WHAT IS THE INTERNET

Internet is a computers that can communicate

Each computer has an address - like mail or deliveries

- local traffic goes to a local hub
- local hub sends stuff for outside to a higher hub
- until a hub has the destination "inside"
- sends to the right lower hub

There is a lot of redundancy, not just one path

Internet was a US Defense Dept exercise, to survive interruptions. (Like nukes)

INTERNET DELIVERY

Internet Protocol (IP) addresses are not for humans

Humans use names like `www.google.com`

Each "dot" separates a level in reverse

- ".com" knows all the domains inside it
- "google.com" is knows all the domains inside it

Three is normal, but more depth is supported.

Domain Name System (DNS) is how to turn a name into an IP address
https://www.ultratools.com/tools/dnsLookupResult

TRAFFIC PROTOCOLS

Looking up DNS entries is one kind of traffic over the Internet

There are many others. Examples:

- Email
- Database
- Many online games
- Web

INTERNET IS DOWN!

When someone says "The internet is down" they are wrong

- Perhaps their local internet routing is down
- Perhaps some local web issue is down
- Perhaps someone providing a lot of infrastructure (example: Amazon) is broken, taking a lot of web sites down

But it's very rare for any notable portion of the Internet itself to be down.

DON'T BE THAT PERSON

But don't tell be that person

The one that tells them they are wrong

We all know what they mean

WHAT IS A SERVER?

This is really hard. **https://jvns.ca/blog/2019/12/26/whats-a-server/**

- A server can be a program that responds to requests
- A server can be the machine that runs that program
- A server can be a virtual machine running on a physical machine

A server can run a server running a server

Generally, for this class a server is the program

WEB REQUEST RESPONSE

For the Web

- A client (usually a browser) makes a **request**
- A server gets the request and gives a **response**

Each request gets one response.

Only with special preparations and moving outside basic web can you get anything else.

BAD CASES ON THE WEB

- A stock-ticker app that is told when stocks change
- A weather app that is told when weather changes

Both of these worked very poorly on the early web.

You couldn't be told of changes, so you had to ask repeatedly and frequently.

That is a lot of pointless traffic.

WEB WAS FOR LINKING SCIENTIFIC PAPERS

- Text
- linking back and forth
- readable on different platforms

Not WYSIWYG

- H - Hyper
- T - Text
- T - Transfer
- P - Protocol

WEB PROVIDED UNIQUE BENEFITS

- Common port (80 for HTTP, 443 for HTTPS)
  - Meant once you got through a firewall, you had access to everything
- Not tied to a particular appearance
- Tolerant of bugs/typos
- Human readable
- Searchable

WEB WAS SEARCHABLE

- A program (crawler, spider, bot) would read a page
- Would get a list of all the links on that page
  - Add anything new to list of pages to crawl
- Would read the text of the page and save info (index)
- Would repeat with next link on list

Users go to site with index, enter search terms

- would return list of matching links

WEB IS STATELESS

Each request is considered without respect to previous requests

Can go straight to any link without passing through others

WHAT ABOUT REQUIRING LOGIN?

Isn't login not-stateless?

Yes and no. The *protocol* doesn't enforce that.

Request comes in and is handed to whatever handler manages the details for that request.

That handler can decide to send you elsewhere (redirect) or provide alternate output (login screen)

Even with this requirement, if the request had all the necessary parts, it would move forward, even if those parts HADN'T come from previous steps

BROWSER RENDERING

Not every web client is a browser, (a spider is a web client and not a browser)

Browsers, after they get the content, will decide what to do with it

Often this means rendering an HTML page

But it could be displaying an image, playing a sound file, showing a PDF, saving a file

