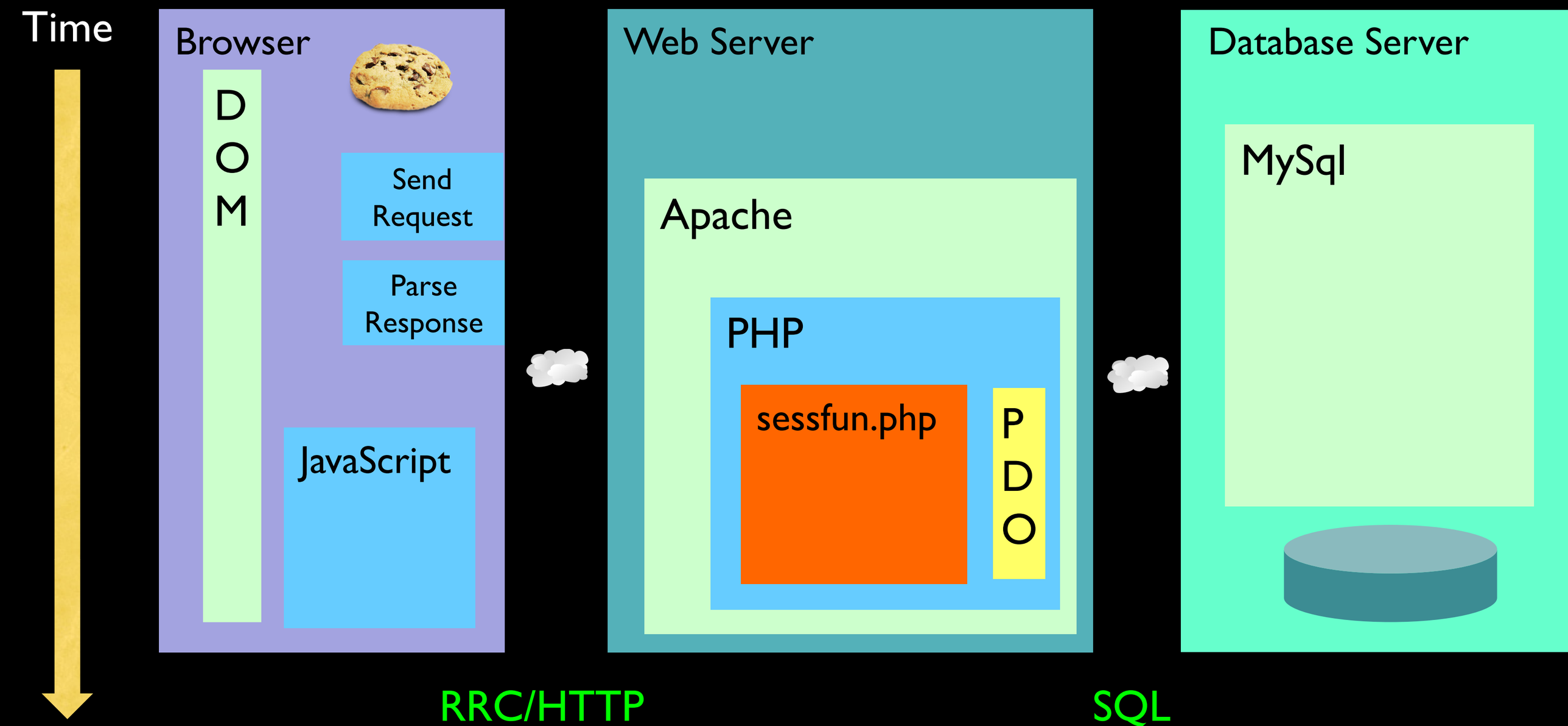
Cookies and Sessions

Dr. Charles Severance
www.wa4e.com

http://www.wa4e.com/code/sessions.zip

HTTP Cookies


Time
Browser
DOM
Send Request
Parse Response
JavaScript
Web Server
Apache
PHP
sessfun.php
PDO
Database Server
MySql
RRC/HTTP
SQL

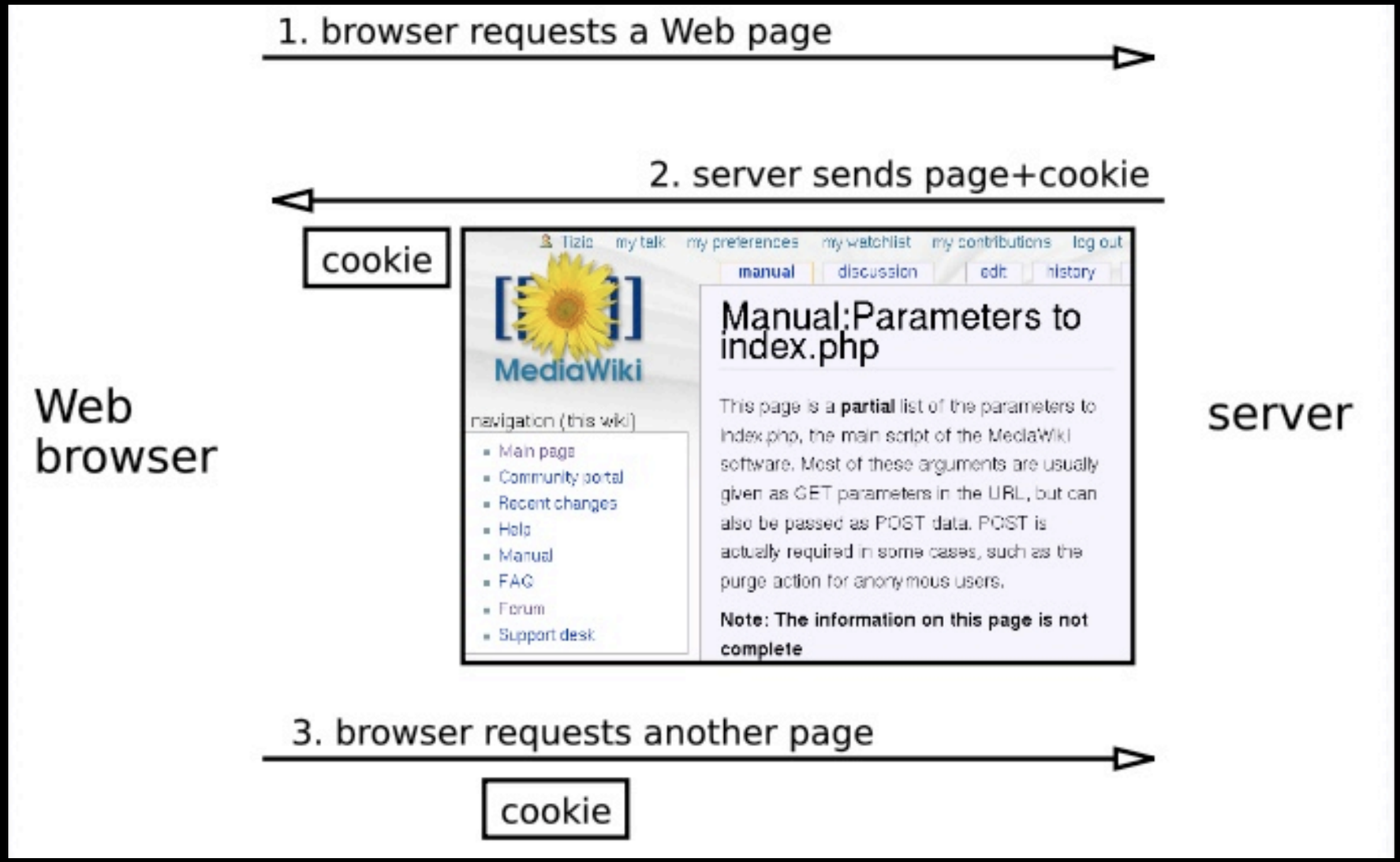
Multi-User / Multi-Browser

- When a server is interacting with many different browsers at the same time, the server needs to know *which* browser a particular request came from.
- Request / Response initially was stateless - all browsers looked identical . This was really bad and did not last very long at all.

