

# Web Techniques Reference

By John Bentley

---

# Non Code

---

## Introduction

All about web technologies, tools, and configurations that are apart from coding issues. Apart, that is from HTML, CSS and JavaScript issues.

## Server Side Considerations

### Basic install, upgrade and setup

Install, upgrade and setup Apache, PHP, MySQL:

*See Brainforest*

+ "C:\Users\John\Documents\CustomData\AppDataUserSaved\Brainforest\Dev and Computer\Setup\Setup - Master - Windows.pdb", Search from "Install WAMP"  
+ "C:\Users\John\Documents\CustomData\AppDataUserSaved\Brainforest\Dev and Computer\WebSda\Web - Certificates.pdb"

### Apache .htaccess

For Apache .htaccess recipes see

C:\Users\John\Documents\Sda\Info\Web\KB\Servers\Apache\Aphorism

### Mime Types

Verify your Web Server will serve standard Mime Types (aka "content type") for the following file extensions:

- ".html": "text/html";
- ".xhtml": "application/xhtml+xml"

*See (Bentley 2019, "Xhtml-PolyglotMarkupRules.Docx",  
\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Xhtml\Reference\Xhtml-PolyglotMarkupRules.docx) "File Extension, Mime Type, and HTML V XHTML Syntax"*

### Search Engine Control

#### Via Robots.txt

robots.txt is a text file that controls how a web page is crawled.

... if you want all your content to be crawled, you don't need a robots.txt file at all (and we recommend that you don't use one).

*(Google 2019, "Getting the Most from Your Google Search Appliance: Crawling and Indexing",  
<https://support.google.com/gsa/answer/6280131>) <https://support.google.com/gsa/answer/6280131>*

The Robots Exclusion Protocol (standard):

*(Illyes et al. 2020, "Robots Exclusion Protocol", <https://www.ietf.org/id/draft-koster-rep-01.txt>)  
<https://www.ietf.org/id/draft-koster-rep-01.txt>*

If robots.txt is used (because you don't want some or all pages to be crawled/indexed), it should be in the root directory, never subdirectories.

```
// OK
http://www.example.com/robots.txt
http://website.example.com/robots.txt
http://example.com:8181/robots.txt

// Not OK
http://example.com/pages/robots.txt
```

(Google 2019, "Getting the Most from Your Google Search Appliance: Crawling and Indexing", <https://support.google.com/gsa/answer/6280131>) <https://support.google.com/gsa/answer/6280131>

### Example robots.txt

```
# Comment lines start with a "#"

# No Googlebot crawlers should crawl the folder http://example.com/nogooglebot/
# and all contained URLs. The line "User-agent: Googlebot" starts the section
# with directives for Googlebots.
User-agent: Googlebot
Disallow: /nogooglebot/

# No other crawlers should crawl the folder http://example.com/onlygooglebot/ and
# all contained URLs. The line "User-agent: *" starts the section for all crawlers
# not otherwise specified.
User-agent: *
Disallow: /onlygooglebot/

Sitemap: http://www.example.com/sitemap.xml
```

Derived from (Google 2019, "Getting the Most from Your Google Search Appliance: Crawling and Indexing", <https://support.google.com/gsa/answer/6280131>) <https://support.google.com/gsa/answer/6280131?hl=en>  
(Google 2020, "Administering Crawl: Preparing for a Crawl, Using Robots.Txt to Control Access to a Content Server", <https://support.google.com/gsa/answer/6329153#92549>) <https://support.google.com/gsa/answer/6329153#92549>

### Clichéd robots.txt files

```
# Allow all content
User-agent: *
Allow: /
```

```
# Disallow crawling of whole website
User-agent: *
Disallow: /
```

```
# Disallow crawling of certain parts of the website.
User-agent: *
Disallow: /calendar/
Disallow: /junk/
```

(Google 2020, "Administering Crawl: Preparing for a Crawl, Using Robots.Txt to Control Access to a Content Server", <https://support.google.com/gsa/answer/6329153#92549>) <https://support.google.com/gsa/answer/6329153#92549>  
 (Google 2019, "Getting the Most from Your Google Search Appliance: Crawling and Indexing", <https://support.google.com/gsa/answer/6280131>) <https://support.google.com/gsa/answer/6280131?hl=en>

## Robots.txt testing.

"Ryte - Robots.txt Test Tool" <https://en.ryte.com/free-tools/robots-txt/>  
 "Test your robots.txt with the robots.txt Tester" <https://support.google.com/webmasters/answer/6062598>

## Via metatag or HTTP header

Indexing can be controlled on a page by page basis, either:

- For webpages, by using `<meta name="robots" ... />`; or
- For non html files, by setting the HTTP header X-Robots-Tag.

```
// Webpages
<meta name="robots" content="noindex, nofollow" />

// Set X-Robots-Tag in the Apache root .htaccess file or httpd.conf file:
<Files ~ "\.pdf$" >
  Header set X-Robots-Tag "noindex, nofollow"
</Files>

// HTTP Header might therefor look like
HTTP/1.1 200 OK
X-Robots-Tag: noindex, nofollow
Content-Type: text/html; charset=UTF-8
```

(Google 2019, "Getting the Most from Your Google Search Appliance: Crawling and Indexing", <https://support.google.com/gsa/answer/6280131>) <https://support.google.com/gsa/answer/6280131?hl=en>

Meta tag and X-Robots-Tag values (at least as Google recognizes them).

Value	Meaning
all	There are no restrictions for indexing or serving. Note: this directive is the default value and has no effect if explicitly listed.
noindex	Do not show this page in search results and do not show a "Cached" link in search results.
nofollow	Do not follow the links on this page
none	Equivalent to noindex, nofollow
noarchive	Do not show a "Cached" link in search results.
nosnippet	Do not show a snippet in the search results for this page

(Google 2019, "Getting the Most from Your Google Search Appliance: Crawling and Indexing", <https://support.google.com/gsa/answer/6280131>) <https://support.google.com/gsa/answer/6280131?hl=en>

Robots meta tags and X-Robots-Tag HTTP headers are discovered when a URL is crawled. If a page is disallowed from crawling through the robots.txt file, then any information about indexing or serving directives will not be found and will therefore be ignored.

(Google 2019, "Getting the Most from Your Google Search Appliance: Crawling and Indexing", <https://support.google.com/gsa/answer/6280131>) <https://support.google.com/gsa/answer/6280131?hl=en>

## HTTP Errors: 404 Not Found

Creating a 404 Not Found web page is server dependent. However it is generally straight forward. You:

- Create your 404NotFound.xhtml page. This is a normal web page with some user friendly message

```
<!-- In 404NotFound.xhtml -->
<p>The page you were looking for doesn't exist.</p>
<p>Use the menu on the left to find what you are looking for.</p>
```

- Then, for example in Apache, in .htaccess set 404 (and optionally other) Errors to point to your 404NotFound.xhtml

```
ErrorDocument 404 /404NotFound.html
```

<https://httpd.apache.org/docs/2.4/custom-error.html>

## Optimization

### Compression

See [\\ATLAS\Documents\Sda\Info\Web\KB\Apache\Aphorism\Enable Compression in Apache 2\\_X with mod\\_deflate.txt](#)

### Caching

#### Overview

A caching policy should be specified on the server for every file.

<https://developers.google.com/speed/docs/insights/LeverageBrowserCaching>

Control caching with:

- HTTP Header fields. Specifically using HTTP 1.1 header directives Cache-Control and Etag; and (optionally)
- URL versioning, also known as URL fingerprinting. Cache control via Url versioning or "Url fingerprinting" is a technique that allows you to instantly update web files that are set to be cached (kept fresh) for a long time.

(Grigorik 2019, "HTTP Caching | Web Fundamentals",  
<https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/http-caching>  
<https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/http-caching>)

Example HTTP 1.1 Header with cache control relevant headers and header values emphasized.

```
HTTP/1.1 200 OK
Date: Fri, 30 Oct 1998 13:19:41 GMT
Server: Apache/1.3.3 (Unix)
Cache-Control: max-age=3600, must-revalidate
Expires: Fri, 30 Oct 1998 14:19:41 GMT
Last-Modified: Mon, 29 Jun 1998 02:28:12 GMT
ETag: "3e86-410-3596fbbc"
Content-Length: 1040
```

Content-Type: text/html

Two types of caches:

- Private cache. For a single user, as is a local web browser cache;
- Shared cache. An intermediate cache for multiple use, as is a Content Delivery Network Server.

