GENERAL ASSEMBLY

PROGRAMMING FOR NON PROGRAMMERS

Jeremiah Alexander | Lead Instructor | WDI

Learning Objectives

Learn the technical vocabulary to communicate effectively with developers

Program your own basic websites in HTML/CSS

Think like a programmer and write code in Javascript

Schedule & Lesson Format

16/11 - Part 1	Programming the World Wide Web
17/11 - Part 2	Building Websites with HTML & CSS
23/11 - Part 3	Adding Interactivity with Javascript
24/11 - Part 4	Deploying Websites using Github

Each Lesson

7:00pm - 7:30pm	Introductions / Recap
7:30pm - 8:00pm	Lecture Topic 1
8:00pm - 8:30pm	Lab Topic
8:30pm - 9:00pm	Lecture Topic 2
9:00pm - 9:30pm	Lab Topic

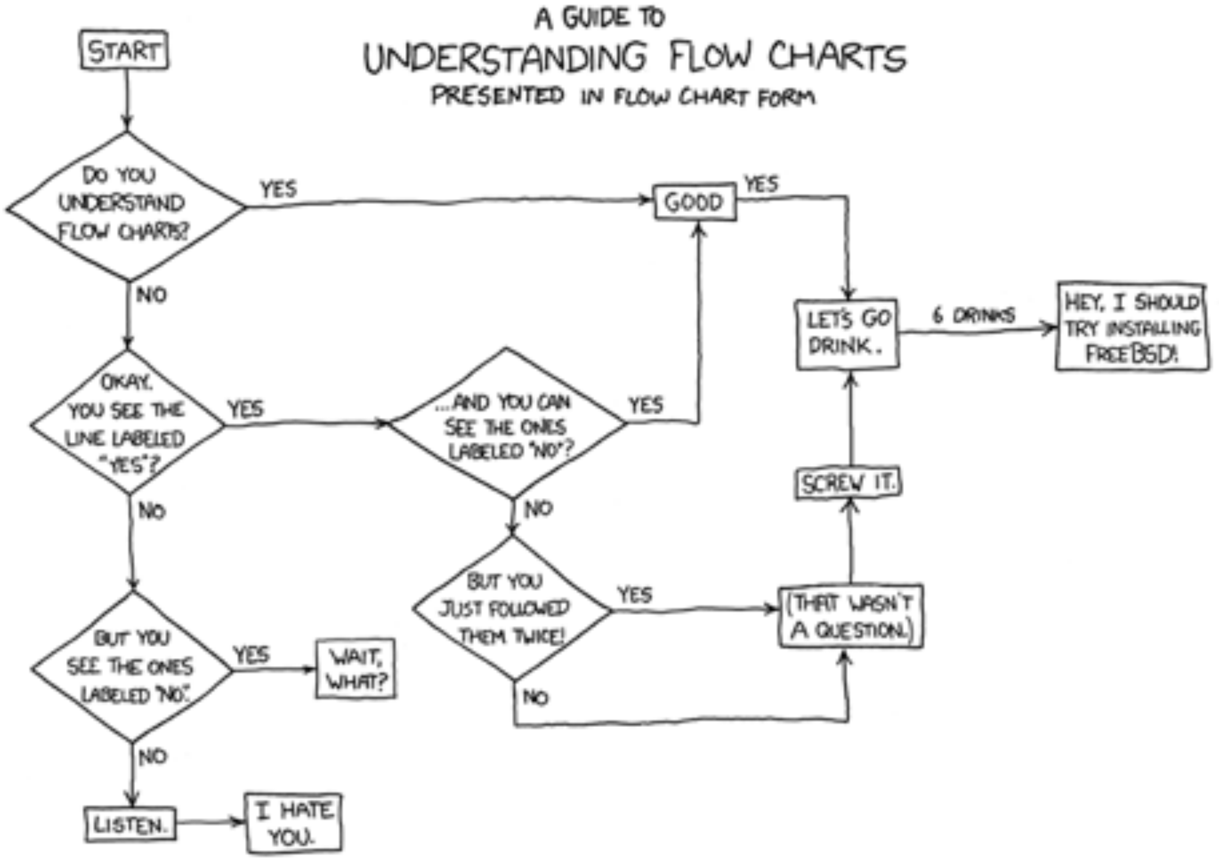
INTRODUCTIONS GOALS & RULES

PART 1
PROGRAMMING THE WORLD WIDE WEB

Learning Objectives

- Define what Programming is
- Give examples of different Programming Languages
- Describe how the Web work
- Identify the stages of Web Development

