Coding for Development and Social Change, Oct 2014

The Internet

What happens when you go to a website

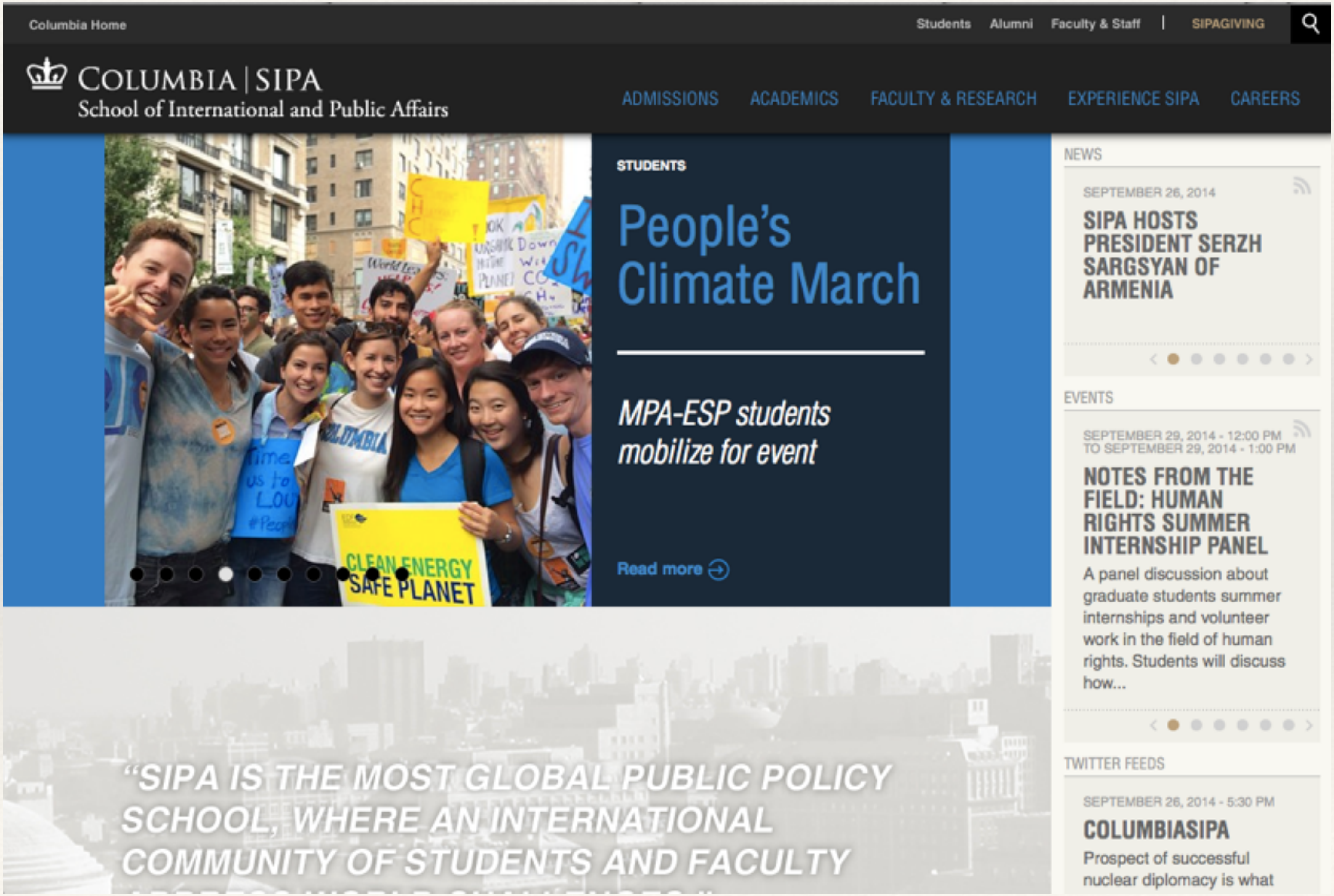
This lecture

- 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



STUDENTS

People's Climate March

MPA-ESP students mobilize for event

SEPTEMBER 26, 2014

SIPA HOSTS PRESIDENT SERZH SARGSYAN OF ARMENIA

SEPTEMBER 29, 2014 - 12:00 PM TO SEPTEMBER 29, 2014 - 1:00 PM

NOTES FROM THE FIELD: HUMAN RIGHTS SUMMER INTERNSHIP PANEL

A panel discussion about graduate students summer internships and volunteer work in the field of human rights. Students will discuss how...

SEPTEMBER 26, 2014 - 5:30 PM

COLUMBIASIPA

Prospect of successful nuclear diplomacy is what

"SIPA IS THE MOST GLOBAL PUBLIC POLICY SCHOOL, WHERE AN INTERNATIONAL COMMUNITY OF STUDENTS AND FACULTY

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”

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/

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

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.