(Internet Engineering Task Force (IETF) et al. 2014, "Hypertext Transfer Protocol (HTTP/1.1): Caching", <https://tools.ietf.org/html/rfc7234>) <https://tools.ietf.org/html/rfc7234>

Caching is relevant only to "equivalent requests" with respect to a previously "stored response" (e.g. a user requests the same html page that they had previously surfed to). The possible actions to be taken on any subsequent request:

- Don't check the server: use the "stored response". This is done for "fresh" pages: those reusable without "validation".
- Check the server (perform "validation"):
  - If the server copy is identical to the cached copy then the response signals (via a "304 Not Modified" http status code) that there is no need to update the body of the cached page.
  - If the sever copy has changed compared to the cached copy then the response will include the entire page.
  - Fetch and display a new page from the server without preforming validation (without comparing stored response and server response).
  - Display an Error.

(Internet Engineering Task Force (IETF), 2014. RFC 7234: Hypertext Transfer Protocol (HTTP/1.1): Caching) <https://tools.ietf.org/html/rfc7234>

A fresh response is one whose age has not yet exceeded its freshness lifetime. Conversely, a stale response is one where it has.

(Internet Engineering Task Force (IETF), 2014. RFC 7234: Hypertext Transfer Protocol (HTTP/1.1): Caching) <https://tools.ietf.org/html/rfc7234#section-4.2>

A response's age is the time that has passed since it was generated by, or successfully validated with, the origin server.

Jlb: Response Age = Now - Date [Response Header]

(Internet Engineering Task Force (IETF), 2014. RFC 7234: Hypertext Transfer Protocol (HTTP/1.1): Caching) <https://tools.ietf.org/html/rfc7234#section-4.2>

## Caching with HTTP Headers

### Using HTTP Headers

HTTP 1.1 response header fields that are cache relevant.

Field	Meaning	Defined in
Date	"the date and time at which the message was originated" ... "A response's age is the time that has passed since it was generated by, or successfully validated with, the origin server."	rfc7231

Age	"sender's estimate of the amount of time since the response was generated or successfully validated at the origin server. ... The presence of an Age header field implies that the response was not generated or validated by the origin server for this request." [That is, it is generated by a cache]	rfc7234
Expires	"the date/time after which the response is considered stale ... A cache recipient MUST interpret invalid date formats, especially the value "0", as representing a time in the past (i.e., "already expired")." [Cache-Control supersedes Expires]	rfc7234
Cache-Control	"directives for caches along the request/response chain"	rfc7234
Vary	"inform cache recipients that they MUST NOT use this response to satisfy a later request unless the later request has the same values for the listed fields as the original request"	rfc7231
Validator Headers		
Etag	"entity-tag" a token that identifies whether a representation (such as a web page) as changed.	rfc7232
Last-Modified		rfc7232

(Internet Engineering Task Force (IETF), Fielding, and Reschke 2014, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", <https://tools.ietf.org/html/rfc7231>) <https://tools.ietf.org/html/rfc7231#section-7.1>  
 (Internet Engineering Task Force (IETF) et al. 2014, "Hypertext Transfer Protocol (HTTP/1.1): Caching", <https://tools.ietf.org/html/rfc7234>) <https://tools.ietf.org/html/rfc7234#section-5>

Servers should send both an Etag and a Last-Modified header.

"the preferred behavior for an origin server is to send both a strong entity-tag and a Last-Modified value in successful responses to a retrieval request."

(Internet Engineering Task Force (IETF), Fielding, and Reschke 2014, "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests", <https://tools.ietf.org/html/rfc7232>) <https://tools.ietf.org/html/rfc7232#section-2.4>

HTTP 1.1 response header field "Cache-Control" directives.

Directive	Directive or (JLB) Clearer Word	Meaning
(JLB) sourcing options.		
no-store	no-store	Never store a page in the cache, always attempt fetch from server.
no-cache	always-validate	Always validate.
must-revalidate	validate-if-stale-error-if-unreachable	For fresh pages: use stored response without server validation. For stale pages and server reachable: validate. For stale pages and server unreachable: error (504 Gateway Timeout).

proxy-revalidate	validate-if-stale-error-if-unreachable-shared-cache	As for "must-revalidate", but not for private caches.
[none. E.g. just use 'max-age=x']	validate-if-stale-stored-response-if-unreachable	For fresh pages: use stored response without server validation. For stale pages and server reachable: validate. For stale pages and server unreachable: possibly use stored response.
(jlb) privacy options.		
public	any-cache	"any cache MAY store the response ..."
private	private-cache-only	"the response message is intended for a single user and MUST NOT be stored by a shared cache."
(jlb) freshness timing		
max-age	fresh-for	" the response is to be considered stale after its age is greater than the specified number of seconds. ... [use the form] 'max-age=5' not 'max-age="5"' ."
s-maxage	fresh-for-on-shared-cache	" in shared caches, the maximum age specified by this directive overrides the maximum age specified by either the max-age directive"
no-transform		
warning		Additional information to the status code.

(Internet Engineering Task Force (IETF) et al. 2014, "Hypertext Transfer Protocol (HTTP/1.1): Caching", <https://tools.ietf.org/html/rfc7234>) <https://tools.ietf.org/html/rfc7234#section-5.2.2>

A stored response is considered "fresh"... if the response can be reused without "validation" (checking with the origin server to see if the cached response remains valid for this request).

... A fresh response is one whose age has not yet exceeded its freshness lifetime. Conversely, a stale response is one where it has.

A response's freshness lifetime is the length of time between its generation by the origin server and its expiration time.

When a response is "fresh" in the cache, it can be used to satisfy subsequent requests without contacting the origin server.

... The primary mechanism for determining freshness is for an origin server to provide an explicit expiration time in the future, using either the Expires header field ... or the max-age response directive ...

(Internet Engineering Task Force (IETF) et al. 2014, "Hypertext Transfer Protocol (HTTP/1.1): Caching", <https://tools.ietf.org/html/rfc7234>) <https://tools.ietf.org/html/rfc7234>



If a response includes a Cache-Control field with the max-age directive (Section 5.2.2.8), a recipient MUST ignore the Expires field. Likewise, if a response includes the s-maxage directive (Section 5.2.2.9), a shared cache recipient MUST ignore the Expires field. In both these cases, the value in Expires is only intended for recipients that have not yet implemented the Cache-Control field.

(Internet Engineering Task Force (IETF) et al. 2014, "Hypertext Transfer Protocol (HTTP/1.1): Caching", <https://tools.ietf.org/html/rfc7234>) <https://tools.ietf.org/html/rfc7234>

Note that freshness applies only to cache operation; it cannot be used to force a user agent to refresh its display or reload a resource.

(Internet Engineering Task Force (IETF) et al. 2014, "Hypertext Transfer Protocol (HTTP/1.1): Caching", <https://tools.ietf.org/html/rfc7234>) <https://tools.ietf.org/html/rfc7234>

When a cached response is not fresh, it might still be reusable if it can be freshened by validation (Section 4.3) or if the origin is unavailable (Section 4.2.4).

(Internet Engineering Task Force (IETF) et al. 2014, "Hypertext Transfer Protocol (HTTP/1.1): Caching", <https://tools.ietf.org/html/rfc7234>) <https://tools.ietf.org/html/rfc7234>

... the preferred behavior for an origin server is to send both a strong entity-tag and a Last-Modified value in successful responses to a retrieval request.

(Internet Engineering Task Force (IETF) et al. 2014, "Hypertext Transfer Protocol (HTTP/1.1): Caching", <https://tools.ietf.org/html/rfc7234>) <https://tools.ietf.org/html/rfc7234>

For setting cache control policy via headers you only need to:

- Set Cache-Control with the right directives.
- Ensure your server sends an Etag header.

"Cache-Control header was defined as part of the HTTP/1.1 specification and supersedes previous headers (e.g. Expires) used to define response caching policies. All modern browsers support Cache-Control, hence that is all we will need."

(Grigorik 2019, "HTTP Caching | Web Fundamentals", <https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/http-caching>)  
<https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/http-caching>

## Vary Response Directive

Vary is a comma separated field.

```
Vary: accept-encoding, accept-language
```

(Internet Engineering Task Force (IETF), Fielding, and Reschke 2014, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", <https://tools.ietf.org/html/rfc7231>) <http://tools.ietf.org/html/rfc7231#section-7.1.4>

Values of a Vary response directive tell the cache to store a separate representation for for each different value type.

For example, a response contains ...

```
Vary: accept-language
```

... Imagine a web page is served either as English or German. The two different language versions of the one url will be stored as two representations in the cache.

