Web Browsers Internals

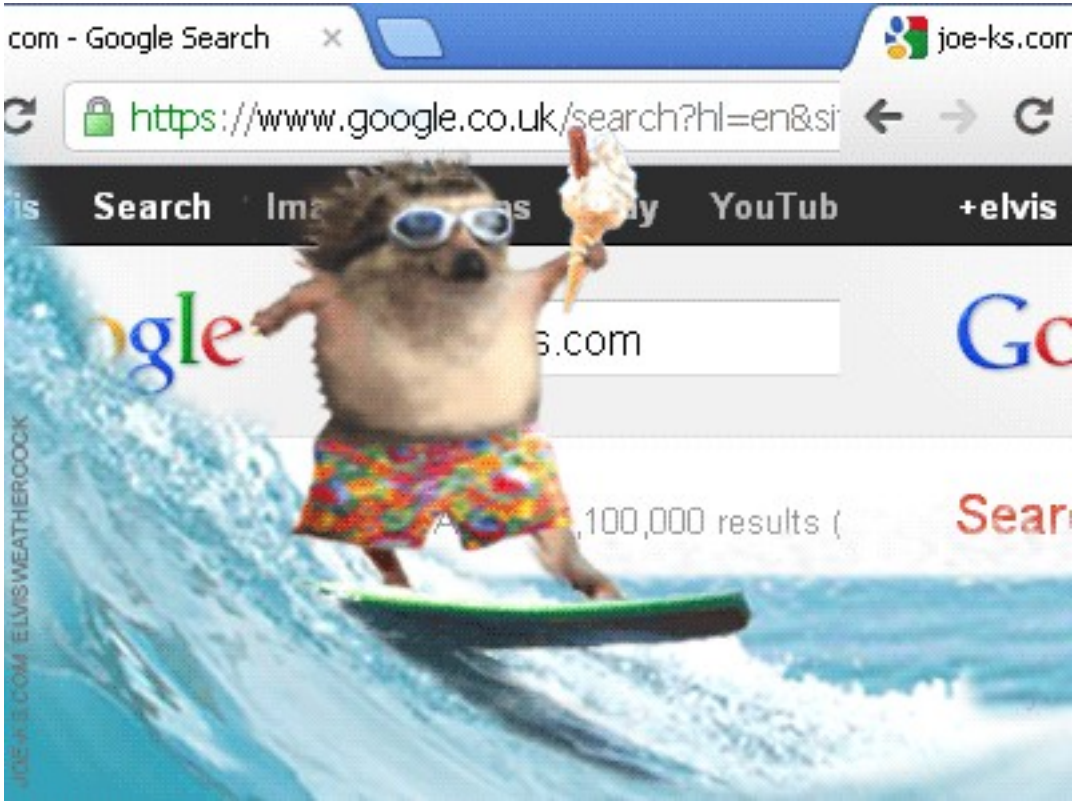
Under the Cover

- Arun Singh | twitter: @aruns89

What happens?
1 Minute rundown

- Parsing the URL
- GET’ting the page
- Browser re-entry
- Parsing HTML
- DOM
- Rendering the page

Surfing www.google.com



Parsing the URL

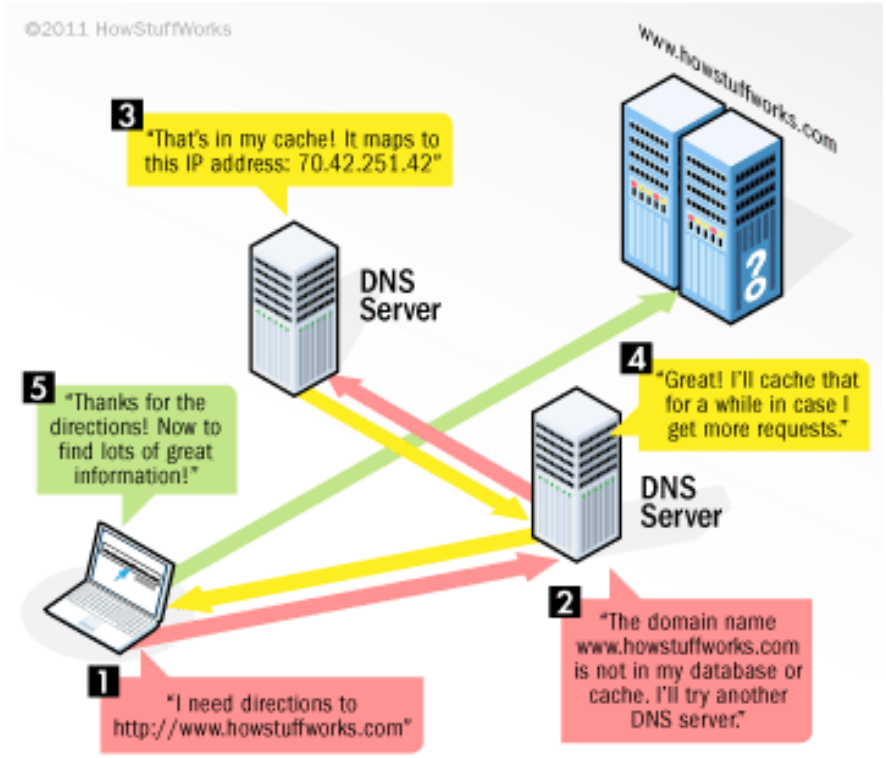
http://www.google.com

- Retrieve
  - **Protocol:** http or https
  - **Resource:** "/": Retrieve main (index) page

