Web Site Tune-Up
London Perl Workshop 2017
{Perl Hacks}

Improve Your
Googlejuice
London Perl Workshop 2017
{Perl Hacks}

Web Site Tune-Up

- Making your site attractive to Google
- Warning: Perl-free zone

Plug

- This is a new course
- I'll be running a longer version in Glasgow in 2018
- All feedback welcome
- dave@perlhacks.com

Web Site Tune-Up

- Most web sites measure success by number of visitors
- Selling products or services
- Promoting ideas
- Raising awareness
- Eyeballs == success
- Google ranking == eyeballs

Google Ranking

- If Google thinks your web site is important or useful ...
- Google will show your site in more search results pages ...
- And that brings more visitors

Improve Your Googlejuice

- Simple techniques to make your site more attractive to Google
- Make it easier for Google to understand what your site is about
- Make it easier for Google to understand the structure of your site
- Anyone can use these techniques
- Most people don't
- Which is why they work

What We Will Cover

Web Site Structure

- Semantic HTML
- URL structure
- Changing URLs
  - Redirections
  - 410
- Structured data
- Sitemaps

Web Site Structure

- Open Graph
- Link tags
- Robots.txt
- Useful HTTP response headers
- 'Nofollow' attributes

Security

- HTTPS
- HSTS

Javascript

- Does Google render Javascript-heavy sites?
- Pre-rendering vs cloaking

Mobile Pages

- Mobile-first index
- Responsive pages
- AMP

Measurement

- Google Webmaster Tools
- Google Analytics

What We Won't Cover

- Keyword research & analysis
- Content marketing

Semantic HTML

London Perl Workshop 2017

{Perl Hacks}

Non-Semantic HTML

```
<html>
  <head><title>Non-Semantic HTML</title></head>
  <body>
    <p><font size="2" color="ref"
face="Arial">Important Header</font></p>
    <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit</p>
    <p>etc…,/p>
  </body>
</html>
```

Semantic HTML

```
<html>
  <head>
    <title>Semantic HTML</title>
    <link rel="stylesheet" type="text/css"
href="style.css">
  </head>
  <body>
    <h1>Important Header</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit</p>
    <p>etc…,/p>
  </body>
</html>
```

Semantic HTML

Wikipedia:

> "Semantic HTML is the use of HTML mark-up to reinforce the semantics, or meaning, of the information in web pages and web applications rather than merely to define its presentation or look."

Content vs Presentation

- Use HTML to mark-up the content of your page
- Use CSS to control the presentation of your content

Semantic Mark-Up

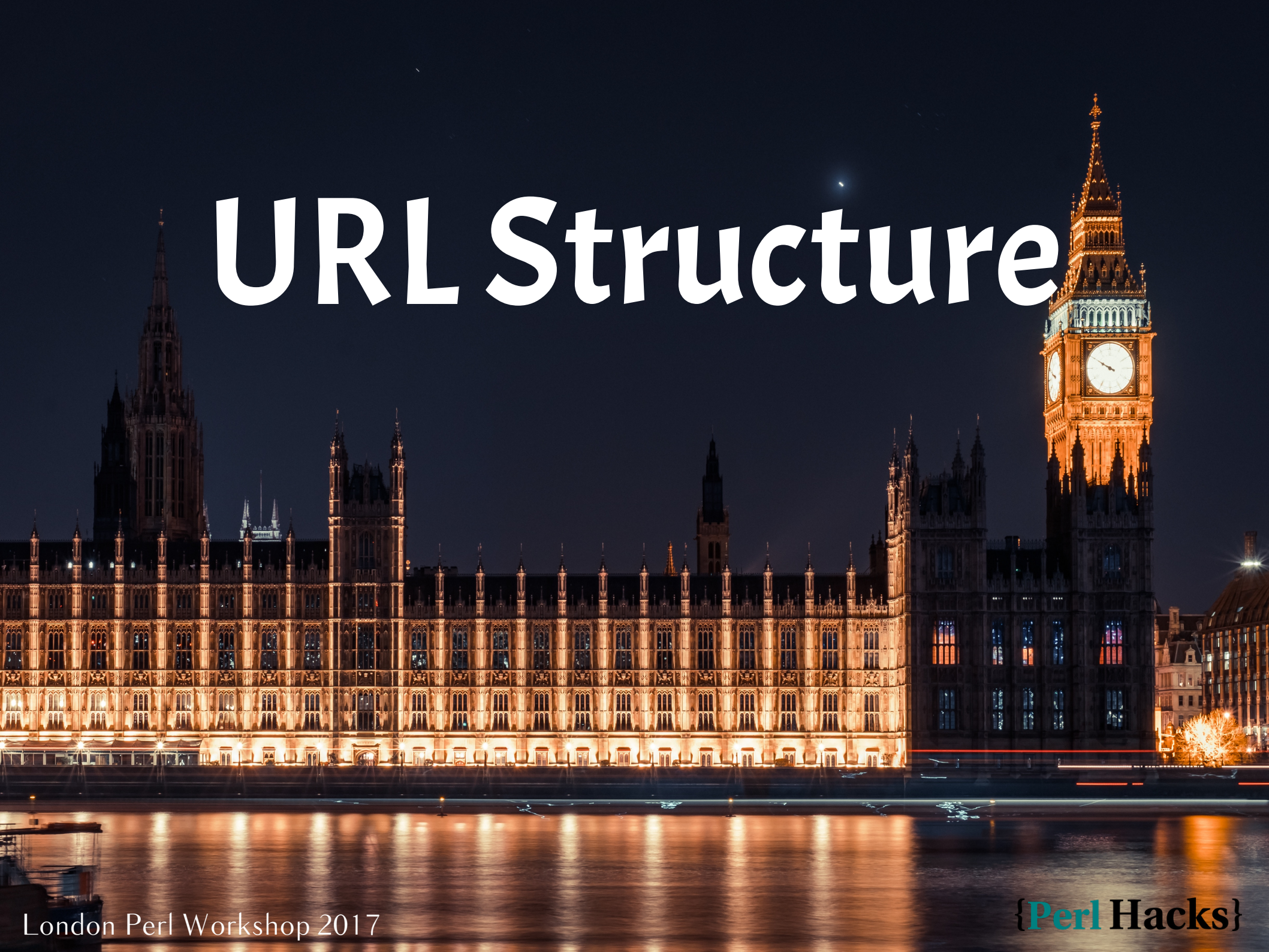
- Mark-up that describes the structure of your content
- Headers, paragraphs, lists, etc
  - Heading levels
- Use `<div class="...">` and `<span class="...">`
- See also: structured data

