



Molecular Biomedical Informatics
分子生醫資訊實驗室

Web Programming
網際網路程式設計

Cookie

The motivation

- HTTP is a **stateless** protocol, each subsequent click to a web site is treated as new by the web server
- Web developers have to put a lot of efforts for this
 - demo with **form**
 - it still has limitations and, more importantly, security issues
- Suppose that we use two tabs to open the same web site twice, we expect that the changes in the first tab should also be applied on the second tab, with a refresh at most
 - can you do this right now?

Cookie is another storage media

- Before this class, you need CGI to make users' input permanent by save them on the server side in either files or database
- Actually, you won't be happy if the web applications can save something on the client site, that is, your hard disk
- As a compromising way, cookie provides a limited storage media on the client side
- References
 - [HTTP cookie](#)
 - [List of HTTP header fields](#)

HTML

```
Content-type: text/html  
Content-Disposition: filename=fname.ext  
Set-Cookie: UserID=dirty  
...
```

Header

A blank line

```
<!DOCTYPE html>  
<html>  
  <body>  
    ...  
  </body>  
</html>
```

Content

Cookie is not 100% safe

- Cannot be programmed, cannot carry viruses, and cannot install malware on the host computer
- However, they can be used by spyware to track user's browsing activities
 - this does be a problem, think about Facebook API
 - a major privacy concern that prompted European and US law makers to take action
- Cookies can also be stolen by hackers to gain access to a victim's web account
 - bad habit breaks any security mechanism

A cookie example

- Workflow
 - use the first tab to open a CGI of setting cookie
 - check the browser to see the cookie
 - use the second tab to open another CGI of getting the cookie
 - use the third tab to open the third CGI of setting cookie with another value
 - refresh the second tab
- Now, it is no longer stateless
- Here is also a JavaScript (thus Ajax-able) version
 - still need CGI to make it permanent

Session

- Cookie is a technology while session is how to use this technology, a concept
- If you have two fields, you may use two cookie variables. But how about 100 fields? This is not ridiculous to, for example, a shopping car
- A better idea is to use only one cookie variable as an session id and the remaining data are stored in that session on the server side
 - even the state is stored on the server side
 - reduce the network traffic, make the data permanent...
 - more important, security

Reminding of session

- Do not name the id according to current user
 - or the can create a session without your authentication
 - the id is for the session, but not for the user
- Session can be done without cookie
 - note that cookie is just a technology
 - you can use the more simple HTML protocol, [form](#), to achieve session
 - [CGI::Session](#)



The difference

between using form and cookie?



Today's assignment

今天的任務

Enhance the user experience

- Make your web site state-sensitive. If you have no such a requirement, please add a “remember me” checkbox or remember the last parameters (such as search keywords)
- Reference
 - [CGI::Cookie](#)
 - [Cookie | jQuery Plugins](#)
- Your web site (<http://merry.ee.ncku.edu.tw/~xxx/cur/>, ex10) will be checked not before 23:59 12/16 (Sun). You may send a report (such as some important modifications) to [me](#) in case I did not notice your features. Or, even better, just explain your modifications in the [homepage](#).

Appendix

附錄

jQuery .on()

- A frequently used scenario
 - when document ready, attach a click event to all posts in a blog
 - add a new post via Ajax
 - we want the click event is attached to the newly added post automatically
- If you use `.click()`, it only affects current elements
- If you want the event is attached to the selector rather than the elements, use `.on()`
 - jQuery API [.on\(\)](#), [.bind\(\)](#), [.delegate\(\)](#) and [.live\(\)](#)
 - [Quick Tip: The Difference Between Live\(\) and Delegate\(\)](#)
 - [The Difference Between jQuery's .bind\(\), .live\(\), and .delegate\(\)](#)