# IERG4210 Web Programming and Security

Course Website: https://course.ie.cuhk.edu.hk/~ierg4210/
Live FB Feedback Group: https://fb.com/groups/ierg4210.2015spring/

# Optimizing Web Applications
## Lecture 11

Dr. Adonis Fung
phfung@ie.cuhk.edu.hk

Information Engineering, CUHK
Product Security Engineering, Yahoo!

# Agenda

- Performance Optimizations
  - Motivation: faster pages get more traffic/sales
  - To minimize above-the-fold time (AFT)
  - Tools: Google PageSpeed Online and Add-on, Critical NPM
- Search Engine Optimizations
  - Motivation: know/test out/meet what people want
  - To maximize your website's exposure
  - Tools: Google Webmaster Tool, Google/Yahoo Analytics,

- Ultimately, it is about driving traffic to your website
  - Venture Capitals (or investors) evaluate a site by its traffic
    - User counts perhaps play a more important role than business model in terms of the valuation metrics

# Why Performance Optimization?

- Best Practice: Fast!
  - Google takes site speed into ranking consideration since 2010
  - To impress the users, make your services responsive

- Effects of being slow:
  - Google: +500ms → -20% traffic
    experimented 30 results/page (instead of 10) in 2006
  - Amazon: +100ms → -1% sales
    experimented by delaying 100ms in its shopping site in 2006

  - Reference: Andrew B. King, Website Optimization: Speed, Search Engine & Conversion Rate Secrets, 2008

# General Considerations

- Benchmark First
  - PageSpeed, JSPerf
  - Chrome Dev Tools: Waterfall, Profiling & Timeline
  - Webmaster Tools, Analytics

- Optimize the part that impact the most
  - Micro-benchmarking and Macro-benchmarking
  - Relative v.s. Absolute comparisons
  - Before and after comparisons
  - With and without CDN

# What to Optimize?

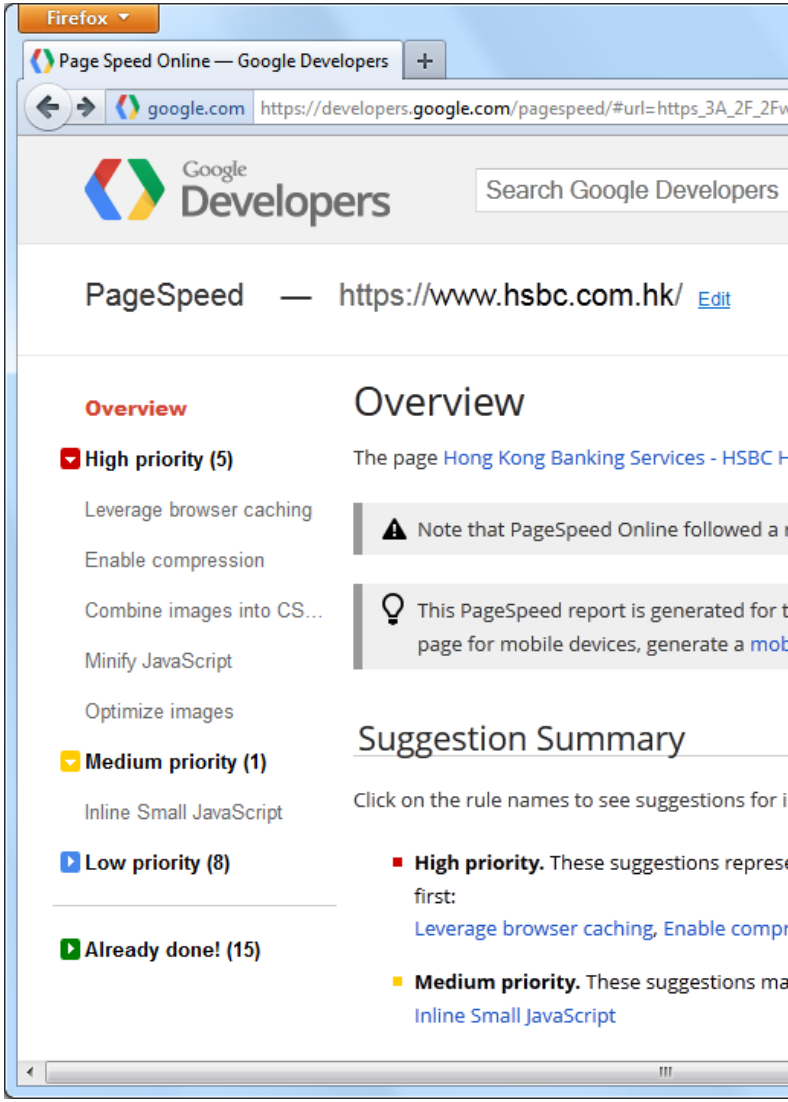
- Priority list for better performance:
  A. Network Latency
     1. Browser Caching – keeping your data and logic *off* the network
     2. Minifications, Compression and Image Optimization: reducing filesizes
     3. HTTP/2.0 – multiplexed streams and less redundancy
  B. Frontend: Browser Rendering and Code Execution Time
     1. Prioritize Visible Content – minimizing above-the-fold time
     2. Optimizing JavaScript – optimizing the way you write faster JS
  C. Backend: Logic, DB Processing (Time permitting?)
     - Load Balancer: Using Nginx to serve static files
     - Node.js: Optimizing JavaScript, Splitting/Distributing tasks (to Workers)
     - DB: MemCache, Combined Queries, Indexing, DynamoDB, MapReduce, …

