

EVERYDAY APACHE MAGIC

# Ryan Smith

*Technical SEO Consultant  
Web Application Developer  
Übergeek*

**HackingSearch.com**

*(These Slides Available There Now)*

# EVERYDAY APACHE MAGIC

- How .htaccess files work
- Password-protected pages
- When URL's change...
- Wordpress permalinks
- Setting PHP options

# HOW .HTACCESS FILES WORK

- Just a regular text file.
- Overrides global / default settings.
- Applies to current dir & all sub-dirs.
- File read again for every new request.
- Enough rope to hang yourself.

# PASSWORD PROTECTED PAGES

- Actually it's the directory.
- The Magic Protection Spell Incantation:

```
AuthType Basic  
AuthName "My Protected Area"  
AuthUserFile /path/to/.htpasswd  
Require valid-user
```

- Won't work without a .htpasswd file:

```
username:[encrypted password]
```

- From the Command Line:

```
htpasswd -c .htpasswd username
```

- <http://www.htaccesstools.com/htpasswd-generator/>



WHEN URL'S CHANGE

# ~~WHEN URL'S CHANGE~~

- Good URL's don't change.
- Apache's "Content Negotiation" = *Pure Magic*

**/products/blue-widgets**

is the same as:

**/products/blue-widgets.html**

# ~~WHEN URL'S CHANGE~~

- Content Negotiation permits other cool tricks:

**/products.php?id=12345**

could be changed to:

**/products/12345**

or even:

**/products/blue-widgets-12345** (*better!*)

- Needs some code to read ID from URL...

# ~~WHEN URL'S CHANGE~~

**Example URL: /products/blue-widgets-12345**

**Example code for products.php:**

```
<?php
```

```
$URLparts = explode ( '/',  
    $_SERVER[ 'REQUEST_URI' ] );
```

```
$nameParts = explode ( '-',  
    $URLparts [ count($URLparts) - 1 ] );
```

```
$id = $nameParts[ count($nameParts) - 1 ];
```

```
?>
```

# WHEN URL'S CHANGE

- Breaks everybody's bookmarks.
- External inbound links broken = *lost PageRank!*
- Search engines won't update immediately.
- **Changed URL's lose ALL search traffic!**
- **Search traffic is *FREE, DAMMIT!***

# WHEN URL'S CHANGE

- **Solution #1:** Don't change the URL (*ha, ha...*)
- **Solution #2:** Redirect requests for all old URL's to the new URL.

# WHEN URL'S CHANGE

- There's more than one way to screw this up.
- META refresh tags don't cut it for SEO:  
`<meta name="refresh" content="0;/new/url">`
- JavaScript redirects don't cut it either:  
`top.location.href='/new/url';`
- Server-side one-liners aren't good enough either:  
PHP: `header('Location: /new/url');`  
ASP: `Response.AddHeader('Location', '/new/url')`  
CF: `<cfheader name="Location" value="/new/url/">`