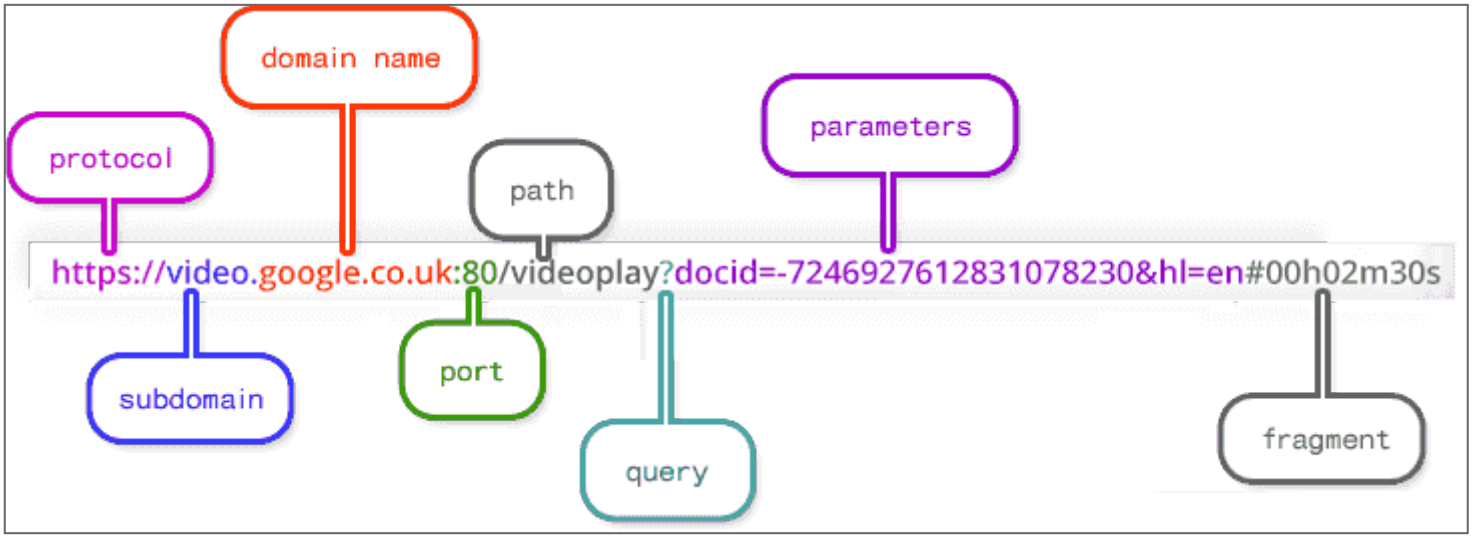
WHAT IS A URL?

A Uniform Resource Locator (URL) tells you what something on the internet is (identifies it) and how to find it (locates it).

http://catalog.northeastern.edu/graduate/engineering/multidisystems-msis/

Often called a `Web Address`, though they cover more than the web

PARTS OF A URL



From **https://doepud.co.uk/blog/anatomy-of-a-url**

FULLY QUALIFIED

A URL with all the parts is known as "Fully-Qualified"

Without all the parts, it might just be a path

That path might be "absolute" or "relative"

ABSOLUTE PATH

Absolute Paths are on the same server, but a different path.

- This involves a different query (params and hash/fragment)

An absolute path is taken from some "root" of the server.

This is NOT the "root" of the file system. The "document root" is how the web server treats requests for the "root".

```
<img src="/images/complex_url.png">
```

RELATIVE PATH

A Relative Path is based on navigation from the path of the currently loaded page.

- `<img src="cat.png"/>`
- `<img src="images/cat.png"/>`
- `<img src="../images/cat.png">`

PATHS

Consider the absolute paths that would match these if you were on `http://example.com/foo/` and `http://example.com/bar/images`

- `<img src="cat.png"/>`
- `<img src="images/cat.png"/>`
- `<img src="/images/cat.png"/>`
- `<img src="../images/cat.png"/>`
- `<img src="images/../cat.png"/>`

SUMMARY - PART 1

- Internet vs Web
- Internet routing
- DNS/Domain names/subdomains
- Web is request/response
- Web is stateless
- Searching isn't built in
- Searching is easy because stateless

SUMMARY - PART 2

- URLs can be fully qualified or not
- A path can be absolute or relative
- Browsers render a page after getting the data
- Not all clients are browsers
- Not all data is rendered