Coding for Development and Social Change, Oct 2014

The Internet

What happens when you go to a website

This lecture

- What happens when you visit a website?
- basic html
- basic css
- requests and responses

Prerequisites

- A text editor
- A web browser (Chrome, Safari, Firefox etc)

Example: SIPA Homepage

What happens when you visit a webpage?

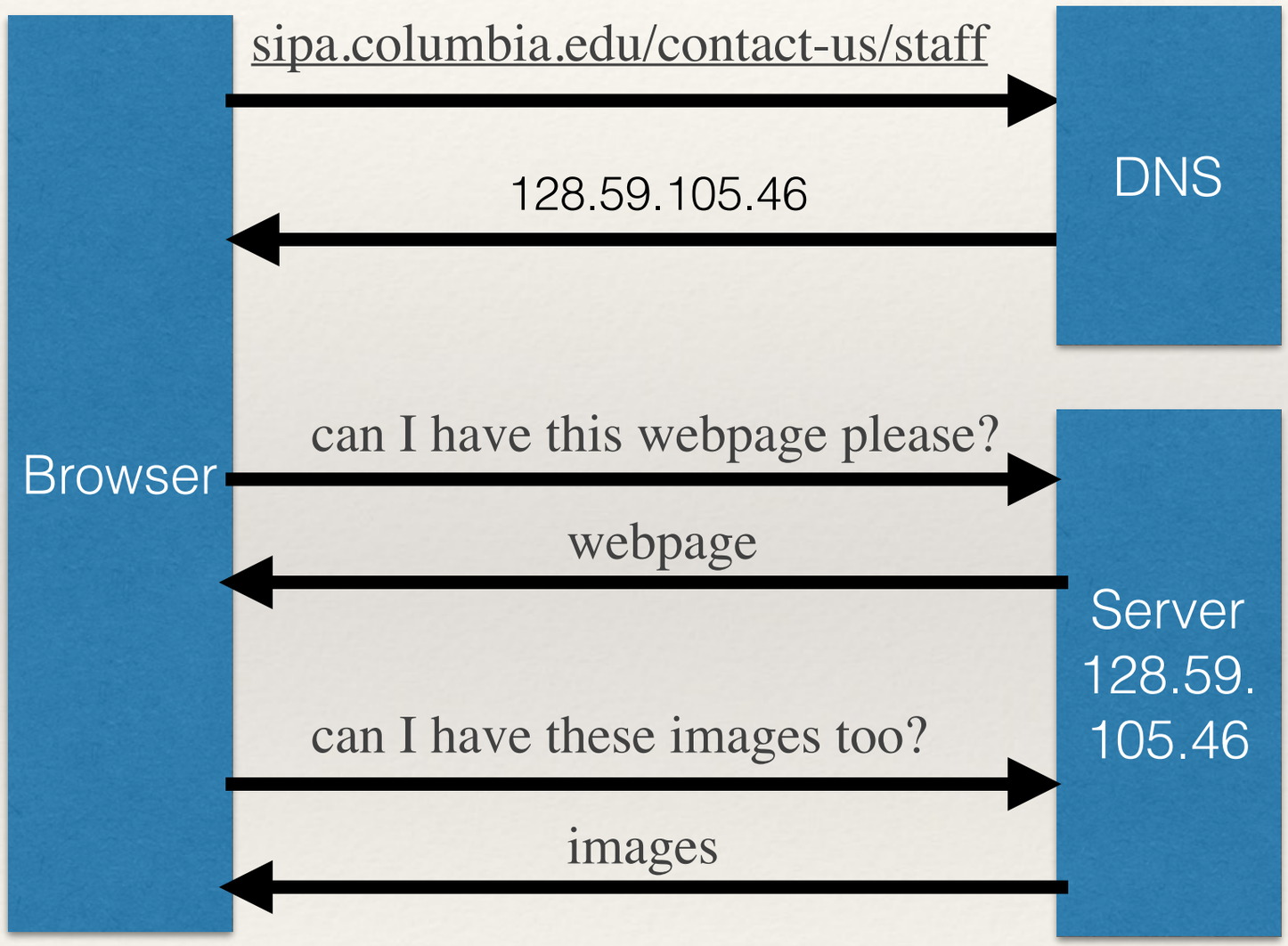
- Web browser:
  - Chrome, Safari, Firefox, Internet Explorer etc
  - Reads files stored on remote computers (“servers”)
  - processes them
  - displays results to you as webpages
  - can also read files on your own computer
    - and display to you in same way