# IT'S ALL ABOUT STATUS [CODES]

Browser Requests URL  
*/products/blue-widgets*



*/home/user/html/products/blue-widgets.html*

Server Returns Contents of  
*blue-widgets.html* with  
Status: 200

# IT'S ALL ABOUT STATUS [CODES]

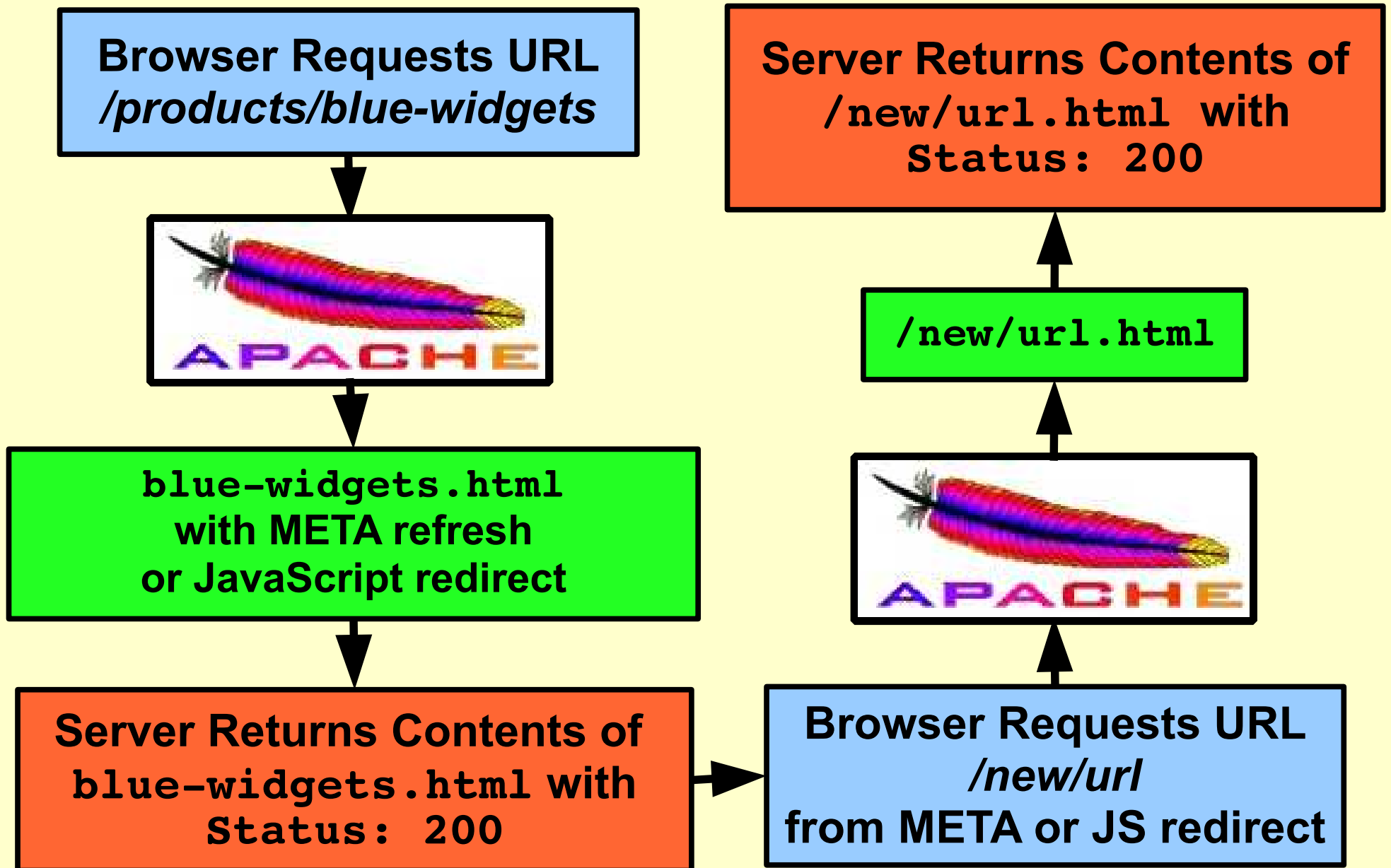
Browser Requests URL  
*/products/blue-widgets*