Non-Semantic Mark-Up

- Mark-up that describes presentation
- `<font>`, `<center>`, `<blink>`, etc.
- Presentation attributes: `background`, `align`, `nowrap`, etc.
- Can all be handled in CSS
- `<i>` vs `<em>`
- `<b>` vs `<strong>`

HTML5

- HTML5 adds many new semantic elements
- `<main>`, `<section>`, `<article>`, `<header>`, `<footer>`, `<aside>`, `<nav>` and `<figure>`
- Well worth using



URL Structure

London Perl Workshop 2017

{Perl Hacks}

URL Structure

- Think about your URL structure
- Mirror the structure of your site

Good URLs are

- Meaningful
- Hackable
- Permanent

Meaningful URLs

- Which is more meaningful?
- `/2017/10/lpw-is-awesome`
- `/?p=12345678`
- Users don't care about URLs(?)
- Google **does** care

Meaningful URL tips

- Titles not IDs
- Use - to separate words
- Minimise parameters
- Avoid session information
- Avoid tech-specific URLs
- `mod_rewrite` is your friend

Hackable URLs

- Users don't care about URLs
- Some do
- Some will edit the URL in the location bar
- Let them
- Ensure pages exist
- `/news/2017/10/lpw-is-awesome`

Permanent URLs

- Cool URLs don't change
- Sometimes they have to
- 404 is bad news
- Use redirections
- Sometimes URLs go away
- HTTP response 410

Redirecting URLs

London Perl Workshop 2017

{Perl Hacks}

Redirecting URLs

- Sometimes URLs have to change
  - Is this change essential?
  - Or is your new CMS too restrictive?
- Return 301 to redirect browsers
- Transfers Google page rank too
- Browsers cache this redirection

Removing URLs

- Sometimes a page goes away for good reasons
- Don't just let it 404
- Redirect somewhere useful
  - Or (at least) the home page
- Or return 410 (Gone away)
- Google will it remove from index

