

Web Development Pre-101

A crash course by this guy ↓

What you'll need

- Laptop (not provided)
- Basic proficiency in Python, HTML, CSS, and JavaScript (or just wing it. That's what most “professional” web developers do anyways)
- Python 2.7.x with **pip** and **virtualenv**
- I'll be using PyCharm, you should too

What we'll cover

- Why is web development so difficult?
- HTTP — the duct tape of the internet
- HTML and CSS
- Flask — a server-side web app framework in Python
- Bootstrap — a CSS framework for passable designs
- No time for JavaScript :(

So... why indeed is it so difficult?

There's so much to do:

- Specifications
- Design
- Server-side development
- Client-side development
- Design implementation
- Social media integration
- Accessibility
- Test on a million different devices
- Deployment

So... why indeed is it so difficult?

There are so many choices:

- **Who am I designing for?**
 - Desktop-only? Desktop and mobile sites? Responsive design? Ads? Social media? Multilingual? Is money involved?
- **Which server-side framework should I pick?**
 - Drupal (PHP), Ruby-on-Rails (Ruby), node.js (JavaScript), Django (Python), ASP.NET MVC (C#), and a million more
- **And what about a client-side framework?**
 - or rather, which Frankensteinian combination of frameworks am I going to summon from the unholy depths of hell to conjure up this illusion of a working whole?

So... why indeed is it so difficult?

There are so many things to think about:

- Browser (in)compatibilities
 - Chrome, Firefox, Safari, Internet Explorer 10, Internet Explorer 9, Internet Explorer 8, Internet Explorer 7, Internet Explorer 6... and we didn't even get to the mobile browsers
- Security & Privacy
 - Cross-Site Scripting (XSS), SQL injections, clickjacking, “oops, I just exposed all my clients' credit card information to someone who signed up to the website with the username Rootie McAdmin”
- i18n & l10n
 - Internationalization (i18n) = “Hello!”, « Bonjour ! », “¡Hola!”
 - Localization (l10n) = “1,000”, “1.000”, “1 000”

So... why indeed is it so difficult?

Just to get started you need to know four different languages:

- HTML
- CSS
- JavaScript
- Server-side language such as PHP

So... why indeed is it so difficult?

When there isn't a choice to make, your only option usually sucks:

- HTML (it sucks)
- CSS (it sucks)
- JavaScript (it sucks)
- Server-side language such as PHP* (it sucks)

* You actually have a choice here. I just wanted to poke fun at PHP.

What can we do then?

Baby steps!

Let's learn some of the really, really basic basics of web development

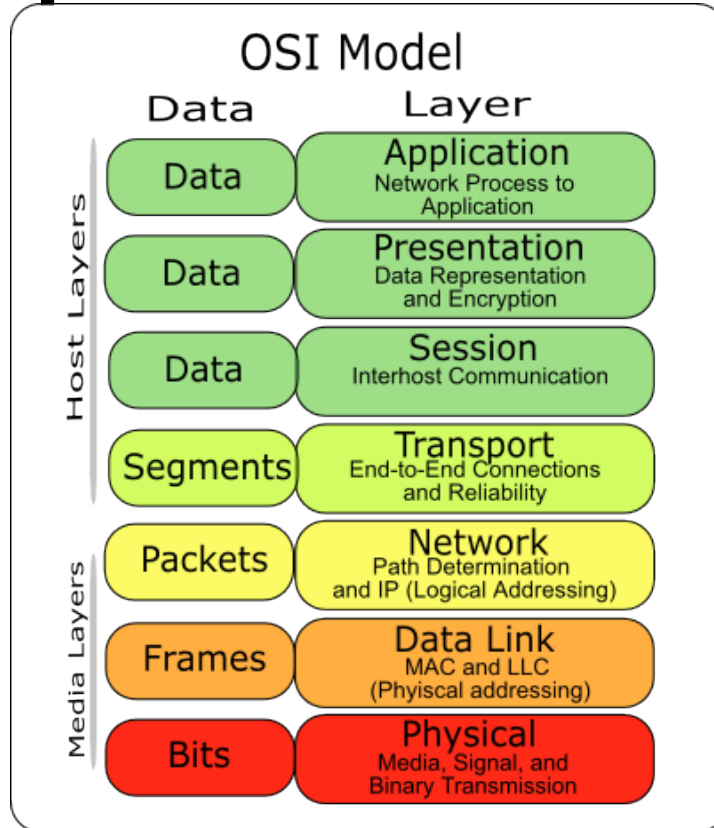
HTTP

HTTP = “HyperText Transfer Protocol”