Error: File Not Found

Server Returns Error Page  
with  
Status: 404

# IT'S ALL ABOUT STATUS [CODES]



# AM I BOT OR NOT?

Bot Requests URL  
`/products/blue-widgets`



`blue-widgets.html`  
with META refresh  
or JavaScript redirect

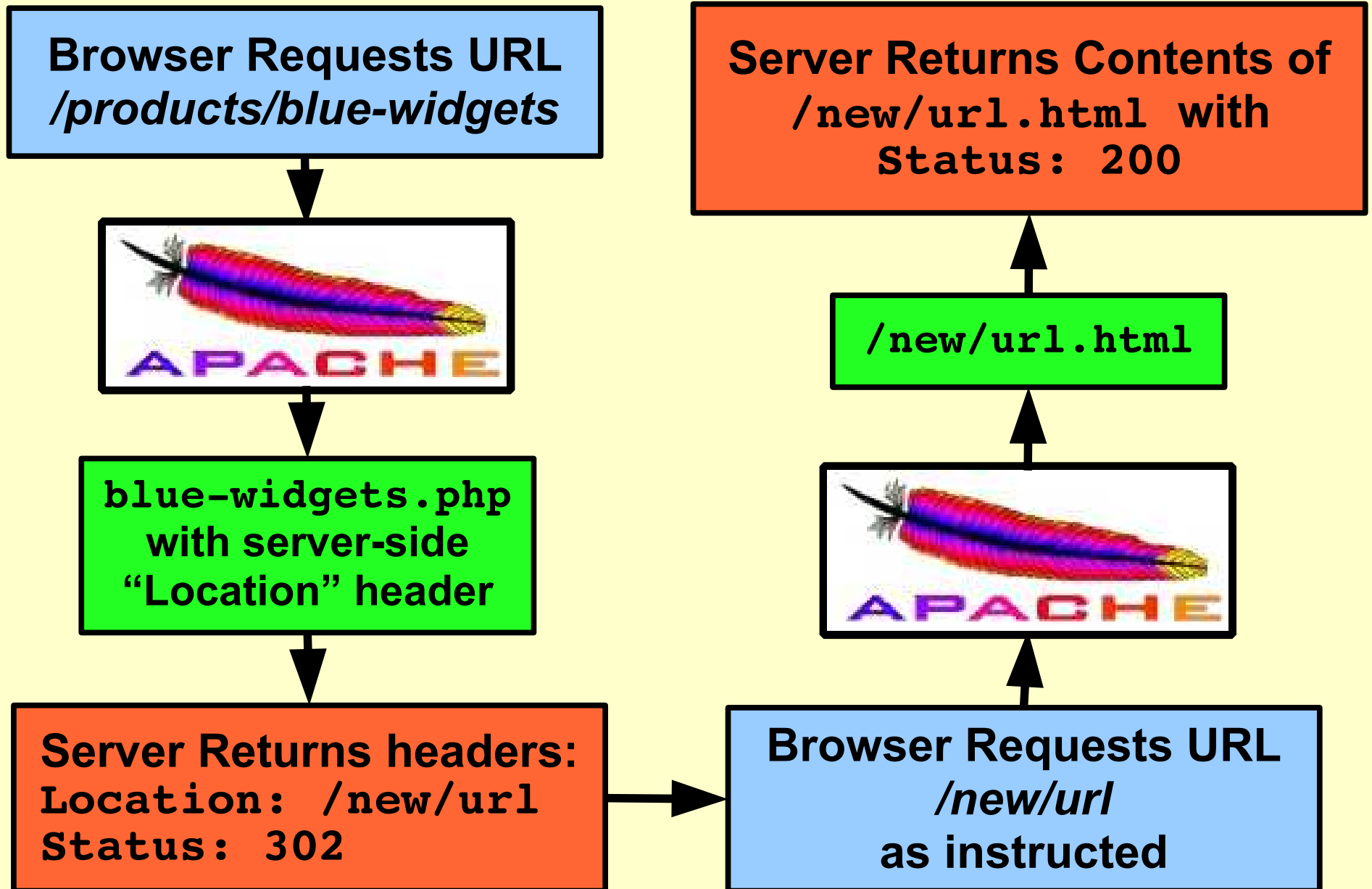
Server Returns Contents of  
`blue-widgets.html` with  
Status: 200

She'll never notice your  
new URL at all.