Web Cookies to the Rescue

Technically, cookies are arbitrary pieces of data chosen by the Web server and sent to the browser. The browser returns them unchanged to the server, introducing a state (memory of previous events) into otherwise stateless HTTP transactions. Without cookies, each retrieval of a Web page or component of a Web page is an isolated event, mostly unrelated to all other views of the pages of the same site.

http://en.wikipedia.org/wiki/HTTP_cookie



http://en.wikipedia.org/wiki/HTTP_cookie

Cookies In the Browser

- Cookies are marked as to the web addresses they come from. The browser only sends back cookies that were originally set by the same web server.
- Cookies have an expiration date. Some last for years, others are short-term and go away as soon as the browser is closed

Cookies

Change language: English

Edit Report a Bug

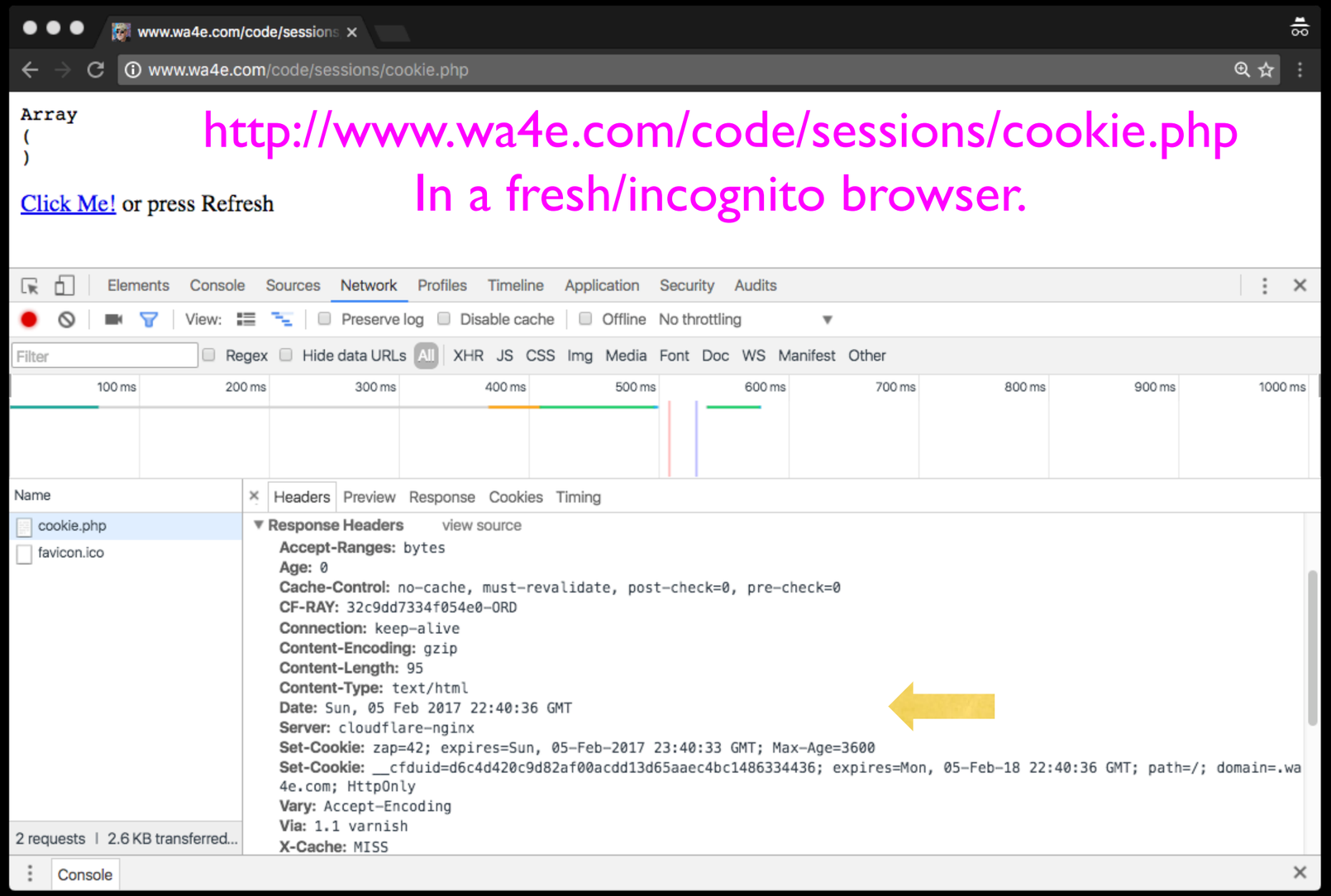
PHP transparently supports HTTP cookies. Cookies are a mechanism for storing data in the remote browser and thus tracking or identifying return users. You can set cookies using the setcookie() or setrawcookie() function. Cookies are part of the HTTP header, so setcookie() must be called before any output is sent to the browser. This is the same limitation that header() has. You can use the output buffering functions to delay the script output until you have decided whether or not to set any cookies or send any headers.

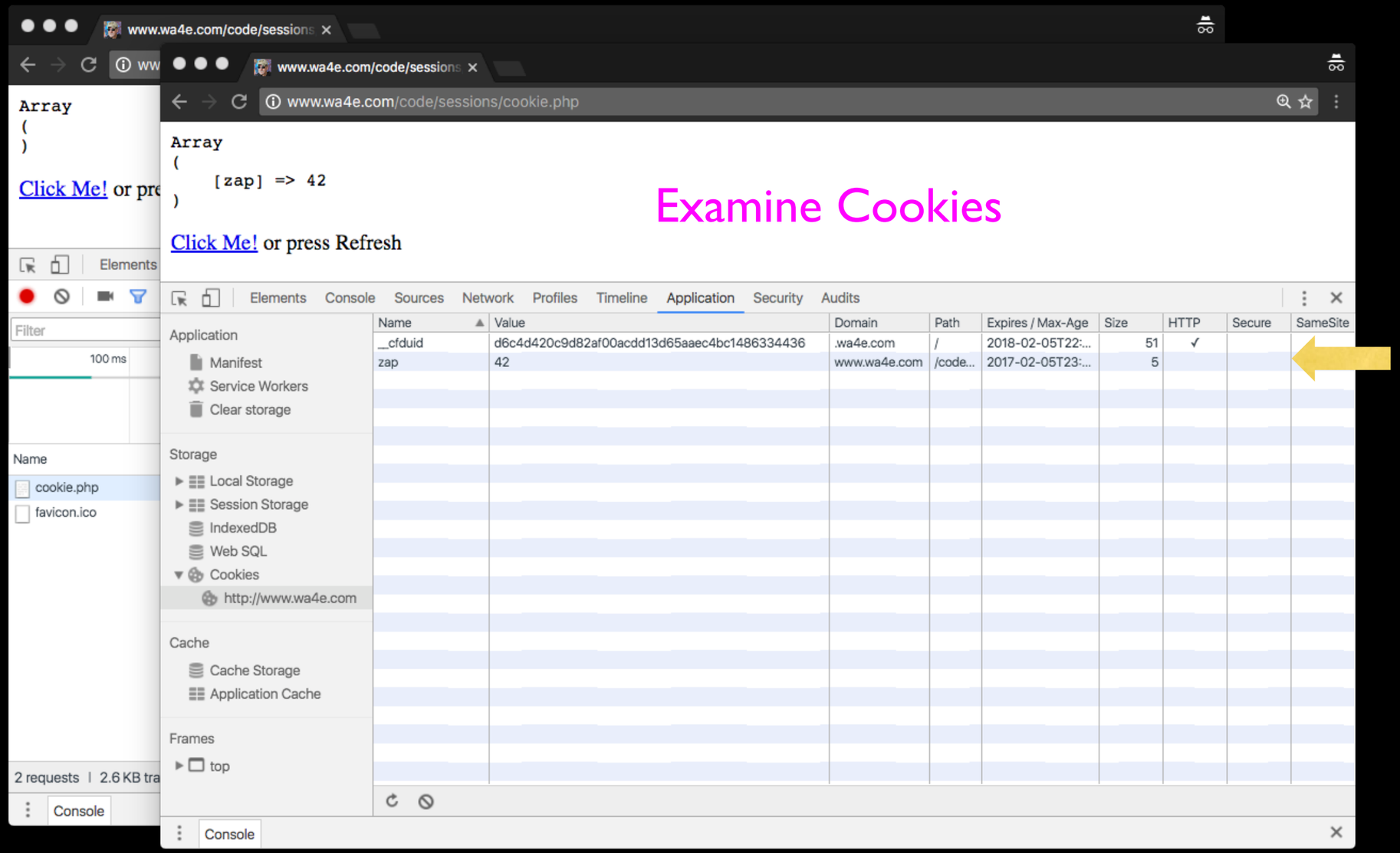
Any cookies sent to you from the client will automatically be included into a `$_COOKIE` auto-global array if variables_order contains "C". If you wish to assign multiple values to a single cookie, just add *[]* to the cookie name.

