EECS 1012: Introduction to Computer Science

September 16, 2016

The Web

- Fundamental concepts
  - Uniform Resource Locator
    - scheme://domain:port/path?query#fragment
      - e.g. http://www.eecs.yorku.ca
  - Standard content
    - Hypertext markup language (HTML)
  - Standard protocol
    - Standard network protocol (HTTP)

HTML

- In essence a text document with ‘markup’ instructions
  - in italics
- Critical markup instruction is a link to another document
  - cs home
- Can be created with any text editor although special purpose tools exist

HTTP

- Standard protocol to request a page from a server
  - GET /images/foo.jpg HTTP/1.1
- If the server has ‘/images/foo.jpg’ then the page is returned, otherwise an error is reported

Net effect

- You see a page and click on a hyperlink
- The browser sends a HTTP request to a machine on the network
  - That sends back the content
  - So your browser loads the next page

And that gets us to the first of two pillars of this course

- Original web pages were static (just display static material).
  - 1995 Brendan Eich (working at Netscape) developed what we now know as JavaScript - a language initially designed to animate web pages.
  - In 1996 Cascading Style Sheets were proposed by the World Wide Web Consortium to standardize the appearance of web pages.
- By 2015 these three tools (HTML, JavaScript and CSS) had evolved into a mature set of tools to develop interactive software capable of running on a wide range of hardware.

The second pillar

- In the mid to late 1990's increases in computing hardware power coupled with miniaturization and battery performance saw the development of mobile computing infrastructure.
- By 2011 it was estimated that more smart phones were sold world wide than PC's annually.

This course

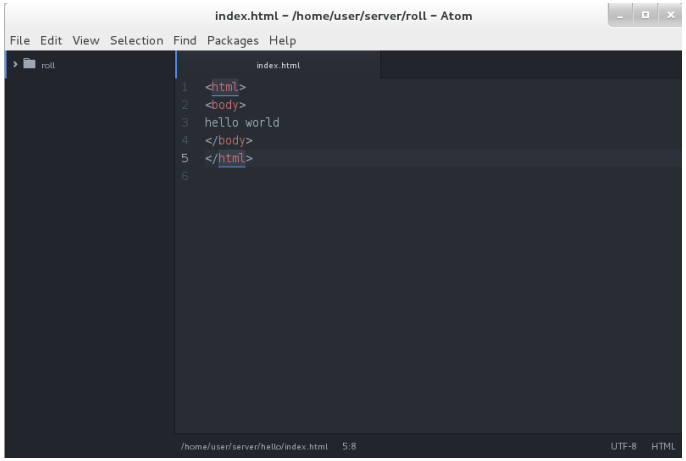
- Software development (programming if you like) in a web-based/cloud-based/net-centric environment.
- HTML, JavaScript, CSS
- Client-server programming
- Databases, JSON, XML

As an example

- Goal -> create a web page (an application) that puts ‘hello world’ on an application on an Android
- To get here
  - Virtual Box running the current version of the appliance
  - An Android device
  - Both on the same network

Step I

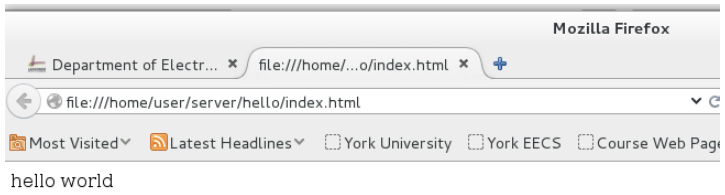
- Fire up the virtual machine
- Use atom to create the file index.html in the ‘server’ directory



control shift h - preview

Step II

- Yes, its a web page
- So any browser you happen to have around will happily ‘render’ the page



Here is firefox, running in the box

Now on to the Android

- We have crafted an application html2apk that takes a zip file of a web site and displays it as an application.
- The virtual box runs a service that will take a folder (directory) under the server directory and automatically create the zip file and upload it to the Android
- Suppose the ip address of the host machine is 1.2.3.4 then on the Android device
  - http://1.2.3.4:8000/zip/hello is the magic url

Android

- Menu items
  - About - an alert
  - Settings - change things about what is happening
  - Contents - what files are there in the archive
  - Alert Log - alerts that have been displayed
  - Web view - refreshes the web view
  - Start - start ‘index.html’

Android Settings

- Autostart - the page starts when loaded
- Clear console on start
- JavaScript alert shows on screen
- Screen orientation (portrait, landscape, rotate)

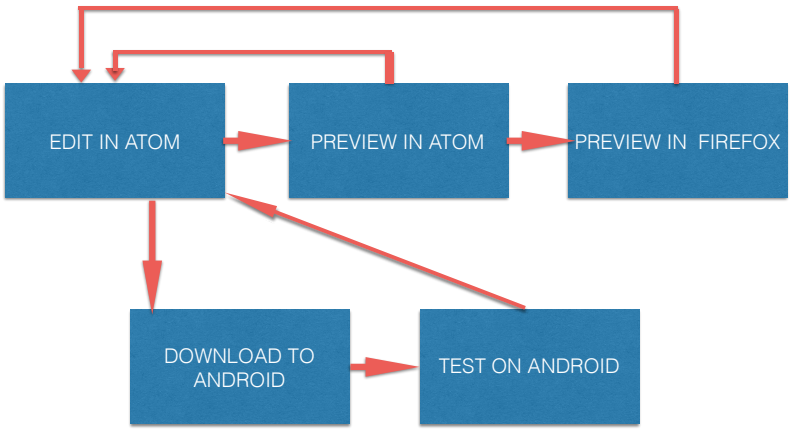
Getting started

- VirtualBox - download as described on web page
- Appliance - download as described on web page
- Android application - download as described on web page
  - Note: for your machine, you need to enable ‘download from untrusted source’

Some ‘gotcha’s’

- You need to know the ip address of the host environment
  - Use the web page at www.eecs.yorku.ca/~jr/server/ip (will not always work if you are behind a firewall)
- If you are doing this outside of the lab, make sure both the Android device and the host machine are on the same subnet.