Requests and Responses

- 1) You **type a url** (e.g. sipa.columbia.edu/contact-us/staff) into your browser
- 2) The browser sends this to a **domain name server** (like a big phonebook), which sends back the IP address (e.g. 128.59.105.46) of the server (remote computer) associated with that url
- 3) Your browser sends a **"request" message to the remote server**, asking for that webpage
- 4) The **remote server sends a "response"** containing the page that you asked for, back to your browser (or rather, to the server that your browser is on)
- 5) Your **browser requests any other information** it needs (e.g. images for the page), receives them, then displays it to you as a webpage.

http://styliii.com/blog/2012/10/17/how-the-internet-works/

Requests and Responses

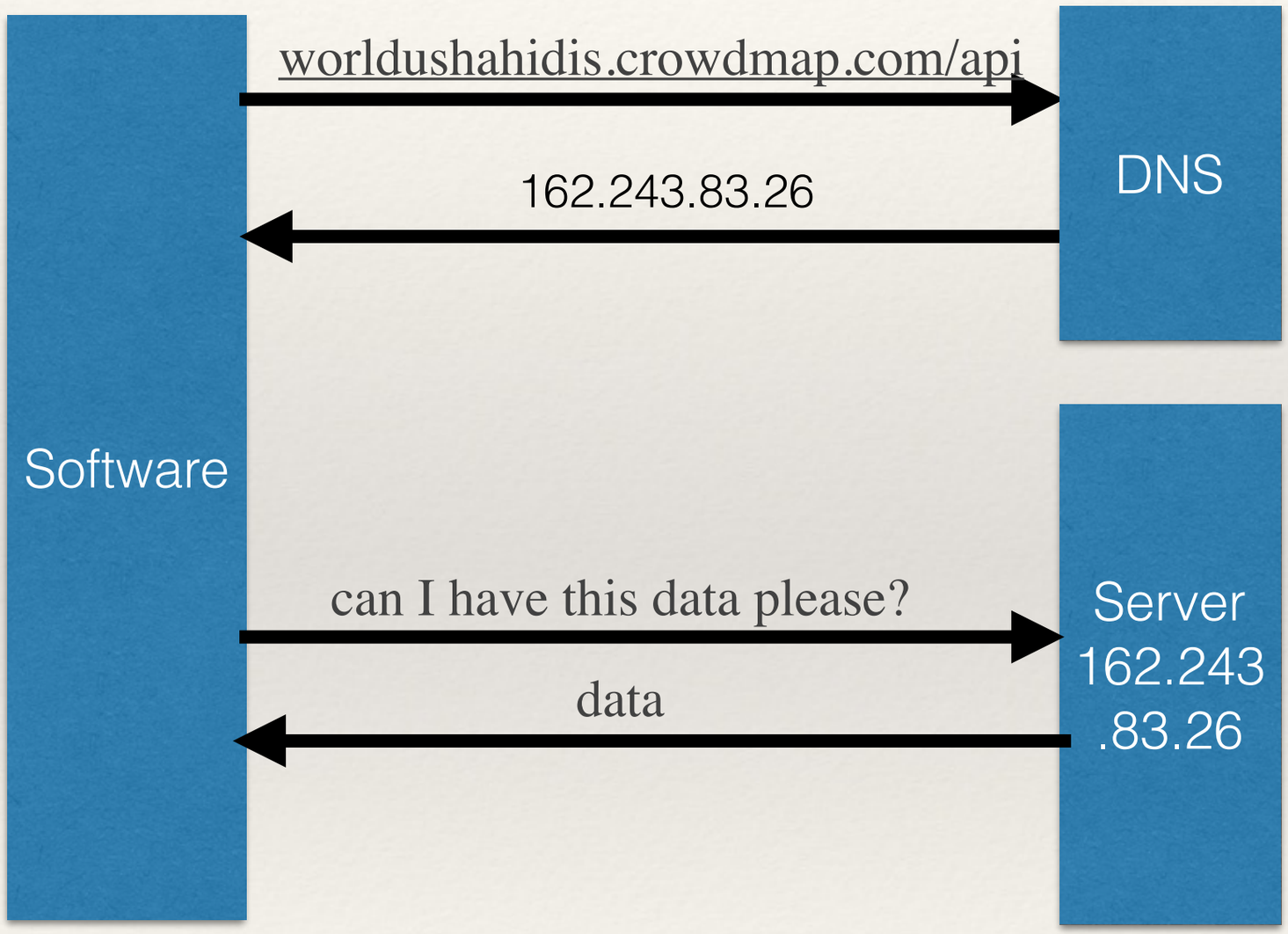


IP Addresses

- When you type in "sipa.columbia.edu", this is converted into an "IP address" - a unique code, like a zipcode, that identifies the server (remote computer) that a website is on.
- SIPA's IP address is 128.59.105.46 - you can find the IP address of a website by typing "**ping sips.columbia.edu**" (or the website you're interested in) into the terminal window. Ping sends a set of short "are you alive" messages to the other computer… use control-c to stop ping!
- Another fun program is **traceroute**: this shows you the route that your message takes across the internet to get to a website's server.

Can also check IP addresses online, e.g. at http://webipaddress.net

APIs


worldushahidis.crowdmap.com/api
162.243.83.26
DNS
Software
can I have this data please?
data
Server
162.243
.83.26

Data formats: JSON

KENplaces.json


```
{
"type": "FeatureCollection",

"features": [
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Mwingi", "NAMEPAR": null, "
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Admin-1 capital", "NAME": "Embu", "NAMEPAR": null, "NA
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Machakos", "NAMEPAR": null,
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Nanyuki", "NAMEPAR": null,
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Maralal", "NAMEPAR": null,
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Konza", "NAMEPAR": null, "N
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Lodwar", "NAMEPAR": null, "
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Eldama Ravine", "NAMEPAR":
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Sotik", "NAMEPAR": null, "N
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Namanga", "NAMEPAR": null,
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Naivasha", "NAMEPAR": null,
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Kericho", "NAMEPAR": null,
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Kitale", "NAMEPAR": null, "
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Bungoma", "NAMEPAR": null,
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Admin-1 capital", "NAME": "Kakamega", "NAMEPAR": null,
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Wajir", "NAMEPAR": null, "N
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Garissa", "NAMEPAR": null,
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Witu", "NAMEPAR": null, "NA