- Check url with HSTS (HTTP Strict transport security) list

- DNS mapping

DNS lookup



Domain Name System

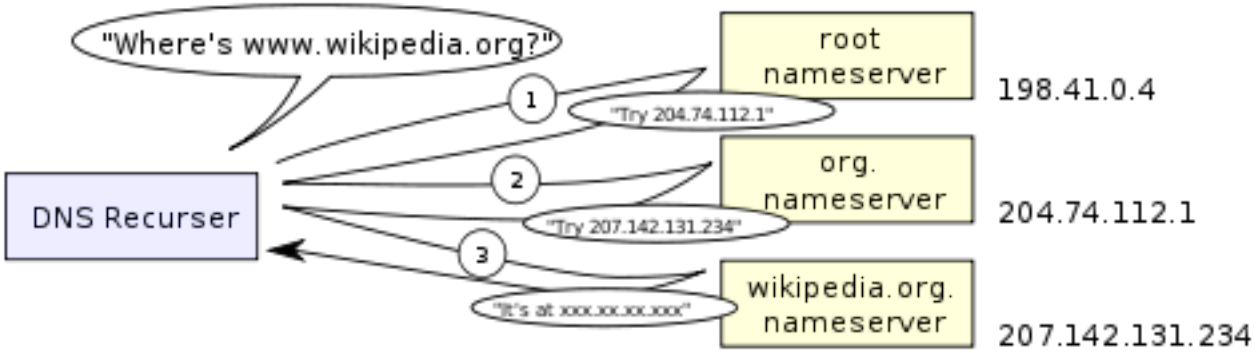
Sequence of checks

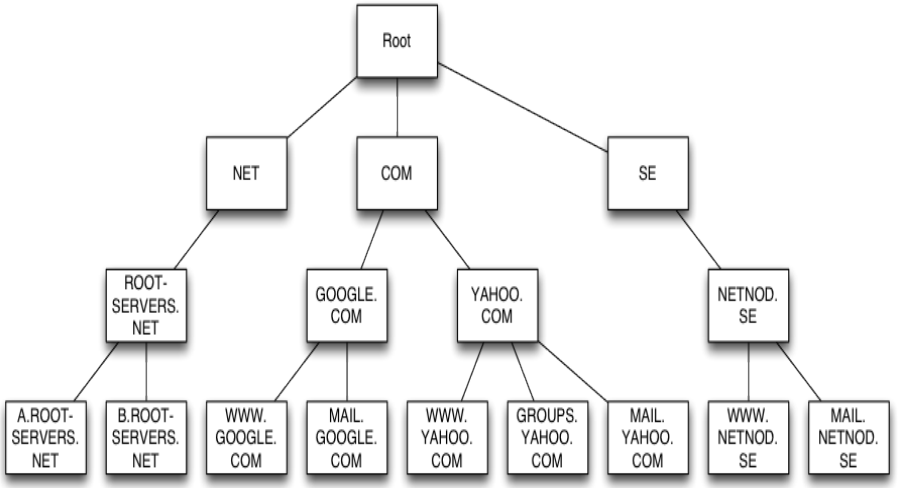
- Browser Cache
- OS Cache
- Router Cache
- ISP DNS cache