- The higher the priority, the more likely is the bottleneck

  > **80%** of the response time is spent on network and client-side

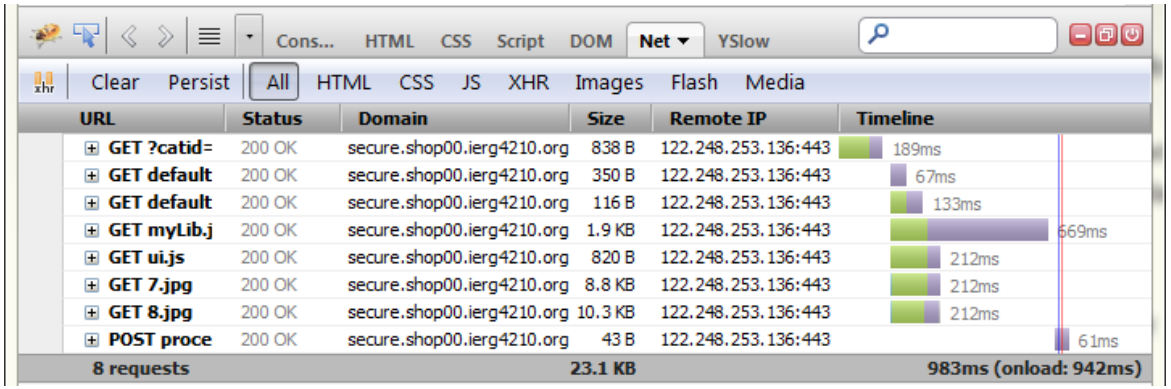
# Demonstrations and Explanations

- Online Google PageSpeed Insights
  - https://developers.google.com/pagespeed/
  - Cannot test pages that require authentication):
- Browser add-ons:
  - Firebug
  - Yahoo YSlow!
  - Google PageSpeed Insights
  - p.s. Chrome also has similar add-ons
- Best Practices
  - Work on the higher prioritized items
  - Optimize those that give you the most benefit (least effort, big impact)

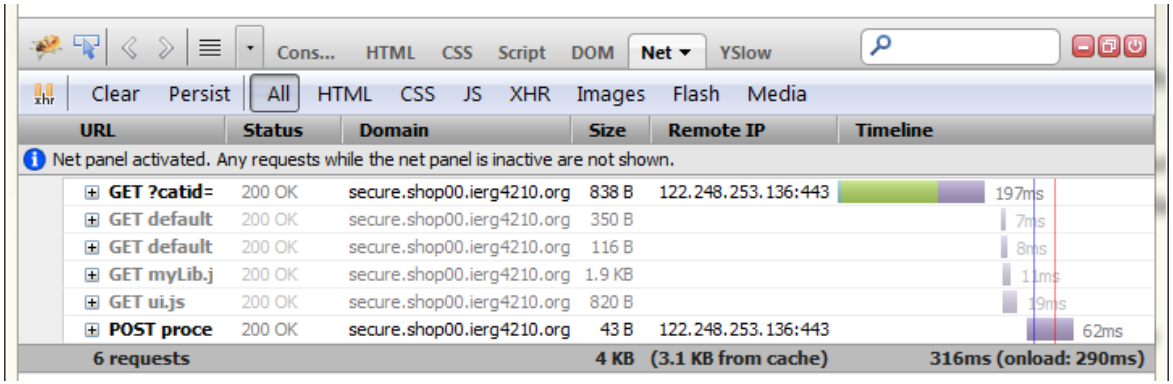


# Optimizing Caching (1/3)

- Network is slow?
  - Would be great to have them "near"
  - Cache static resources in browser or thru proxy
- Experiment:
  - How long to load:


First Visit: a full load takes 942ms for 23.1KB (Ctrl+F5)




Subsequent Visits: takes only 290ms for 0.881KB

# Optimizing Caching (2/3)

- Best Practices to Leverage Browser Caching:
  - Set caching headers aggressively for static resources
    - Static: those do not change over time, e.g. site's logo, JS, CSS, etc…
  - Set `Cache-Control: public` header to enable caching even HTTPS

```
# Apache: Make static content expire after one month
ExpiresActive on
<FilesMatch "\\.(ico|jpe?g|png|gif|flv|swf|pdf|js|css)$">
  ExpiresDefault "access plus 1 month"
  Header merge Cache-Control "public"
</FilesMatch>
```

`Expires` and `Cache-Control` headers are generated

```
HTTP/1.1 200 OK
Date: Sat, 14 Apr 2012 08:49:49 GMT
Cache-Control: max-age=2592000, public
Expires: Mon, 14 May 2012 08:49:49 GMT
...
```