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Tsavo", "NAMEPAR": null, "N
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Voi", "NAMEPAR": null, "NAM
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Kilifi", "NAMEPAR": null, "
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Thika", "NAMEPAR": null, "N
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Kendu Bay", "NAMEPAR": null
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Karungu", "NAMEPAR": null,
{ "type": "Feature", "properties": { "SCALERANK": 7, "NATSCALE": 20, "LABELRANK": 5, "FEATURECLA": "Populated place", "NAME": "Kisii", "NAMEPAR": null, "N
```

Data formats: XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<osm version="0.6" generator="SMAP Server">
<node id="-1" action="modify" visible="true" >
    <tag k='i' v='Q1. We are working with World Vision Malawi collecting information about our ADP program. We
        will use this information to make sure that the activities in our program are useful to people in this
        village.' />
    <tag k='q' v='Todays Date?' />
    <tag k='name' v='todays_date' />
    <tag k='type' v='date' />
    <tag k='dateType' v='D' />
    <tag k='a' v='now' />
</node>
<node id="-2" action="modify" visible="true" >
    <tag k='i' v='Q2' />
    <tag k='q' v='Group Number?' />
    <tag k='name' v='Group_Number' />
    <tag k='type' v='text' />
    <tag k='textType' v='N' />
</node>
<node id="-3" action="modify" visible="true" >
    <tag k='i' v='Q3' />
    <tag k='q' v='Name of Interviewer?' />
    <tag k='name' v='Interviewer' />
    <tag k='type' v='text' />
</node>
<node id="-4" action="modify" visible="true" >
    <tag k='i' v='Q4' />
    <tag k='q' v='Head Village Code?' />
    <tag k='name' v='Head_Village' />
    <tag k='type' v='text' />
</node>
<node id="-5" action="modify" visible="true" >
    <tag k='i' v='Q5' />
    <tag k='q' v='Village Name Code?' />
```

Dynamic Webpages

- Requests and responses take time. Loading a whole page every time takes time. We get round this by using "dynamic" webpages
- Client-side: your code changes the webpage in response to user actions (e.g. the drop-down menus on the SIPA site) without asking the remote server for information
- Server-side: your code asks the server for information that's smaller than the whole webpage, then changes the webpage to match
- Javascript is really useful for this.

Cookies and Sessions

- **Cookie**: piece of text, stored on user's *client* machine
  - Created by code
  - Stays on the machine until removed

- **Session**: information stored on *server*
  - With "session id" stored on *client* machine
  - Created by code
  - Removed when user stops interacting with website

http://www.lassosoft.com/Tutorial-Understanding-Cookies-and-Sessions
http://www.tutorialspoint.com/ruby-on-rails/rails-session-cookies.htm

Looking at examples

- Chrome:
  - right-click page, then “view page source”
  - right-click anything on page, then “inspect element”
- Safari:
  - right-click page, select “show page source”

Most webpages contain…

- Html - content and structure
- Css - look and feel
- Javascript - actions
- Media files - images, video, audio etc

HTML