DNS Lookup

Finding the IP

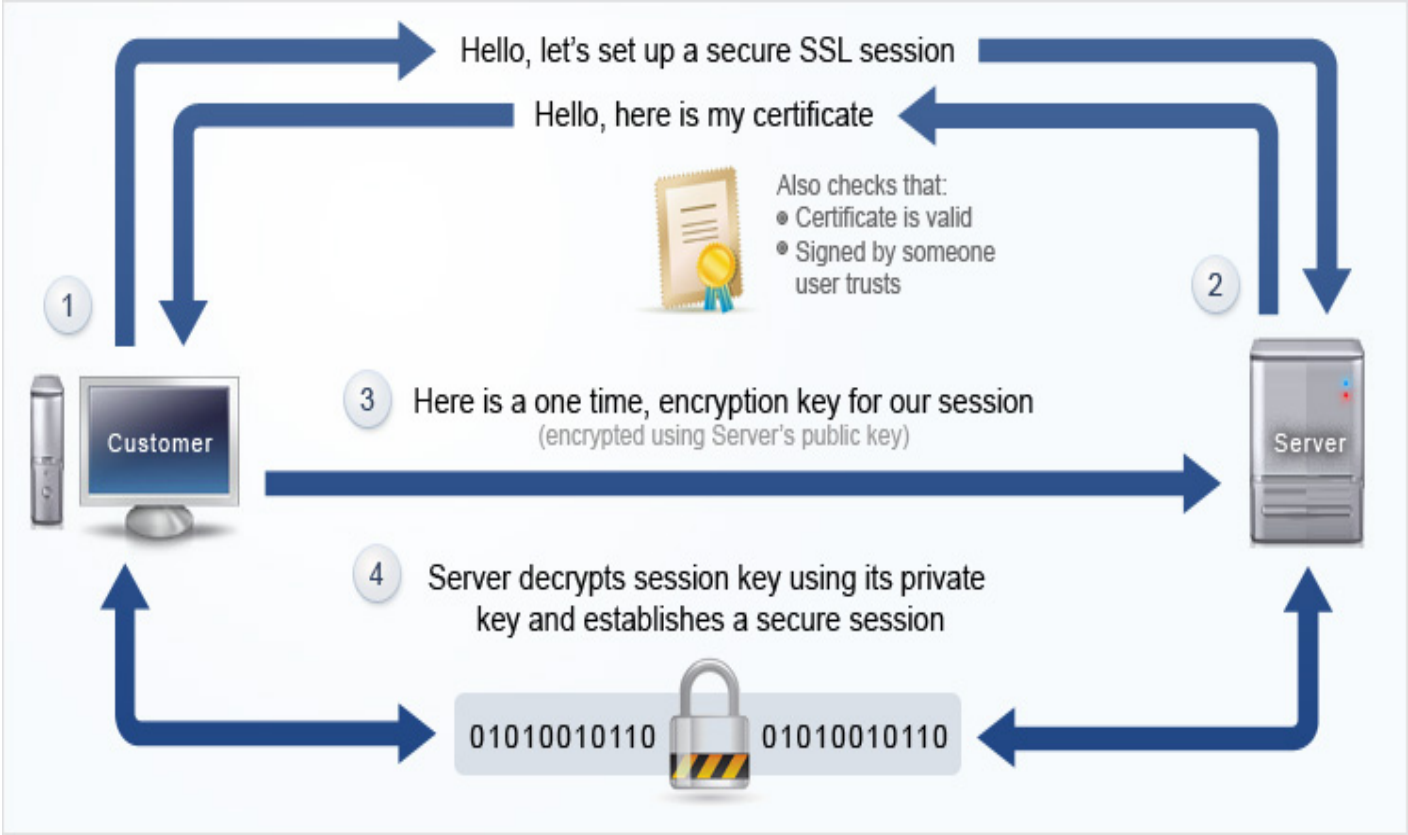
DNS hierarchy





The handshake…

The handshake



GET’ting the page

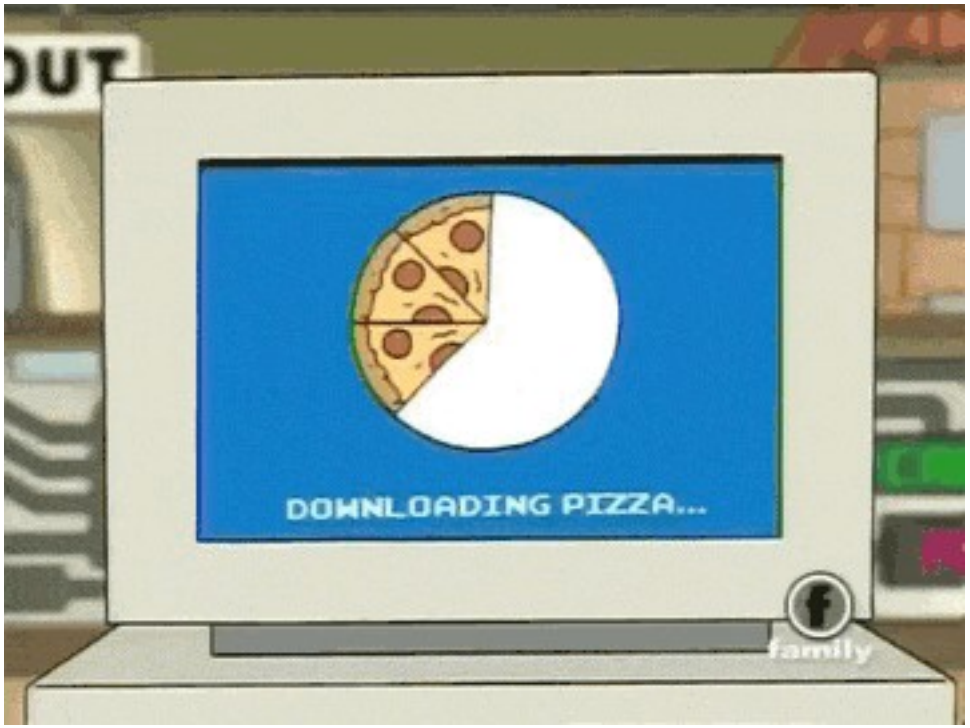
```
GET http://www.google.com/ HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, [...]
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; [...]
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
Host: google.com
Cookie: datr=1265876274-[...]; locale=en_US; lsd=WW[...]; c_user=2101[...]
```