References: https://developers.google.com/speed/docs/best-practices/caching

# Optimizing Caching (3/3)

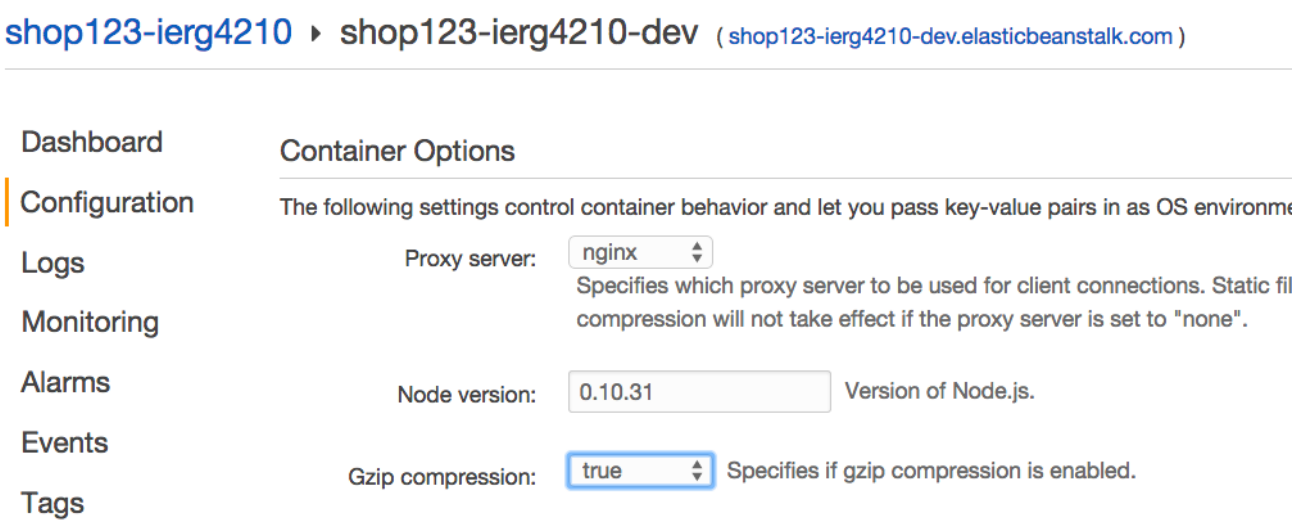
- Best Practices to Leverage Proxy Caching:
  - Proxies cache only resources with `Cache-Control: public`
    - Personalized resources (e.g. which set cookies) should not be cached
  - Don't include a query string in the URL for static resources
    - Most proxies disregard the Expiration headers if it sees ? in URLs
- How about those resources that change occasionally (e.g. JS)?
  - Trick: Still cache them aggressively as if they are "static"
  - When changed, change also the URL (differs by >8 chars for Firefox)!!

- Add the following (w/RewriteEngine On) in Apache to enable file mapping:
  `RewriteRule  ^(.*)-\w+\.js$  $1.js`
- When updated `myLib.js`, include the new one by updating your HTML:
  `<script type="text/javascript" src="myLib-61ad0d9b0.js"/>`
  → `<script type="text/javascript" src="myLib-957f639ef6.js"/>`
- Browser loads the updated JS even if a non-expiring cached copy exists

References: https://developers.google.com/speed/docs/best-practices/caching

# Minimizing Payload Size (1/2)

- Best Practices to minimize request overhead:
  - Enable compression
    - Browsers declares support by the request header `Accept-Encoding: deflate, gzip`
    - Servers returns the response header `Content-Encoding: gzip` and gzipped content (headers are not gzipped)




References: https://developers.google.com/speed/docs/best-practices/payload

# Minimizing Payload Size (2/2)

- Best Practices to minimize request overhead:
  - Minify Javascript, CSS, and HTML
    - Minify: Strip whitespaces, Rewrite equiv. code with less characters
    - Automatic Tools: YSlow, Google Closure Compiler, etc
  - Optimize images
    - Choose the right format: Use PNG over GIF for figures, Use JPG for photos
    - Compress images: flatten layers, remove meta-information, etc
  - Serve scaled images
    - Big files take browser's time to download and resize
    - Hence, generate and produce a thumbnail (smaller sizes) of images
  - Serve resources from a consistent URL
    - e.g. the same logo should be served from a single location even if it is used across subdomains

References: https://developers.google.com/speed/docs/best-practices/payload

# More Network Optimizations (1/2)

- HTTP/2.0
  - Loading subresources in parallel over a single TCP connection
  - Headers redundancy reduced and Compression
  - Server push
- More Efficient HTTPS Ciphers
  - ECDSA faster than RSA
  - Recommended Best Practice:
    - https://mozilla.github.io/server-side-tls/ssl-config-generator/
- Use Content Delivery Networks
  - Edge closer to clients
  - Amazon Cloudfront, Akamai, Cloudflare, etc...