```
<!DOCTYPE html>
<html>
<head>
<title>My Hello World</title>
</head>
<body>
<p>Hello World!</p>
</body>
</html>
```

HTML tags

- Headings and paragraphs:
  - <h1> <h2> <h3> <h4> <h5> <h6> <p>
- Text: <b> <i> <u>
- Weblinks:
  - SIPA Columbia
- Images:
  - <img src="http://en.wikipedia.org/wiki/Kitten#mediaviewer/File:Cat-MaineCoon-Cookie.png" alt="This is a kitten" width=150 height=100>
- Tables: <tr> <th> <td>

- CSS: <style> <link>
- Javascript: <script>
- Meta:
  - <meta name="description" content="A webpage about open data">
  - <meta name="keywords" content="development,data,learning">
  - <meta name="author" content="Sara Terp">

HTML Tag Attributes

- Attributes tell the browser more about a tag:
  - <a **href**="https://sipa.columbia.edu/">SIPA
  - <img **src**="http://en.wikipedia.org/wiki/Kitten#mediaviewer/File:Cat-MaineCoon-Cookie.png" **alt**="this is a kitten" **width**=300 **height**=200>
- Other attributes:
  - id - tag identifier
  - style - CSS style for this tag
  - value - form input value
  - disabled - hide this button or form

Html: Your Turn

- Write an html page that describes your new system. Call it “about.html”. Make sure it has all these pieces:

<!DOCTYPE html>

<html>

<head>

<title>My Hello World</title>

</head>

<body>

** Hello World! **

</body>

</html>

And add in anything else you need to tell this story:

- text (use **<h1></h1>, <h2></h2>, ,
, , ** etc as appropriate)
- links (**link text**)
- images (**<img src=“url” alt=“alternative text” width=100 height=150>**)

- You have 15 minutes… go!

CSS

- CSS makes your html pretty. It can be "**inline**" (part of your html code):

  ** This text is red **

- or a set of CSS commands **inside <style> tags** in your html code:

  <head>

  <style>