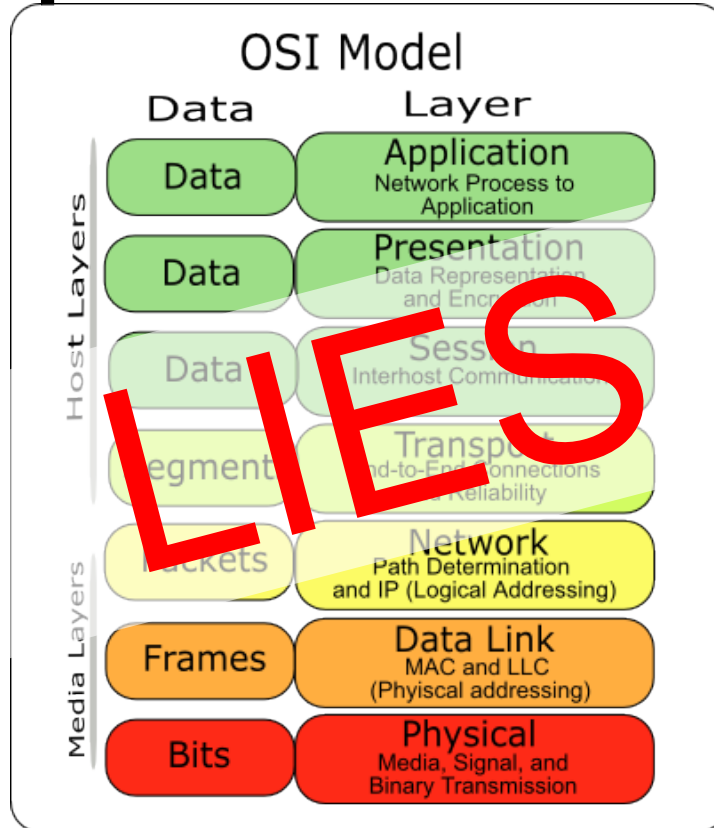
A client-server plain-text application-level network communications protocol

HyperText is structured text that uses logical links (hyperlinks) between nodes containing text. HTTP is the protocol to exchange or transfer hypertext.

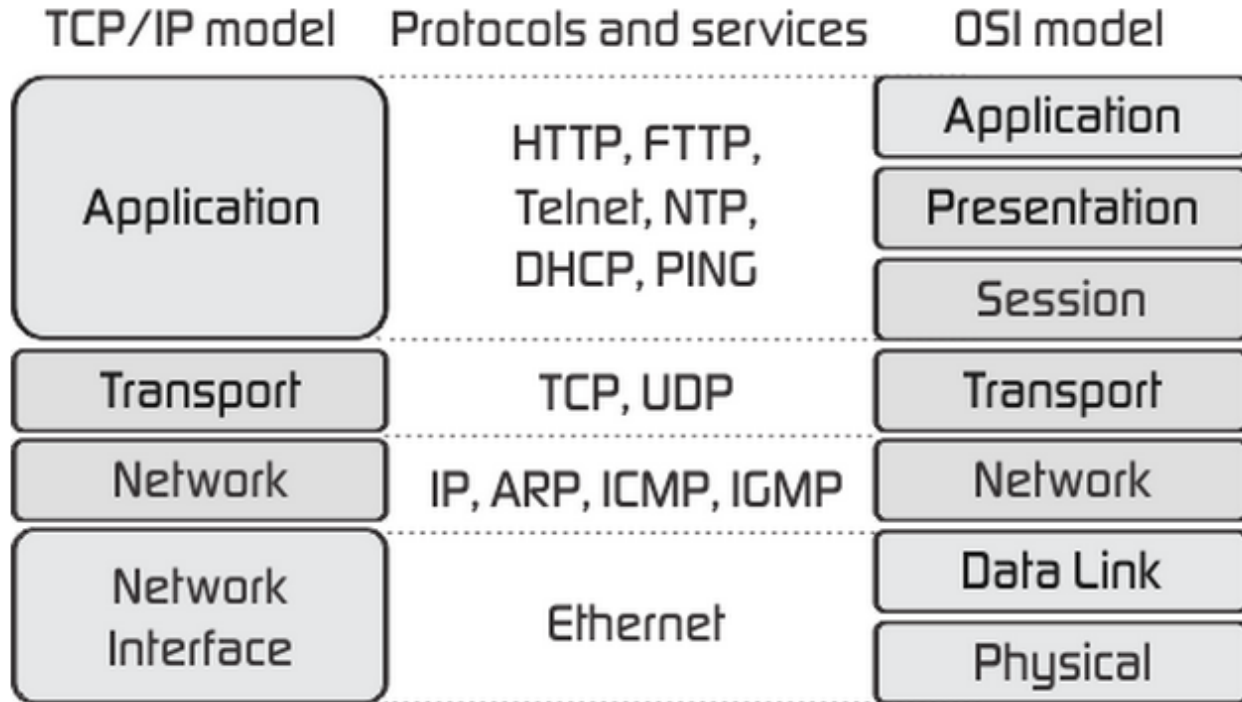
HTTP — application-level



HTTP — application-level



HTTP — application-level



HTTP — plain-text

Request

```
GET /watch?v=QH2-TGulwu4 HTTP/2.0
Host: www.youtube.com
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:36.0) Gecko/20100101 Firefox/36.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-CA,en;q=0.7,en-US;q=0.3
Accept-Encoding: gzip, deflate
Cookie: ...
Connection: keep-alive
```