What a *Frigid Bitch!*



# IT'S ALL ABOUT STATUS [CODES]



# AM I BOT OR NOT?

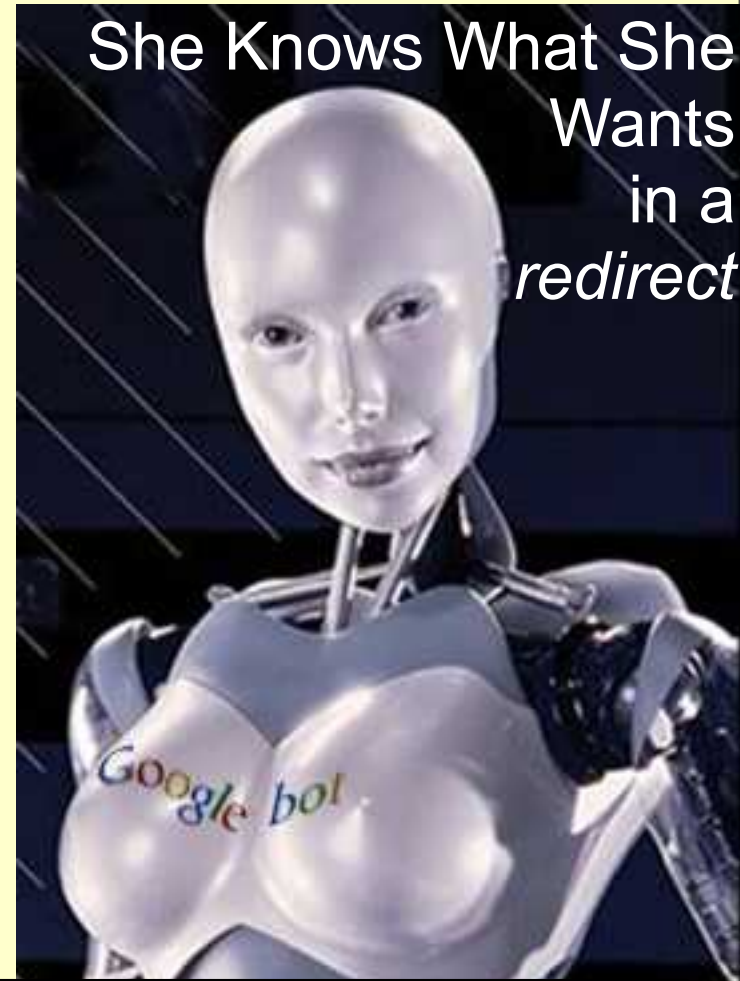
Bot Requests URL  
`/products/blue-widgets`



`blue-widgets.php`  
with server-side  
"Location" header

Server Returns headers:  
Location: `/new/url`  
Status: 302

Turned off by your 302 as  
too *Temporary*, she ignores  
your new URL entirely.



# AM I BOT OR NOT?

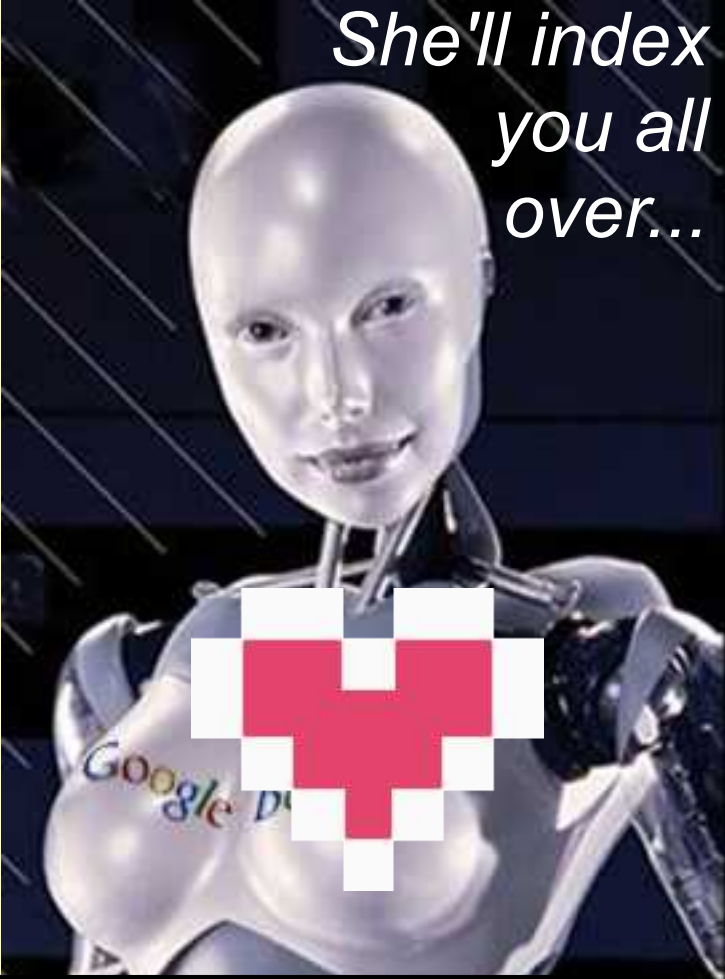
Bot Requests URL  
`/products/blue-widgets`



Proper 301 Redirect

Server Returns headers:  
Location: `/new/url`  
Status: 301

She digs your groovy 301  
*Permanent* Redirect and  
wants to see your `/new/url`



# WHEN URL'S CHANGE

- Yes, this is mildly ridiculous.
- Can't do 301's in META or JavaScript.
- Options: Web scripting (PHP, etc.) or `.htaccess`