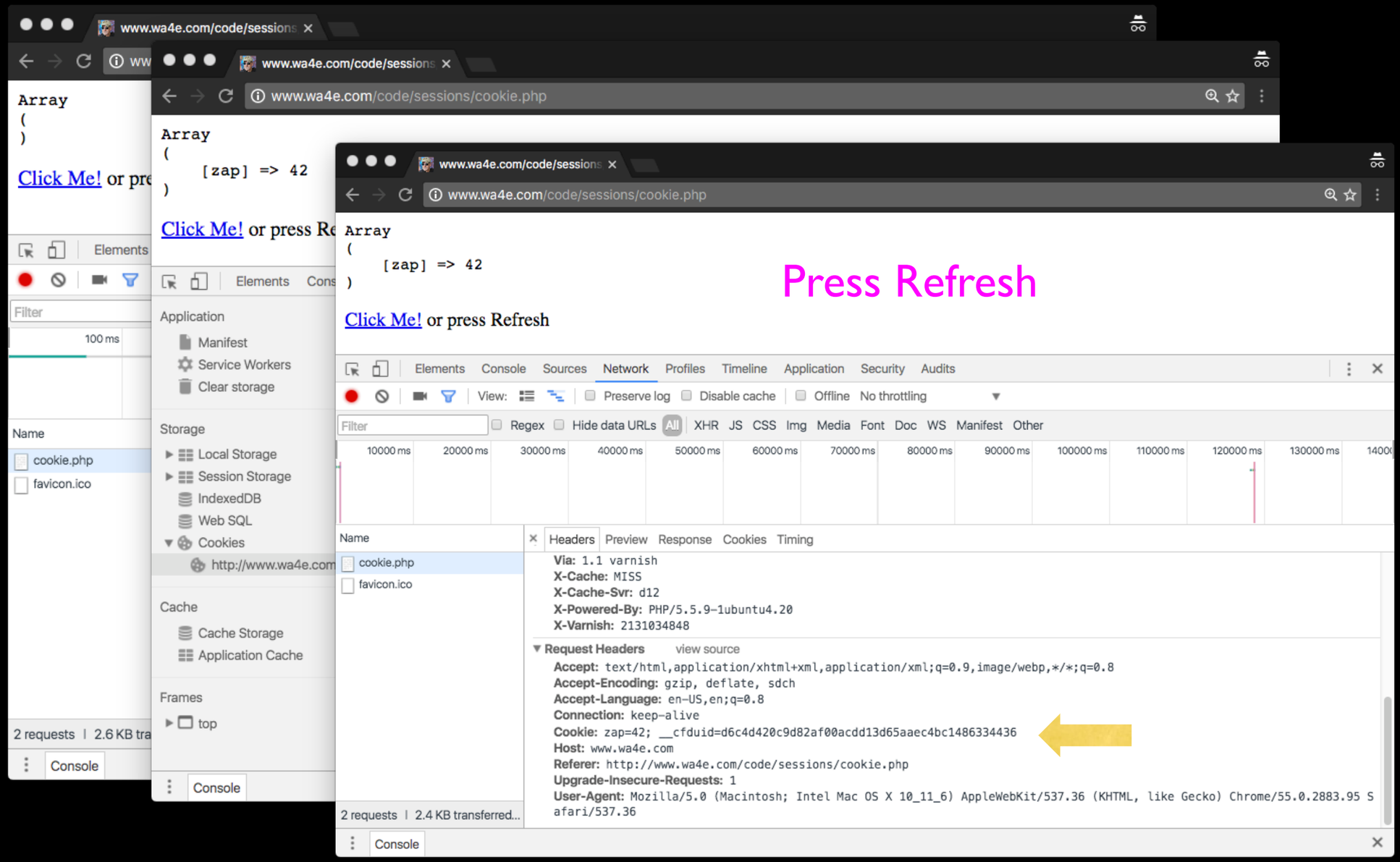
http://php.net/manual/en/features.cookies.php

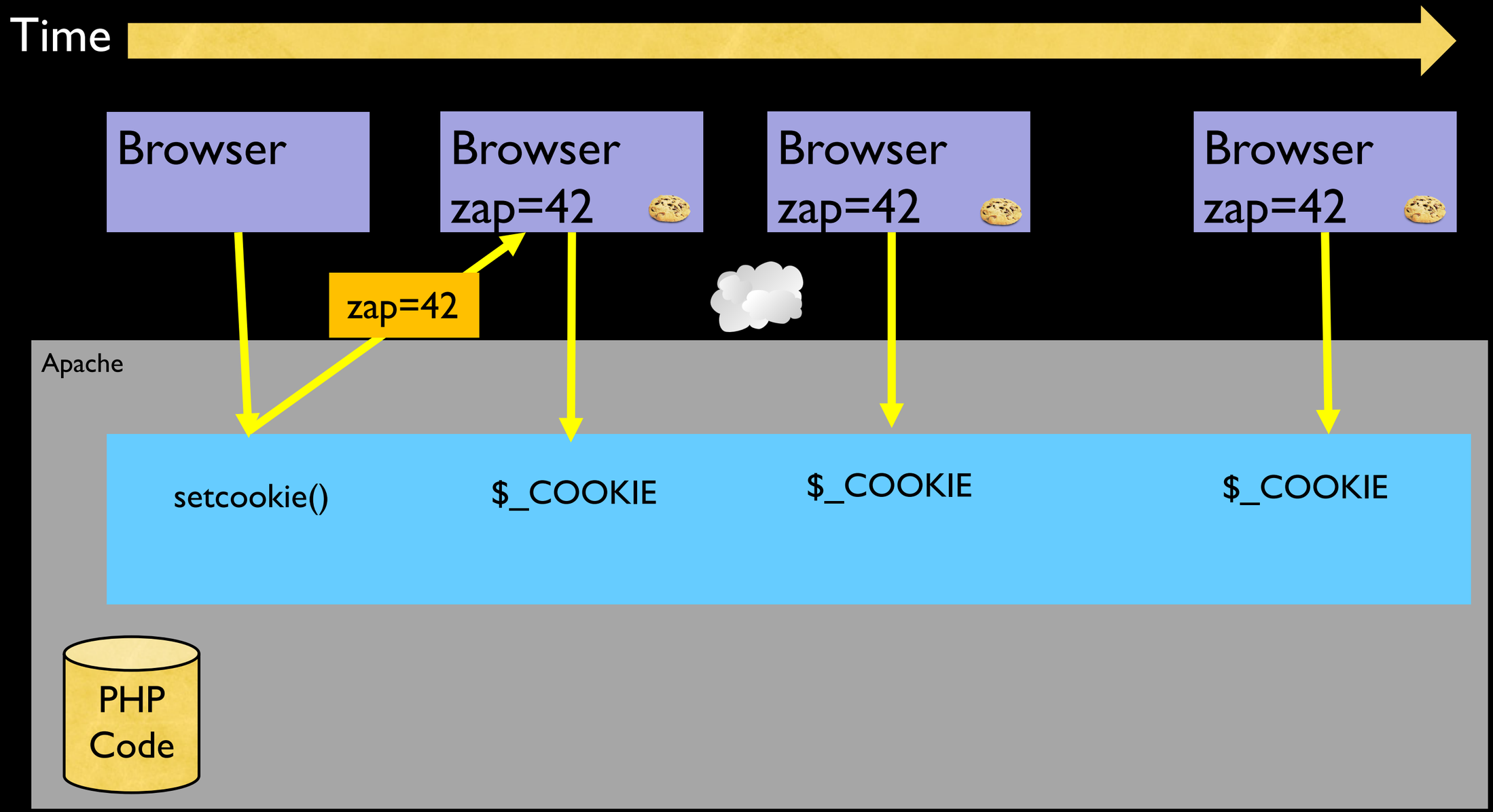
```php
<?php
// Note - cannot have any output before setcookie
if ( ! isset($_COOKIE['zap']) ) {
    setcookie('zap', '42', time()+3600);
}
?>
<pre>
<?php print_r($_COOKIE); ?>
</pre>
<p><a href="cookie.php">Click Me!</a> or press Refresh</p>
```

http://www.wa4e.com/code/sessions/cookie.php


www.wa4e.com/code/sessions
www.wa4e.com/code/sessions/cookie.php
Array
(
)
Click Me! or press Refresh
http://www.wa4e.com/code/sessions/cookie.php
In a fresh/incognito browser.
Elements Console Sources Network Profiles Timeline Application Security Audits
View: Preserve log Disable cache Offline No throttling
Filter Regex Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other
100 ms 200 ms 300 ms 400 ms 500 ms 600 ms 700 ms 800 ms 900 ms 1000 ms
Name
cookie.php
favicon.ico
2 requests | 2.6 KB transferred...
Headers Preview Response Cookies Timing
Response Headers view source
Accept-Ranges: bytes
Age: 0
Cache-Control: no-cache, must-revalidate, post-check=0, pre-check=0
CF-RAY: 32c9dd7334f054e0-ORD
Connection: keep-alive
Content-Encoding: gzip
Content-Length: 95
Content-Type: text/html
Date: Sun, 05 Feb 2017 22:40:36 GMT
Server: cloudflare-nginx
Set-Cookie: zap=42; expires=Sun, 05-Feb-2017 23:40:33 GMT; Max-Age=3600
Set-Cookie: __cfduid=d6c4d420c9d82af00acdd13d65aaec4bc1486334436; expires=Mon, 05-Feb-18 22:40:36 GMT; path=/; domain=.wa4e.com; HttpOnly
Vary: Accept-Encoding
Via: 1.1 varnish
X-Cache: MISS
Console