- Identify the browser
- Accepted response
- Supported encoding
- Cookie info

Server response handler

- HTTP Daemon server (Apache or nginx)
- Breaks down the request headers (methods/path)
- Pull content based on requested path (/)
- The server parses the file according to the request handler.
  - Eg. PHP, ASP.NET etc
- In the case of PHP interpret the index file and stream the output

Downloading page……



Server Response

```
HTTP/1.1 200 OK
Cache-Control: private, no-store, no-cache, must-revalidate, post-check=0,
    pre-check=0
Expires: Sat, 01 Jan 2000 00:00:00 GMT
P3P: CP="DSP LAW"
Pragma: no-cache
Content-Encoding: gzip
Content-Type: text/html; charset=utf-8
X-Cnection: close
Transfer-Encoding: chunked
Date: Fri, 12 Feb 2010 09:05:55 GMT

2b3
[illegible]T[illegible]n[illegible]@[illegible][...]
```

Decompressed gzip

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
      lang="en" id="google" class=" no_js">
<head>
<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
<meta http-equiv="Content-language" content="en" />
...
```

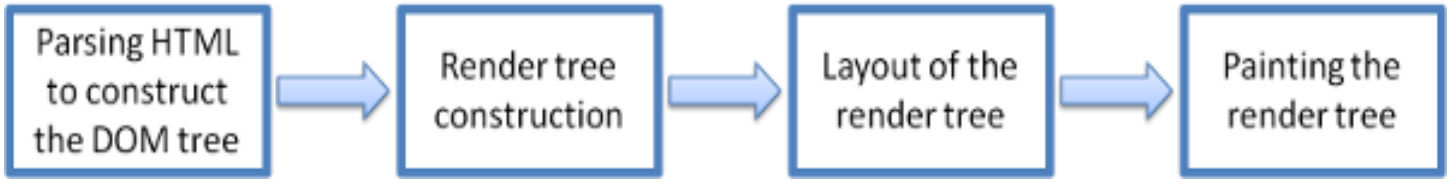
Browser Re-entry

Tasks

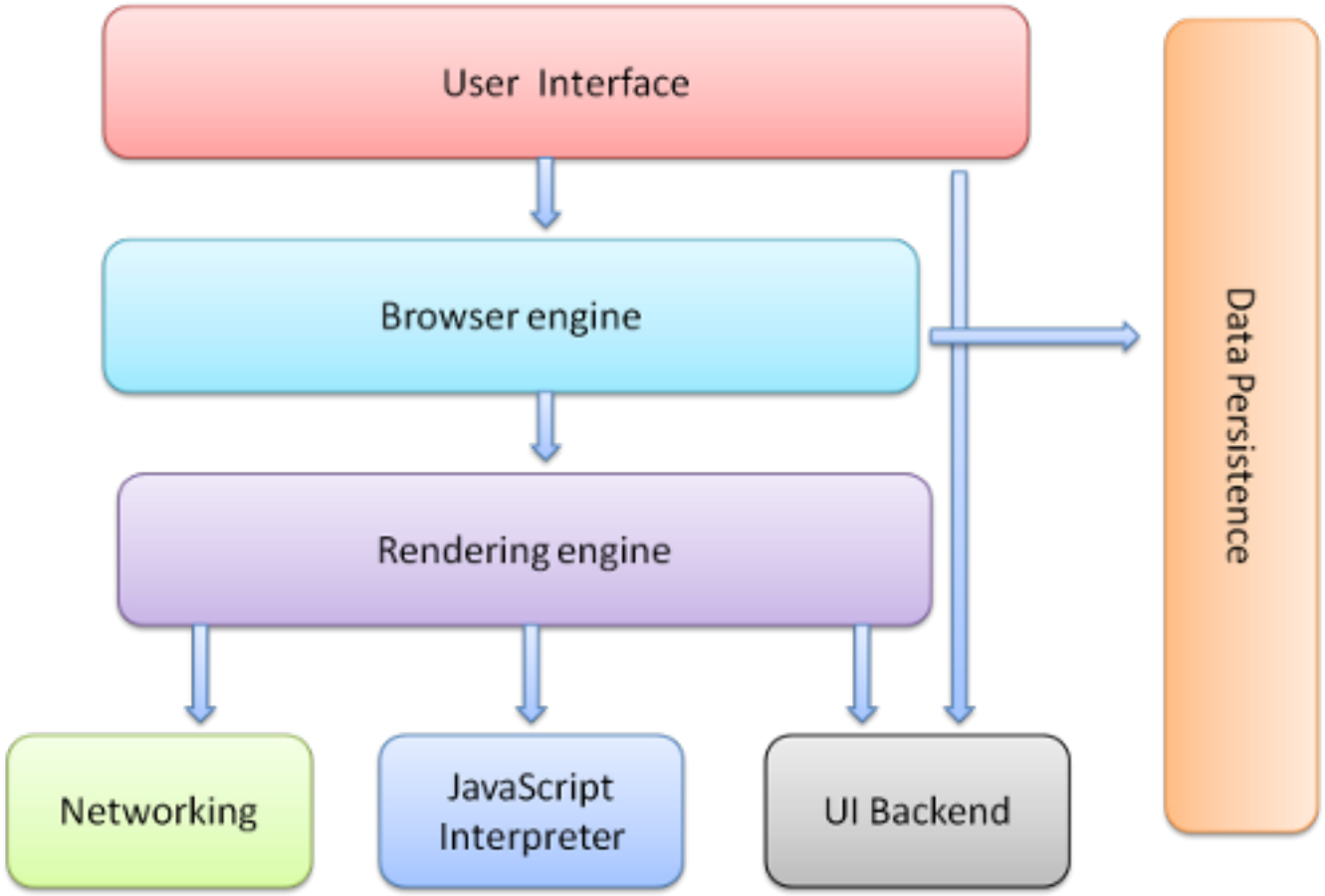
•Parsing

- HTML, CSS, JS

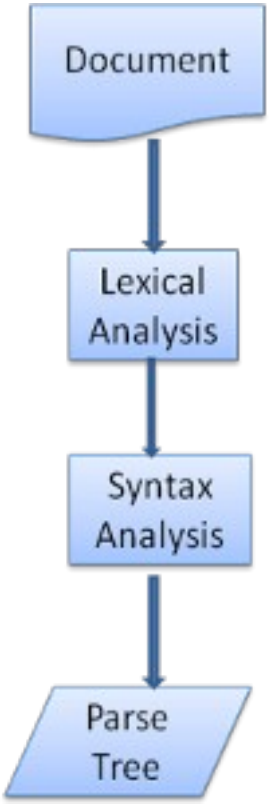
•Rendering

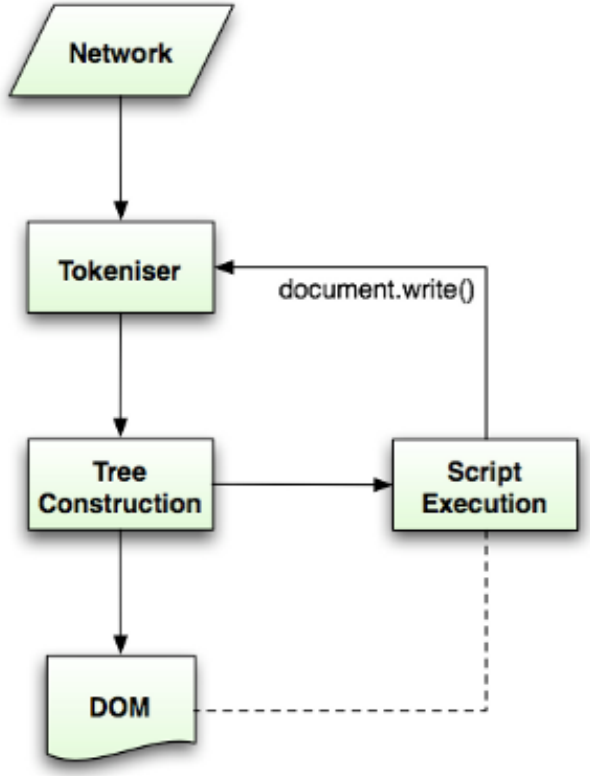


Browser: High Level



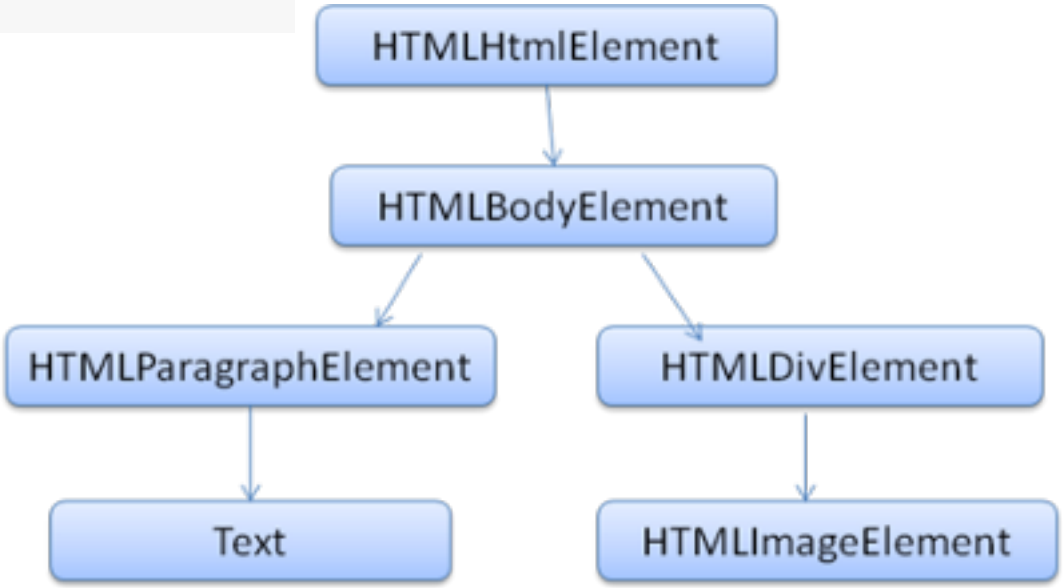
HTML Parsing





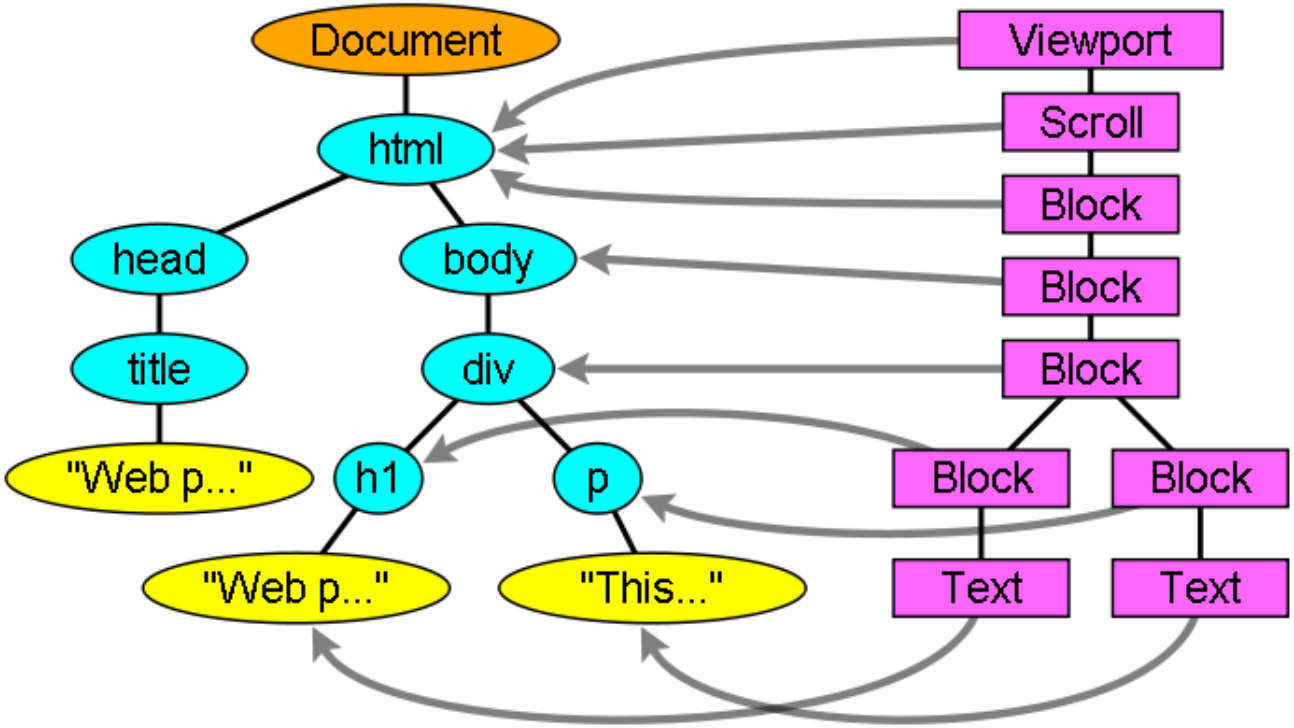
DOM tree

