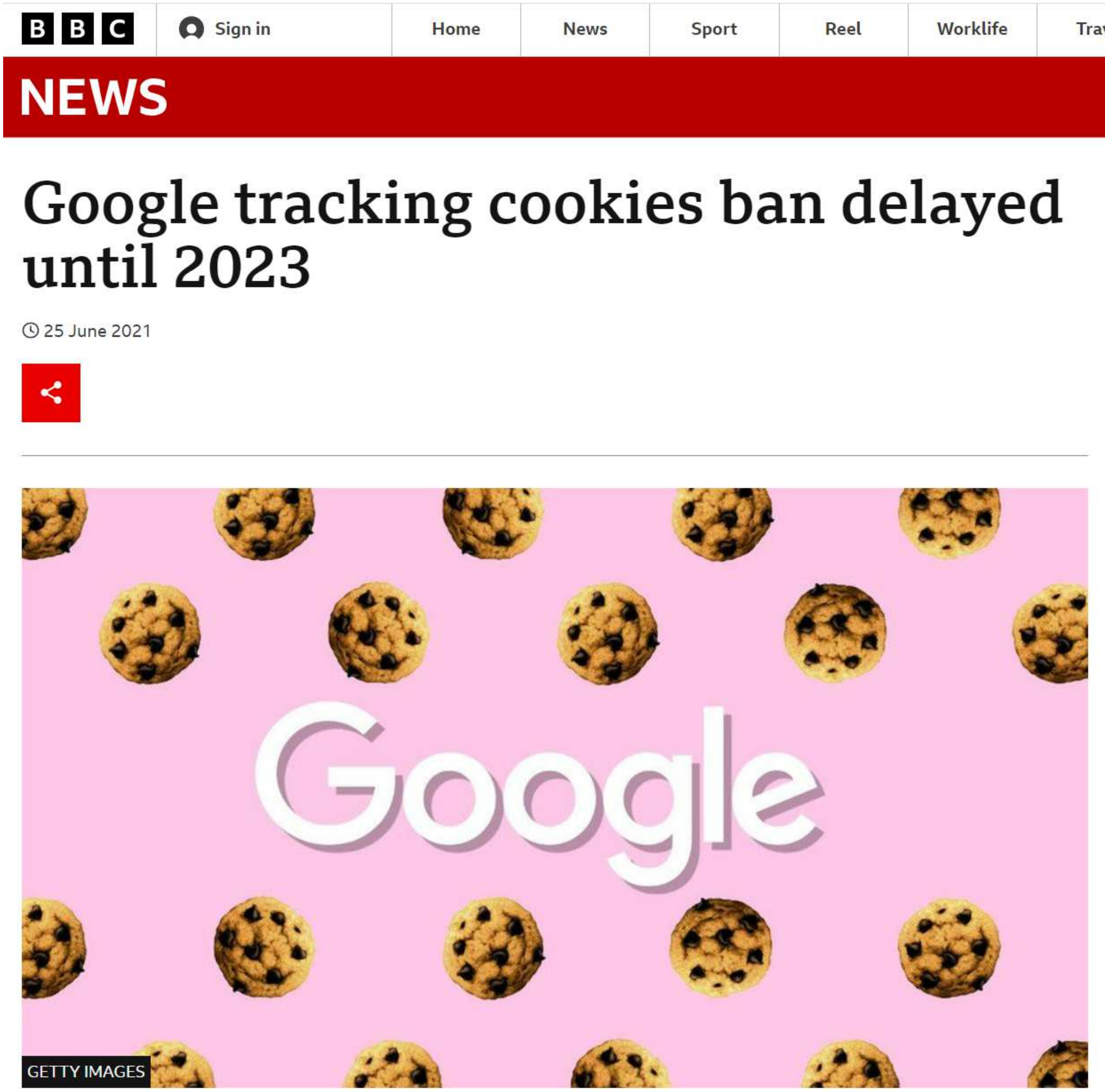
Cookies

- A cookie is a small piece of text, which is stored in the browser when you go to a web site
- After that, when you load a web page from the same web site, the cookie information is automatically given back to the web site
- In this way the web site can ‘remember’ things about your visit

Use of Cookies

- Cookies can be used in many ways
- Here are some examples:
  - Storing data for a web page, e.g. storing the highest score of a game
  - Maintaining login session between the browser and the server
  - Activity tracking, e.g. Google can track the pages and ads that you have visited and show you similar ads later

No more tracking cookies, i.e. third-party cookies?