When using compression and cache control always include "accept-encoding" in the `Vary` response header directive for both compressed and non-compressed responses.

```
# Response header for compressed response.
HTTP/1.1 200 OK
Content-Length: 3458
Cache-Control: max-age=86400
Content-Encoding: gzip
Vary: accept-encoding

# Response header for uncompressed response
HTTP/1.1 200 OK
Content-Length: 8365
Cache-Control: max-age=86400
Vary: accept-encoding
```

(Mulhuijzen 2014, "Best Practices for Using the Vary Header", <https://www.fastly.com/blog/best-practices-using-vary-header>) <https://www.fastly.com/blog/best-practices-for-using-the-vary-header>

### Common HTTP Header settings

Basic caching behaviour. Keep stored response fresh for 1 day (60s \* 60m \* 24h = 8640), then revalidate, possibly use stored response if the server can't be contacted. Rule for private and shared caches.

```
Cache-Control: max-age=86400, public
ETag: "3e86-410-3596fbbc"
Last-Modified: Mon, 29 Jun 1998 02:28:12 GMT
```

Force the resource to always validate with the server. Rule for private and shared caches.

```
Cache-Control: no-cache, public
ETag: "3e86-410-3596fbbc"
Last-Modified: Mon, 29 Jun 1998 02:28:12 GMT
```

### Caching with URL fingerprinting

Cache control via Url versioning or "Url fingerprinting" is a technique that allows you to instantly update web files that are set to be cached (kept fresh) for a long time. The technique entails that you:

1. Set some child web files to be cached (kept fresh) for a long time. E.g. You might set \*.css files to be served with a response header directive ...

```
"Cache-Control: max-age=3153600, public" (a year).
```

2. Include a version code (datetime stamps are a good candidate) in the file name of the target, child, long-term cached file and update this version code everytime you edit the file. E.g.

```
my-style-20151008-1904.css
```

3. The parent web file, the web file that references the target, child, long-term cached file, must be:
  - o Updated to reference the new child file name; and
  - o Set to be cached with relatively short freshness lifetimes, or not cached at all.

```
# parent.xhtml ...
```

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-gb" xml:lang="en-gb">
<head>
...
  <link rel="stylesheet" href="../my-style-20151008-1904.css" />

# parent.xhtml response header set to always revalidate with the server ...
Cache-Control: no-cache, public
```

(Ellingwood 2015, "Web Caching Basics: Terminology, HTTP Headers, and Caching Strategies", <https://www.digitalocean.com/community/tutorials/web-caching-basics-terminology-http-headers-and-caching-strategies>) <https://www.digitalocean.com/community/tutorials/web-caching-basics-terminology-http-headers-and-caching-strategies>

## General Caching Strategy

The default general strategy for specifying the freshness lifetime of web files (and therefore the time of which a web files will come from the local cache), uses HTTP headers only (no URL fingerprinting), with Cache-Control directives as follows:

File Set	Cache-Control value	Reason
Default	"max-age=0, public" (or could use "no-cache, public") - Always validate.	We want users to always validate with the sever unless intentionally overridden.
Supporting web resourcess: like javascript, css, logos, favicon	<ul style="list-style-type: none"> <li>Under development: "max-age=0, public" - Always validate.</li> <li>When stable (in production after beta): "max-age=259200, public" - Validate after 3 days.</li> </ul>	In production we don't want a local copy to be fresh for too long, in case there are changes to those supporting files.
General markup files: html, xhtml, php.	<ul style="list-style-type: none"> <li>Under development: "max-age=0, public" - Always validate.</li> <li>When stable (in production after beta): "max-age=259200, public" - Validate after 3 days.</li> </ul>	As for supporting web resources. But target them separately in order to facilitate cache strategy changes (e.g. a shorter freshness lifetime for general markup files).
Frequently updated markup files: html, xhtml, php	"max-age=0, public" (or could use "non-cache, public") - Always validate.	Frequently updated (by the developer) mark up files ought checked with the server to see if the local copy is different from the server copy.

## Implementation techniques

### Example Apache and Litespeed .htaccess file

Example Apache and Litespeed .htaccess file settings for cache control [See C:\Users\John\Documents\Sda\Code\web\Apps\Philorum\source\.htaccess]:

```
##### Compression #####

AddOutputFilterByType DEFLATE text/html text/plain text/xml text/css application/x-
javascript application/javascript text/javascript application/xml image/jpeg image/gif
image/png image/svg+xml

##### Caching #####

# -----
```

```

# General
# -----
# Designed to work under Litespeed and Apache 2.4

# -----
# LiteSpeed Notes
# -----
# Litespeed doesn't support "Header set" or "Header merge" very well, or at all.
# We therefore use something like:
# Header set Cache-Control "max-age=0, public"

# -----
# We use compression with caching despite no "304 Not Modified" in Apache
# -----
# We use compression with caching even though there is Apache Bug 45023 (of Version 2.4):
# Serving compressed files (using mod_deflate, aka "gzip") prevents "304 Not Modified"
# responses. Solution: use compression and cache control anyway. Compress your files
# anyway and bite the bullet that when your pages are stale they will be fully served
# rather than merely re-freshed. Even under these circumstances you benefit from cache
# control in virtue of facilitating using a copy of fresh pages from the local cache; and
# when you do serve a page it will be reduced in size in virtue of compression. See
# \\ATLAS\Documents\Sda\Info\Web\KB\Xhtml\QuickReference\Html5-PolyglotMarkup-
QuickReference.docx

# Litespeed servers "304 Not Modified" when appropriate, with compression turned on.

# -----
# Deprecated: mod_expires
# -----
# mod_expires sets both the Expires header and the max-age value in the Cache-Control
# Header. We don't care about the Expires header because Cache-Control and Etags supercede
# it. But setting max-age with mod_expires is superior than setting max-age with Header (
# directly), as setting max-age with mod_expires affords the opportunity to use plain
# english time intervals.
# E.g. "access plus 1 month 15 days 2 hours", or "access plus 1 days", etc.
# ExpiresActive On
# ExpiresDefault "access plus 2 hours"

# -----
# Etag Header
# -----
# Include here as some Web Hosts have turned off Etags by default.
# Don't use INode as part of the Etag as that will falsely flag a changed document
# when a file is served from different server nodes (in a cluster)
# http://httpd.apache.org/docs/2.4/mod/core.html#fileetag
FileETag -INode +Size +MTime

# -----
# Default Cache-Control
# -----
# Always Validate
# We want users to always validate with the sever unless intentionally overridden.
Header set Cache-Control "max-age=0, public"

# -----
# Seconds Conversion
# -----
# 259200 seconds = 3 days.
# 86400 seconds = 1 day.
# 3600 seconds = 1 hour.
# 900 seconds = 15 minutes.

# -----
# Supporting Web Resources Cache-Control
# -----
# In production we don't want a local copy to be fresh for too long, in case
# there are changes to those supporting files.
# In production set to 259200 seconds = 3 days.
<FilesMatch "^.+\. (css|js|ico|png|jpg|kml)$">
Header set Cache-Control "max-age=259200, public"
</FilesMatch>

```

```
# -----
# General Markup Files
# -----
# As for supporting web resources. But target them separately in order to
# facilitate cache strategy changes (e.g. a shorter freshness lifetime for
# general markup files).
# In production set to 259200 seconds = 3 days.
<FilesMatch "^.+\.(xhtml|html|php)$">
  Header set Cache-Control "max-age=259200, public"
</FilesMatch>

# -----
# Frequently Updated Markup files
# -----
# Frequently updated (by the developer) mark up files ought checked with the
# server to see if the local copy is different from the server copy.
# That is, always validate.
<FilesMatch "^(index|centralNextMeeting|centralPastCalendar|centralFutureCalendar|)\.+$">
  Header set Cache-Control "max-age=0, public"
</FilesMatch>
```

(Bentley, n.d., "Experimentally Verified")

## Apache - Browser Caching

If the Expires and Cache-Control headers are set by mod\_expires, an Apache module, then they will override the corresponding headers set by Php (or other script).

(Apache Software Foundation 2019, "Apache HTTP Server Version 2.4 Documentation", <http://httpd.apache.org/docs/current/>) "Apache Module mod\_expires", [http://httpd.apache.org/docs/current/mod/mod\\_expires.html](http://httpd.apache.org/docs/current/mod/mod_expires.html)

Technique for editing max-age in Cache-Control header, if Cache-Control already has many values. Uses regex search string and replace string.

```
Header set Cache-Control "max-age=7200"

# We must merge, rather than set, as Cache-Control "max-age=X" has
# been set by mod_expires above.
# So the resulting header will be something like:
# Cache-Control: max-age=x, public
Header merge Cache-Control "public"
Header edit Cache-Control "(max-age=\d+)" "max-age=2"
```

(Bentley, n.d.) 2015-09-30 22:19

Only use the `Header` directive to set (or merge, edit, ect) the `max-age` property of the `Cache-Control` header. Don't use `mod_expires` to set `Expires` and `max-age` (of `Cache-Control`) as any subsequent overriding of `max-age` (as when using regexes with `Header edit`) will:

- Be difficult to read as a matter of reading code; and
- Will make the `Expires` header out of sync with the value overridden for `max-age`.