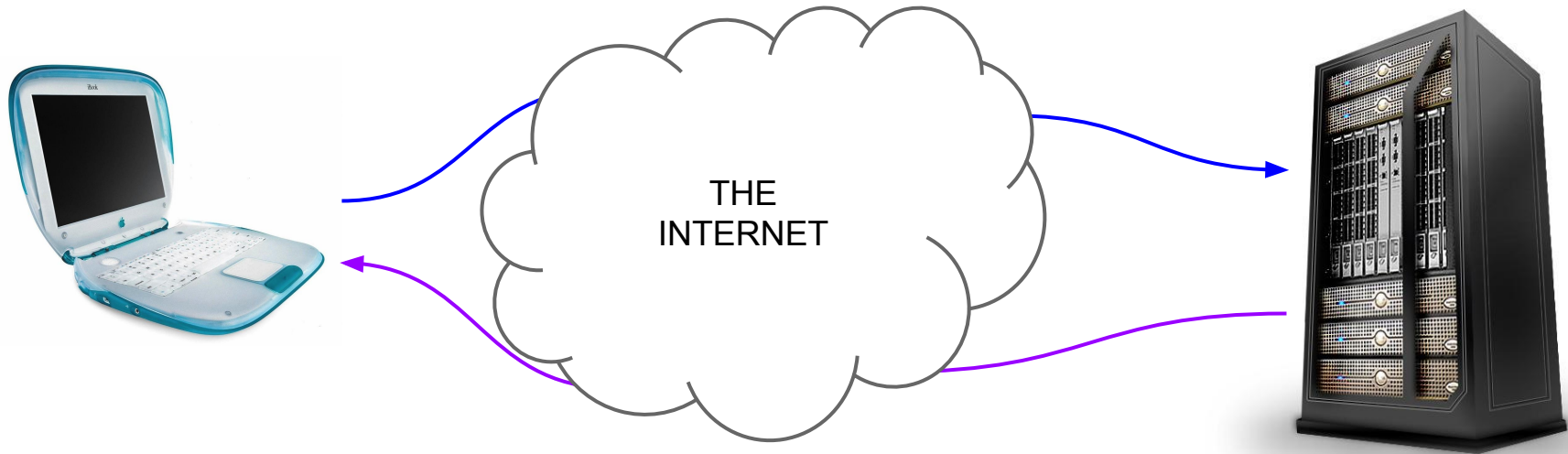
HTTP — plain-text

Response

```
Cache-Control: no-cache
Content-Encoding: gzip
Content-Type: text/html; charset=utf-8
Date: Sat, 28 Mar 2015 19:12:11 GMT
Expires: Tue, 27 Apr 1971 19:44:06 EST
Server: gwiseguy/2.0
X-Firefox-Spdy: h2-15
X-Frame-Options: SAMEORIGIN
x-content-type-options: nosniff
```

```
<!DOCTYPE html><html lang="en" data-cast-api-enabled="true"><head><script>var ytcsi =
{gt: function(n) {n = (n || '') + ...
```

HTTP — client-server



HTTP methods

- GET = “give me this thing”
- POST = “take this thing”
- HEAD = “tell me about this thing”
- PUT = “change this thing”
- DELETE = “delete this thing”

HTTP is...

- stateless
- insecure
- verbose
- slow

HTML

HTML = “HyperText Markup Language”

The standard (read: only) markup language for documents on the web.

The relationship between SGML, HTML, and XML is complicated

- HTML extends SGML, but HTML5 does not
- XML extends SGML, but HTML and HTML5 are not XML
- XHTML extends HTML (but is dead), XHTML5 is the XML variant of HTML5 which means it extends SGML

HTML — example

```
<!DOCTYPE html>
<html>
<head>
  <title>Example</title>
</head>
<body>
  <h1>Example</h1>
  <p>This is an amazing <span>example</span></p>
</body>
</html>
```



CSS

CSS = “Cascading Style Sheets”

A language used to give look and feel to HTML documents (or even XML!)

CSS — example

```
<!DOCTYPE html>
<html>
<head>
  <title>Example</title>
  <link rel="stylesheet" href="/style.css">
</head>
<body>
  <h1>Example</h1>
  <p>This is an amazing <span>example</span></p>
</body>
</html>
```

```
body {
  background-color: black;
  color: white;
}

h1 {
  color: pink;
  border: 1px dashed yellow;
}

p {
  text-align: center;
}

p span {
  display: inline-block;
  transform: rotate(-10deg);
}
```



Introducing: Flask!



Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions. And before you ask: It's [BSD licensed!](#)

Let's write a Hello, World!

Switch to your favourite IDE

Mine is PyCharm