# More Network Optimizations (2/2)

- Minimize request size
  - Keep the size of Cookies compact
    - Use localStorage for supported browsers
    - Host users' information in server-side session instead of Cookies
  - Use the `path` config to avoid sending Cookies to meaningless locations
  - Keep the length of URL compact, as it's used in Referer header
- Serve static content from a cookieless domain
  - Host static content in a separate (sub-)domain of your own (e.g. static1.ierg4210.org, gstatic.com used by google)

References: https://developers.google.com/speed/docs/best-practices/request

# Prioritize Visible Content

- AFT: time for contents loaded into screen before scrolling
  - Do not exceed 14-16KB. NPM that can help: Critical
- Inline render-blocking CSS to reduce AFT
  - Network delay is too much. Other CSS loaded in a file
- Prefer Async and deferred JavaScript Resources
  - Async JavaScript load after AFT
  - Non-visual processing deferred after AFT
- Avoid Long Running JavaScript
  - Defer them where possible
  - If not, for non-visual processing, put them in worker
  - If not, splitting it into smaller tasks to allow processing of other events
- Reference: https://developers.google.com/web/fundamentals/performance/

# Optimizing Browser Rendering

- Compact CSS selectors are more efficient. Less HTML Nodes.
  - Each node asks if any CSS selectors are matched.
- Specify image dimensions and Put CSS in the document head
  - To eliminate the need for unnecessary reflows and repaints
- Animations/Scrolling frame rates
  - Use of requestAnimationFrame() to execute before next repaint
  - Rendering/Paint/Scripts running longer than 10ms introduce janks
- Specify the correct Content-Type and Encoding
  - NodeJS has defaulted to use text/html and utf8
  - Otherwise, browsers could waste time guessing the right content type
  - Add the response header `X-Content-Type-Options: nosniff`

References: https://developers.google.com/speed/docs/best-practices/payload

# Optimizing Javascript

- Best Practices to optimize Javascript:
  - Use Array.push() and Array.join() when concatenating strings
    - Introduced in Lecture 03
  - Use prototype when defining class methods and instance variables
    - Discussed in class and Lecture 03

Bad

```
var Person = function(name, sex){
  // this.name = name || 'Unnamed'; ...
  // one assignment per instantiation
  this.getName = function(){}
};
```

Good

```
var Person = function(name, sex){
  // this.name = name || 'Unnamed'; ...
};
// One assignment per declaration
Person.prototype.getName = function(){}
```

  - Leverage event bubbling to reduce number of event handlers
    - `Event.target` tells the element that triggers the event (Midterm/Final)
  - Using Promises for parallelization (see the reading)
    - http://www.html5rocks.com/en/tutorials/es6/promises/

# Lazy and Cached Loading

- On top of those mentioned, additional optimizations:
  - Defer parsing of Javascript
    - Defer loading of JS and async JS may not be sufficient
    - Google conducted an experiment in early 2011:
      On modern mobile devices, 1 KB of JS → 1 ms of parse time
    - Mobile Gmail loads JS in comments; parse it with `eval()` when needed
    - https://developers.google.com/speed/docs/insights/BlockingJS
  - Make landing page redirects cacheable
    - E.g. cache the redirection from http://www.example.com to http://m.example.com
    - Apply a 302 redirection with Cache-Control: private

# Readings and Reference

- Readings:
  - Google Web Performance Best Practices
    https://developers.google.com/speed/docs/best-practices/rules_intro
  - Yahoo! High Performance Web Pages

For those who skipped lectures… :)

- Reference:
  - Stanford CS193H: High Performance Web Sites
    (its equiv. online course XCS193H is priced at US$600)

# SEARCH ENGINE OPTIMIZATIONS

# Why Search Engine Optimization?

- Technical Limitations of Crawlers
  - Don't understand images/videos, Javascript, Flash
  - Don't know how to fill out forms
  - Don't know a site's presence from day one
- Marketing: SEO a.k.a. Search Engine Marketing
  - To target what users would put in the search bar
  - To boost the traffic and sales of your website
- You want traffic but don't want to pay for ads
  - Organic v.s. Paid Traffic

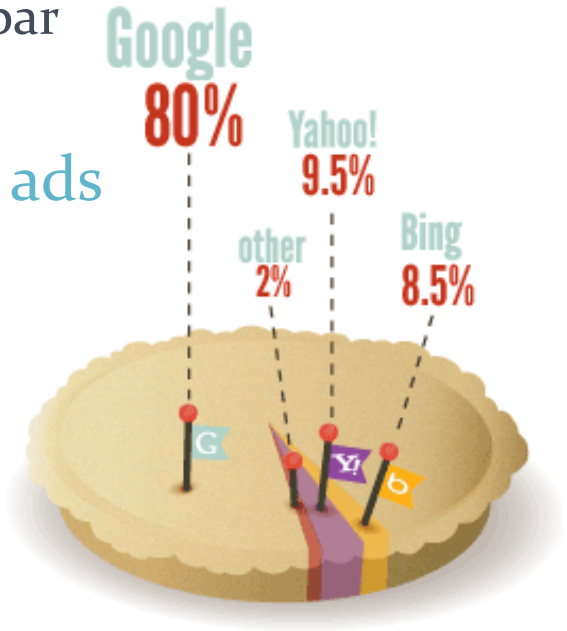



Image Source: http://www.seomoz.org/beginners-guide-to-seo

# Organic v.s. Paid Traffic

- If you are rich enough to pay, then forget about SEO ☺
  - Bid on keywords, which are priced by click-through rates (CTR)