```
# In .htaccess
#####
# Do this
#####
# 7200 seconds = 2 hours * 60 minutes * 60 seconds.
Header set Cache-Control "max-age=7200"
```

```
# We must merge, rather than set, as Cache-Control "max-age=X" has
# been set by mod_expires above.
# So the resulting header will be something like:
# Cache-Control: max-age=x, public
Header merge Cache-Control "public"

# Overriding
Header edit Cache-Control "(max-age=\d+)" "max-age=2"

#####
# Don't do this
#####
# Also sets Cache-Control max-age.
ExpiresActive On
ExpiresDefault "access plus 2 hours"

Header merge Cache-Control "public"

# Overriding
Header edit Cache-Control "(max-age=\d+)" "max-age=2"
```

(Bentley, n.d., "Experimentally Verified") 2015-09-30 22:19

Apache Bug 45023 (of Version 2.4): Serving compressed files (using mod\_deflate, aka "gzip") prevents "304 Not Modified" responses. Solution: use compression and cache control anyway.

Bug explanation: When a stale local copy is validated against the server the normal, required, caching behaviour is to return a "304 Not Modified" status message, without the payload server copy. Instead a full copy of the server version is served, along with a "200 OK" status message.

Request Header:

```
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:41.0) Gecko/20100101 Firefox/41.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://localhost:8080/web/Libraries/Html5Library/caching-test-02.xhtml
Connection: keep-alive
If-Modified-Since: Thu, 01 Oct 2015 14:17:47 GMT
If-None-Match: "8981-5210bb625995e-gzip"
```

Response Header:

```
Accept-Ranges: bytes
Cache-Control: max-age=10, public
Connection: Keep-Alive
Content-Encoding: gzip
Content-Length: 10806
Content-Type: application/xhtml+xml
Date: Tue, 06 Oct 2015 07:36:25 GMT
Etag: "8981-5210bb625995e-gzip"
Keep-Alive: timeout=5, max=100
Last-Modified: Thu, 01 Oct 2015 14:17:47 GMT
Server: Apache/2.4.12 (Win32) OpenSSL/1.0.2a PHP/5.6.10
Vary: Accept-Encoding
```

Response Status Code [Even though the values in Etag and If-None-Match are matching (and the stored copy is stale at the time of the request)]:

200 OK [And the entire payload is served]

Expected Response Code:

304 Not Modified [And only headers, not the payload is served]

Solution explanations: Compress your files anyway and bite the bullet that when your pages are stale they will be fully served rather than merely re-freshed. Even under these circumstances you benefit from cache control in virtue of facilitating using a copy of fresh pages from the local cache; and when you do serve a page it will be reduced in size in virtue of compression. Turning off compression for additional caching benefits is generally not the better trade off.

(Apache Software Foundation n.d., "Bugzilla", Accessed 2019-02-15. <https://bz.apache.org/bugzilla/>) Bug 45023 - DEFLATE preventing 304 NOT MODIFIED response, [https://bz.apache.org/bugzilla/show\\_bug.cgi?id=45023](https://bz.apache.org/bugzilla/show_bug.cgi?id=45023)  
(Apache Software Foundation n.d., "Bugzilla", Accessed 2019-02-15. <https://bz.apache.org/bugzilla/>) Bug 39727 - Incorrect ETag on gzip:ed content, [https://bz.apache.org/bugzilla/show\\_bug.cgi?id=39727](https://bz.apache.org/bugzilla/show_bug.cgi?id=39727)

Full browser caching .htaccess template.

```
##### Compression #####

AddOutputFilterByType DEFLATE text/html application/xhtml+xml text/plain text/xml text/css
application/x-javascript text/javascript
# Disable Compression in .htaccess.
# SetEnv no-gzip 1

##### Caching #####

# -----
# General
# -----
# Designed to work under Litespeed and Apache 2.4

# -----
# LiteSpeed Notes
# -----
# Litespeed doesn't support "Header set" or "Header merge" very well, or at all.
# We therefore use something like:
# Header set Cache-Control "max-age=0, public"
<IfModule LiteSpeed>
    CacheEnable public /
    CacheEnable private /
</IfModule>

# -----
# We use compression with caching despite no "304 Not Modified" in Apache
# -----
# We use compression with caching even though there is Apache Bug 45023 (of Version 2.4):
# Serving compressed files (using mod_deflate, aka "gzip") prevents "304 Not Modified"
# responses. Solution: use compression and cache control anyway. Compress your files
# anyway and bite the bullet that when your pages are stale they will be fully served
# rather than merely re-freshed. Even under these circumstances you benefit from cache
# control in virtue of facilitating using a copy of fresh pages from the local cache; and
# when you do serve a page it will be reduced in size in virtue of compression. See
# \\ATLAS\Documents\Sda\Info\Web\KB\Xhtml\QuickReference\Html5-PolyglotMarkup-
QuickReference.docx

# Litespeed servers "304 Not Modified" when appropriate, with compression turned on.

# -----
# Vary
# -----
# When using compression and cache control always include "accept-encoding" in the
# Vary response header directive for both compressed and non-compressed responses.
Header set Vary "accept-encoding"

# -----
# Deprecated: mod_expires
# -----
# mod_expires sets both the Expires header and the max-age value in the Cache-Control
# Header. We don't care about the Expires header because Cache-Control and Etags supercede
```

```

# it. But setting max-age with mod_expires is superior than setting max-age with Header (
# directly), as setting max-age with mod_expires affords the opportunity to use plain
# english time intervals.
# E.g. "access plus 1 month 15 days 2 hours", or "access plus 1 days", etc.
# ExpiresActive On
# ExpiresDefault "access plus 2 hours"

# -----
# Etag Header
# -----
# Include here as some Web Hosts have turned off Etags by default.
# Don't use INode as part of the Etag as that will falsely flag a changed document
# when a file is served from any of several servers (in a cluster)
# http://httpd.apache.org/docs/2.4/mod/core.html#fileetag
FileETag -INode +Size +MTime

# -----
# Default Cache-Control
# -----
# Always Validate
# We want users to always validate with the sever unless intentionally overridden.
Header set Cache-Control "max-age=10, public"

# -----
# Supporting Web Resources Cache-Control
# -----
# In production we don't want a local copy to be fresh for too long, in case
# there are changes to those supporting files.
# 259200 seconds = 3 days.
<FilesMatch "^.\.(css|js|ico|png|jpg|kml)$">
  Header set Cache-Control "max-age=259200, public"
</FilesMatch>

# -----
# General Markup Files
# -----
# As for supporting web resources. But target them separately in order to
# facilitate cache strategy changes (e.g. a shorter freshness lifetime for
# general markup files).
# 259200 seconds = 3 days.
<FilesMatch "^.\.(xhtml|html|php)$">
  Header set Cache-Control "max-age=10, public"
</FilesMatch>

# -----
# Frequently Updated Markup files
# -----
# Frequently updated (by the developer) mark up files ought checked with the
# server to see if the local copy is different from the server copy.
# That is, always validate.
<FilesMatch "^(index|centralNextMeeting|centralPastCalendar|centralFutureCalendar|)\..+$">
  Header set Cache-Control "max-age=0, public"
</FilesMatch>

```

(Bentley, n.d., "Experimentally Verified") 2015-10-26 18:32

## Apache - Server Caching

Be aware of the difference between a browser cache and a server cache. Apache (should) support browser caching by default. The Apache server cache is supported through mod\_cache. For the Apache server cache see "Caching Guide"

(Apache Software Foundation 2019, "Apache HTTP Server Version 2.4 Documentation", <http://httpd.apache.org/docs/current/>) Caching Guide, <http://httpd.apache.org/docs/2.4/caching.html>



We are generally interested in Apache `mod_cache` HTTP caching (using the `mod_cache` dependency: `mod_cache_disk`). For SSL sessions we might want to use shared object caching (using the `mod_cache_socache` dependency: `mod_cache_socache`).

(Apache Software Foundation 2019, "Apache HTTP Server Version 2.4 Documentation", <http://httpd.apache.org/docs/current/>) Caching Guide, <http://httpd.apache.org/docs/2.4/caching.html>

Generally turning on server side Apache HTTP caching (using `mod_cache` and `mod_cache_disk`, for example) entails placing directives in the server config, or virtual host contexts.