HOW TO PROGRAM A COMPUTER


A GUIDE TO
UNDERSTANDING FLOW CHARTS
PRESENTED IN FLOW CHART FORM
START
DO YOU UNDERSTAND FLOW CHARTS?
YES
GOOD
YES
NO
OKAY. YOU SEE THE LINE LABELED "YES"?
YES
...AND YOU CAN SEE THE ONES LABELED "NO"?
YES
LET'S GO DRINK.
6 DRINKS
HEY, I SHOULD TRY INSTALLING FREEBSD!
NO
NO
BUT YOU JUST FOLLOWED THEM TWICE!
YES
(THAT WASN'T A QUESTION.)
SCREW IT.
NO
BUT YOU SEE THE ONES LABELED "NO".
YES
WAIT, WHAT?
NO
LISTEN.
I HATE YOU.

Programming Languages

- Human
- SQL, HTML & CSS
- JavaScript / Python / Ruby
- Java / C#
- C/C+
- Assembly
- Machine

High Level vs. Low Level

Ease of Use vs. Performance

Portability vs. Specificity

Machine Code

```
8B542408 83FA0077 06B80000 0000C383
FA027706 B8010000 00C353BB 01000000
C9010000 008D0419 83FA0376 078BD98B
B84AEBF1 5BC3
```

Assembly

```
fib:
    mov edx, [esp+8]
    cmp edx, 0
    ja @f
    mov eax, 0
    ret

    @@:
    cmp edx, 2
    ja @f
    mov eax, 1
    ret
```

C

```
   if (n <= 0)
         return 0;
     else if (n <= 2)
         return 1;
     else {
         unsigned int a,b,c;
         a = 1;
         b = 1;
         while (1) {
             c = a + b;
             if (n <= 3) return c;
             a = b;
             b = c;
             n--;
         }
     }
}
```

SQL

```
SELECT title
 FROM   Book
 WHERE price > 100.00
 ORDER BY author;
```

Same Same but Different

Receiving Input	Text, Clicks, Speech, Location, Big Data
Remembering	Data Stores: Variables, Cookies, Files, Databases
Making Decisions	Conditionals: if, else, then
Repeating Processes	Iteration/Loops: for, each, while
Producing Output	Text, Image, Sound, Movement

Programming Languages on the Web

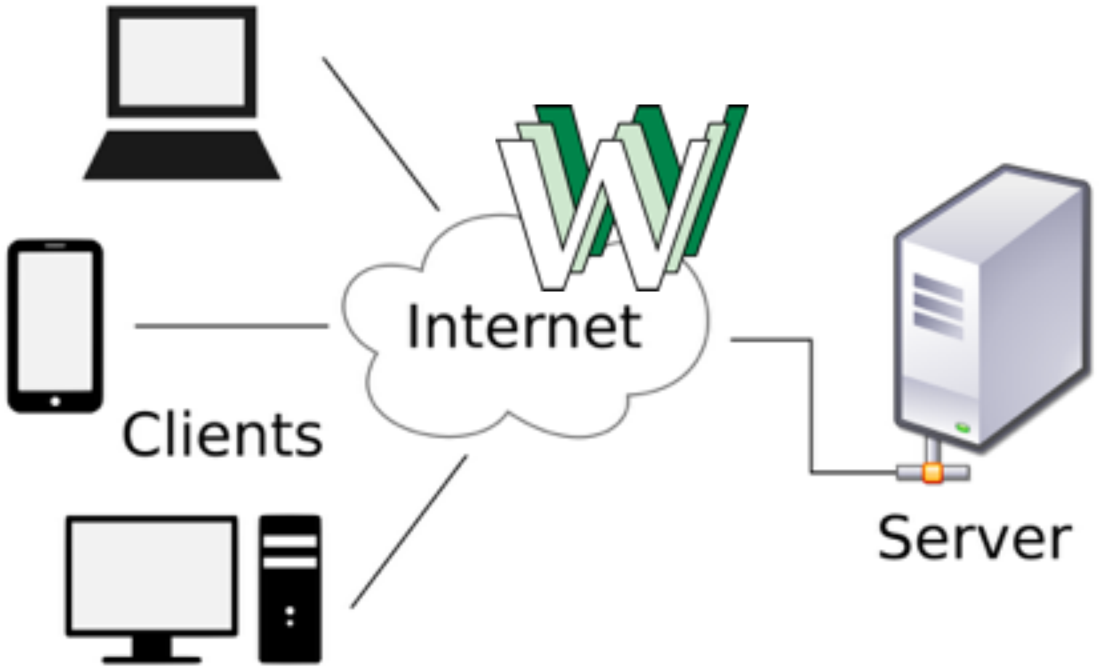
- Human
- SQL, HTML & CSS
- JavaScript / Python / Ruby
- Java / C#
- C/C+
- Assembly
- Machine

