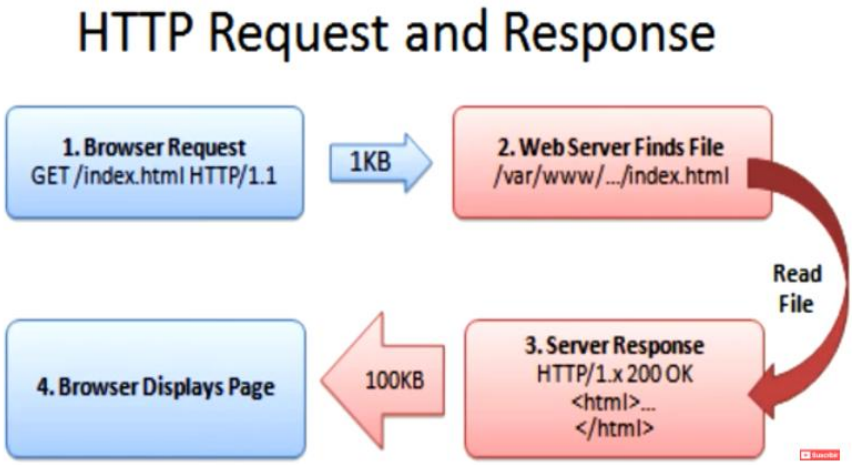
PHP Headers

What are PHP headers?

- The header() is a pre-defined function.
- The header() function sends raw HTTP header to the client before sending any other output.
- HTTP header manipulates the information sent by the web-server to the client.
- Note: header() function is that it must be called before sending any actual output.



- **Syntax:** void header ($header, $replace,$http_response_code)

1) $header (string): This function has two types of special header calls.

   The first header starts with an **"HTTP/"** string, which locates the HTTP status code for sending.

   The second special case of header starts with **"Location:"** header. It does not only sends back the header to the browser, but it also provides a REDIRECT (302) status code to the browser, until the 201 or 3xx status code has already been set.

2) $replace (Boolean) (optional)

   This parameter is used for specifying whether a previous same header should be replaced by the header or add another header of same type. The $replace is a boolean type optional parameter.

   The default value is TRUE, which means it replaces the previous same header. But you can bind several headers of same type if FALSE is passed as second argument.

3) http_response_code (int) (optional)

   The $http_response_code is an **optional** parameter, which forces the HTTP response code to a specified value.

Changes:

- After PHP version 5.4.x, this function stops sending more than one header to prevent the header injection attacks. It allows only one header at a time.

Application:

- It changes the page location.
- It sets the time zone.
- It sends the HTTP status.
- This function sets the caching control.
- It initiates the force download.
- It refreshes the page.

1) header('location:')

- Redirecting a user to another page

Example 1: Redirecting page

```
<?php
header('Location: URL/');
exit;
?>
```

Note: If any line of code is written after the header(), it will not execute.

2) header('Refresh:')

- Redirecting a user to another page can be confusing for them if there is no warning at all.
- Using a header refresh you can give some time for the user to see a message before it sends them off to another page.

Example 2: Redirection interval (redirect user to another page after 10 seconds)

```
<?php
    // This will redirect after 10 seconds
    header('Refresh: 10; url = http://www.google.com/');
    echo 'You will be redirected to Google in 10 seconds.';
    exit;
?>
```

The redirect above will refresh the page in ten seconds and when it refreshes, it will send the user to Google. It prints out a message to the user that will be redirected.

3) header('Cache-Control:')

- When web pages are viewed, browser might be storing them in cache somewhere to reference later. When you come back to the site, it might load faster because it is just being loaded right from your computer and not from a server like it was the first time. This is great for the most part. It cuts down on the amount of data being sent and reduces the time it takes to load the page.
- The problem is that if you have a site that constantly udpates, like a news site, you don't want it to be cached. Otherwise the page your users will keep loading will have all the old content on it and no recent news.
- To make sure your page doesn't get cached, you use the Cache-Control header. With this you can tell the browser that it should never cache your site. You specify no-cahce, and then tell the browser to revalidate the page with the original server.

4) header('Expires:')

- The Expires header is used along with the Cache-Control header.
- With Expires, We can set the date for when the pages cache is to expire.
- To make sure that the page is never cached, set past date so that will always expire and must reload with new content.

5) header('Pragma:')

- Pragma headers allow a browser to send information to a server.
- Assigning a 'Pragma: no-cache' to their headers will make their page uncacheable.
- Note: This is an older method of controlling caching in web development.

Example 3: Don't cache pages (prevent the browser to cache pages)

```
<?php

    // Set a past date

  header("Expires: Tue, 03 March 2022 04:00:00 GMT");

  header("Cache-Control: no-cache");

  header("Pragma: no-cache");

?>
<html>
  <body>
   <?php
     print_r(headers_list());
  ?>
  </body>
</html>
```

6) header('Content-Type:')

- The PHP header Content-Type should look familiar to you. You see the meta tag in HTML document that tells the browser what type of document it is and what to expect.

  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

- With a PHP Content-Type header, you can change how you want the browser to read the page. For normal HTML pages, the Content-Type will be the text/html. But, you could change that in the header to be text/plain and the browser will display the source code of your site.

  Example:

  CSS: header('Content-Type: text/css');

  Javascript: header('Content-Type: text/javascript');

  JPEG Image: header('Content-Type: image/jpeg');

  JSON: header('Content-Type: application/json');

  PDF: header('Content-Type: application/pdf');

  RSS: header('Content-Type: application/rss+xml; charset=ISO-8859-1');

  Unknown binary files: header('Content-Type: application/octet-stream');

  Text (Plain): header('Content-Type: text/plain');

  XML: header('Content-Type: text/xml');

7) header('Content-Disposition:')

- Content-Type tell the browser what type of document to expect.
- Content-Disposition tells the browser how to handle the document.
- If you have a PDF that you want users to download, you can use this Content-Disposition to make the browser display a save dialog. You set the Content-Disposition to attachment and then you must put the filename so that it can be downloaded.

  header("Content-disposition: attachment; filename=\"" . $file_url . "\"");

  Example:

  <?php

  header("Content-type:application/pdf");

  header("Content-Disposition:attachment;filename='downloaded.pdf'");

  readfile("original.pdf");

  ?>

8) Example: Set HTTP Status in the header response.

header("HTTP/1.0 404 Not Found");