Maintaining Redirections

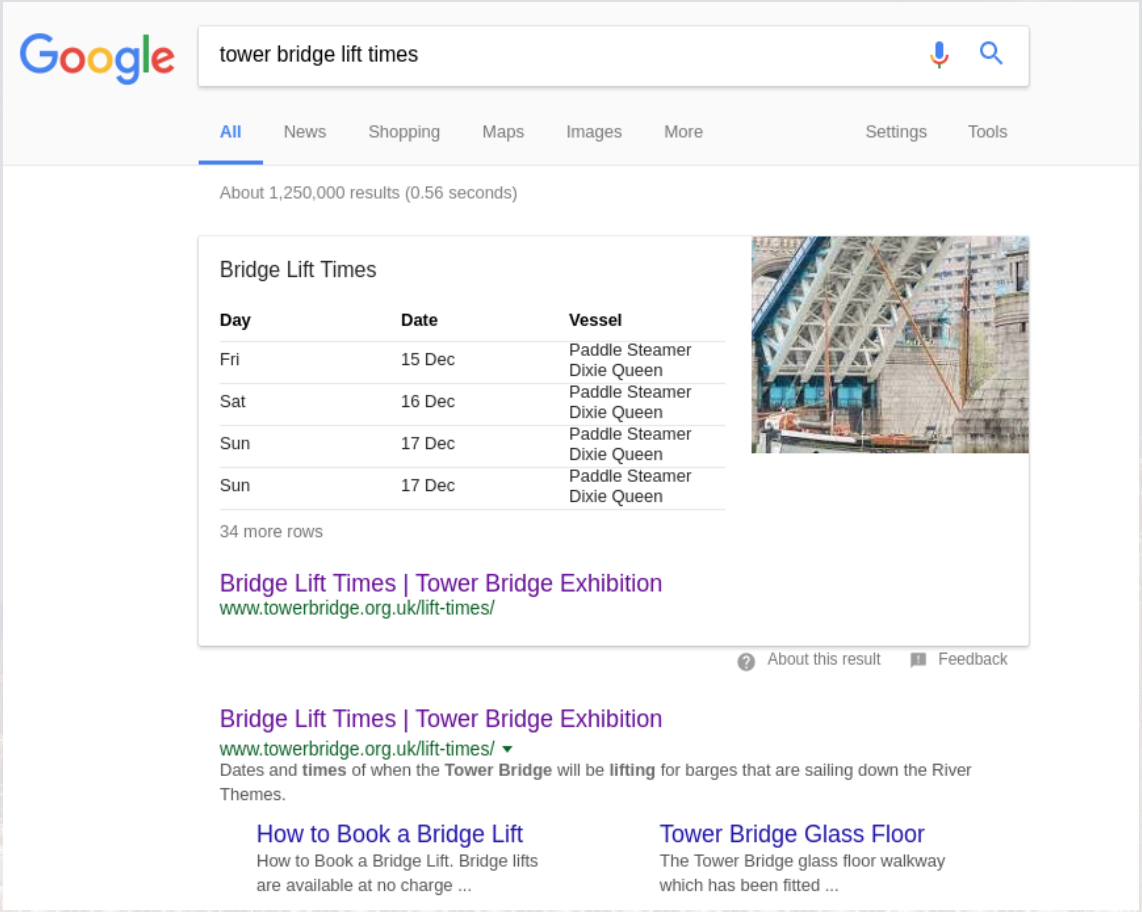
- We remove `/foo` and redirect to `/bar`
- Time passes
- We remove `/bar` and redirect to `/`
- Requesting `/foo` redirects twice
- Redirection chains are bad
- Update the `/foo` redirection
- Can of worms

Structured Data
London Perl Workshop 2017
{Perl Hacks}

Structured Data

- Google needs to parse your content
- In order to understand it
- Make it easier
- Include structured versions of your data
- Google Knowledge Graph
- Position Zero

Position Zero



Types of Structured Data

- Microdata
  - Built into your HTML
- JSON-LD ("Linked Data")
  - In a separate `<script>` tag

Microdata

- Uses extensions to HTML mark-up
- New HTML attributes `itemscope`, `itemtype` and `itemname`

Microdata Example

```
<div itemscope itemtype="http://schema.org/Movie">
  <h1 itemprop="name">Avatar</h1>
  <div itemprop="director" itemscope
       itemtype="http://schema.org/Person">
    Director: <span itemprop="name">James Cameron</span>
    (born <time itemprop="birthDate"
      datetime="1954-08-16">August 16, 1954</time>)
  </div>
  <span itemprop="genre">Science fiction</span>
  <a href="../movies/avatar-theatrical-trailer.html"
     itemprop="trailer">Trailer</a>
</div>
```

JSON Linked Data

- Separate listing of data
- Embedded with a `<script>` tag
- **Note:** Google doesn't like external files