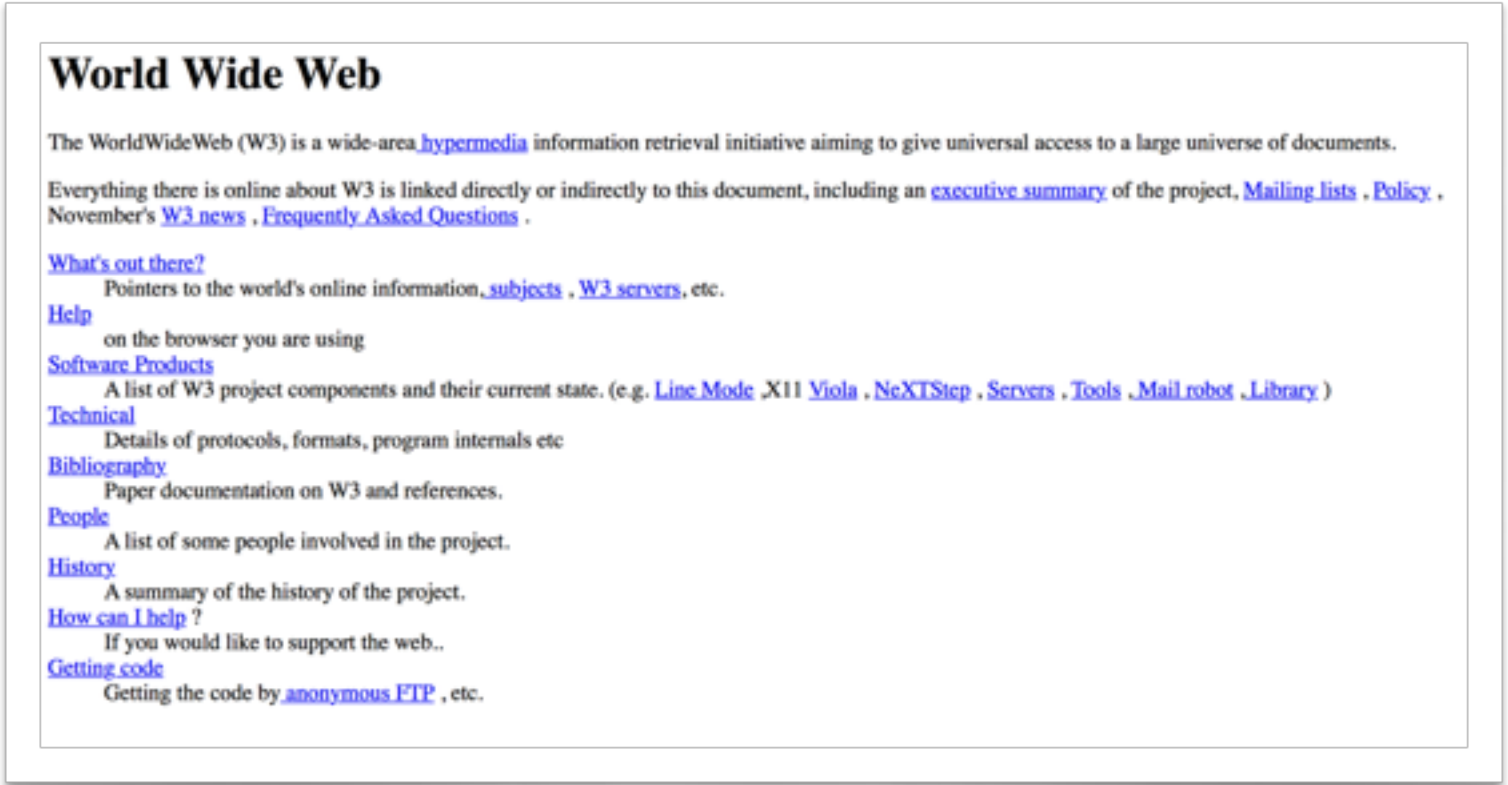
High Level vs. Low Level

Ease of Use vs. Performance

Portability vs. Specificity

HOW THE WEB WORKS