Google tracking cookies ban delayed until 2023

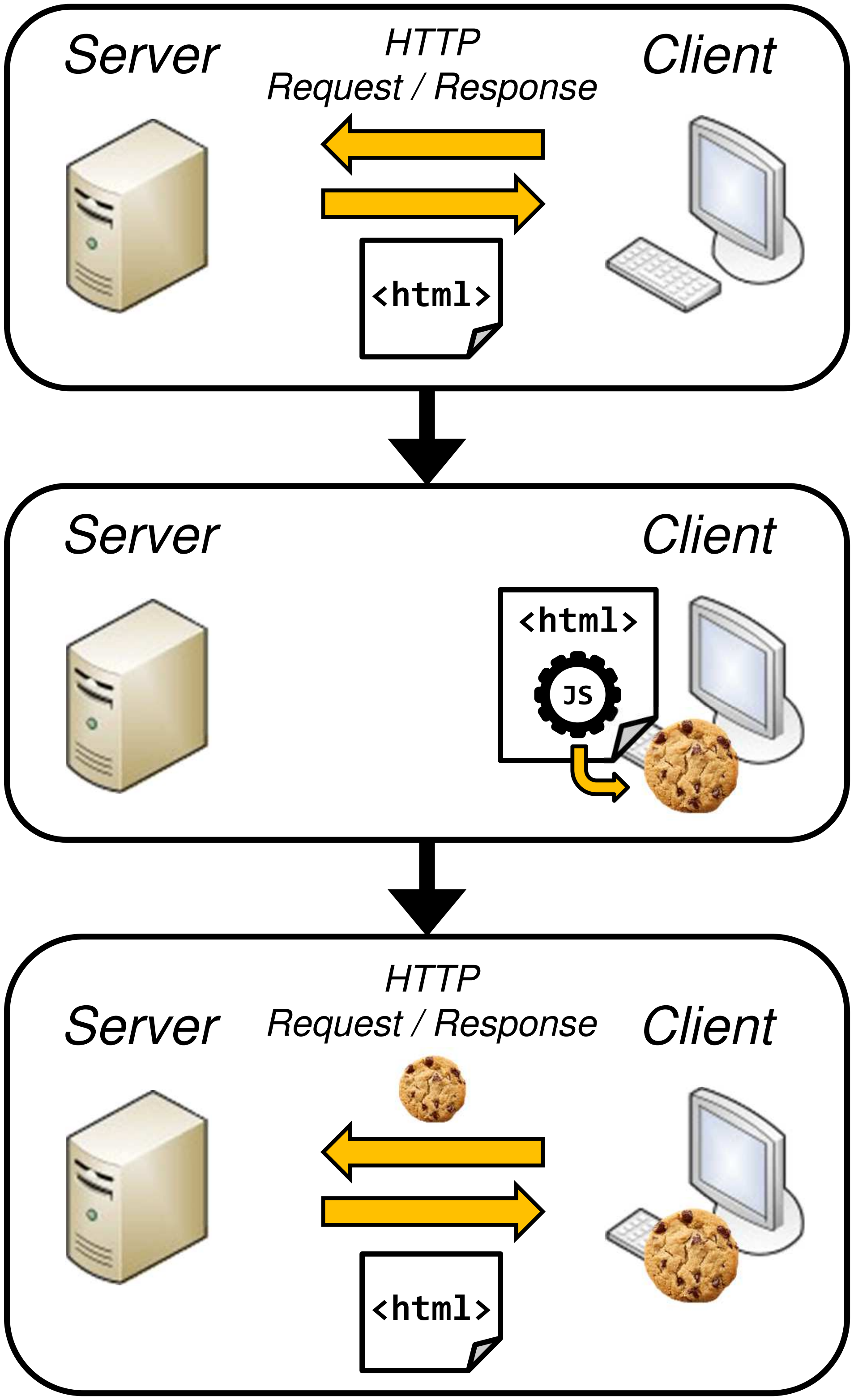
25 June 2021

Creating Cookies

- Cookies can be created by:
  - The browser using JavaScript, or
  - The server using HTTP headers
- Cookies created by a web site are not shared to another web site
  - For example, if you have a cookie from starwars.com, then you visit disney.com, it will not be sent to disney.com