```
<html>
  <body>
    <p>
      Hello World
    </p>
    <div> <img src="example.png"/></div>
  </body>
</html>
```



Render tree

Visual representation of DOM
(Boxes with geometry)



Layout

CSS-OM + Layout

Layout

- Maps elements to co-ordinates in the viewport
- When the renderer is created and added to the tree, it does not have a position and size. Calculating these values is called layout or reflow.

- In order not to do a full layout for every small change, browsers use a "dirty bit" system. A renderer that is changed or added marks itself and its children as "dirty": needing layout.

Painting

Painting

- The render tree is traversed and the renderer's "paint()" method is called to display content on the screen. **Painting uses the UI infrastructure component.**

- Like layout, painting can also be global–the entire tree is painted–or incremental.

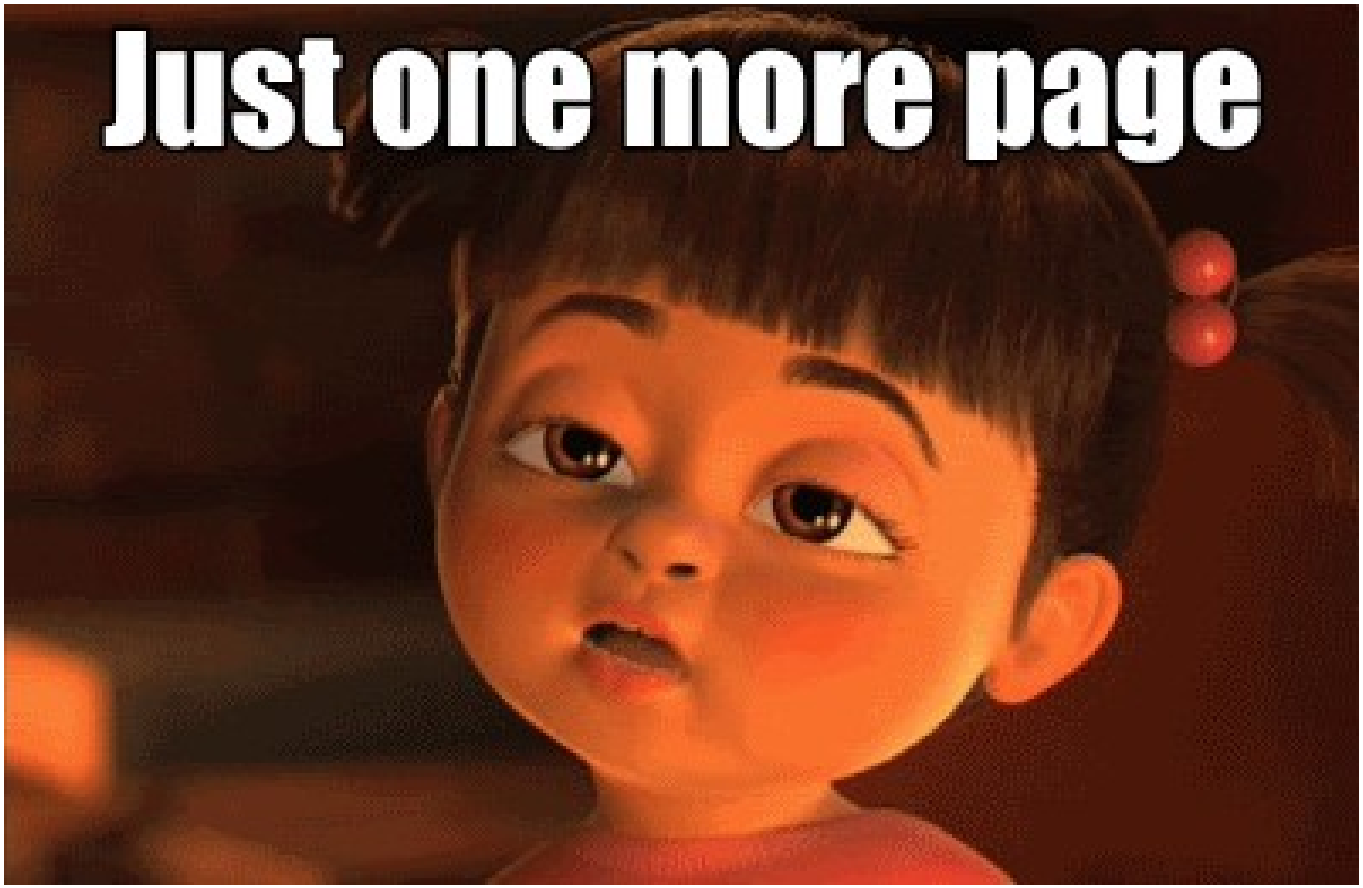