```
#
# Sample Cache Configuration in httpd.conf
#
##### CACHING - SERVER SIDE #####
LoadModule cache_module modules/mod_cache.so
<IfModule mod_cache.c>
    # Enable cache lock.
    CacheLock on
    CacheLockPath "C:/Program Files (x86)/Apache Software
Foundation/Apache24/cache/tmp/mod_cache-lock"
    CacheLockMaxAge 5

    # For safety. 259200 seconds = 3 days.
    # A response header like <Cache-Control "max-age=10, public"> still controls the browser
    # Cache.
    CacheMaxExpire 259200

    # Enable the X-Cache header
    CacheHeader on

    # Enable the X-Cache-Detail header
    CacheDetailHeader on

    LogFormat "%{cache-status}e %h %l %u %t \"%r\" %>s %b" cachelog
    CustomLog "C:/Program Files (x86)/Apache Software Foundation/Apache24/logs/cache.log"
cachelog
    # When acting as a proxy, don't cache the list of security updates
    #CacheDisable "http://security.update.server/update-list/"

    LoadModule cache_disk module modules/mod_cache_disk.so
    <IfModule mod_cache_disk.c>
        CacheRoot "C:/Program Files (x86)/Apache Software
Foundation/Apache24/cache/cacheroot"
        CacheEnable disk "/"
        CacheDirLevels 2
        CacheDirLength 1
    </IfModule>
</IfModule>
```

(Apache Software Foundation 2019, "Apache HTTP Server Version 2.4 Documentation", <http://httpd.apache.org/docs/current/>), `mod_cache`, [http://httpd.apache.org/docs/2.4/mod/mod\\_cache.html](http://httpd.apache.org/docs/2.4/mod/mod_cache.html)

Cache locks prevent the backend server from being hit too much. It tells other requests not to contact the backend server as a request has already been made.

```
Enabling the cache lock

#
# Enable the cache lock
#
<IfModule mod_cache.c>
    CacheLock on
    CacheLockPath "/tmp/mod_cache-lock"
    CacheLockMaxAge 5
</IfModule>
```

(Apache Software Foundation 2019, "Apache HTTP Server Version 2.4 Documentation", <http://httpd.apache.org/docs/current/>) `mod_cache`, [http://httpd.apache.org/docs/2.4/mod/mod\\_cache.html](http://httpd.apache.org/docs/2.4/mod/mod_cache.html)

By default the server side cache will be handled with a "quick handler". In this mode the cache "bolt onto" the front of the server, offering best performance. If you want to control when caching takes place turn the quick handler off and specify the process order. This is called specifying the "filter chain".

```
# Cache content before optional compression
CacheQuickHandler off
AddOutputFilterByType CACHE;DEFLATE text/plain

# Cache content between mod_include and mod_deflate
CacheQuickHandler off
AddOutputFilterByType INCLUDES;CACHE;DEFLATE text/html
```

(The Apache Software Foundation, 2015. Version 2.4 Documentation), *mod\_cache*, Fine Control with the CACHE Filter, [http://httpd.apache.org/docs/2.4/mod/mod\\_cache.html](http://httpd.apache.org/docs/2.4/mod/mod_cache.html)

If you change the order of the filter chain, clean your cache.

(Apache Software Foundation 2019, "Apache HTTP Server Version 2.4 Documentation", <http://httpd.apache.org/docs/current/>) *mod\_cache*, Fine Control with the CACHE Filter [http://httpd.apache.org/docs/2.4/mod/mod\\_cache.html](http://httpd.apache.org/docs/2.4/mod/mod_cache.html)

Clean your cache with `htcacheclean`. This can be run ad hoc, from the command line, or has a periodic job.

```
# Stop the apache service
.\htcacheclean.exe -r -v -p "C:/Program Files (x86)/Apache Software
Foundation/Apache24/cache/cacheclean" -l 1K
```

(Bentley, n.d., "Experimentally Verified") 2015-10-25 09:27  
(Apache Software Foundation 2019, "Apache HTTP Server Version 2.4 Documentation", <http://httpd.apache.org/docs/current/>) Caching Guide, <http://httpd.apache.org/docs/2.4/caching.html>

Analyze caching decisions either:

- By altering the output of your LogFormat; and/or
- Including a `X-Cache-Detail` response directive.

```
LogFormat "%{cache-status}e ..."

# Enable the X-Cache-Detail header
CacheDetailHeader on

# We get back something like
X-Cache-Detail: "conditional cache hit: entity refreshed" from localhost
```

(Apache Software Foundation 2019, "Apache HTTP Server Version 2.4 Documentation", <http://httpd.apache.org/docs/current/>) *mod\_cache*, Cache Status and Loggin, [http://httpd.apache.org/docs/2.4/mod/mod\\_cache.html](http://httpd.apache.org/docs/2.4/mod/mod_cache.html)

## Litespeed Notes

Litespeed @ VentrailP missing etag bug. With cache control is setup in `.htaccess` and when a "304 Not modified" response header status code is returned (through surfing between web pages), an Etag directive should be included in the response header. But it isn't: the Etag directive is missing.

Steps to reproduce:

1. `.htaccess` with `cache-control` set to: always validate browser cached version against server representation.

```
# -----
```

```
# Etag Header
# -----
# Include here as some Web Hosts have turned off Etags by default.
# Don't use INode as part of the Etag as that will falsely flag a changed document
# when a file is served from any of several servers (in a cluster)
# http://httpd.apache.org/docs/2.4/mod/core.html#fileetag
FileETag -INode +Size +MTime

# -----
# Browser Caching Rule
# -----
# Files ought checked with the
# server to see if the local copy is different from the server copy.
# That is, always validate.
<FilesMatch "^(index|centralNextMeeting|centralPastCalendar|centralFutureCalendar|)\.+$">
  Header set Cache-Control "max-age=0, public"
</FilesMatch>
```

## 2. In Firefox ...

3. Surf to <http://www.philorum.org/index.html>. Press F12 for developer tools > Network Tab. Click "reload" in developer tools Network tab.
4. Ctrl + F5 for a forced retrieval of files from the server. In Developer Tools > Network > Click on the index.html record > Headers [Tab]
5. Observe Status Code "200 OK" and, in the response header, (something like) Etag: "'19ac-5618a670-0'". This all looks good so far.
6. On the webpage <http://www.philorum.org/index.html> click on "Draft Constitution" link, to surf away from index.html.
7. Click on "Home" to surf back <http://www.philorum.org/index.html>. Observe, in Developer Tools > Network > Index.html > Headers .... Status Code "304 Not Modified" and, in the response header, no Etag.

Expected:

7. I expected Status Code "304 Not Modified" and, in the response header, an Etag (with a value identical to the previously cached Etag, to assert that index.html has not been modified). In short, there is a missing Etag.

Other notes:

- \* This is identified at <https://redbot.org/?uri=http%3A%2F%2Fwww.philorum.org> when mousing over the warnings for "The If-None-Match response is missing required headers" and "The If-Modified-Since response is missing required headers" there is information about the missing Etag.
- \* This problem occurs whether the web pages are compressed (gzip) or not compressed.
- \* In firefox (at least) a "304 not modified" response correctly does not come with a message payload (the full webpage) - it is taken from the cache. This is ultimately what the "304 not modified" response status code is for ... so the caching \*is\* behaving as expected and desired - however without the Etag as is required. (So the caching works as desired but by luck rather than design).

mistwang, LiteSpeed Staff, reports

This is caused by bug introduced in LSWS 5.0, it will be addressed in next build. Anyway, browser today can tolerant that, it does not affect caching function.

(Bentley, n.d., "Experimentally Verified") 2015-10-12 20:17

Litespeed support forums > '[Bug?] Browser caching. Missing Etag on "304 Not Modified".'

<https://www.litespeedtech.com/support/forum/threads/bug-browser-caching-missing-etag-on-304-not-modified.13261/>

## Php files

Php files served to the browser can be cached like regular HTML pages. For example, you can set it that the php file should be fresh for 15 minutes and therefore the browser will not check the server to validate for 15 minutes.

## Caching Behaviour

To ensure you retrieve web files as currently stored on the server use Ctrl + F5. Other kinds of commands (E.g. In-Page Link clicking, or F5) generally invoke normal caching behaviour, validating local stored copies against the sever and using that local version if it has not been modified at the server.

Old school caching behaviour uses "If-Modified-Since" and other headers. New School caching essentially relies on comparisons between the Etag sent in Request header "If-None-Match" and the Etag of the server copy (as identified when sent back in a Response Etag header). Therefore the absence "If-None-Match" is key to determining whether a

Served from Apache, without compression, after the local stored copy has been allowed to become stale (Waiting for the number of seconds from the response "Cache-Control: max-age=x; public" to elapse), with the following response code effects:

- 304 Not Modified: entails that the stored version in the local cache has passed validation with the server and the response only includes the header, not the payload. The local stored version is therefore loaded from the local cache and made fresh once again.
- 200 OK: a payload from the server is sent.

Action	Firefox 41.0.1	Chrome 45.0.2454.101	IE 11.0.10240.16431
In-Page Link Click	[Request: Not sent ~ Cache-Control; <b>If-None-Match</b> sent] [Reponse: 304 Not Modified.]	[Request: Not sent ~ Cache-Control; <b>If-None-Match</b> sent] [Reponse: 304 Not Modified.]	[Request: Not sent ~ Cache-Control; <b>If-None-Match</b> sent] [Reponse: 304 Not Modified.]
Address bar + Enter	[Request: Not sent ~ Cache-Control; <b>If-None-Match</b> sent] [Reponse: 304 Not Modified.]	[Request: Cache-Control: " <b>max-age=0</b> "; <b>If-None-Match</b> sent] [Response: 304 Not Modified]	[Request: Not sent ~ Cache-Control; <b>If-None-Match</b> sent] [Reponse: 304 Not Modified.]
F5	[Request: Cache-Control: " <b>max-age=0</b> "; <b>If-None-Match</b> sent] [Response: 304 Not Modified]	[Request: Cache-Control: " <b>max-age=0</b> "; <b>If-None-Match</b> sent] [Response: 304 Not Modified]	[Request: Not sent ~ Cache-Control; <b>If-None-Match</b> sent] [Reponse: 304 Not Modified.]
Ctrl + R	[Request: Cache-Control: " <b>max-age=0</b> "; <b>If-None-Match</b> sent] [Response: 304 Not Modified]	[Request: Cache-Control: " <b>max-age=0</b> "; <b>If-None-Match</b> sent] [Response: 304 Not Modified]	[Request: Cache-Control: " <b>no-cache</b> "; Not set ~ If-None-Match] [Response: 200 OK]
Refresh Button	[Request: Cache-Control: " <b>max-age=0</b> "; <b>If-None-Match</b> sent] [Response: 304 Not Modified]	[Request: Cache-Control: " <b>max-age=0</b> "; <b>If-None-Match</b> sent] [Response: 304 Not Modified]	[Request: Not sent ~ Cache-Control; <b>If-None-Match</b> sent] [Reponse: 304 Not Modified.]
Ctrl + F5	[Request: Cache-Control: " <b>no-cache</b> "; Not set ~ If-None-Match] [Response: 200 OK]	[Request: Cache-Control: " <b>no-cache</b> "; Not set ~ If-None-Match] [Response: 200 OK]	[Request: Cache-Control: " <b>no-cache</b> "; Not set ~ If-None-Match] [Response: 200 OK]

In Firefox 41.0.1 fresh pages are served from the local cache, with the headers like this:

```
[Request:
Not sent ~ Cache-Control;
Not set ~ If-None-Match]
[Response:
200 OK.]
```

In Firefox developer tools see Network > Transferred = "cached".

Note that serving a compressed page from Apache prevents a response of "304 not modified" and header only. See "Apache Bug 45023" above.

*(Bentley, n.d., "Experimentally Verified") 2015-10-06 19:03  
(Stack Exchange Inc n.d., "Stack Overflow", Accessed 2019-02-15. <https://stackoverflow.com/>) "What requests do browsers' "F5" and "Ctrl + F5" refreshes generate?", <http://stackoverflow.com/questions/385367/what-requests-do-browsers-f5-and-ctrl-f5-refreshes-generate>*

Local/Browser caching seems to work identically under https as http

*(Bentley, n.d., "Experimentally Verified") 2019-08-02 with <https://atlas/web/Libraries/Html5Library/html5-polyglot-template-040-project/>*

## Caching Tools

Analyze your cache settings: <https://redbot.org>

## Range requests

Range requests occur when a client has stored only a partial representation, and needs the rest from the server. Servers indicate their support for ranged request through the `Accept-Ranges` response directive. When serving partial content, in response to a range request, a "206 Partial Content" status code is sent.

```
# Server supports ranged requests.
# Generally only "bytes" type ranged requests are supported.
Accept-Ranges: bytes

# The server does not support ranged requests.
Accept-Ranges: none
```

*(Internet Engineering Task Force (IETF) et al. 2014, "Hypertext Transfer Protocol (HTTP/1.1): Range Requests", <https://tools.ietf.org/html/rfc7233>) <https://tools.ietf.org/html/rfc7233>*

Ranged requests are supported and enabled by default in:

1. Litespeed. <https://www.litespeedtech.com/support/forum/threads/byte-range-requests.5946/>; and
2. [\*fix: apache?]

## Tools for HTTP Traffic.

Developer tools of various browsers (F12).

Microsoft Message Analyzer.

## cURL

cURL is a command line client that is an alternative to telnet, that works well with http and https (Telnet doesn't support https). Useful for debugging page fetch errors and problems.

[http://httpd.apache.org/docs/2.4/ssl/ssl\\_faq.html](http://httpd.apache.org/docs/2.4/ssl/ssl_faq.html)

cURL home: <http://curl.haxx.se/>

cURL download for Windows.

3. From "cURL for Windows", by " Confused by Code", <http://www.confusedbycode.com/curl/>
4. Choose: 64-Bit; "With Administrator Privileges [only works in a command line opened with admin privileges]" - e.g. curl-7.43.0-win64.msi:

cURL usage. Open a windows command line with admin privileges then type "curl -h" for help.

cURL Examples.

```
// Return HTML body via standard HTTP
>curl http://localhost:8080/

// Return HTML body via HTTPS
>curl https://localhost:443/

// (or) defaults to port 443 for https.
>curl https://localhost/
```

## Powershell Invoke-WebRequest

As of Powershell 5.0, if not before, curl is an alias for Invoke-WebRequest.

```
PS> Get-Alias -Definition Invoke-WebRequest | Format-Table -AutoSize

CommandType Name                Version Source
-----
Alias        curl -> Invoke-WebRequest
Alias        iwr -> Invoke-WebRequest
Alias        wget -> Invoke-WebRequest

# To use the unaliased command ...

PS> Invoke-WebRequest -Uri https://atlas:443/
PS> Invoke-WebRequest -Uri https://localhost:443/
PS> Invoke-WebRequest -Uri https://www.google.com
```

*(Bentley, n.d.) 2015-10-31 10:23*

## Abbreviations

JLB  
jlb

Stipulations of a John Bentley convention (above what Polyglot markup or the HTML spec provides).

---

# Code (HTML, CSS)

---

## HTML References

(Bentley 2019, "Xhtml-Reference.Docx",  
\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Xhtml\Reference\Xhtml-Reference.docx)  
<\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Xhtml\Reference\Html-AsPolyglotMarkup-Reference.docx>

(Bentley 2019, "Xhtml-PolyglotMarkupRules.Docx",  
\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Xhtml\Reference\Xhtml-PolyglotMarkupRules.docx)  
<\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Xhtml\Reference\Html-PolyglotMarkupRules.docx>

## Css Reference

(Bentley 2018, "Cascading Style Sheets - Css -Reference",  
\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Css\Reference\CascadingStyleSheets-Css-Reference.docx)  
<\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Css\Reference\CascadingStyleSheets-Css-Reference.docx>

## Mobile friendliness and responsive design

### Overview

You ought build web pages to be mobile friendly. That entails the web page:

- Has "responsive page design": readable regardless of device screen size (and indeed regardless of whether the device has a screen), sometimes known as "liquid in layout"; and
- Has a responsive menu: the menu works regardless of device screen size (and indeed regardless of whether the device has a screen).
- Takes touch commands.

### Responsive page design

#### Principles

Implementing responsive design entails:

1. Using a specific `meta name="viewport" code.`

```
<!-- Make page mobile friendly -->  
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

(Google 2018, "Be Mobile Friendly > Welcome", <https://developers.google.com/search/mobile-sites/>) *Be Mobile-Friendly, Choose a configuration, Responsive Web Design*, <https://developers.google.com/search/mobile-sites/mobile-seo/responsive-design>

2. Applying the right CSS so that all the page elements resize and flow as you change (decrease) screen size.
  - a. Use relative width values (e.g. width: 100%) rather than large fixed width values (e.g. width: 330px). (Google, 2016)

<https://developers.google.com/web/fundamentals/design-and-ui/responsive/>.

- b. Use media queries to change styles based on media type.

```
<link rel="stylesheet" href="print.css" media="print">

// or

<style>
  /*  */
  @media print {
    /* print style sheets go here */
  }
  /*  */
</style>

// or

// In an external *.css file
@media print {
  /* print style sheets go here */
}
```

(LePage 2019, "Responsive Web Design Basics | Web Fundamentals",  
<https://developers.google.com/web/fundamentals/design-and-ux/responsive/>)  
<https://developers.google.com/web/fundamentals/design-and-ui/responsive/>

- c. Use media queries to change styles based on browser window size or orientation. Use min-width over min-device-width (do not use min-device-width).

min-width	Rules applied for any browser width greater than the value defined in the query.
max-width	Rules applied for any browser width less than the value defined in the query.
min-height	Rules applied for any browser height greater than the value defined in the query.
max-height	Rules applied for any browser height less than the value defined in the query.
orientation=portrait	Rules applied for any browser where the height is greater than or equal to the width.
orientation=landscape	Rules for any browser where the width is greater than the height.