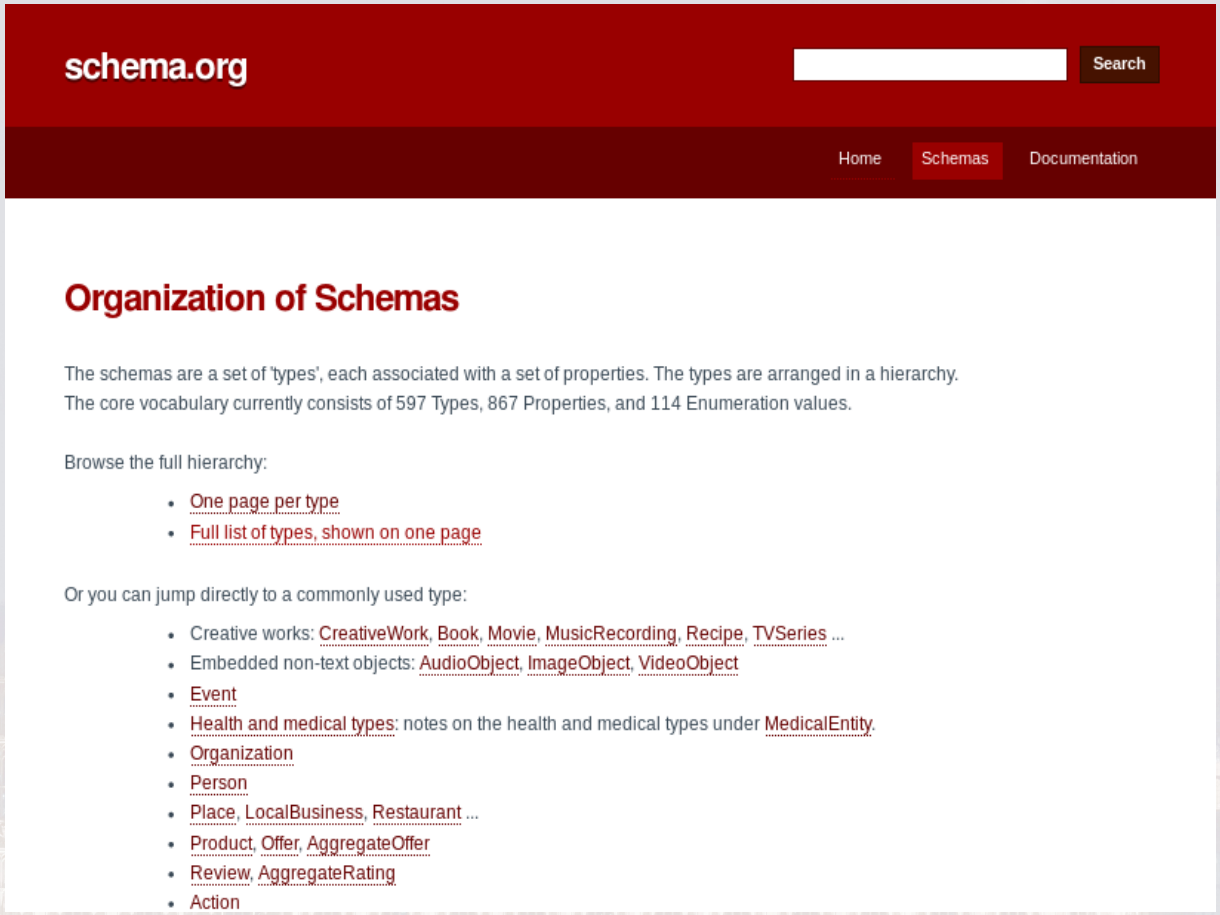
JSON-LD Example

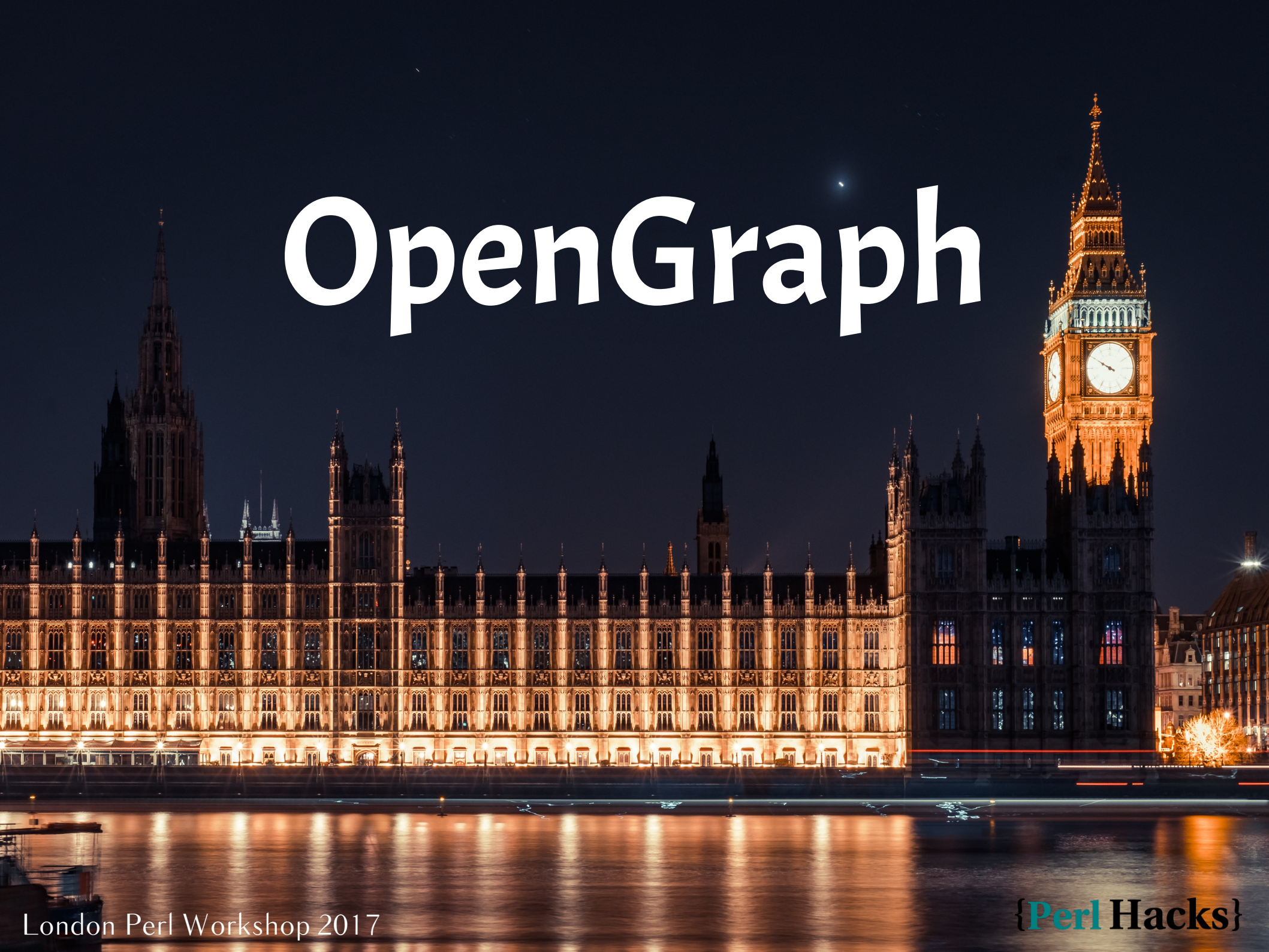
```
<script type="application/ld+json">
{
  "@context": "http://schema.org/",
  "@type": "Movie",
  "name": "Avatar",
  "director":
  {
    "@type": "Person",
     "name": "James Cameron",
     "birthDate": "1954-08-16"
  },
  "genre": "Science fiction",
  "trailer": "../movies/avatar-theatrical-trailer.html"
}
</script>
```