# WEB SCRIPTING 301'S

- **PHP:**

```
header('HTTP/1.1 301 Moved Permanently');  
header('Location: /new/url');
```

- **ASP:**

```
Response.Status = '301 Moved Permanently'  
Response.AddHeader('Location', '/new/url')
```

- **ColdFusion:**

```
<cfheader statuscode="301" statustext="Moved  
  Permanently">  
<cfheader name="Location" value="/new/url/">
```

# THE .HTACCESS METHOD

- **Just A Single URL:**

```
Redirect 301 /old/url /new/url
```

**OR**

```
Redirect Permanent /old/url /new/url
```

- **All URL's Matching a Pattern:**

```
RewriteEngine On
```

```
RewriteCond %{REQUEST_URI} ^/(.*)-old\.html$
```

```
RewriteRule . http://%{HTTP_HOST}//$1-new [R=301,L]
```

- **Consolidating multiple old URL's into new URL:**

```
RewriteCond %{REQUEST_URI} ^/oldpage\.html$ [OR]
```

```
RewriteCond %{REQUEST_URI} ^/other-old-page\.html$
```

```
RewriteRule . http://%{HTTP_HOST}/newpage.php [R=301,L]
```

# WHY DUPLICATE URL'S SUCK

- Search engines treat them as distinct pages.
- Divides your PageRank & precious “link juice”.
- Engines consider it duplicate content, so your rankings take a hit from anti-spam algorithms.

# THE .HTACCESS METHOD

- **Canonical Hostname Redirect (*the 'www' thing...*)**

```
RewriteCond %{HTTP_HOST} !^www\.
```

```
RewriteRule ^(.*)$ http://www.%{HTTP_HOST}/$1 [R=301,L]
```

- **Or, to enforce a *no-dubyas* policy:**



```
RewriteCond %{HTTP_HOST} ^www\.
```

```
RewriteRule ^(.*)$ http://%{HTTP_HOST}/$1 [R=301,L]
```

- **Make Slash your official homepage URL:**

```
RewriteCond %{REQUEST_URI} ^/index\.html [OR]
```

```
RewriteCond %{REQUEST_URI} ^/some-dead-page\.html
```

```
RewriteRule . http://%{HTTP_HOST}/ [R=301,L]
```



# WORDPRESS PERMALINKS

- WTF is a “permalink” anyway?

**Wikipedia:** “a URL that points to a specific blog or forum entry after it has passed from the front page to the archives.”

Default WordPress story URL:

**<http://yourblog.com/?p=12345>**

Permalink URL (*search friendly*):

**<http://yourblog.com/post-title/>**

# WORDPRESS PERMALINKS

- Settings > Permalinks

## Common settings

Default

`http://hackingsearch.com/?p=123`

Day and name

`http://hackingsearch.com/2009/06/15/sample-post/`

Month and name

`http://hackingsearch.com/2009/06/sample-post/`

Numeric

`http://hackingsearch.com/archives/123`

Custom Structure

`/%year%/monthnum%/postname%/`

# WORDPRESS PERMALINKS

- Wordpress gives you the `.htaccess` code:

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
</IfModule>
```

- Don't forget to install the All-In-One SEO Pack or Platinum SEO Pack.

# SETTING PHP OPTIONS

- Can be useful either per-site or per-directory.
- Look for PHP code in all .html and .htm files:

```
AddType application/x-httpd-php .html .htm
```

- Include site-wide headers and footers:

```
php_value auto_prepend_file header.php
```

```
php_value auto_append_file none
```

# SETTING PHP OPTIONS

- Error Handling:

php\_value display\_errors 1 = **ON**

php\_value display\_errors 0 = **OFF**

- Max Execution Time:

php\_value max\_execution\_time 300 (**5 mins**)

- Memory Limit:

php\_value memory\_limit 1024M (**1 GB**)

# SETTING PHP OPTIONS

Complete List of Options:

<http://www.php.net/manual/en/ini.list.php>

# IN SUMMARY

- .htaccess files aren't so scary.
- 301-redirect all changed/dupe URL's.
- Use permalinks for blogs & forums.
- Bully PHP around as needed.
- With great power comes great responsibility. So don't screw up.

EVERYDAY APACHE MAGIC

**Thank You**

Ryan Smith

ryan@hackingsearch.com

Twitter: @hackingsearch

*Slides Available at:*

**HackingSearch.com**