

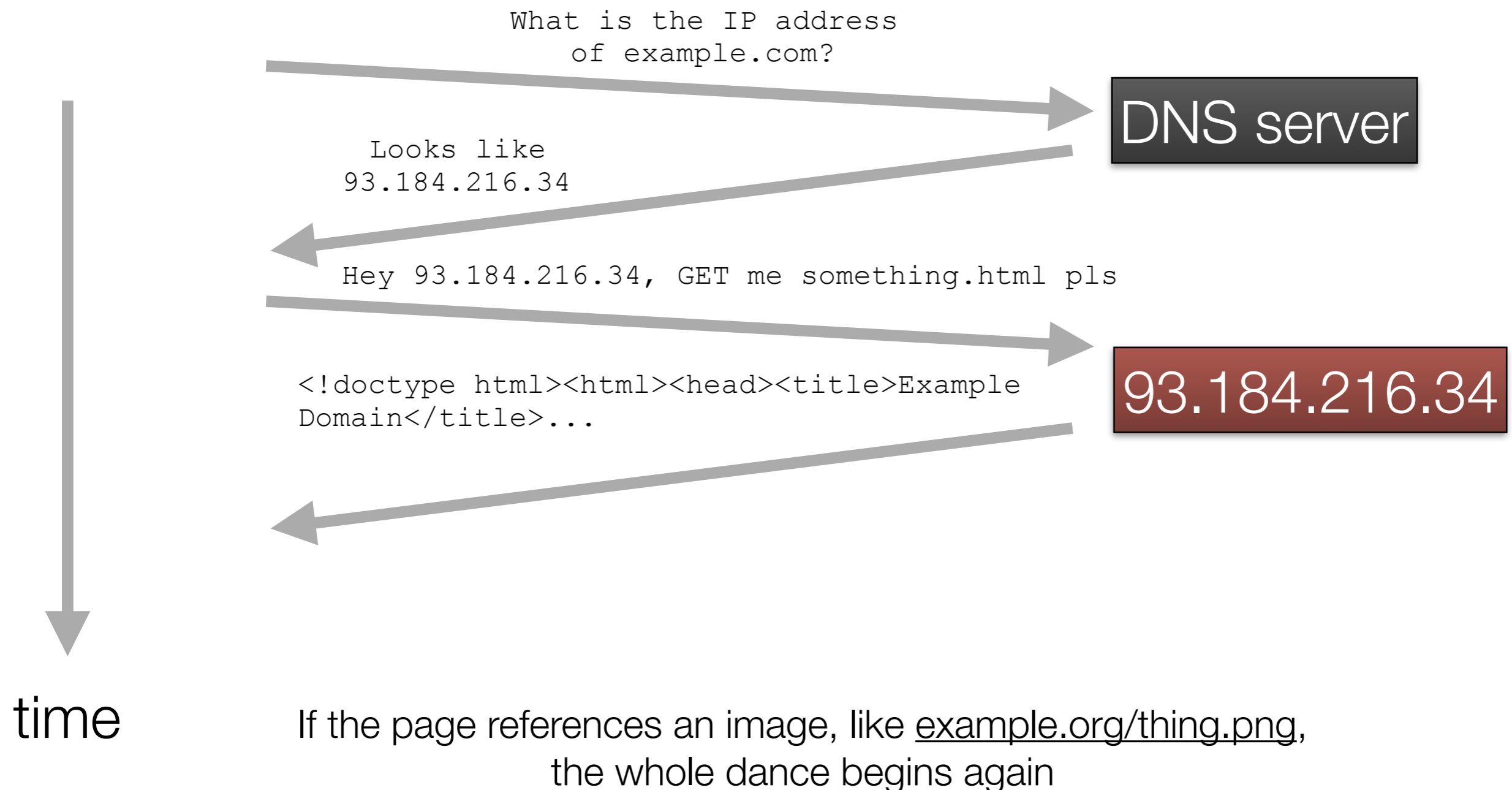
Achieving maximum website

What happens when you visit a webpage?