Image Source: https://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf

# General Recommendations from Google

- Make pages primarily for users, not for search engines
  - Don't deceive your users or present different content to search engines than you display to users, which is commonly referred to as cloaking.
- Make a site with a clear hierarchy and text links
  - Every page should be reachable from at least one static text link.
- Create a useful, information-rich site
  - Write pages that clearly and accurately describe your content. Make sure that your <title> elements and ALT attributes are descriptive and accurate.
- Keep the links on a given page to a reasonable number (fewer than 100)

Reference: http://www.seomoz.org/beginners-guide-to-seo

# SEO Best Practices

A. Setting Meta Information
   1. Create unique and accurate page <title>s
   2. Make use of the description meta tag

B. Improving Site Structure
   1. Improve the structure of URLs
   2. Make a site easier to navigate

C. Optimizing Content
   1. Offer quality content and services
   2. Write better anchor text
   3. Optimize the use of images
   4. Use heading tags appropriately

D. Dealing with Crawlers
   1. Make effective use of robots.txt
   2. Be aware of rel="nofollow" for links

E. Promotions and Analysis
   1. More incoming links can result in higher rank
   2. Make use of Google Webmaster, Adwords Keyword, and Analytics (Demo)

Reference: https://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf

# A. Setting Meta Information

- Best Practices:
  - Include <title> and <meta name="description"> in <head>

```
<html>
<head>
<title>Brandon's Baseball Cards - Buy Cards, Baseball News, Card Prices</title>
<meta name="description=" content="Brandon's Baseball Cards provides a large
selection of vintage and modern baseball cards for sale. We also offer daily
baseball news and events in"/>
</head>
<body>
```

Brandon's **Baseball Cards** - Buy **Cards**, **Baseball** News, Card Prices
Brandon's **Baseball Cards** provides a large selection of vintage and modern **baseball cards** for sale. We also offer daily **baseball** news and events in ...
www.brandons**baseballcards**.com/ - Cached - Similar

  - Choose an accurate and unique title that describe the topic of a page
  - Use brief, but descriptive titles
  - Description should accurately summarize the page's content
  - Again, use description that is unique to every page

Reference: https://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf

# B1. Improving Structure of URLs

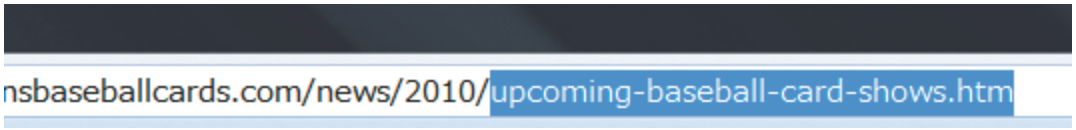
- Best Practices:
  - Use keywords in URLs (i.e., domains and path)
  - Create a simple directory structure
    - E.g. `/2-Fruits/9-Apple/` instead of `?catid=2&pid=9`
  - Provide one version of a URL to reach a document
    - Use 301 redirect (the status code means permanent redirect) to inform search engines, which will update the search results
    - Examples:

Reference: https://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf

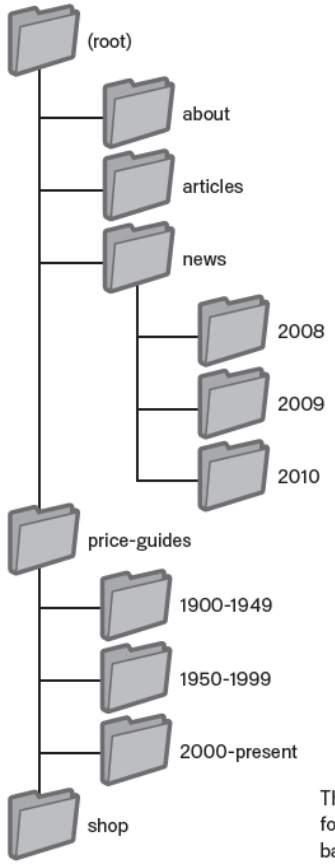
# B2. Make a site easier to navigate (1/2)

- Best Practices:
  - Maintain a natural hierarchical structure
    - Example shown on RHS
  - Assist page navigation by "breadcrumb lists"


  - Allow for the possibility of a part of the URL being removed




  - Serve a custom and useful 404 error page
    - Help users get back to the root or one-level up
    - Tool: Google 404 widget

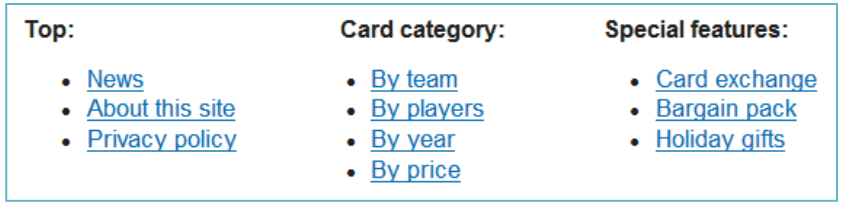



The directory structure for our small website on baseball cards.