- Before repainting, WebKit saves the old rectangle as a **bitmap**. It then paints only the **delta** between the new and old rectangles.

- Color changes: only repaint
- Element position changes: Layout and repaint of the el + children

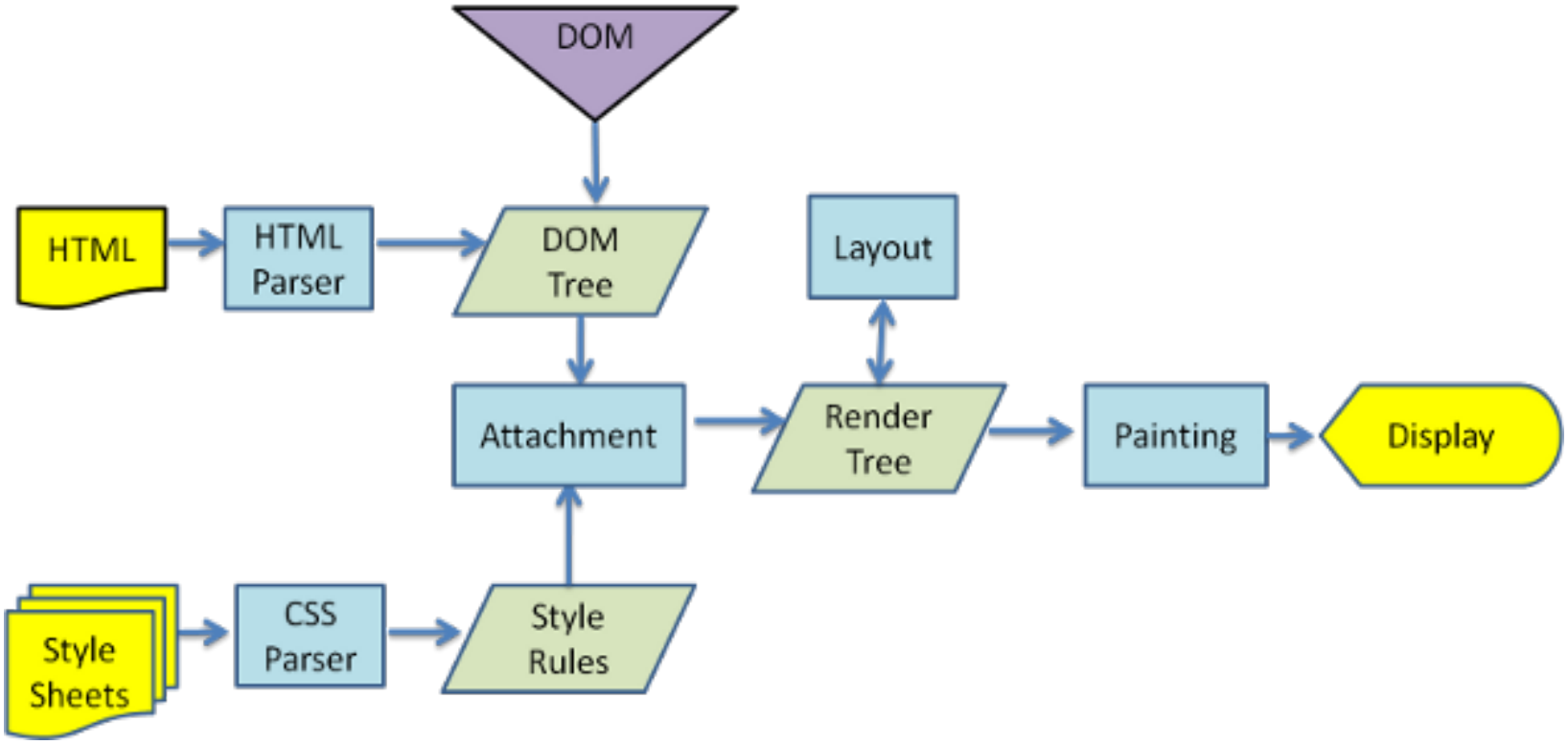
- Major changes, like increasing font size of the "html" element, will cause invalidation of caches, relayout and repaint of the entire tree.

Smooth rendering

60fps = ~~16ms~~ 10-12ms


Just one more page

Browser Rendering recap

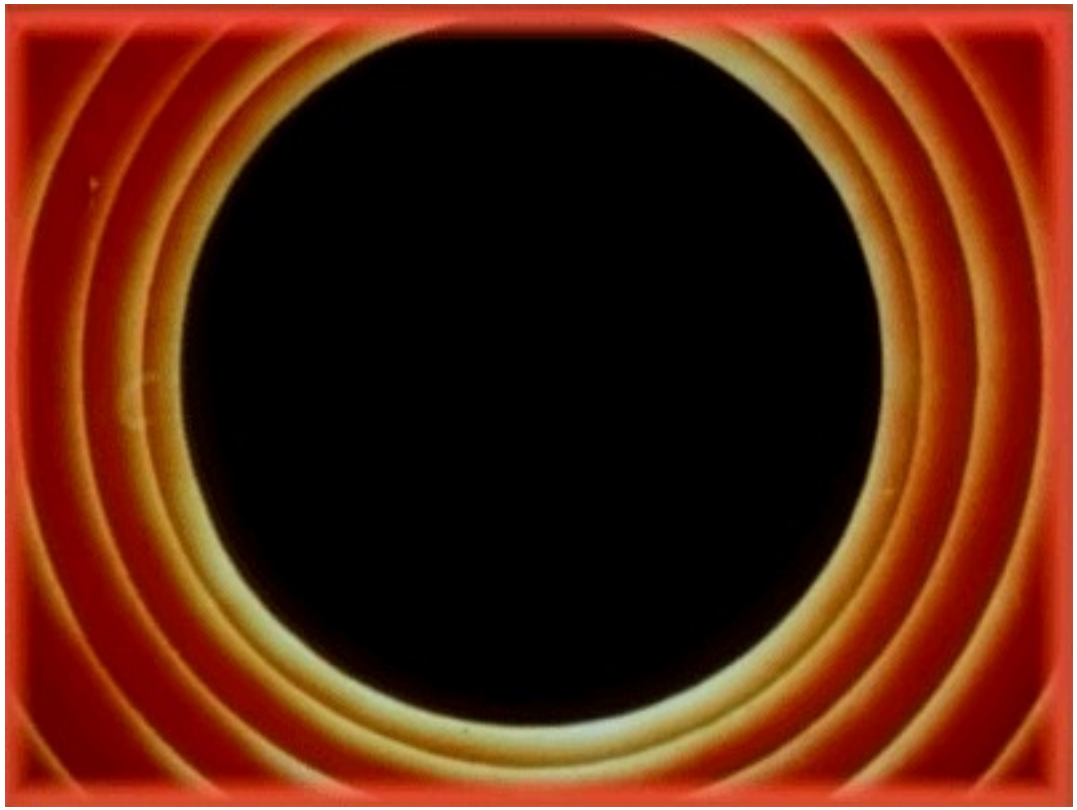


References

- What really happens when you navigate to a URL
  http://igoro.com/archive/what-really-happens-when-you-navigate-to-a-url
  /

- How Browsers Work: Behind the scenes of modern web browsers
- http://www.html5rocks.com/en/tutorials/internals/howbrowserswork/

- What happens when
  https://github.com/alex/what-happens-when

- **Github public Repo of my detailed learnings on this topic**

  - **How Web works?**
  - **https://github.com/arunsingh/web_underhood**