Internet
Clients
Server



World Wide Web

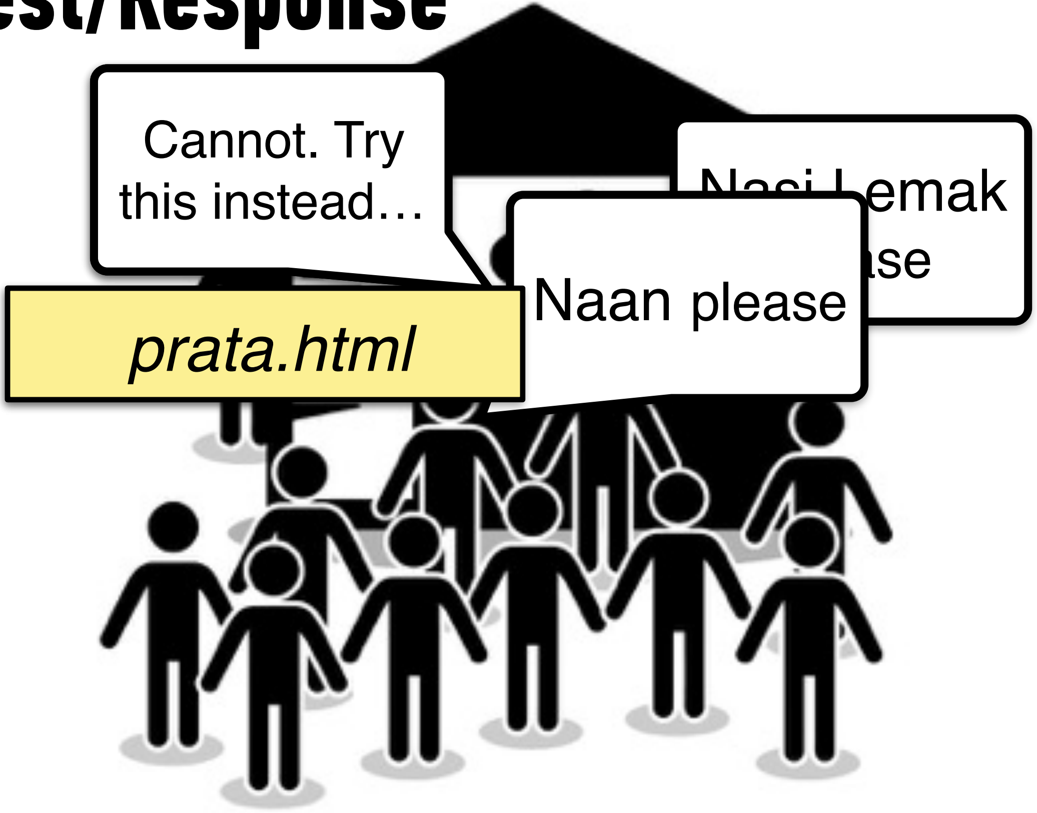
The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