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:
  - <a href=“https://sipa.columbia.edu/“>SIPA Columbia</a>
- 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</a>
  - <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>
<p>Hello World!</p>
</body>
</html>
```

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

- text (use **<h1></h1>, <h2></h2>, <p></p>, <br>, <b></b>, <i></i>** etc as appropriate)
- links (**<a href=“url”>link text</a>**)
- 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):

```
<p style=“color:red”>This text is red</p>
```

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

```
<head>
<style>
your css commands
</style>
</head>
```

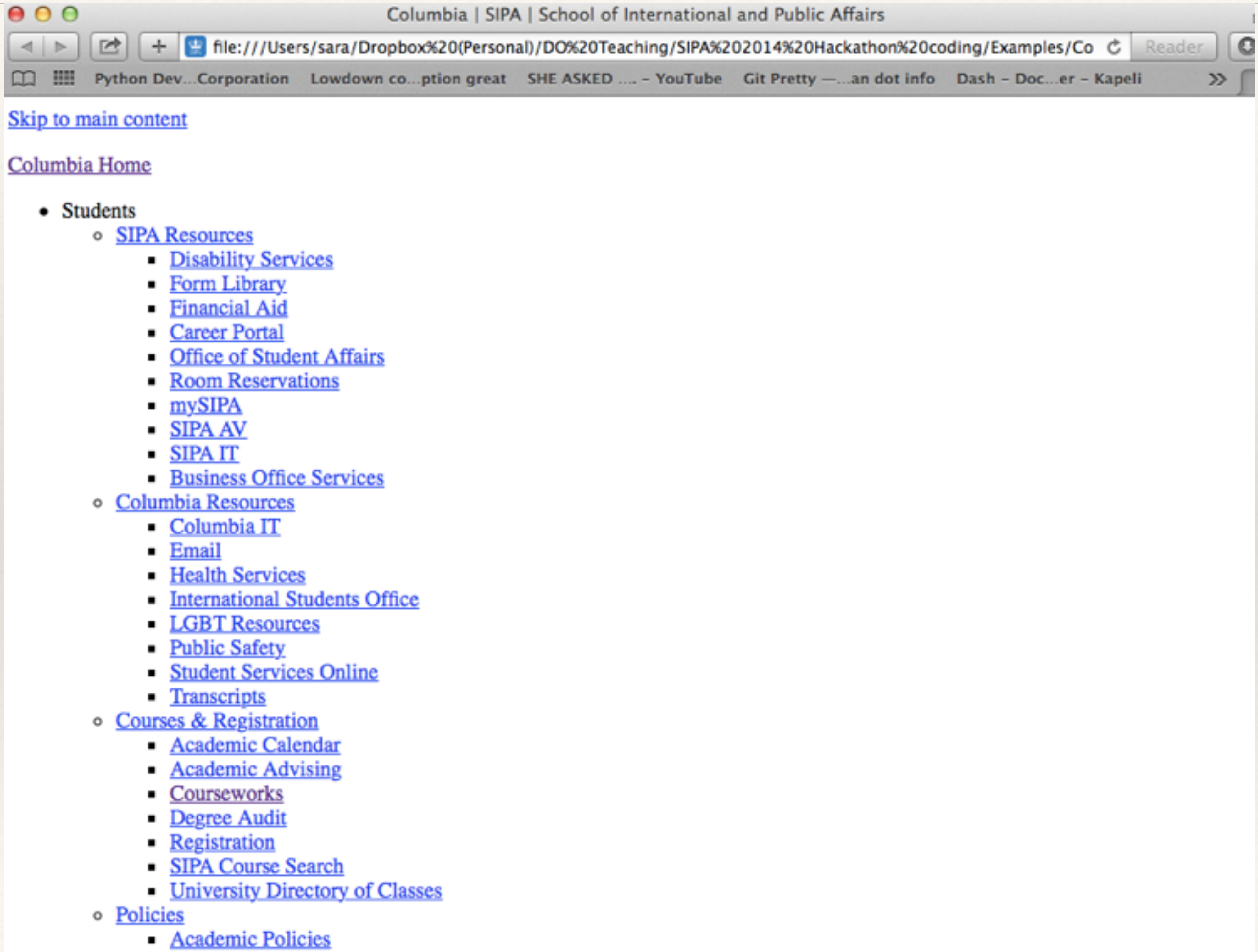
- Or in a file (an “external stylesheet”) outside your html code, e.g.:

```
<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)

SIPA without the CSS


Columbia | SIPA | School of International and Public Affairs
file:///Users/sara/Dropbox%20(Personal)/DO%20Teaching/SIPA%202014%20Hackathon%20coding/Examples/Co
Reader
Python Dev...Corporation
Lowdown co...ption great
SHE ASKED .... – YouTube
Git Pretty —...an dot info
Dash – Doc...er – Kapeli

Skip to main content

Columbia Home

- Students
  - SIPA Resources
    - Disability Services
    - Form Library
    - Financial Aid
    - Career Portal
    - Office of Student Affairs
    - Room Reservations
    - mySIPA
    - SIPA AV
    - SIPA IT
    - Business Office Services
  - Columbia Resources
    - Columbia IT
    - Email
    - Health Services
    - International Students Office
    - LGBT Resources
    - Public Safety
    - Student Services Online
    - Transcripts
  - Courses & Registration
    - Academic Calendar
    - Academic Advising
    - Courseworks
    - Degree Audit
    - Registration
    - SIPA Course Search
    - University Directory of Classes
  - Policies
    - Academic Policies

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.

```
<p class=“redp”>some text</p>

<p>some more text</p>

<p class=“redp”>Yet more text</p>
```

- with a CSS command that looks like this:

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

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