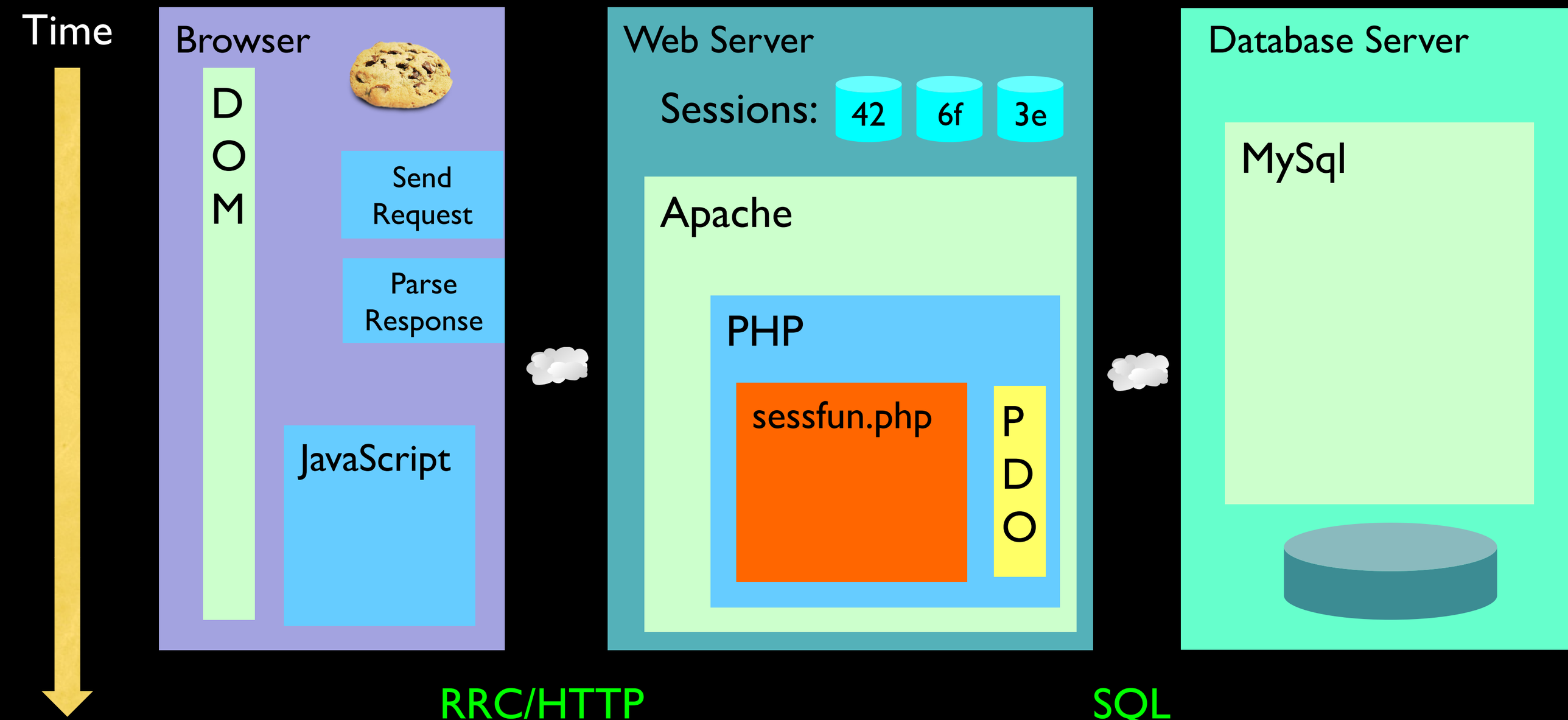

www.wa4e.com/code/sessions
Array
(
)
Click Me! or pre
Elements
Filter
100 ms
Name
cookie.php
favicon.ico
2 requests | 2.6 KB tra
Console
www.wa4e.com/code/sessions
www.wa4e.com/code/sessions/cookie.php
Array
(
[zap] => 42
)
Click Me! or press Refresh
Examine Cookies
Elements Console Sources Network Profiles Timeline Application Security Audits
Application
Manifest
Service Workers
Clear storage
Storage
Local Storage
Session Storage
IndexedDB
Web SQL
Cookies
http://www.wa4e.com
Cache
Cache Storage
Application Cache
Frames
top
Name Value Domain Path Expires / Max-Age Size HTTP Secure SameSite
__cfduid d6c4d420c9d82af00acdd13d65aaec4bc1486334436 .wa4e.com / 2018-02-05T22:... 51 ✓
zap 42 www.wa4e.com /code... 2017-02-05T23:... 5
Console


www.wa4e.com/code/sessions
www.wa4e.com/code/sessions/cookie.php
Array
(
[zap] => 42
)
Click Me! or press Refresh
Press Refresh
Elements
Console
Sources
Network
Profiles
Timeline
Application
Security
Audits
Preserve log
Disable cache
Offline
No throttling
Filter
Regex
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
Name
cookie.php
favicon.ico
Headers
Preview
Response
Cookies
Timing
Via: 1.1 varnish
X-Cache: MISS
X-Cache-Svr: d12
X-Powered-By: PHP/5.5.9-1ubuntu4.20
X-Varnish: 2131034848
Request Headers
view source
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
Connection: keep-alive
Cookie: zap=42; __cfduid=d6c4d420c9d82af00acdd13d65aaec4bc1486334436
Host: www.wa4e.com
Referer: http://www.wa4e.com/code/sessions/cookie.php
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.95 Safari/537.36
2 requests | 2.4 KB transferred...
Application
Manifest
Service Workers
Clear storage
Storage
Local Storage
Session Storage
IndexedDB
Web SQL
Cookies
Cache
Cache Storage
Application Cache
Frames
top


Time
Browser
Browser
zap=42
Browser
zap=42
Browser
zap=42
zap=42
Apache
setcookie()
$_COOKIE
$_COOKIE
$_COOKIE
PHP
Code

PHP Sessions

http://www.wa4e.com/code/sessions.zip


Time
Browser
DOM
Send Request
Parse Response
JavaScript
Web Server
Sessions:
42
6f
3e
Apache
PHP
sessfun.php
PDO
Database Server
MySql
RRC/HTTP
SQL