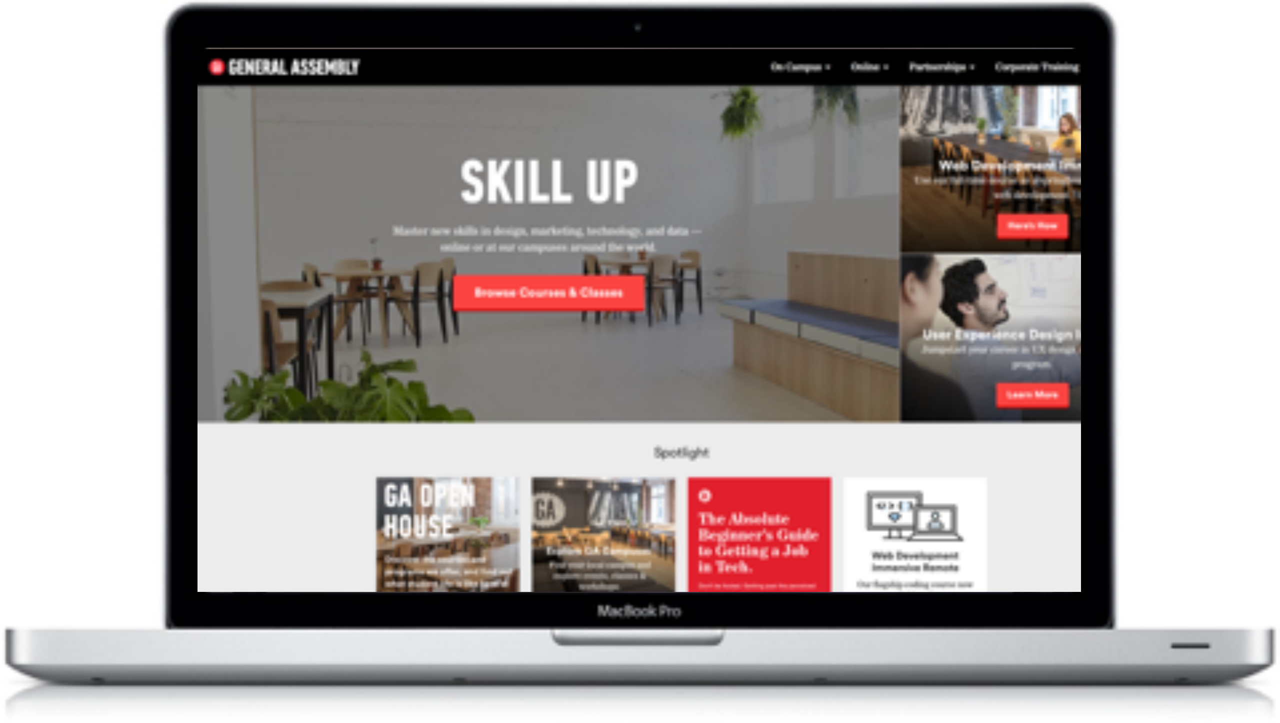
What's out there?
  Pointers to the world's online information, subjects , W3 servers, etc.
Help
  on the browser you are using
Software Products
  A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)
Technical
  Details of protocols, formats, program internals etc
Bibliography
  Paper documentation on W3 and references.
People
  A list of some people involved in the project.
History
  A summary of the history of the project.
How can I help ?
  If you would like to support the web..
Getting code
  Getting the code by anonymous FTP , etc.

http://info.cern.ch/

HTTP Request/Response



What is a Web Client?



Client-side / Front-end Development

HTML	CSS	JS
structure	look	dynamism

Server-side / Back-end Development

php

JS

Java

PostgreSQL

mongoDB®

LAB
PAPER PROGRAMMING

ATM Program Design

Design a basic computer program for an ATM: PIN validation, Balance Enquiries, Withdrawals etc

