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 a new request by the web server
- Web developers have to put a lot of efforts for this

State using forms

Use a parameter from HTML form to maintain state. If specified, operate accordingly; otherwise initialize. The parameter should match the term used in `<input />`.

```
my $cgi = new CGI;
my $state = $defined $cgi->param('state')
  ? $cgi->param('state')+1 : 0;

print <<END;
Content-type: text/html

The state is $state.
<form action="form.cgi" method="post">
  <input name="state" type="hidden" value="$state">
  <button type="submit">Next</button>
</form>
END
```

Remember the empty line in between HTTP header and body.

The actual HTML. In addition to show the state, we also need to send the state to the server. The `hidden` type is exactly for this purpose.

Can we use `get` here?

The form solution

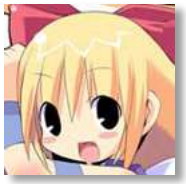
- Still has limitations and, more importantly, security issues
- Suppose that we use two tabs to open the same web site twice, we expect that the changes in the first tab should also be applied on the second tab, with a refresh at most
  - can you do this?

The cookie solution

- You need HTML form because you need to send users' input to the server side and save them in either files or database
- Before this class, this is the only way to make users' input permanent
- Cookie is another storage, on the client side

Cookie

- Conventionally,
  applications can
  is, your hard disk
  - HTML5 Web S
  - security issues
- Cookie, as a compromising way, is a limited storage
  - only 4 kb
  - cannot access cookies from other domains
  - cannot be programmed, cannot carry viruses, and cannot install malware on the host computer



Can you

figure out something evil to do to your classmates

Cookie is not 100% safe

- They can be used by spyware to track users' activities
  - this does be a problem, think about Fabecook API
  - Facebook新Graph API推出 – 你打算跟魔鬼做交易了嗎？
  - 當Facebook統治了世界，你還有隱私可言嗎？
  - 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

State using cookie

Include `CGI::Cookie` module and use `cookie()` instead `param()`. The cookie name should match the `Set-Cookie` filed of the HTTP header.

```
use CGI::Cookie;
my $cgi = new CGI;
my $state = defined $cgi->cookie('state') ? $cgi->cookie('state') : 1;
1 == $cgi->param('next') and ++$state
  and print "Set-Cookie: state=$state\n";
1 == $cgi->param('clean') and $state =
print <<END;
Content-type: text/html

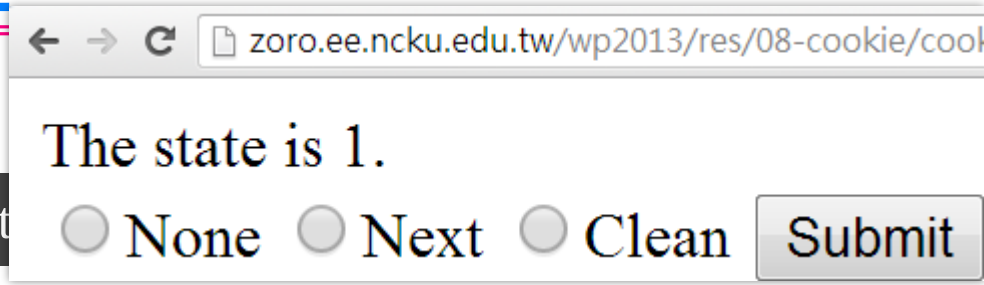
The state is $state.
<form action="cookie.cgi" method="post">
  <input name="next" type="hidden" value="1" />
  <button type="submit">Next</button>
</form>
<form action="cookie.cgi" method="post">
  <input name="clean" type="hidden" value="1" />
  <button type="submit">Clean</button>
</form>
END
```

The state is either from cookie or the initial value.

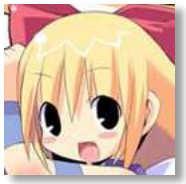
`param()` is still useful for actions instead of states. Update cookie if required.

Set cookie to a null string to clean it.

Remember that out



Use two forms to reduce extra actions, such as a radio button.



Is anything

wrong of the two-form solution

Cookie in JavaScript (jQuery)

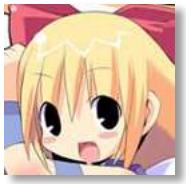
```
<head>
  <script src="jquery.min.js"></script>
  <script src="jquery.cookie.js"></script>
  <script>
    $(document).ready(function(){
      $.cookie('state') && $('#state').text($.cookie('state'));
      $('#next').click(function(){
        var state = $.cookie('state') ? parseInt($.cookie('state'))+1 : 2;
        $.cookie('state',state); $('#state').text(state);
      });
      $('#clean').click(function(){
        $.cookie('state',null); $('#state').text(1);
      });
    });
  </script>
</head>
<body>
  The state is <span id="state">1</span>
  <button id="next">Next</button> <button id="clean">Clean</button>
</body>
```

Include `jquery.cookie` plugin and use `$.cookie()` to access cookie. It's obvious that the code of the the JS version is longer (more things are done by yourself).

Show state if existed. When clicking next, change the state, save it and show it. Watch the `parseInt()` and the weird `2` here.

When clicking clean, clean the cookie (by setting it to null) and show the initial state.

The HTML. You need a place to show the state, but the code is usually more clear.



Is there

any benefits to develop an Ajax version?

I think you can figure out it by yourself :p

A review

- Workflow
  - use a tab to open a URL and change something
  - check the browser to see the cookie
  - use a second tab to open the same URL, see the changes and change something again
  - refresh the first tab → no longer stateless!
- Set-Cookie in HTTP header
  - HTTP cookie
  - List of HTTP header fields
- There is also a JavaScript (thus Ajax-able) version

Any Questions?

about cookie

HTML5 web storage

With HTML5, web pages can store data locally

HTML5 Web Storage

- Earlier, this was done with cookies
- More secure and faster
  - the data is not included with every server request, but used ONLY when asked for
- 5MB (vs. 4kb)
- Better JS integration
- Similarly, the data is stored in key/value pairs, and a web page can only access data stored by itself
- ```
  var state = localStorage.state || 1;
  localStorage.state = ++state;
  ```
  ```

Session

When too many variables to store

- 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
- Cookie, web storage or even HTML form is technology while session is how to use these technologies
- It's a concept (an application of above technologies)

Session

- Only one variable as an session ID, access the remaining data (include the state, which might be complex) with it
  - reduce the network traffic
  - make the data really permanent
  - more important, security
- CGI::Session using cookie
- How many issues are gone in web storage?
- `var state = sessionStorage.state || 1;`
  `sessionStorage.state = ++state;`
  - disappear after closing the browser

Any Questions?

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
  - HTML5 Web Storage
- Your web site (http://merry.ee.ncku.edu.tw/~xxx/cur/, ex7) will be checked not before 23:59 11/19 (Tue). 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.