Creating a Cookie Using JavaScript

1. The browser requests and gets an HTML page from the server
2. JavaScript in the HTML page create a cookie in the browser
3. Later, if the browser requests for a page in the same web site, the cookie content will be sent to the server too



Creating Cookies in JavaScript

- You can easily create a cookie using `document.cookie` in JavaScript
  - For example, if you want to create a cookie called “`name`” with a value of “`Paul`”, you can do this:

```
document.cookie = "name=Paul";
```

A cookie always has a name and value pair

Using a Web Server

- Since cookies are associated with a web site, they do not work if you use your HTML page locally
  - e.g. double-clicking your HTML file to show it inside a browser
- You must put the HTML page on a web server if you want to use the page to create and use cookies

Creating Multiple Cookies

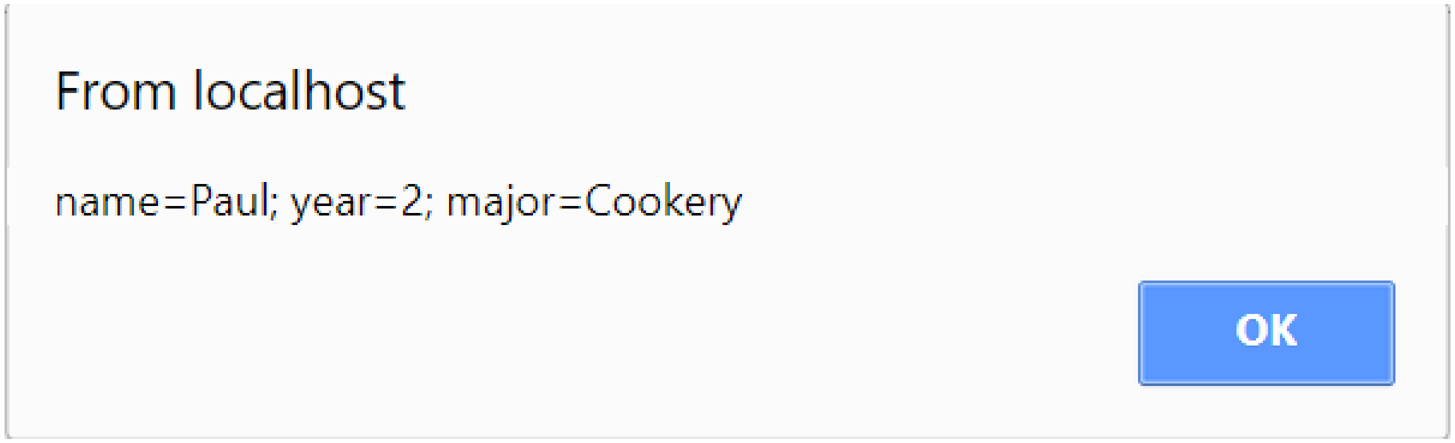
- You can create many cookies using multiple lines of `document.cookie`:

```
document.cookie = "name=Paul";
document.cookie = "year=2";
document.cookie = "major=Cookery";
```

- The above code then creates three cookies with different names

Reading Cookies

- To read the cookies, you simply read the `document.cookie`
- For example, if you show an alert box with `document.cookie`, after creating the cookies shown in the previous slide, you will see this:

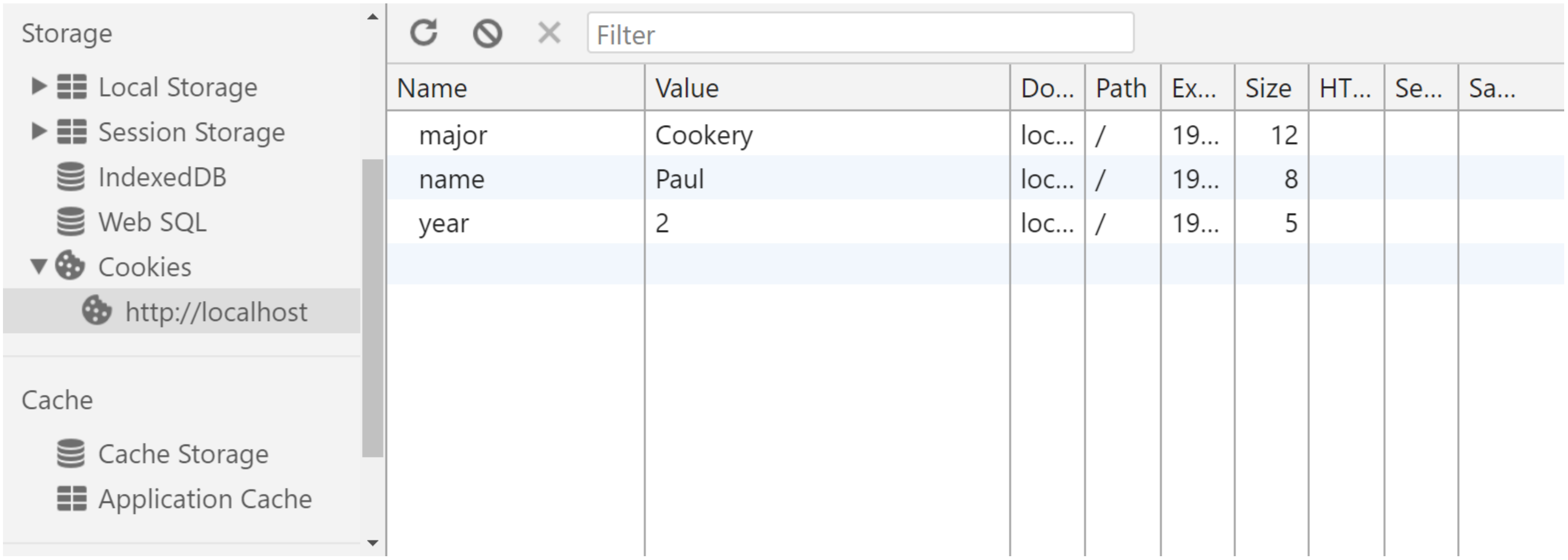


From localhost

```
name=Paul; year=2; major=Cookery
```

Viewing Cookies in Chrome

- You can also use the Chrome Developer Tool (Application > Cookies) to view the cookies:



Name	Value	Do...	Path	Ex...	Size	HT...	Se...	Sa...
major	Cookery	loc...	/	19...	12			
name	Paul	loc...	/	19...	8			
year	2	loc...	/	19...	5			

Cookie Expiry Time

- A cookie may ‘expire’ at a certain time
- In our examples so far, we never do anything about the expiry time
- In these examples, the cookies expire when you close the browser, i.e. the cookie are destroyed after the ‘session’
- If you don’t want them to be destroyed, you will need to set the cookie expiry time

Setting the Expiry Time

- You set a cookie’s expiry time by doing this when you create the cookie, e.g.:

```
document.cookie =
   "year=2;
    expires=Wed, 01 Jan 2023 00:00:00 GMT";
```

The expiry date and time

- If you set the expiry time of a cookie using a time in the past, the cookie will then be deleted

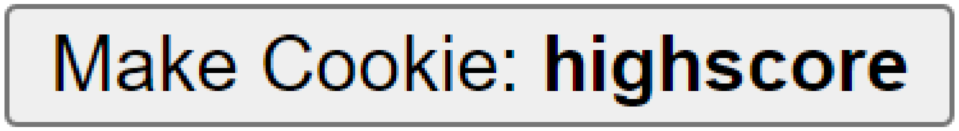
Using Cookie Example 1/3

- Here is some code to create a cookie:

```
...
<script>
function makeCookie() {
  document.cookie =
    "highscore=100;
     expires=Wed, 01 Jan 2023 00:00:00 GMT";
}
</script>
```

Create using a future date/time

```
...
<button onclick="makeCookie()">
  Make Cookie: <b>highscore</b>
</button>
...
```