- Domain name lookup
- Browser contacts server and requests page by URL
- Browser finds resources (images, stylesheets) referenced by URL in the page and requests those
 - This can happen recursively, e.g. for a page that references a stylesheet that references an image

What happens when you visit a webpage?

"I want to see example.com/something.html"



Domain names

- The only prerequisite you cannot get entirely for free
 - (Unless you just put your site on Lavinia, which already has a hostname: `lavinia.as.arizona.edu`)
- Check a site like domainr.com for availability
- Registration involves specifying a computer by IP address as well as human contact information (sometimes called "whois" info)
 - Try `whois arizona.edu` in your terminal

URLs

- The thing that separates the Web from something like FTP is the concept of the *URL* (Uniform Resource Locator)
- Once you have a standardized way to refer to a resource on another computer, you can build webs of links between resources (hypertext)

URLs

`http://user:password@www.example.com:8000/path/to/resource?name=Shirley&gender=not%20applicable#about-our-site`

URLs

`http://user:password@www.example.com:8000/path/to/resource?name=Shirley&gender=not%20applicable#about-our-site`

1. `http://` - transport
2. `user:password@` - credentials
3. `www.example.com` - hostname
4. `:8000` - port number
5. `/path/to/resource` - path to resource
6. `?name=Shirley&gender=not%20applicable` - query string
7. `#about-our-site` - fragment identifier

URLs

`http://user:password@www.example.com:8000/path/to/resource?name=Shirley&gender=not%20applicable#about-our-site`

Some of these are optional:

1. `http://` - transport
2. ~~`user:password@` - credentials~~
3. `www.example.com` - hostname
4. ~~`:8000` - port number~~
5. `/path/to/resource` - path to resource
6. ~~`?name=Shirley&gender=not%20applicable` - query string~~
7. ~~`#about-our-site` - fragment identifier~~

URLs

`http://user:password@www.example.com:8000/path/to/resource?name=Shirley&gender=not%20applicable#about-our-site`

When linking *within* a website, more can be omitted:

1. ~~http://~~ - transport
2. ~~user:password@~~ - credentials
3. ~~www.example.com~~ - hostname
4. ~~:8000~~ - port number
5. **/path/to/resource** - path to resource
6. ~~?name=Shirley&gender=not%20applicable~~ - query string
7. ~~#about-our-site~~ - fragment identifier