Reference: https://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf

# B2. Make a site easier to navigate (2/2)

- Best Practices:
  - Provide sitemaps to expose all pages
    - Good for pages that use AJAX and not easily discoverable by crawlers
    - For users: sitemap.html



**Top:**
- News
- About this site
- Privacy policy

**Card category:**
- By team
- By players
- By year
- By price

**Special features:**
- Card exchange
- Bargain pack
- Holiday gifts

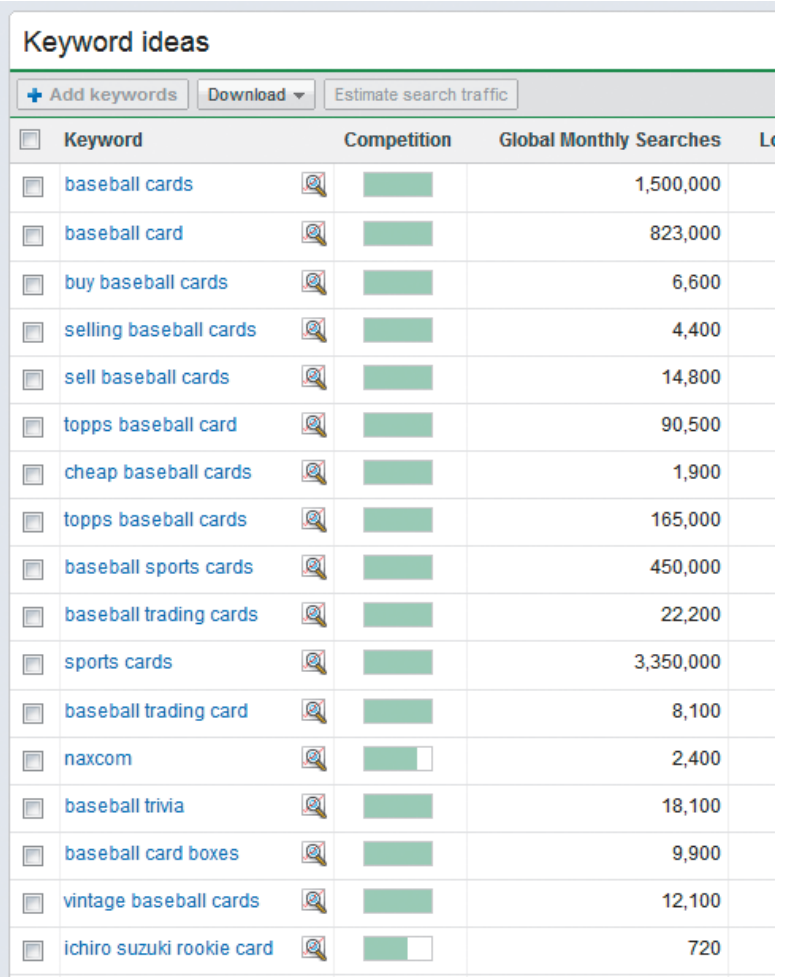
    - For search engines: sitemap.xml
      - Example shown on RHS
      - Specify the location using `Sitemap: <location>` in robots.txt, or submit it to different search engines (e.g. Google Webmaster)
      - Autogen Tool: Google Sitemap Generator