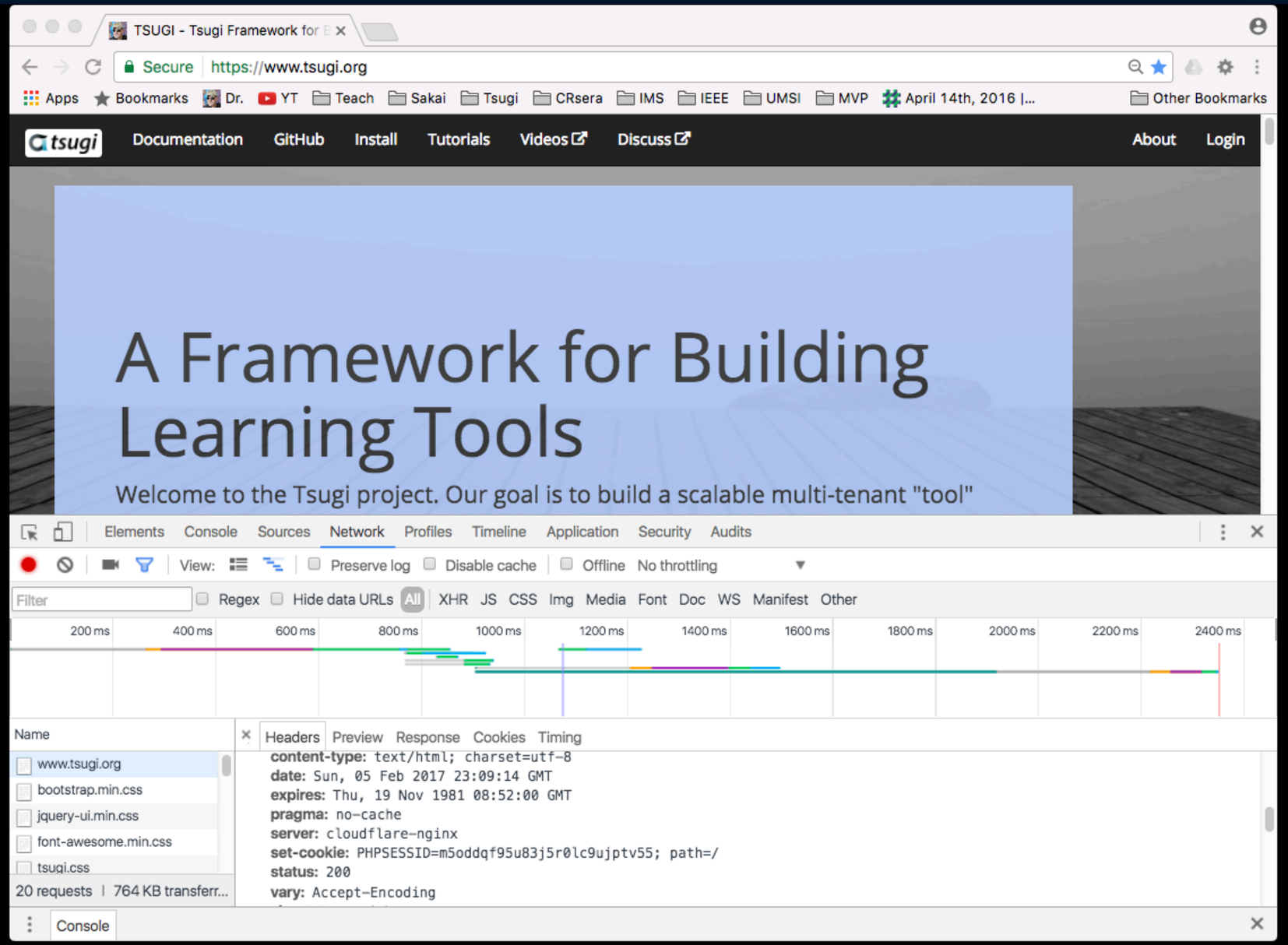
In the Server - Sessions

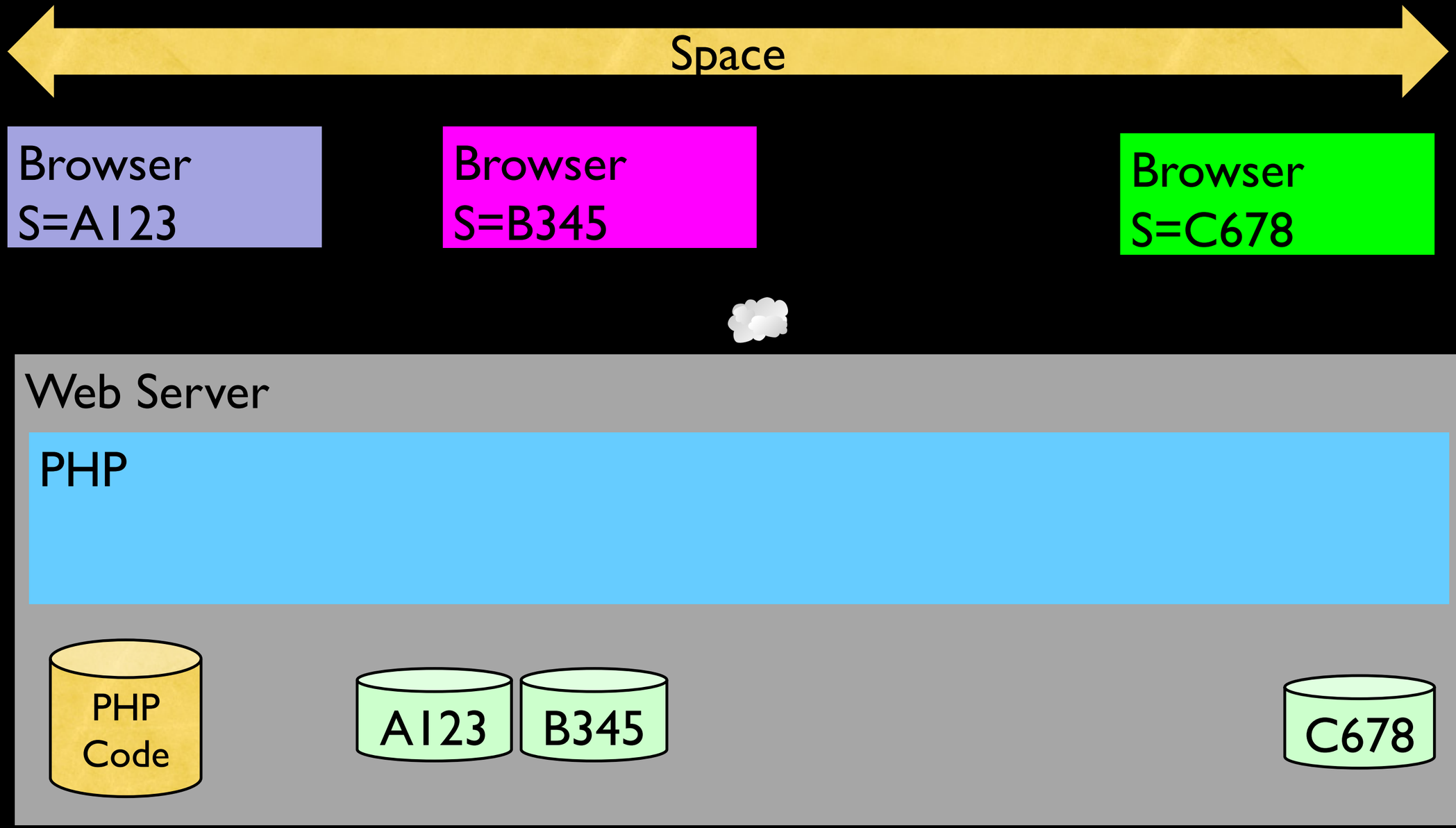
- In most server applications, as soon as we meet a new (unmarked) browser we create a session.
- We set a session cookie to be stored in the browser, which indicates the session id in use – gives this browser a unique "mark".
- The creation and destruction of sessions is handled by a web framework or some utility code that we use in our applications.

Session Identifier

- A large, random number that we place in a browser cookie the first time we encounter a browser
- This number is used to pick from the many sessions that the server has active at any one time.
- Server software stores data in the session that it wants to have from one request to another from the same browser.
- Shopping cart or login information is stored in the session in the server.