Structured Data Types

- See schema.org
- Joint project by Google, Bing, Yahoo!
- Defines dozens of data types
- Gives examples of their use

Structure Data Types




OpenGraph
London Perl Workshop 2017
{Perl Hacks}

OpenGraph

- Special type of structured data
- Used when sharing on social media
- Controls how your link appears
  - Include an image
  - Stop random images

OpenGraph Example

```
<meta property="og:title" content="Title Goes Here">
<meta property="og:site_name" content="My Cool Website">
<meta property="og:type" content="article">
<meta property="og:url"
content="https://example.com/some-article">
<meta property="og:image"
content="https://example.com/thumbnail.jpg">
<meta property="og:image"
content="https://example.com/logo.png">
<meta property="og:description"
      content="This is the description of the article.">
```

Twitter Cards

- OpenGraph is a Facebook invention
- Twitter Cards are supposed to use the same(ish) tags
- Except when they don't
- Add Twitter-specific tags

Twitter Card Additions

```
<meta name="twitter:card"
content="summary" />
<meta name="twitter:image"
content="https://example.com/logo.p
ng"/>
```

Sitemaps
London Perl Workshop 2017
{Perl Hacks}

Sitemaps

- XML file containing a list of your pages
- Easy way to ensure that Google finds them all
- Generated automatically by most CMSs
- Publish location in `robots.txt`
- `Sitemap: https://example.com/sitemap.xml`

Example Sitemap

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>http://www.example.net/?id=who</loc>
    <lastmod>2009-09-22</lastmod>
    <changefreq>monthly</changefreq>
    <priority>0.8</priority>
  </url>
  <url>
    <loc>http://www.example.net/?id=what</loc>
    <lastmod>2009-09-22</lastmod>
    <changefreq>monthly</changefreq>
    <priority>0.5</priority>
  </url>
</urlset>
```

Sitemap Elements

- Each `<url>` element has nested elements
- `<loc>` - the actual URL
- `<lastmod>` - last modification date
- `<changefreq>` - frequency of changes
- `<priority>` - priority of this page (0 - 1)
- Hints for Google crawler

Sitemap Limits

- 50,000 URLs
- 50 MB
- But you can have multiple sitemaps
- List sitemaps in a sitemap index

Sitemap Index