Workflow



Questions???

(in the lab, TA’s to help)

So lets do something

Lets do something else
Display a picture of a cat*
(what else is the internet for)

*If you don't like cats, use a picture of something else.

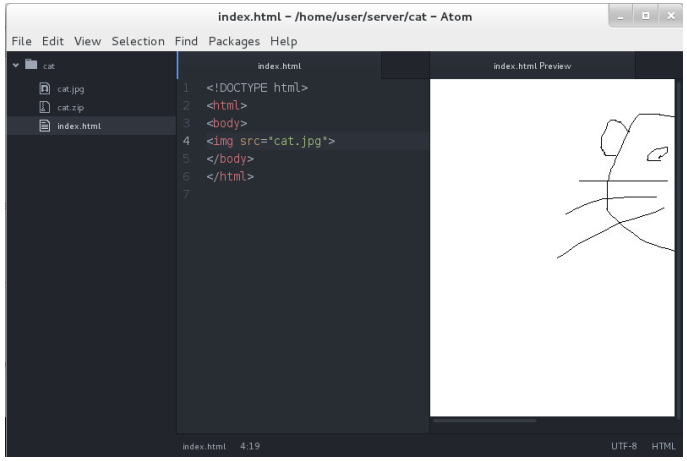
Project

- We need a folder under the 'server' directory in the virtual box.
- Linux (like almost all modern OS's) stores files in a folder, which can be in another folder, and so on.
- Lets put the project in the folder server/cat

Displaying an image in HTML

- HTML (Hypertext markup language)
- Like many (all?) computer languages there is a syntax "is this a valid program" and each piece of syntax has an associated semantics "what does this mean".
- But lets do that later, lets start with some examples

Displaying an image in HTML



And onto the Android device

- http://ip:8000/zip/cat
  - Try it yourself

"Historically" (aka 5 years ago)

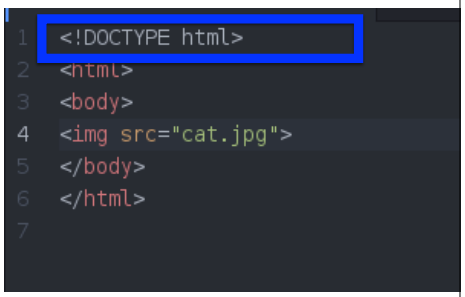
- Typically an entire web site would be encoded as a collection of HTML pages and images with text style (colour, fonts, etc.) included in the HTML.
- Now (HTML5) it is considered better style to separate the definition of content/structure (HTML) from the definition of style (using CSS : Cascading Style Sheets).

Lab 1 examines this concept in detail

HTML syntax

Document type (html)

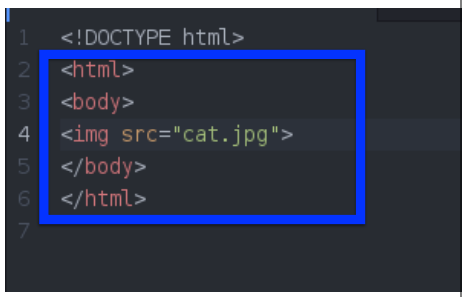
- A text file
- Processed character by character
- First sequence of characters 'should be'
  - <!DOCTYPE html>



HTML syntax

<html> …. </html>

- A text file
- Processed character by character
- First sequence of characters 'should be'
  - <!DOCTYPE html>



<html> tag entire page exists within it
<html> open … </html> close

HTML syntax

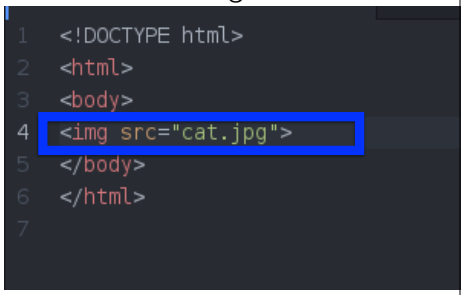
<body> …. </body>

- A text file
- Processed character by character
- First sequence of characters ‘should be’
  - <!DOCTYPE html>

<body> tag all of the body exists in it
<body> open … </body> close

HTML syntax

- A text file
- Processed character by character
- First sequence of characters ‘should be’
  - <!DOCTYPE html>



Its an image

Elements

- Large number.
  - (You will learn many of them, but you can look them up too.)
- <tagname> … </tagname>
- Some tags are “empty” (like img). They can have no content
  - <img src=“cat.jpg”>

Elements can have attributes

- Add ‘information’ to the element
  - <img src=“cat.jpg”>
- Different elements have different attributes
  - **NB: HTML tries to ignore things that are wrong**

http://validator.w3.org