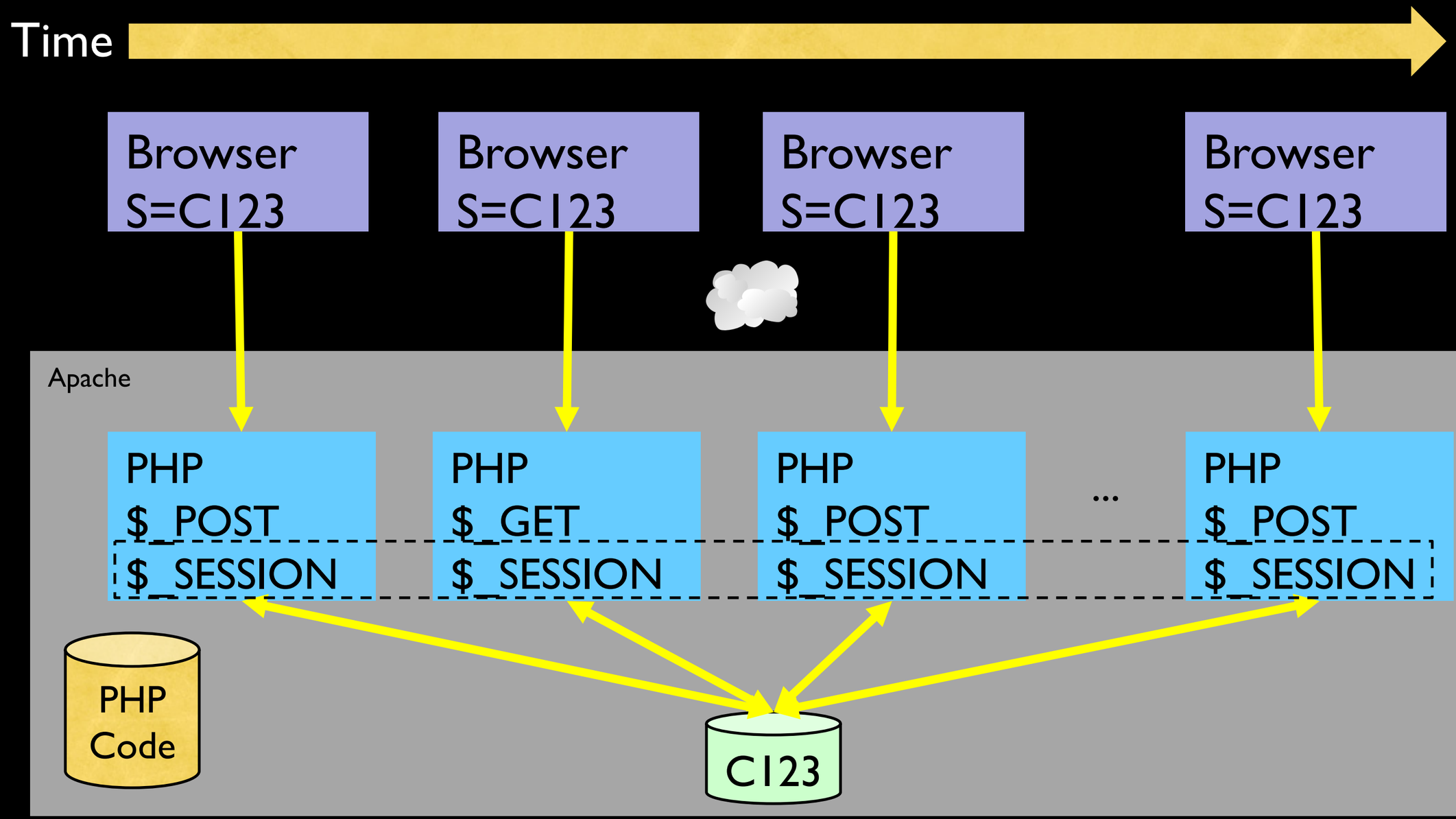
1. Open a fresh browser.
2. Turn on network tab of developer console.
3. Go to www.tsugi.org
4. Find the first page retrieved.
5. Look at the response headers and find set-cookie.




Space
Browser
S=A123
Browser
S=B345
Browser
S=C678
Web Server
PHP
PHP
Code
A123
B345
C678

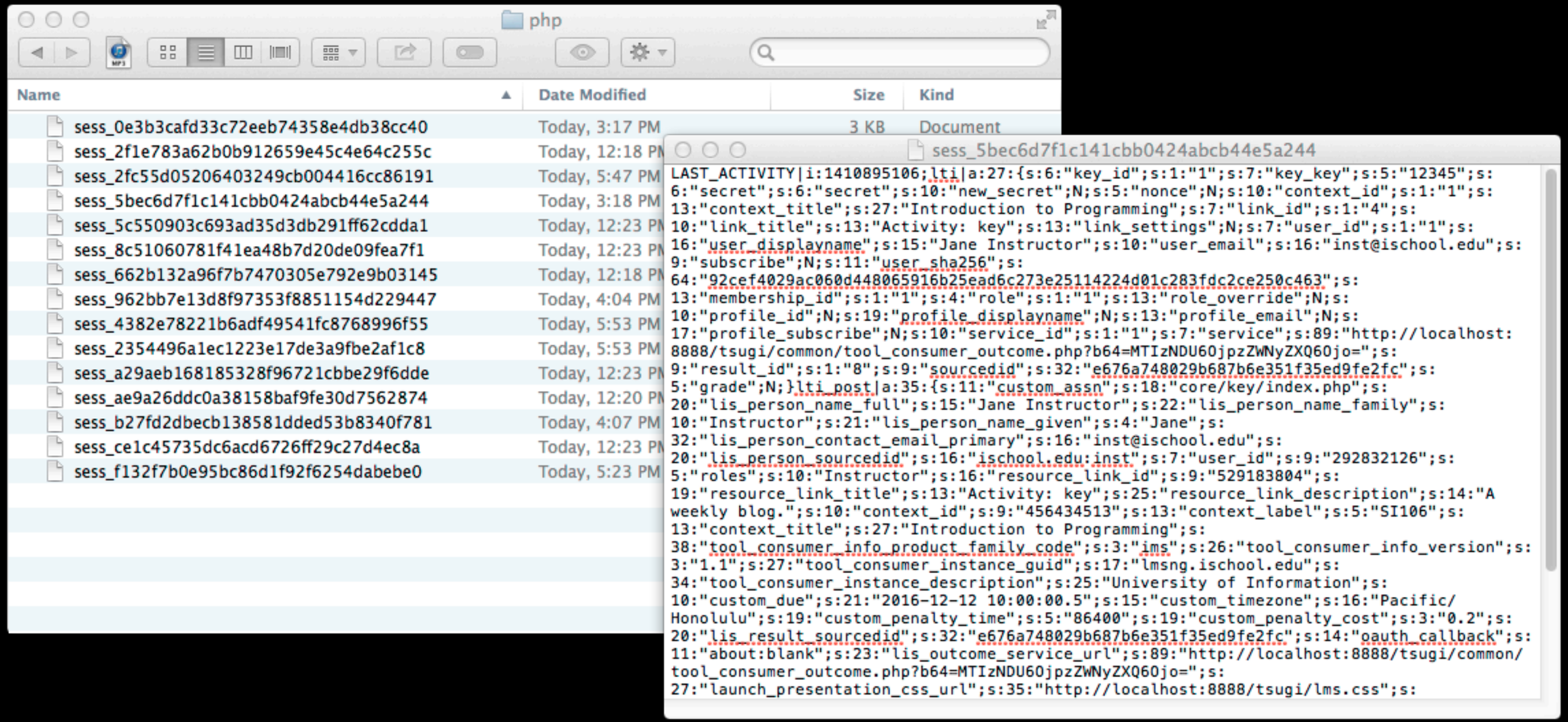
PHP Sessions

- We can establish / initialize a PHP session by calling session_start() before any output has come out.
- If the user has cookies set, we can use the array $_SESSION to store data from one request to the next with a particular browser.
- We have a bit of data that persists from one request to the next.
- By default, these are stored in a temporary folder on disk.


Time
Browser S=C123
Browser S=C123
Browser S=C123
Browser S=C123
Apache
PHP $_POST $_SESSION
PHP $_GET $_SESSION
PHP $_POST $_SESSION
...
PHP $_POST $_SESSION
PHP Code
C123

PHPInfo – session.save_path

session.name	PHPSESSID	PHPSESSID
session.referer_check	*no value*	*no value*
session.save_handler	files	files
session.save_path	/Applications/MAMP/tmp/php	/Applications/MAMP/tmp/php
session.serialize_handler	php	php
session.upload_progress.cleanup	On	On
session.upload_progress.enabled	On	On
session.upload_progress.freq	1%	1%



(On a Mac) /Applications/MAMP/tmp/php

session_start

(PHP 4, PHP 5)

session_start — Initialize session data

Description

Report a bug

bool **session_start** (void)

session_start() creates a session or resumes the current one based on a session identifier passed via a GET or POST request, or passed via a cookie.

To use a named session, call session_name() before calling **session_start()**.

When session.use_trans_sid is enabled, the **session_start()** function will register an internal output handler for URL rewriting.

If a user uses *ob_gzhandler* or similar with ob_start(), the function order is important for proper output. For example, *ob_gzhandler* must be registered before starting the session.

http://php.net/manual/en/function.session-start.php

session_destroy

(PHP 4, PHP 5)

session_destroy — Destroys all data registered to a session

Description

Report a bug

bool **session_destroy** (void)

session_destroy() destroys all of the data associated with the current session. It does not unset any of the global variables associated with the session, or unset the session cookie. To use the session variables again, session_start() has to be called.

In order to kill the session altogether, like to log the user out, the session id must also be unset. If a cookie is used to propagate the session id (default behavior), then the session cookie must be deleted. setcookie() may be used for that.