```
<?xml version="1.0" encoding="UTF-8"?>
<sitemapindex
xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <sitemap>
    <loc>http://www.example.com/sitemap1.xml.gz</loc>
    <lastmod>2004-10-01T18:23:17+00:00</lastmod>
  </sitemap>
  <sitemap>
    <loc>http://www.example.com/sitemap2.xml.gz</loc>
    <lastmod>2005-01-01</lastmod>
  </sitemap>
</sitemapindex>
```

Link Tags
London Perl Workshop 2017
{Perl Hacks}

Link Tags

- `<link rel="..." href="...">`
- Links to related pages
- HTML elements in <head> section
- `rel="canonical"`
- `rel="prev"` / `rel="next"`

Canonical Tag

- Sometimes two URLs might point to the same content
- Google doesn't like duplicate content
- `<link rel="canonical" href="https://example.com/foo">`
- Tells Google which URL is definitive

Reasons For Duplication

- Request parameters
  - Pagination
  - Searches
- Switching protocols
- Switching domains

Canonical Effects

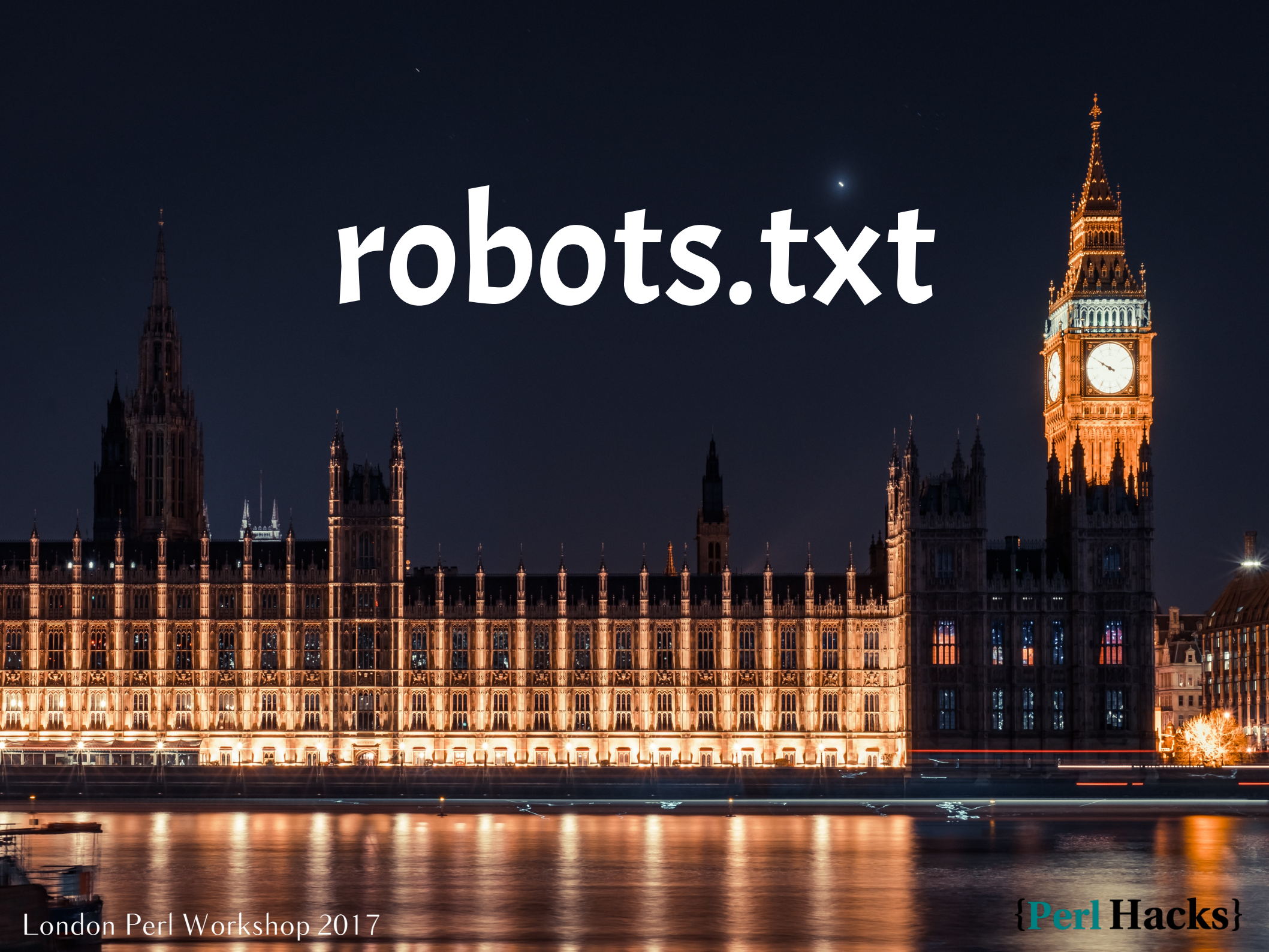
- Don't return this link for search queries
- Pass any page rank to canonical version
- Minimises duplicate content errors

Redirects Are Better

- Switching protocols or domains
- Use a 301 in preference to canonical link

Next and Previous Tags

- Link to other pages in a sequence
- `/news/2017/11/page/2`
- `<link rel="prev" href="/news/2017/page/1">`
- `<link rel="next" href="/news/2017/page/3">`
- Google recognises the sequence
- Browsers could do something clever
  - But don't


robots.txt
London Perl Workshop 2017
{Perl Hacks}

Robots Exclusion Standard

- Tell crawlers about your site
- Ban certain crawlers from certain areas
- Various other pieces of information
  - Non-standard extensions
- `robots.txt`
- Site root directory

Controlling crawlers

```
User-agent: googlebot
Disallow: /private/

User-agent: googlebot-news
Disallow: /no-news-here/

User-agent: *
Disallow: /nothing-interesting-here
```

Allow Directive

```
User-agent: *
Disallow: /private-stuff
Allow: /private-stuff/except-this
```

Default robots.txt

```
User-agent: *
Disallow:
```

Crawl Delay

- Tell crawlers to throttle visits
- `Crawl-delay: 10`
- Results vary

Sitemap

- Define where your sitemaps are
- `Sitemap: https://example.com/mysitemap.xml`
- `Sitemap: https://example.com/another-sitemap.xml`
- Also sitemap indexes
- `Sitemap: https://example.com/sitemap-index.xml`

Meta Tags &
Response Headers
London Perl Workshop 2017
{Perl Hacks}

Meta Tags & Response Headers

- Finer control over indexing
- Either meta tags or response headers
  - `<meta name="robots" content="...">`
  - `X-Robots-Tag: ...`

Index

- Do you want this page to be indexed?
- `<meta name="robots" content="index">`
- `<meta name="robots" content="noindex">`
- `X-Robots-Tag: index`
- `X-Robots-Tag: noindex`
- Default is `index`

Follow

- Do you want Google to follow links on this page?
- `<meta name="robots" content="follow">`
- `<meta name="robots" content="nofollow">`
- `X-Robots-Tag: follow`
- `X-Robots-Tag: nofollow`
- Default is `follow`

Combinations

- Combine both in one tag or header
- `<meta name="robots" content="index,nofollow">`
- `X-Robots-Tag: noindex,follow`

Nofollow
Attributes
London Perl Workshop 2017
{Perl Hacks}

Nofollow Attributes

- Even finer control
- Attribute on the <a> tag
- `rel="nofollow"`
- `<a href="https://example.com/" rel="nofollow">`
- Don't follow this link
- Don't pass page rank to linked page
- e.g. Links in blog comments

Security
London Perl Workshop 2017
{Perl Hacks}

Security

- Google gives ranking boost to secure sites
- `https`
- Do users care?
- Green padlocks
- Browsers warn on insecure forms

HTTPS

- Need an HTTPS certificate
- Can be expensive
- Let's Encrypt

Let's Encrypt

Mixed Content Warnings

- Images, Javascript, stylesheets, etc
- Are secure versions available?
- Can you host locally?
- Protocol-less URLs
  - `<image src="//example.com/pic.png>`

HTTPS Redirections

- You redirect from `http` to `https`
- Users have bookmarks
- Other sites have old links to you
- That's a lot of redirection

HSTS

- HTTP Strict Transport Security
- HTTP response header
- `Strict-Transport-Security: max-age=31536000`
- "Only make HTTPS requests to this domain"
- Information cached in browser

HSTS Preload

- Still need to make a first HTTP request
- Browsers have "pre-load" list
- Never make HTTP requests
- See https://hstspreload.org/

Javascript
London Perl Workshop 2017
{Perl Hacks}

Javascript Pages