(LePage 2019, "Responsive Web Design Basics | Web Fundamentals",  
<https://developers.google.com/web/fundamentals/design-and-ux/responsive/>)  
<https://developers.google.com/web/fundamentals/design-and-ui/responsive/>

```
<link rel="stylesheet" media="(max-width: 640px)" href="max-640px.css">
<link rel="stylesheet" media="(min-width: 640px)" href="min-640px.css">
<link rel="stylesheet" media="(orientation: portrait)" href="portrait.css">
<link rel="stylesheet" media="(orientation: landscape)" href="landscape.css">

// Or

<style>
  /*  */
  @media (min-width: 500px) and (max-width: 600px) {
    h1 {
      color: fuchsia;
    }
  }
  /*  */
</style>
```



```
.desc:after {
  content:" In fact, it's between 500px and 600px wide.";
}
}
/* ]]> */
</style>
```

3. Design for the smallest mobile device first, then "progressively enhance".
4. Make layout decisions based on screen size, not device brand.
5. Keep lines of text to about 70 to 80 characters.
6. Never make invisible elements just because it can't fit on the screen (just let it not fit on the screen). E.g. Don't remove pollen count on a weather forecast when on a mobile on the basis that pollen count is less important than other pieces of data.
7. Use flex-box, the in-vogue css layout technique for the page.

(LePage 2019, "Responsive Web Design Basics | Web Fundamentals",  
<https://developers.google.com/web/fundamentals/design-and-ux/responsive/>  
<https://developers.google.com/web/fundamentals/design-and-ui/responsive/>

8. Testing your website at various screen sizes (minimally a typical mobile screen size). Google's Desktop Chrome browser, developer tools (F12) in Device Mode is ideal for this:
  - a. Choose the responsive option and resize.
  - b. Choose "Show Media Queries" from the menu.

## With flex-box

For using flex-box to achieve responsive design see

(LePage 2019, "Responsive Web Design Basics | Web Fundamentals",  
<https://developers.google.com/web/fundamentals/design-and-ux/responsive/>  
<https://developers.google.com/web/fundamentals/design-and-ui/responsive/>,  
<https://developers.google.com/web/fundamentals/design-and-ui/responsive/patterns>

(Bentley 2018, "Cascading Style Sheets - Css -Reference",  
 \\Atlas\Users\John\Documents\Sda\Info\Web\KB\Css\Reference\CascadingStyleSheets-Css-Reference.docx),  
 Flexbox <\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Css\Reference\CascadingStyleSheets-Css-Reference.docx>

## Holy grail layout

(Bentley 2018, "Cascading Style Sheets - Css -Reference",  
 \\Atlas\Users\John\Documents\Sda\Info\Web\KB\Css\Reference\CascadingStyleSheets-Css-Reference.docx),  
 Flexbox > Flexbox techniques > Holy Grail Layout  
<\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Css\Reference\CascadingStyleSheets-Css-Reference.docx>

## Responsive Menus

Menu using collapsed flex item, leaving an invisible "strut" behind that keep's the flex item's cross-size.

(Bentley 2018, "Cascading Style Sheets - Css -Reference",  
 \\Atlas\Users\John\Documents\Sda\Info\Web\KB\Css\Reference\CascadingStyleSheets-Css-Reference.docx)  
<\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Css\Reference\CascadingStyleSheets-Css-Reference.docx>,  
 "Flexbox", "Collapsed Flex Item".

Sliding (and hamburger) menus using the "off canvas" technique. See

(LePage 2019, "Responsive Web Design Basics | Web Fundamentals",  
<https://developers.google.com/web/fundamentals/design-and-ux/responsive/>)  
<https://developers.google.com/web/fundamentals/design-and-ui/responsive/patterns>, "Off canvas" technique.

Bentley's sliding/hamburger menu using left setting and css transitions.

```
function showNavigation(show) {
  var navElement = document.getElementsByTagName("nav")[0];
  var bodyElement = document.getElementsByTagName("body")[0];
  var scrimElement = document.getElementById("scrim");
  if (show) {
    // Transitions speeds set according to Material Design > Material Foundation > Motion >
    // Speed > Duration > Exits and Closing
    // https://material.io/design/motion/speed.html#duration
    scrimElement.classList.add("scrim");
    navElement.style.transitionDuration = "0.250s";
    navElement.style.left = "0px";
  } else {
    scrimElement.classList.remove("scrim");
    navElement.style.transitionDuration = "0.200s";
    navElement.style.left = "-200px";
  }
}

window.onload = function() {
  document.getElementById("hamburger-icon").addEventListener('click', function(e) {
    showNavigation(true);
  }, true);
  document.getElementsByTagName("article")[0].addEventListener('click', function(e) {
    showNavigation(false);
  }, true);
}

<body>
  <main>
    <nav>
      <ul>
        <li><a href="#">Cumque minus </a></li>
        <li><a href="#">Laboriosam optio</a></li>
        <li><a href="#">Provident delectus</a></li>
        <li><span class="selected">Aliquam aperiam</span></li>
      </ul>
    </nav>
    <article>
      <div id="scrim">
        
      </div>
    </article>
  </main>
</body>
```

C:\Users\John\Documents\Sda\Code\web\Libraries\CssLibrary\menu-hamburger-off-canvas.xhtml

"Ideas tree menu". An infinitely hierarchical menu tree with expandable and collapsible submenus, all done on the client side (with javascript).

C:\Users\John\Documents\Sda\Code\web\Libraries\Html5Library\html5-polyglot-template-040-project\ideasTreeMenu

## Touch

Then test for touch.

## Lists

To markup lists generally do so with the list element marked up as an independent entity, e.g. outside of a paragraph.

```
<p>For instance, this fantastic sentence has bullets relating to</p>
<ul>
  <li>wizards,
  <li>faster-than-light travel, and
  <li>telepathy,
</ul>
<p>and is further discussed below.</p>
```

(W3C The Latest, "HTML 5.3, the Latest", <https://www.w3.org/TR/html53/>), <https://www.w3.org/TR/html53/grouping-content.html#the-p-element>

Optionally, if you want to treat the list as part of the preceding or subsequent text, e.g. for styling purposes, wrap it in a div ...

```
<div>
  <p>For instance, this fantastic sentence has bullets relating to</p>
  <ul>
    <li>wizards,</li>
    <li>faster-than-light travel, and</li>
    <li>telepathy,</li>
  </ul>
  <p>and is further discussed below.</p>
</div>
```

(W3C The Latest, "HTML 5.3, the Latest", <https://www.w3.org/TR/html53/>), <https://www.w3.org/TR/html53/grouping-content.html#the-p-element>

To collapse the distance between a list introduction and the list items do this (except do it in the style sheet):

```
<p style='margin-bottom: 0;'>To collapse the distance between a list introduction and the
list items do this:</p>
<ul style='margin-top: 0;'>
  <li>The first item</li>
  <li>The second item</li>
</ul>
```

C:\Data\Sda\Code\Web\Examples\XhtmlExamples\quickReferenceExamples.html

## Common Idioms without dedicated elements

### Overview

See the spec for:

- Subheadings, subtitles, alternative titles and taglines;
- Bread crumb navigation;
- Tag clouds;
- Conversations;
- Footnotes

(W3C The Latest, "HTML 5.3, the Latest", <https://www.w3.org/TR/html53/>), "4.13. Common idioms without dedicated elements", <https://www.w3.org/TR/html53/common-idioms-without-dedicated-elements.html#common-idioms-without-dedicated-elements>

### Logo

Logo with anchor, hover, and embedded svg

## SCALABLE VECTOR GRAPHICS



Lorem ipsum dolor sit, amet consectetur adipiscing el

```

<a id="logo" href="#">
  
  <p>
    My Website <br />
    Name
  </p>
</a>

<style>
/* <![CDATA[ */
#logo {
  width: min-content;
  display: flex;
  align-items: center;
}
#icon, #logo p {
  flex: 1 1 auto;
}
#icon {
  height: 100px;
}
#logo p {
  width: max-content;
  height: max-content;
  padding: 0 1em;
  font-weight: bold;
  font-size: 150%;
  color: black;
}

a#logo:link {
  text-decoration: none;
}
a#logo:hover {
  background-color: whitesmoke;
}
/* ]]> */
</style>