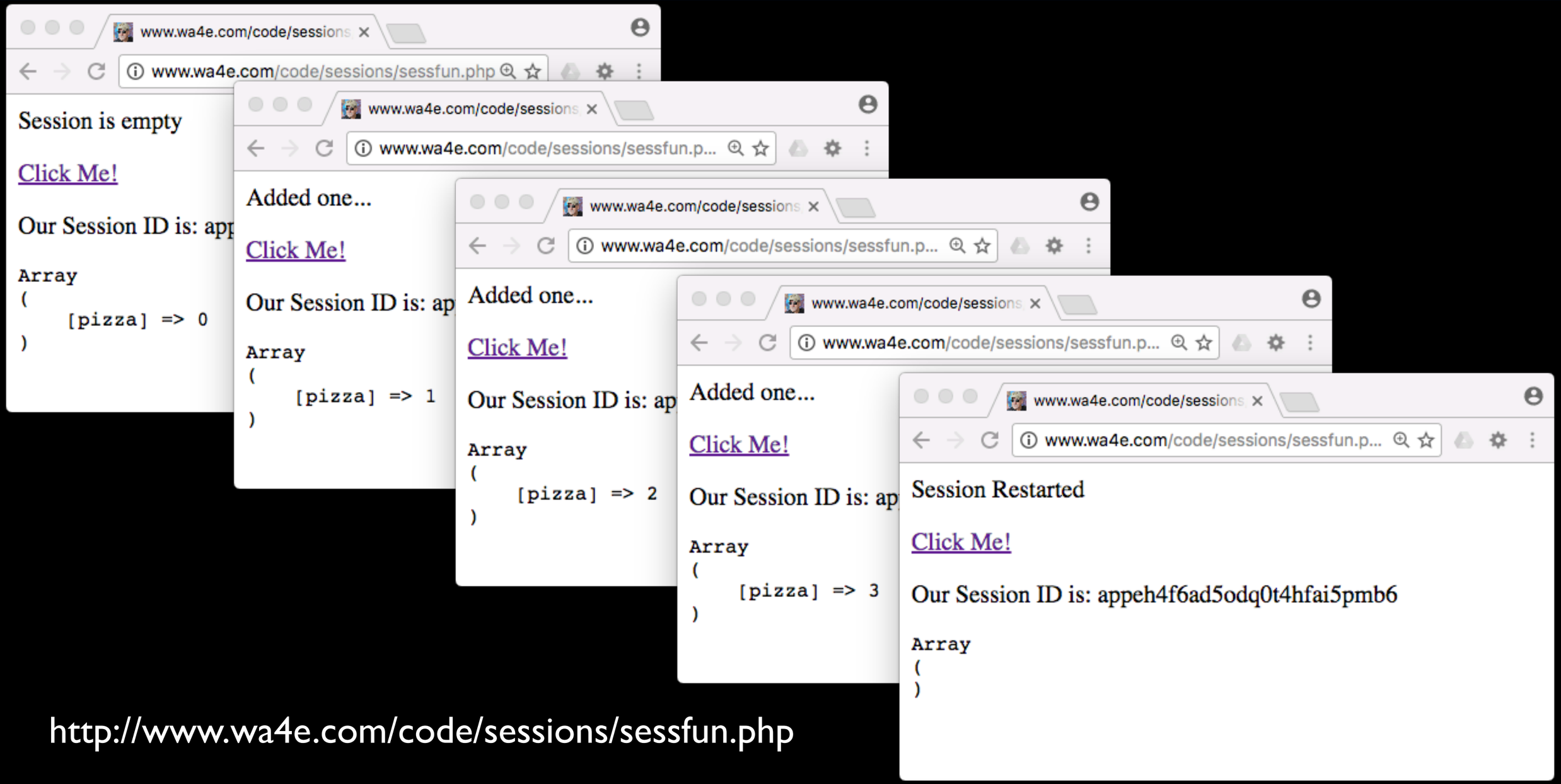
http://php.net/manual/en/function.session-destroy.php

```php
<?php
  // Note - cannot have any output before this
  session_start();

  if ( ! isset($_SESSION['pizza']) ) {
     echo("<p>Session is empty</p>\n");
     $_SESSION['pizza'] = 0;
   } else if ( $_SESSION['pizza'] < 3 ) {
     $_SESSION['pizza'] = $_SESSION['pizza'] + 1;
     echo("<p>Added one...</p>\n");
  } else {
    session_destroy();
    session_start();
    echo("<p>Session Restarted</p>\n");
  }
?>
<p><a href="sessfun.php">Click Me!</a></p>
<p>Our Session ID is: <?php echo(session_id()); ?></p>
<pre>
<?php print_r($_SESSION); ?>
</pre>
```

http://www.wa4e.com/code/sessions/sessfun.php

http://www.wa4e.com/code/sessions/sessfun.php



http://www.wa4e.com/code/sessions/sessfun.php

Sessions Without Cookies

PHP Sessions Without Cookies

- For a simple application handling login, logout, and shopping cart-like information, cookie sessions are sufficient.
- But if an application needs to function within an iframe, or have more than one session active (i.e., multiple tabs to the same site), we cannot use session cookies.
- PHP has nice support for maintaining a session without a cookie.

nocookie.php

nocookie.php

```php
<?php
    // Tell PHP we won't be using cookies for the session
    ini_set('session.use_cookies', '0');
    ini_set('session.use_only_cookies',0);
    ini_set('session.use_trans_sid',1);

    session_start();

// Start the view
?>
<p><b>No Cookies for You!</b></p>
```