```
<html>
  <head>
    <script src="cool-app.js"></script>
  </head>
  <body>
    <div id="content-goes-here" />
  </body>
</html>
```

Does Google Render JS Pages?

- Kind of
- Slowly
- And it's buggy
  - Chrome 41
- Depends on your framework
  - AngularJS is the worst

Javascript Suggestions

- Don't make Javascript essential
- Serve useful HTML content
- Progressive enhancement
- Test your site with Javascript disabled

Pre-Rendering

- Have a pre-rendered version of your site
- PhantomJS
- Serve that to search engine bots
- Hard to get right
- https://prerender.io/
- Is this cloaking?

Mobile Pages
London Perl Workshop 2017
{Perl Hacks}

Mobile Pages

- Mobile-first index
- Responsive pages
- AMP

Mobile-First Index

- Most Google searches are on mobile devices
- Google prioritising mobile sites
- Crawling mobile sites first
- Boosting sites with good mobile version
- Full implementation "soon"

Mobile Sites

- Your site must work on mobile devices
- Equal to desktop site

Separate Mobile Site

- Two domains
- www vs m
- Two sets of templates
- Redirect as appropriate
- Increased maintenance load

Serving Mobile Sites

- Trust the user
- Assume the user knows what site the want
- URL sharing problems

Serving Mobile Sites

- Browser sniffing
- Redirect mobile UA to mobile site
- Redirect desktop UA to desktop site
- Override cookies
- Where do tablets fit?

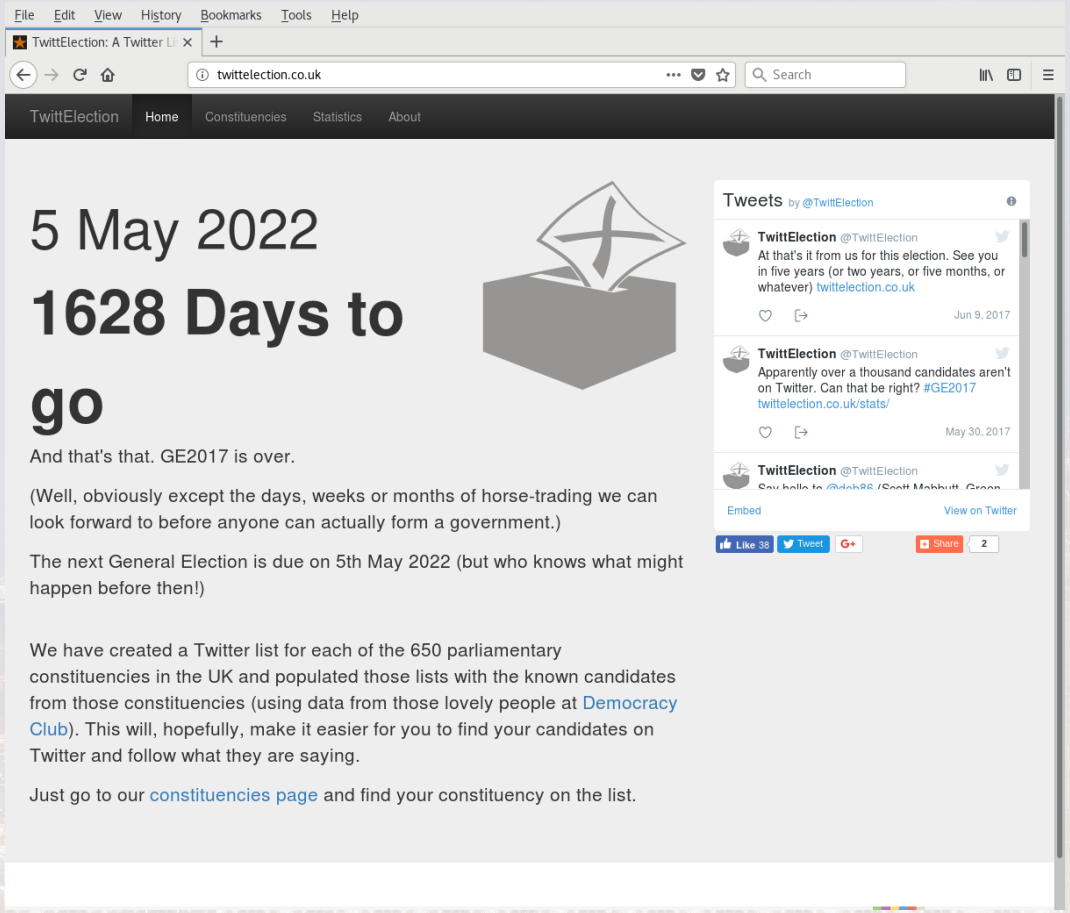
Responsive Pages

- CSS3 media queries
- Change layout as appropriate
- Reflow as window size changes
- Menu becomes burger, etc.
- Very hard to get right
- Single domain

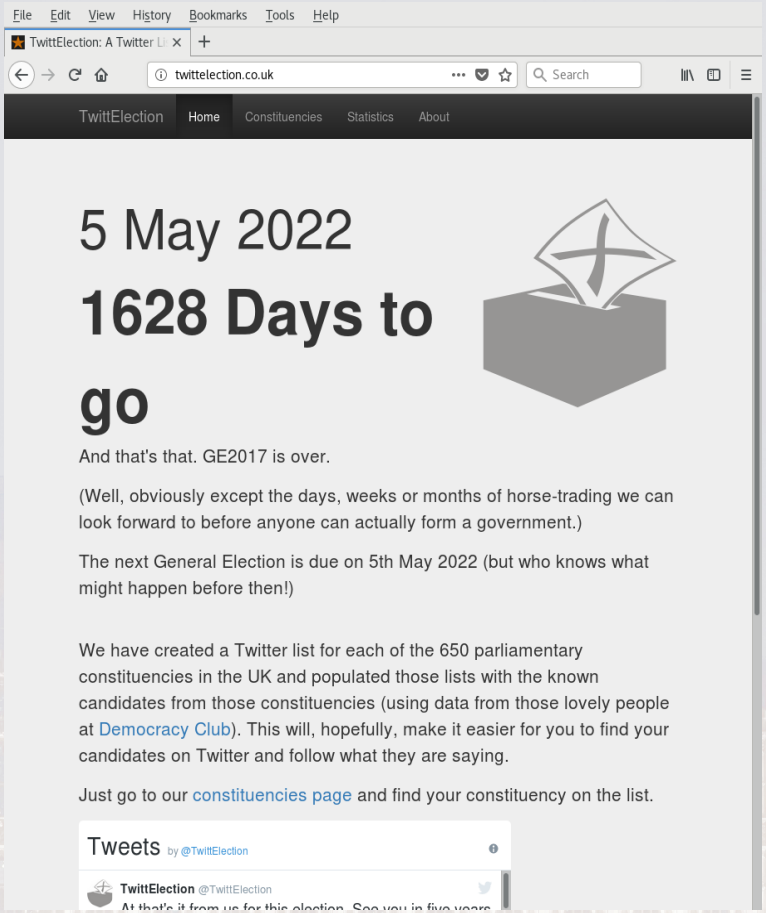
Bootstrap

- CSS framework from Twitter
- Responsive classes built-in