HTML

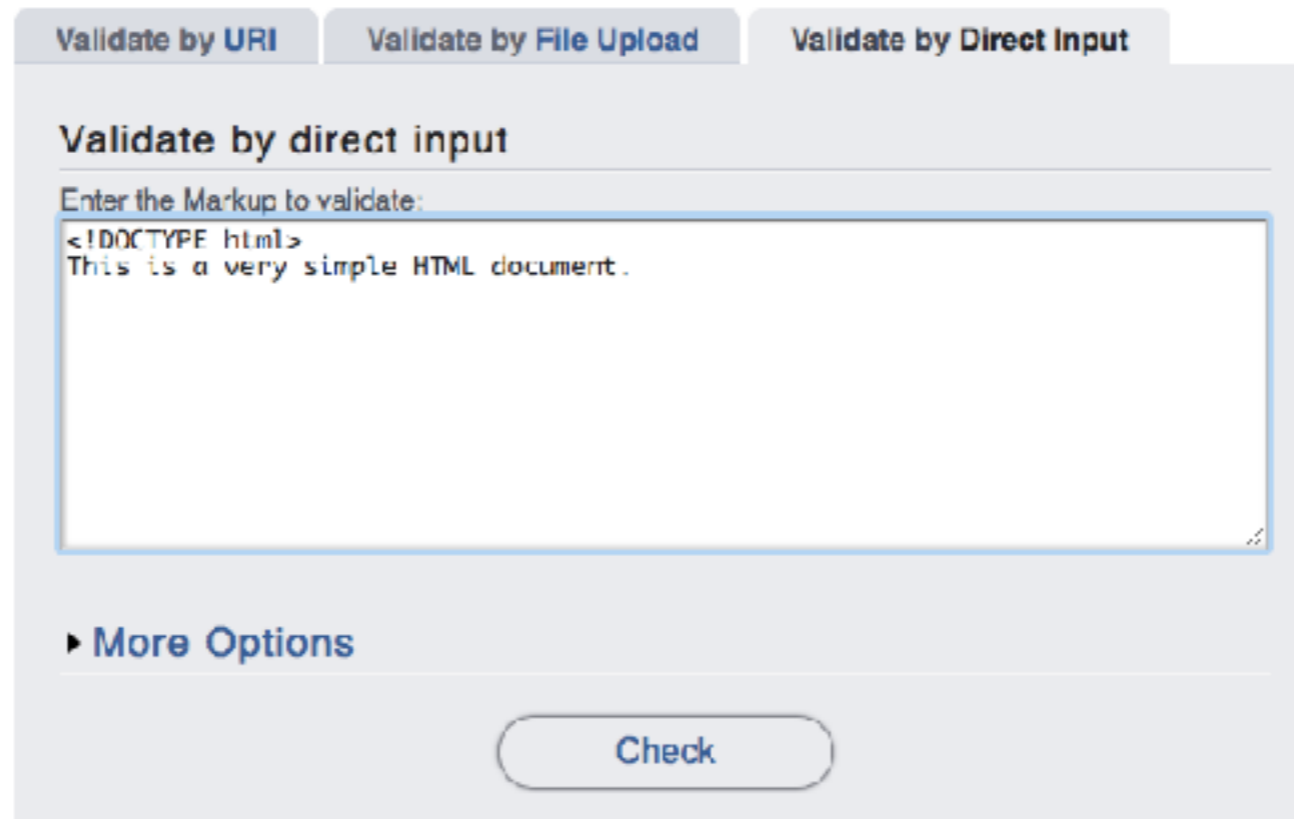
- 90s kids remember this
- HyperText *Markup* Language
- No compilation step, just plain text files saved somewhere
- Open a text editor and save this as "example.html":

```
<!DOCTYPE html>
```

```
This is a very simple HTML document.
```

HTML

- Open it in a browser
- It looks fine, but is it?
- Ask validator.w3.org!



The image shows a screenshot of the W3C HTML Validator interface. At the top, there are three tabs: 'Validate by URI', 'Validate by File Upload', and 'Validate by Direct Input'. The 'Validate by Direct Input' tab is selected. Below the tabs, the heading 'Validate by direct input' is displayed. Underneath, there is a label 'Enter the Markup to validate:' followed by a text area containing the following HTML code: `<!DOCTYPE html>`
`This is a very simple HTML document.` Below the text area, there is a link '► More Options'. At the bottom center, there is a 'Check' button.

HTML

- Oh no

Message Filtering

- Error** Element `head` is missing a required instance of child element `title`.
[From line 1, column 16; to line 2, column 35](#)
`TYPE html>↵This is a very simple HTML document.`
Content model for element `head`:
If the document is [an iframe srcdoc document](#) or if title information is available from a higher-level protocol: Zero or more elements of [metadata content](#), of which no more than one is a `title` element and no more than one is a `base` element.
Otherwise: One or more elements of [metadata content](#), of which exactly one is a `title` element and no more than one is a `base` element.

HTML

- I lied. This is the simplest HTML document.

```
<!DOCTYPE html>
<title>Simple HTML</title>
This is the simplest HTML document.
```

Document checking completed. No errors or warnings to show.

Source

1. `<!DOCTYPE html>`
2. `<title>Simple HTML</title>`
3. `This is the simplest HTML document.`

Used the HTML parser.

Total execution time 2 milliseconds.

HTML

- It's not really HTML without hypertext
- Links look like this: `link text`
- This is one of those cases where you can omit everything but the `/path/to/resource`
- Copy example.html to example2.html and add a link:

```
<!DOCTYPE html>  
<title>Simple HTML</title>  
This is the <a href="example.html">second-  
simplest</a> HTML document.
```

That's it!

- Mostly.
- Everything from Wordpress to facebook.com is just a matter of serving different specially-formatted text files at various URLs.
- These dynamic web applications (a subset of websites) are doing creative things, like generating HTML on the fly, but are ultimately subject to the same constraints as any other web page

Demo: <https://pages.github.com>