  your css commands

  </style>

  </head>

- Or **in a file** (an "external stylesheet") outside your html code, e.g.:

  <link rel="stylesheet" type="text/css" href="name of your css file" />

  (you might also see <style type="text/css">@import url(name of your css file)</style> in code)

(External stylesheets are best if you have lots of pages to style the same way)

CSS code

- CSS code looks like this:

tag {property : value; property : value}

- Where tag is one of the html tags we saw earlier
- You can also use CSS “classes” to select the tags you want a CSS command to affect, e.g.

** some text **

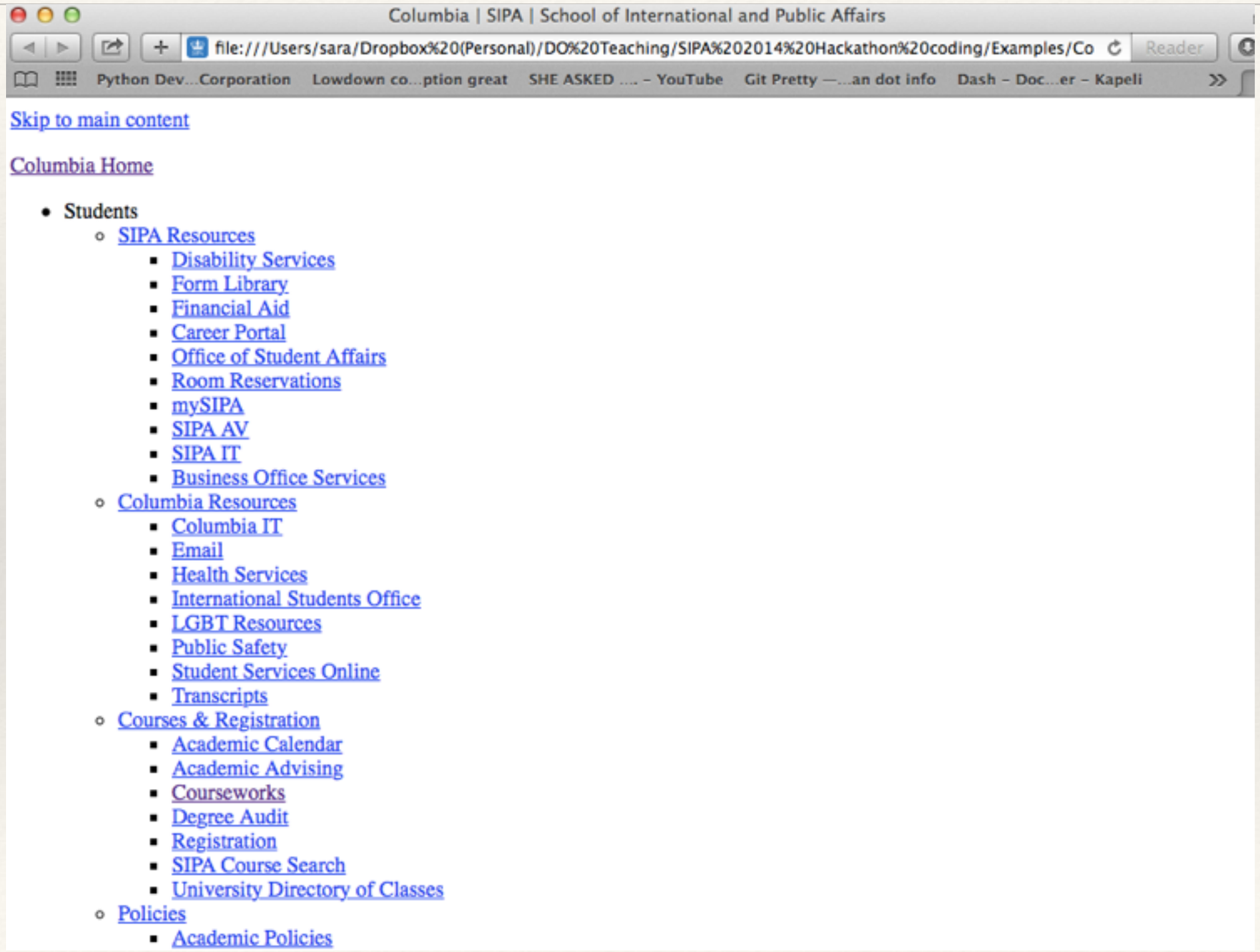
** some more text **

** Yet more text **

- with a CSS command that looks like this:

.redp { color: #ff0000; font-size: small;}

SIPA without the CSS



Javascript

- Javascript is a programming language that you can use to change the html on a webpage, *without* sending requests to the server

- You can use Javascript to check inputs (e.g. “is this a number?”, make parts of the page visible or invisible, load just *part* of a webpage at a time (using something called Ajax)

- Javascript is also behind many popular data visualisation languages (e.g. the D3 library, d3js.org)

Good tutorial: http://www.w3schools.com/js/js_intro.asp and Book: Javascript the good parts

Stacks

- Stack consists of:
  - operating system: Linux, Mac OSX, Windows: the software that controls your hardware
  - Web server: the software that connects your code to a browser
  - Database: the software that makes data available to your code
  - Programming language: Ruby on Rails etc.

- Well-known stacks:
  - LAMP (Linux/Apache/MySQL/Php).

- This weekend's stack:
  - (MacOS or Windows or Linux)/WEBrick/Sqlite/Ruby on Rails