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>http://www.brandonsbaseballcards.com/</loc>
    <changefreq>daily</changefreq>
    <priority>0.8</priority>
  </url>
  <url>
    <loc>http://www.brandonsbaseballcards.com/news/</loc>
  </url>
  <url>
    <loc>http://www.brandonsbaseballcards.com/news/2008/</loc>
  </url>
  <url>
    <loc>http://www.brandonsbaseballcards.com/news/2009/</loc>
  </url>
  <url>
    <loc>http://www.brandonsbaseballcards.com/news/2010/</loc>
  </url>
</urlset>
```

Reference: https://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf

# C1. Offer quality content and services

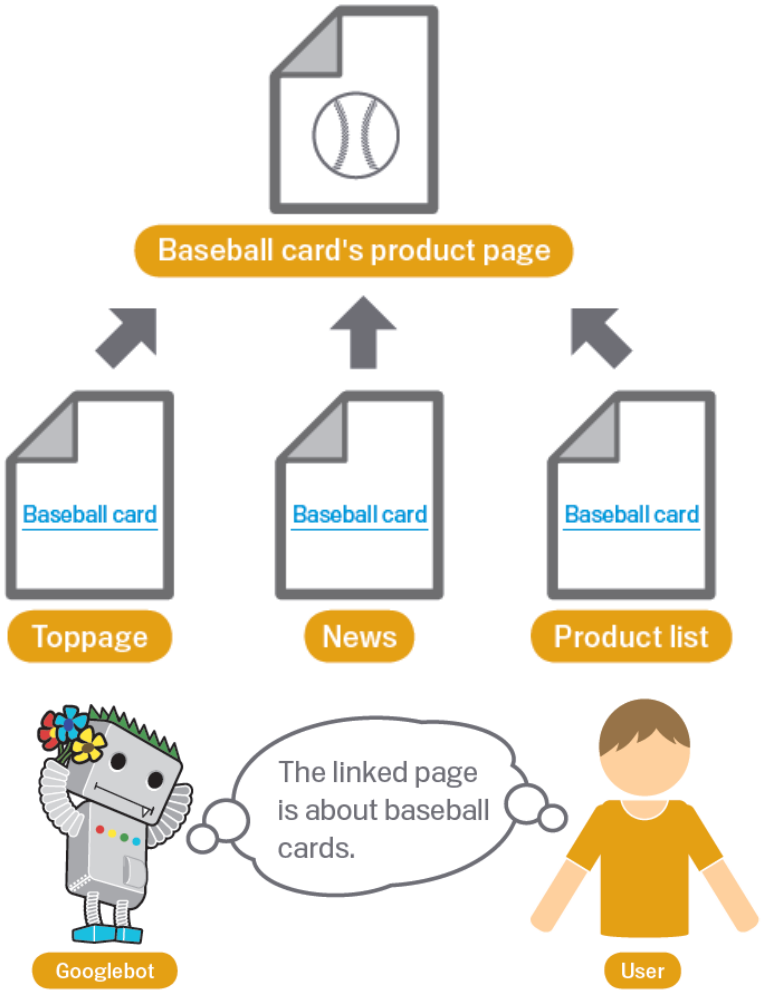
- Best Practices:
  - Know what users expect on a topic and offer unique, exclusive and easy-to-read content
    - Writing your content that uses a good mix of keywords produce +ve results
    - RHS: Adwords Keyword Tool
  - Useful content itself attracts linkages and likes
    - e.g. Put a "Like/+1" box to facilitate publicizing in social networks
  - Keep your site fresh and new
    - Users' tastes could change
    - Keep users engaging and coming back




Reference: https://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf

# C2. Write better anchor text

- Best Practices:
  - Use descriptive and concise text
    - `<a href="…">[Concise Text]</a>`
    - Generally, naming is like the title tag
    - Avoid: Click here, URLs, long text
  - Format links so they're easy to spot
    - Ensure users understand which are hyperlinks or clickable
      - Mouse pointer for clickable elements (CSS style: `cursor:pointer`)
    - Search Engines don't know JS, so do onclick handlers (e.g. loads a page over AJAX)
      - Set a URL at href that gives equiv. content




Reference: https://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf

# C3. Optimize the use of images

- Best Practices:
  - Describe images using the alt attribute
    - E.g. `<img src="/i/prod/2.jpg" alt="Apple"/>`
    - When used in anchor, alt is treated as important as anchor text
    - Avoid using generic filenames (e.g. 2.jpg) when possible
      - At least, describe those manually added images (e.g. logo.png)
  - Avoid using only image links for page navigation
  - Consider using an image sitemap

Reference: https://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf

# C4. Use heading tags appropriately

- Best Practices:
  - Use heading tags to emphasize the keywords
    - <h1> to <h5>, where <h1> is the most important
    - Proper organization and appropriate use of heading tags aids both users' and search engines' understanding
  - Avoid overuse when  and  is more appropriate

```
</head>
<body>
<h1>Brandon's Baseball Cards</h1>
<h2>News - Treasure Trove of Baseball Cards Found in Old
Barn</h2>
<p>A man who recently purchased a farm house was pleasantly
surprised ... dollars worth of vintage baseball cards in the
barn. The cards were ... in news papers and were thought to be in
near-mint condition. After ... the cards to his grandson instead
of selling them.</p>
```

Reference: https://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf

# D1. Make effective use of robots.txt

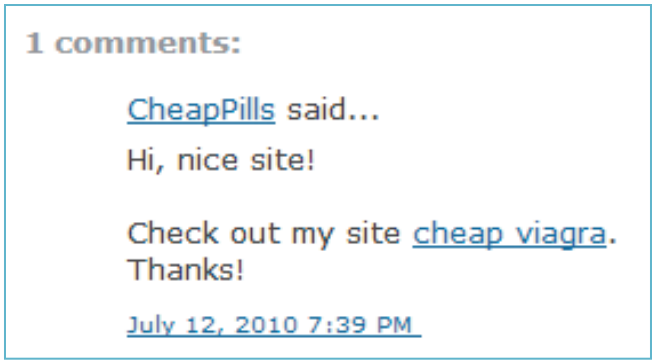
- Best Practices:
  - Restrict crawling where it's not needed with robots.txt
    - Disable indexing pages that you don't want them appear in search results, e.g.
      - Image and Javascript files
      - Pages that requires authentication
    - Example file shown on RHS
    - Another Example: http://www.google.com/robots.txt
    - Host robots.txt under root, i.e. http://example.com/robots.txt
    - Google Webmaster provides a robots.txt generator
  - Note: NOT all crawlers obey what you defined in robots.txt
    - Security by obfuscation is always not recommended