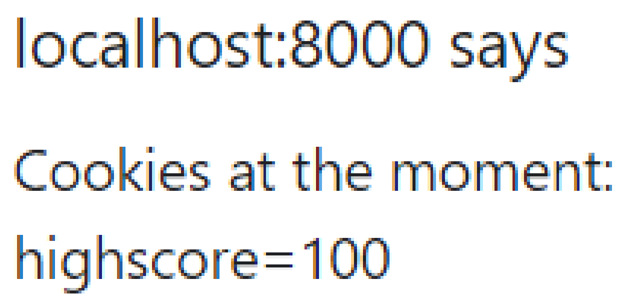


Using Cookie Example 2/3

- Here is some code to view the cookie:

```
...
<script>
function viewCookies() {
  alert("Cookies at the moment:\n" +
        document.cookie);
}
</script>

...
<button onclick="viewCookies()">
  View Cookies
</button>
...
```



```
localhost:8000 says

Cookies at the moment:
highscore=100
```

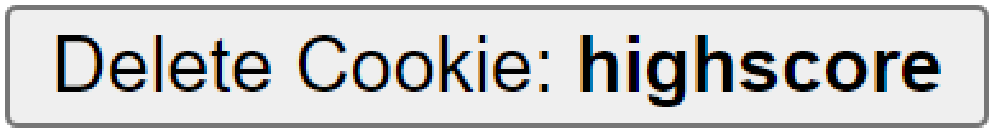
Using Cookie Example 3/3

- Here is some code to delete the cookie:

```
...
<script>
function deleteCookie() {
  document.cookie =
    "highscore=100;
     expires=Thu, 01 Jan 1970 00:00:00 GMT";
}
</script>
```

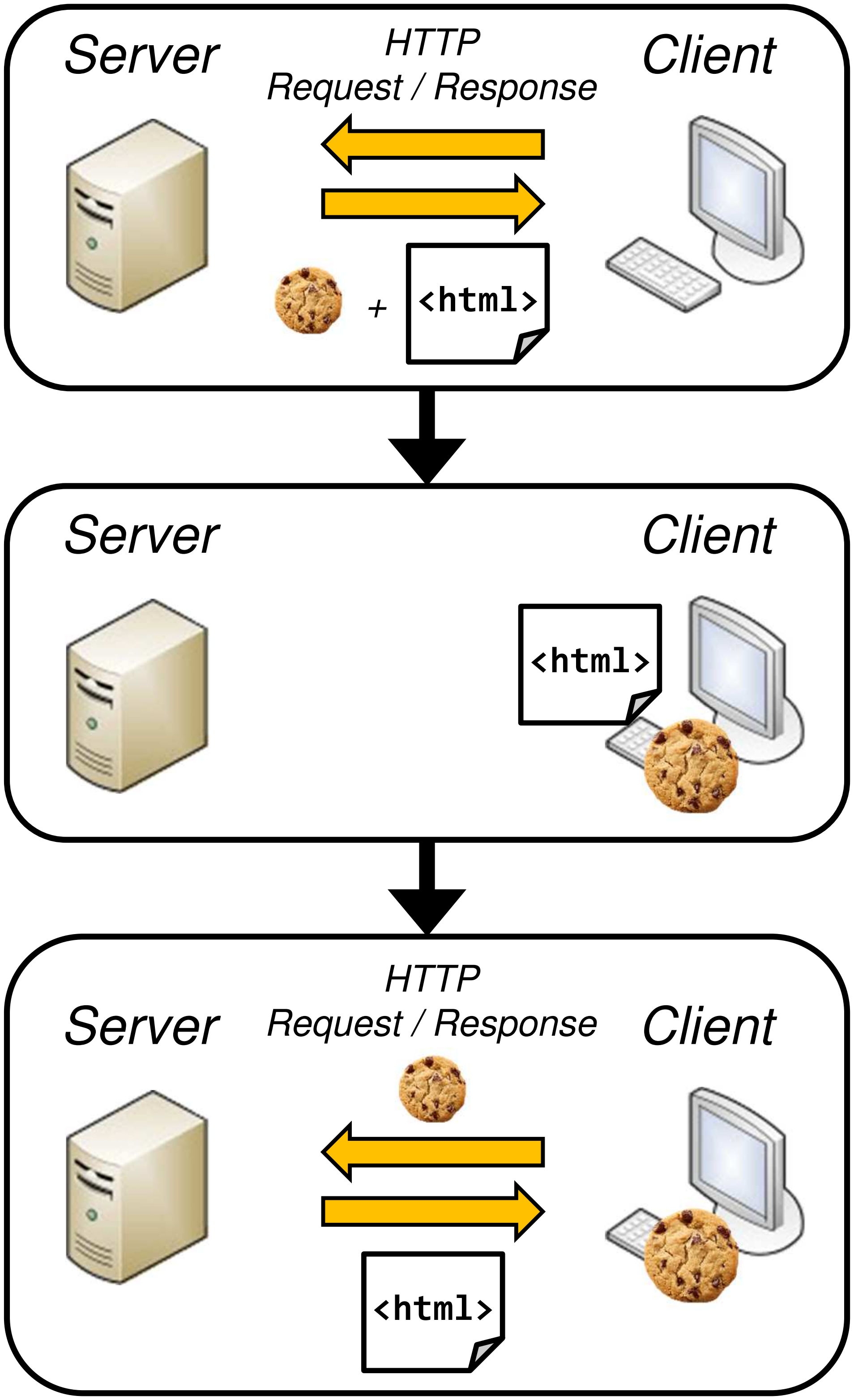
Create using a past date/time

```
...
<button onclick="deleteCookie()">
  Delete Cookie: <b>highscore</b>
</button>
...
```