```

C:\Users\John\Documents\Sda\Code\web\Examples\SvgExamples\svg-logo-hover.xhtml

## Breadcrumbs

Breadcrumbs.

```

<header>
  <div id="skipper"><a href="#main-content">Skip to main content</a></div>
  <nav>
    <ul class="bread-crumbs">
      <li><a href="//www.loc.gov/">The Library of Congress</a> >></li>
      <li><a href="//www.loc.gov/standards/">Standards</a> >></li>
      <li><a href="//www.loc.gov/standards/datetime">Extended Date Time Home</a> >></li>
      <li><span class="selected">EDTF Spec</span></li>
    </ul>
  </nav>
</header>

ul.bread-crumbs {
  margin: 0 10;
  padding: 0;
  list-style: none;
}

```

```
ul.bread-crumbs > li {
  display: inline;
}
```

C:\Users\John\Documents\Sda\Info\EveryApplication\System\Conventions\Datetimes\Iso8601\LibraryOfCongress-Edtf\WebPurify\index.xhtml

C:\Users\John\Documents\Sda\Info\EveryApplication\System\Conventions\Datetimes\Iso8601\LibraryOfCongress-Edtf\WebPurify\datetime-base-pure.css

## Skip to main content

Skip to main content. Use an internal hyperlink with the text "Skip to main content".

```
<header>
<div id="skipper"><a href="#main-content">Skip to main content</a></div>
<nav>
  <ul class="bread-crumbs">
    <li><a href="//www.loc.gov/">The Library of Congress</a> >></li>
    <li><a href="//www.loc.gov/standards/">Standards</a> >></li>
    <li><a href="//www.loc.gov/standards/datetime">Extended Date Time Home</a> >></li>
    <li><span class="selected">EDTF Spec</span></li>
  </ul>
</nav>
...
<main>

  <article id="main-content">

    <header>

      <h1>Purified Extended Date/Time Format (EDTF) Specification</h1>
```

## Self anchored headings

Selt anchored headings.

```
<!-- HTML -->
<h2 id="meta"><a href="#meta"><span>#</span> Meta</a></h2>

<h3 id="meta-introduction"><a href="#meta-introduction"><span>#</span> Introduction</a></h3>

<p>The Extended Date/Time Format (EDTF) was create

<!-- CSS -->
/* Heading Anchors */
h2, h3 {
  position: relative;
  left: -0.9em;
  /* Shrink headings to fit contents so that the anchor isn't triggered if you scroll past a
  a heading but not directly over it */
  display: table;
}
h2 > a > span, h3 > a > span {
  visibility: hidden;
}
h2:hover, h3:hover {
  text-decoration: underline;
}
h2:hover > a, h3:hover > a {
  background-color: white;
}
h2:hover > a > span, h3:hover > a > span {
  visibility: visible;
}
h2, h2 > a:link, h2 > a:visited {
  color: #7b8cc6;
  text-decoration: none;
```

```

}
h3, h3 > a:link, h3 > a:visited {
  color: #817a41;
  text-decoration: none;
}

```

C:\Users\John\Documents\Sda\Info\EveryApplication\System\Conventions\Datetimes\Iso8601\LibraryOfCongress-Edtf\SourceCodePurified\index.xhtml  
 datetime-base-pure.csss

## Animations

(Bentley 2019, "ScalableVectorGraphics-Svg-Reference.Docx",  
 \\Atlas\Users\John\Documents\Sda\Info\Web\KB\Xhtml\Reference\ScalableVectorGraphics-Svg-Reference.docx)  
<\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Xhtml\Reference\ScalableVectorGraphics-Svg-Reference.docx>

## CSS image replacement

Replace an img's src via CSS.

```

' Xhtml


' Css
img#aboutPortrait {
  --new-image-width: 125px;
  background-size: cover;
  box-sizing: border-box;
  width: var(--new-image-width);
  height: calc(var(--new-image-width)/0.74); /* Aspect ratio for image */
  padding-left: var(--new-image-width);
  background-image: url(images/paul-bentley--black-and-white.png);
}

```

Derived from (Sheilds 2013, "Replace the Image in an <img> with CSS", <https://css-tricks.com/replace-the-image-in-an-img-with-css/>)

## Browser Hacks

Microsoft IE unsupported conditional warning.

```

<!--[if lte IE 6]>
  <div class='warning'>
    <p><strong>You are viewing this site with Internet Explorer 6.0 or lower.</strong></p>

    <p>Our website does not support your Internet Browser. Consider upgrading to a modern Internet Browser. Some free modern browsers include:</p>
    <ul>
      <li><a href='http://www.mozilla-europe.org/en/firefox/'>Mozilla Firefox</a></li>
      <li><a href='http://www.google.com/chrome'>Google Chrome</a></li>
    </ul>
  </div>
<![endif]-->

```

---

## References (Zotero)

---

- Apache Software Foundation. 2019. "Apache HTTP Server Version 2.4 Documentation".  
<http://httpd.apache.org/docs/current/>. 2019.
- — —. n.d. "Bugzilla". Accessed 2019-02-15. <https://bz.apache.org/bugzilla/>. n.d.
- Bentley, John. 2018. "Cascading Style Sheets - Css -Reference".  
\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Css\Reference\CascadingStyleSheets-Css-Reference.docx. 2018.
- — —. 2019. "ScalableVectorGraphics-Svg-Reference.Docx".  
\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Xhtml\Reference\ScalableVectorGraphics-Svg-Reference.docx. 2019.
- — —. 2019. "Xhtml-PolyglotMarkupRules.Docx".  
\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Xhtml\Reference\Xhtml-PolyglotMarkupRules.docx. 2019.
- — —. 2019. "Xhtml-Reference.Docx".  
\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Xhtml\Reference\Xhtml-Reference.docx. 2019.
- — —. n.d. "Experimentally Verified". n.d.
- Ellingwood, Justin. 2015. "Web Caching Basics: Terminology, HTTP Headers, and Caching Strategies". DigitalOcean. <https://www.digitalocean.com/community/tutorials/web-caching-basics-terminology-http-headers-and-caching-strategies>. 2015-04-01.
- Google. 2018. "Be Mobile Friendly > Welcome". Google Developers.  
<https://developers.google.com/search/mobile-sites/>. 2018-07-16.
- — —. 2019. "Getting the Most from Your Google Search Appliance: Crawling and Indexing".  
<https://support.google.com/gsa/answer/6280131>. 2019.
- — —. 2020. "Administering Crawl: Preparing for a Crawl, Using Robots.Txt to Control Access to a Content Server". <https://support.google.com/gsa/answer/6329153#92549>. 2020.
- Grigorik, Ilya. 2019. "HTTP Caching | Web Fundamentals". Google Developers.  
<https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/http-caching>. 2019-02-12.
- Illyes, Gary, Henner Zeller, Lizzi Harvey, and Martijn Koster. 2020. "Robots Exclusion Protocol".  
<https://www.ietf.org/id/draft-koster-rep-01.txt>. 2020-01-08.
- Internet Engineering Task Force (IETF), Roy Fielding, and Julian Reschke. 2014. "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests". <https://tools.ietf.org/html/rfc7232>. 2014-06.
- — —. 2014. "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content".  
<https://tools.ietf.org/html/rfc7231>. 2014-06.

- Internet Engineering Task Force (IETF), Yves Lafon, Roy Fielding, and Julian Reschke. 2014. "Hypertext Transfer Protocol (HTTP/1.1): Range Requests". <https://tools.ietf.org/html/rfc7233>. 2014-06.
- Internet Engineering Task Force (IETF), Mark Nottingham, Roy Fielding, and Julian Reschke. 2014. "Hypertext Transfer Protocol (HTTP/1.1): Caching". <https://tools.ietf.org/html/rfc7234>. 2014-06.
- LePage, Pete. 2019. "Responsive Web Design Basics | Web Fundamentals". Google Developers. <https://developers.google.com/web/fundamentals/design-and-ux/responsive/>. 2019-02-14.
- Mulhuijzen, Rogier. 2014. "Best Practices for Using the Vary Header". <https://www.fastly.com/blog/best-practices-using-vary-header>. 2014-08-18.
- Sheilds, Marcel. 2013. "Replace the Image in an <img> with CSS". CSS-Tricks. <https://css-tricks.com/replace-the-image-in-an-img-with-css/>. 2013-05-20.
- Stack Exchange Inc. n.d. "Stack Overflow". Stack Overflow. Accessed 2019-02-15. <https://stackoverflow.com/>. n.d.
- W3C. The Latest. "HTML 5.3, the Latest". <https://www.w3.org/TR/html53/>. The Latest.



---

## Document Licence

---

[Web Techniques Reference](#) © 2021 by [John Bentley](#) is licensed under [Attribution-NonCommercial-ShareAlike 4.0 International](#)