```
User-agent: *
Disallow: /images/
Disallow: /search
```

Reference: https://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf

# D2. Be aware of rel="nofollow" for links

- Best Practices:
  - Combat comment spam with "nofollow"
    - A user could input a spammy link if he is so allowed
    - Guard it using nofollow:
      ```
      <a href="//spammy.com/"
      rel="nofollow">Earn 10X/mth</a>
      ```
    - To tell search engines not to follow spammy links
    - To ensure that you're not giving your page's hard-earned reputation to a spammy site




Reference: https://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf

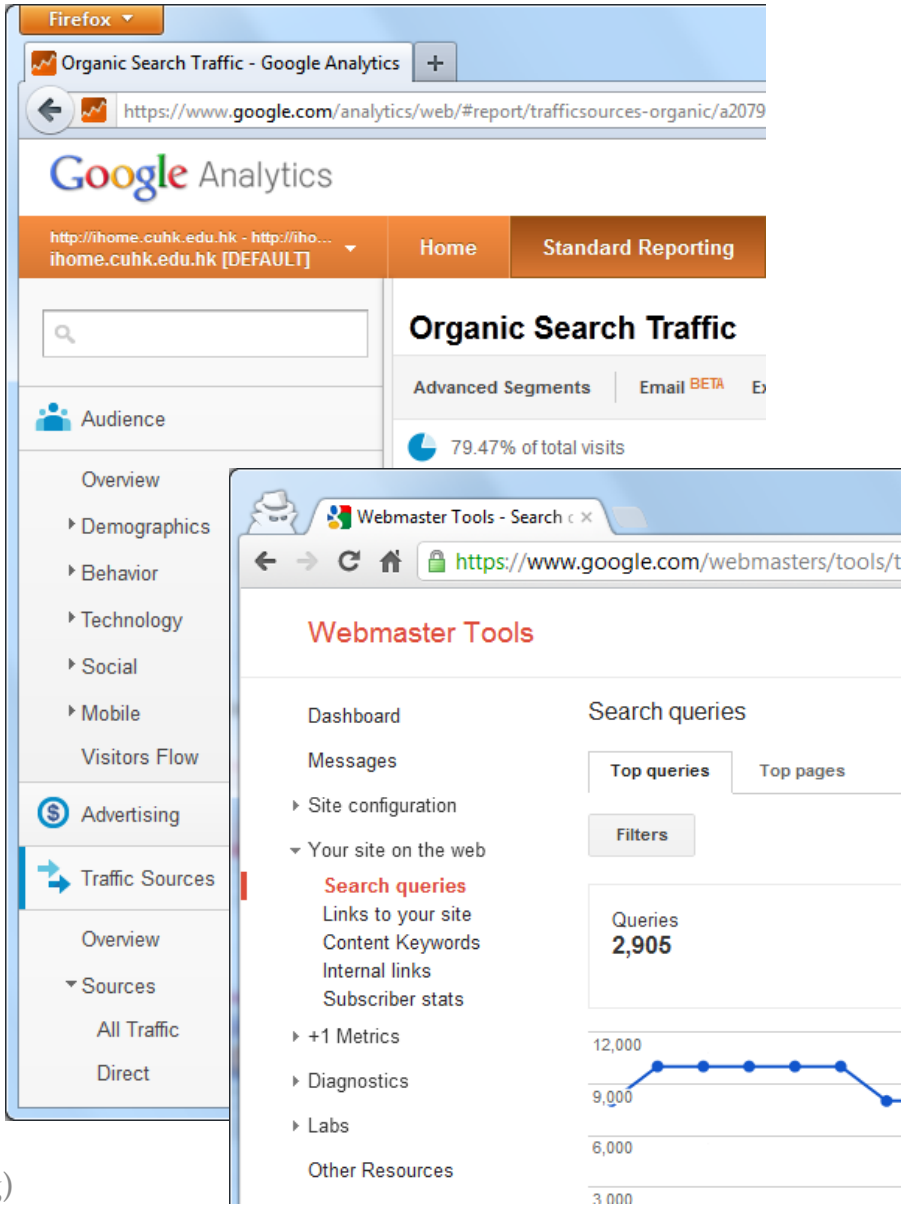
# E. Promotions and Analysis

- Best Practices:
  - PageRank: More incoming links can result in higher rank
    - Promote in Wikipedia, Youtube, Social networks, Online Directories, etc…
    - Having a .edu domain pointing to your site is a sign of credit
    - Make the promotion or links relevant to users
    - Avoid spamming others, or you will get penalties
  - Make good use of Google Webmaster, Adwords Keyword, and Analytics (Demo)
    - Google Webmaster help better control how Google interacts with your websites and get useful information from Google
    - Google Adwords Keyword Tool gives you a better understanding on the popularity of keywords and those related keywords used in Google
    - Google Analytics gives you an extensive analysis on the site's traffic, e.g., traffic volumes, new v.s. re-visiting users, entry page, staying for how long, bounce (leave a site) rate, CTR, users' geolocation, etc…

Reference: https://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf

# Demonstrations

- Search Engine Tools:
  - Google Webmaster Tools
  - Google Adwords: Keyword Tool
  - Google Analytics
- More to read:
  - Chapter 4 SEO Basics
  - Chapter 5 Keyword Research
  - Chapter 7 Growing Popularity
- SEO Trends and Social Media Optimization
  - (time-permitting)



# Readings and Reference

- Readings:
  - Performance: https://developers.google.com/web/fundamentals/performance/
  - Google Search Engine Optimization Starter Guide https://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf
  - SEOmoz: The Beginner's Guide to SEO http://www.seomoz.org/beginners-guide-to-seo
- Reference:
  - KEO: The Future of Search Engine Optimization and Social Media http://www.eciaonline.org/meetings/ExecutiveConference/2011/Presentation-Kloefkorn-SocialMedia%2010-25-2011.pdf