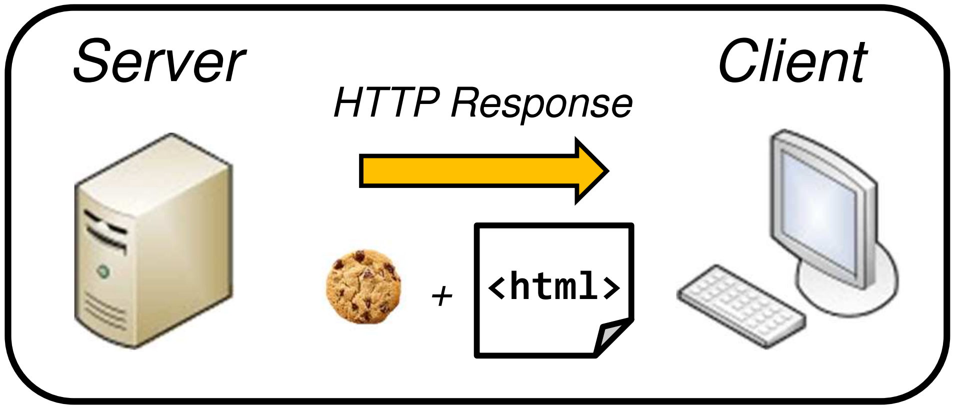
Creating a Cookie From Server Code

1. The browser requests and gets an HTML page from the server; the server also sends a cookie to the browser
2. The browser stores the cookie locally
3. Later, if the browser requests for a page in the same web site, the cookie content will be sent to the server too



Cookies in an HTTP Response

- Here is what happens in an HTTP response:
  - A server responds to an HTTP request with the requested HTML file
  - The server can create a cookie in the browser by putting a `Set-Cookie` header in the response



```
HTTP/1.1 200 OK
...some HTTP headers...
Set-Cookie: username=pchan
Content-Type: text/html

...the HTML content...
```

Using Cookies in Express

- If you want to use cookies in Express, you will need to install a package called `cookie-parser` using npm, i.e:

```
C:\Users\Gibson>npm install cookie-parser
```

- After that, you need to ask the Express app to use the cookie parser like this:

```
const cookieParser =
      require("cookie-parser");
app.use(cookieParser());
```

Using Cookies in the Server

- After setting things up correctly, you can then do these in Express:
  - Read cookies using `req.cookies`, e.g.:

    ```
    const { username } = req.cookies;
    ```

  - Create cookies using `res.cookie()`, e.g.:

    ```
    res.cookie("username", "pchan");
    ```

  - Delete cookies using `res.clearcookie()`, e.g.:

    ```
    res.clearCookie("username");
    ```