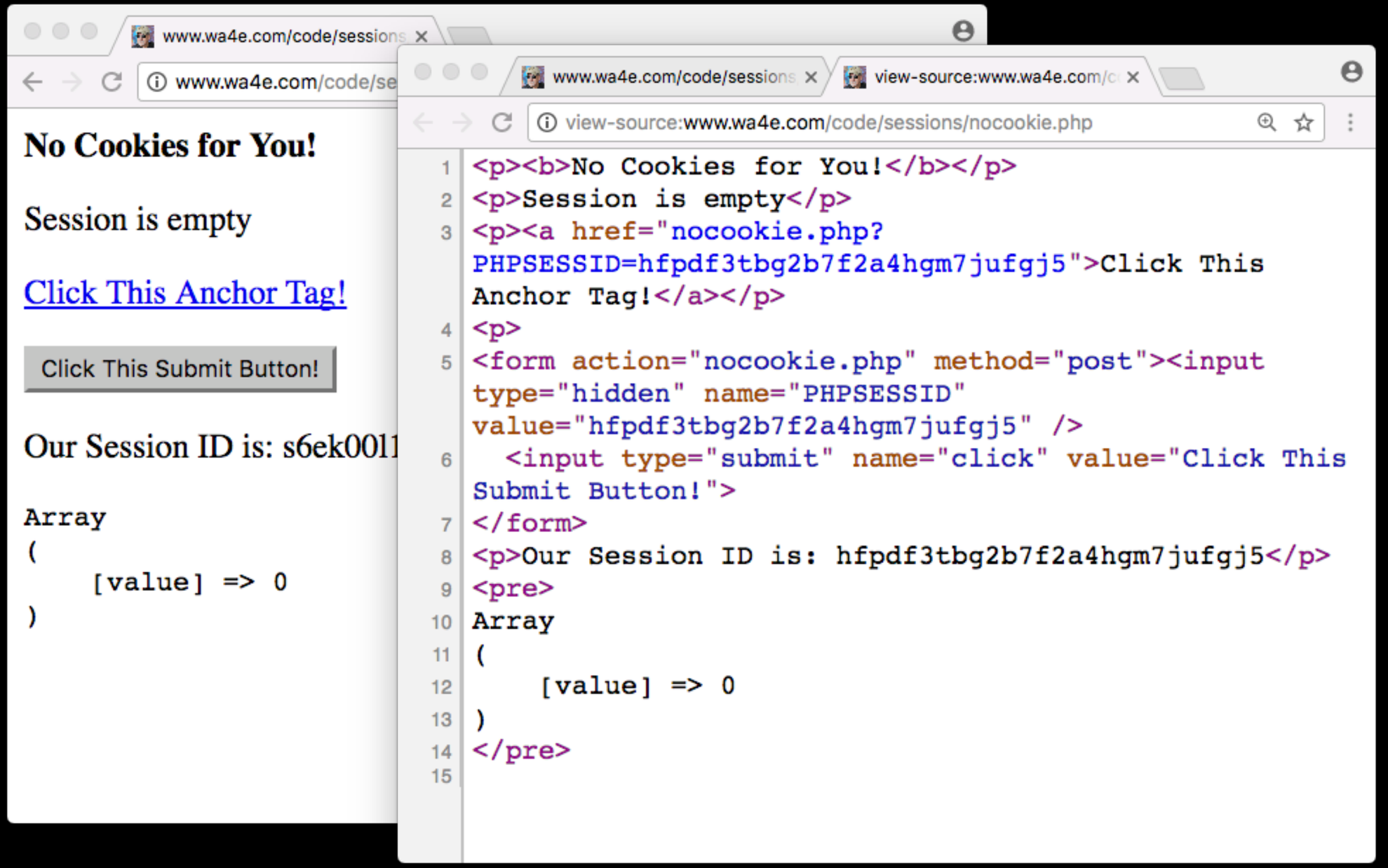
nocookie.php

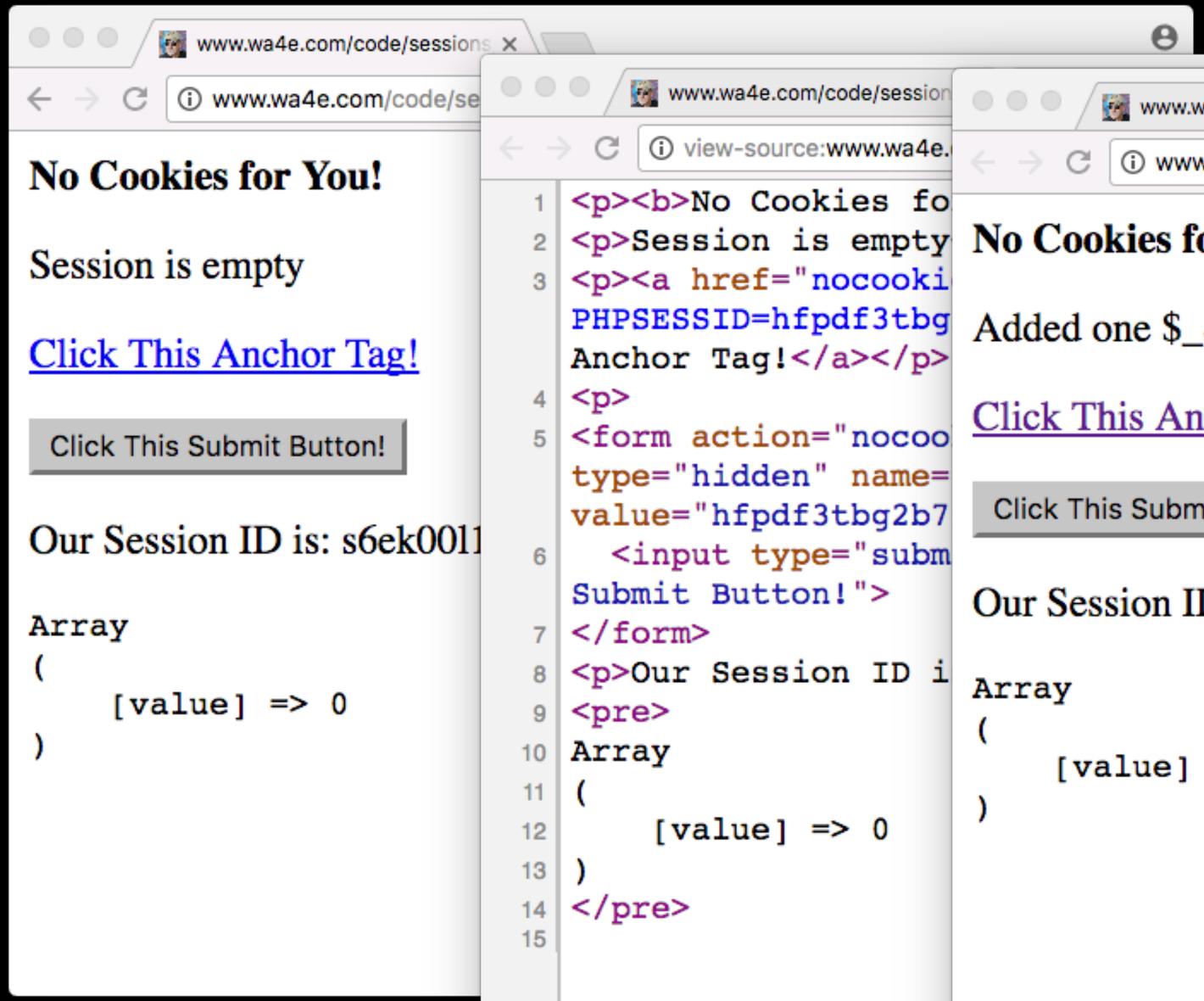
```
<p><b>No Cookies for You!</b></p>
<?php
    if ( ! isset($_SESSION['value']) ) {
        echo("<p>Session is empty</p>\n");
        $_SESSION['value'] = 0;
    } else if ( $_SESSION['value'] < 3 ) {
        $_SESSION['value'] = $_SESSION['value'] + 1;
        echo("<p>Added one \$_SESSION['value']=".$_SESSION['value'].
            "</p>\n");
    } else {
        session_destroy();
        session_start();
        echo("<p>Session Restarted</p>\n");
    }
?>
<p><a href="nocookie.php">Click This Anchor Tag!</a></p>
```

nocookie.php

```
?>
<p><a href="nocookie.php">Click This Anchor Tag!</a></p>
<p>
<form action="nocookie.php" method="post">
  <input type="submit" name="click" value="Click This Submit Button!">
</form>
<p>Our Session ID is: <?php echo(session_id()); ?></p>
<pre>
<?php print_r($_SESSION); ?>
</pre>
```


www.wa4e.com/code/sessions
www.wa4e.com/code/sessions/nocookie.php
No Cookies for You!
Session is empty
Click This Anchor Tag!
Click This Submit Button!
Our Session ID is: s6ek00l1md7n2v8smpfo7b3lk0
Array
(
[value] => 0
)


www.wa4e.com/code/sessions
www.wa4e.com/code/se
No Cookies for You!
Session is empty
Click This Anchor Tag!
Click This Submit Button!
Our Session ID is: s6ek0011
Array
(
[value] => 0
)
www.wa4e.com/code/sessions
view-source:www.wa4e.com/c
view-source:www.wa4e.com/code/sessions/nocookie.php
1 No Cookies for You!
2 Session is empty
3 Click This Anchor Tag!
4
5 <form action="nocookie.php" method="post"><input type="hidden" name="PHPSESSID" value="hfpdf3tbg2b7f2a4hgm7jufgj5" />
6 <input type="submit" name="click" value="Click This Submit Button!">
7 </form>
8 Our Session ID is: hfpdf3tbg2b7f2a4hgm7jufgj5
9 <pre>
10 Array
11 (
12 [value] => 0
13)
14 </pre>
15


www.wa4e.com/code/sessions
www.wa4e.com/code/se
No Cookies for You!
Session is empty
Click This Anchor Tag!
Click This Submit Button!
Our Session ID is: s6ek001
Array
(
[value] => 0
)

www.wa4e.com/code/session
view-source:www.wa4e.
1 <b>No Cookies fo
2 Session is empty
3 <a href="nocooki
PHPSESSID=hfpdf3tbg
Anchor Tag!
4
5 <form action="nocoo
type="hidden" name=
value="hfpdf3tbg2b7
6 <input type="subm
Submit Button!">
7 </form>
8 Our Session ID i
9 <pre>
10 Array
11 (
12 [value] => 0
13)
14 </pre>
15

www.wa4e.com/code/
view-source:www.wa4
www.wa4e.com/code/
www.wa4e.com/code/sessions/nocookie.php?PHPSESSID=hfpdf3tbg2b...
No Cookies for You!
Added one $_SESSION['value']=1
Click This Anchor Tag!
Click This Submit Button!
Our Session ID is: hfpdf3tbg2b7f2a4hgm7jufgj5
Array
(
[value] => 1
)

A Whole Host of Issues...

- Session id is not automatically added in JavaScript, Ajax, Redirect, or other elements of HTML.
- With the session id on the URL, folks can email URLs or even bookmark them and be logged in.
- We will come back to these...

Summary

- HTTP Cookies
- Sessions
- Using Sessions in PHP

Acknowledgements / Contributions

Initial Development: Charles Severance, University of Michigan School of Information

Insert new Contributors and Translators here including names and dates

Continue new Contributors and Translators here

Copyright Attribution

- Cookie Image: By brainloc on sxc.hu (Bob Smith) (stock.xchng) [CC BY 2.5 (http://creativecommons.org/licenses/by/2.5)], via Wikimedia Commons