- Easier to get responsive layouts working
- Not easy
- But easier
- Just use it

Bootstrap Example



Bootstrap Example



Bootstrap Example

Accelerated Mobile Pages

- Google-led initiative
- "The project enables the creation of websites and ads that are consistently fast, beautiful and high-performing across devices and distribution platforms"
- Optimised pages for mobile devices
- Slower network connection
- https://www.ampproject.org

AMP Components

- AMP HTML is HTML with some restrictions for reliable performance
- The AMP JS library ensures the fast rendering of AMP HTML pages
- The Google AMP Cache can be used to serve cached AMP HTML pages

London Perl Workshop 2017

{Perl Hacks}

Measuring Your Success

- Measure stuff so you can improve it
- Google supplies free tools
- Google Analytics
- Google Webmaster Tools

Google Analytics

- Add a small amount of Javascript to your web site
- Get a ridiculous amount of data about your visitors
- All sites should use it
- (Assuming you trust Google with your data)

Google Webmaster Tools

- Validate your ownership of a domain
  - Add a DNS record
- Get data about Google's index of your site
- HTML errors
- Crawl stats
- Crawl errors

Paid Tools

- Pay for more detailed reports
- Loads to choose from
- Moz Pro
- SEMRush
- etc...

Conclusion
London Perl Workshop 2017
{Perl Hacks}

Conclusion

- Improve On-Page SEO
- ...
- Profit!

eligo

PerlCareers

CVLibrary

WCN

adzuna

BYTEMARK

OPUS VL

Booking.com

sureVoIP
Open Telecoms for Business

MAGNUM
SOLUTIONS LIMITED

Geekuni

UNIVERSITY OF
WESTMINSTER

Cogendo

SCIENCEphotoLIBRARY

EVOZON

O'REILLY®

Questions?
London Perl Workshop 2017
{Perl Hacks}

Questions?

- Any questions?
- Coffee break