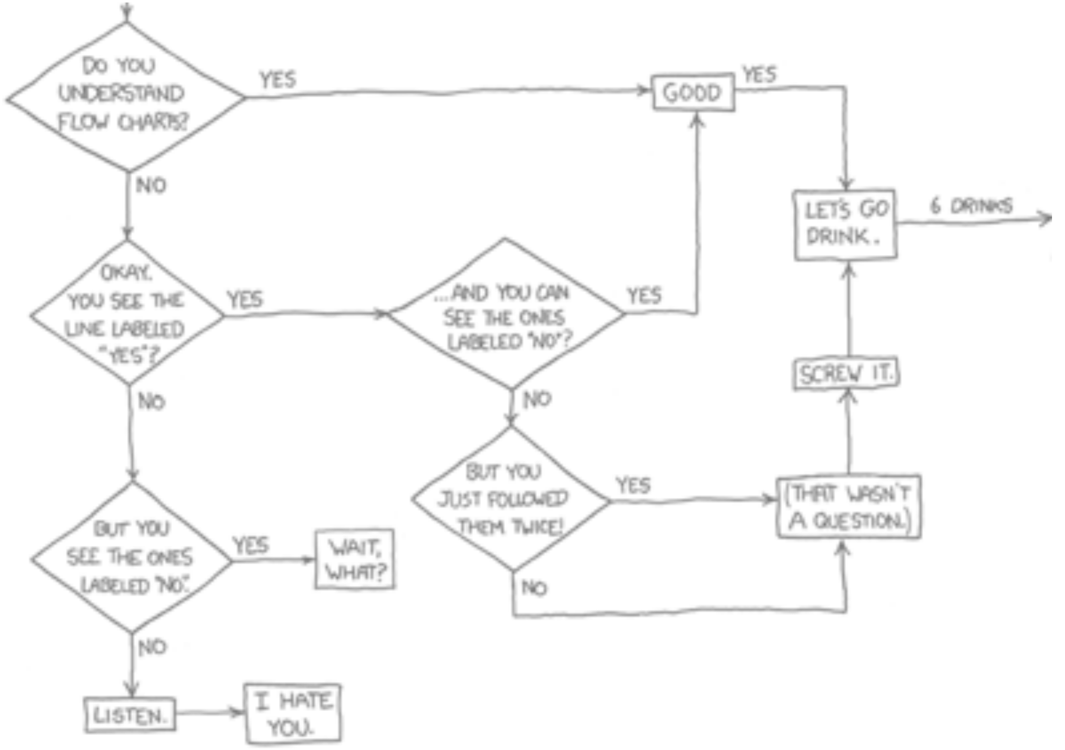
Receiving Input

Remembering

Making Decisions

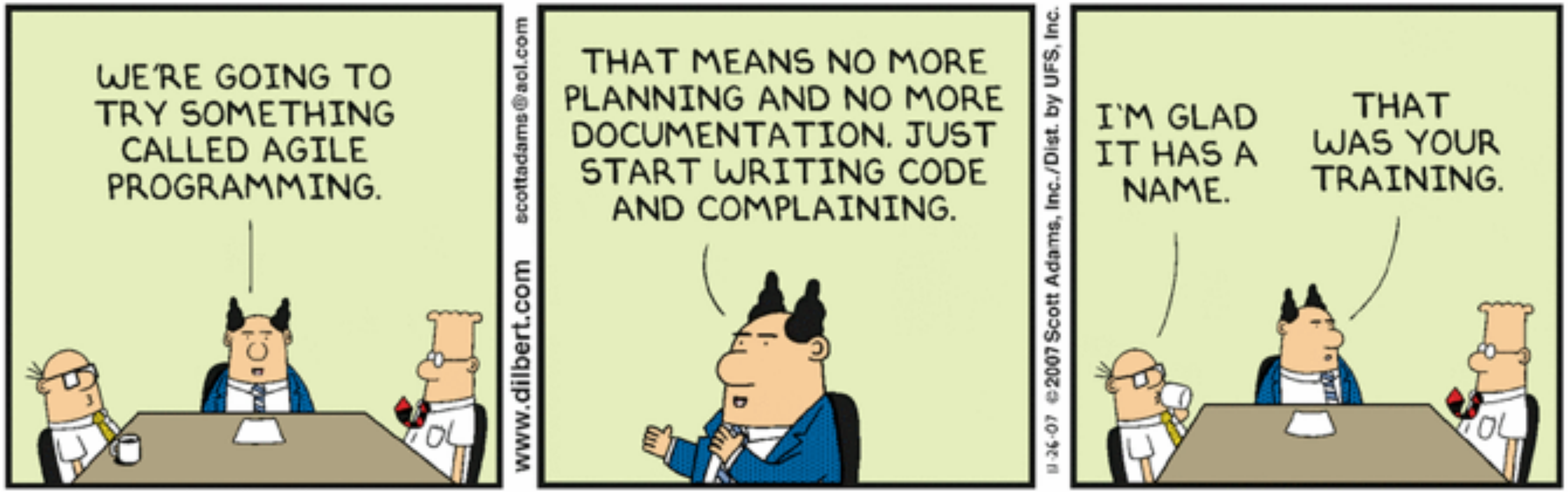
Repeating Processes

Producing Output

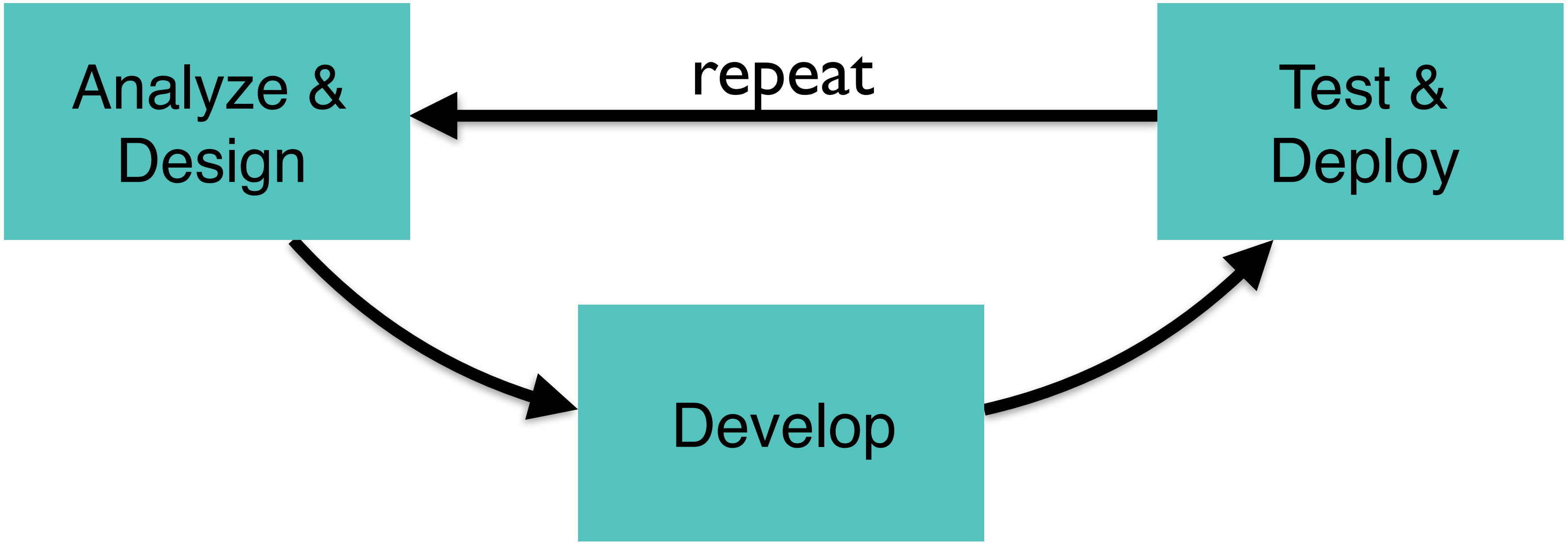

DO YOU UNDERSTAND FLOW CHARTS?
YES
GOOD
YES
LET'S GO DRINK.
6 DRINKS
NO
OKAY. YOU SEE THE LINE LABELED "YES"?
YES
...AND YOU CAN SEE THE ONES LABELED "NO"?
YES
NO
NO
BUT YOU JUST FOLLOWED THEM TWICE!
YES
(THAT WASN'T A QUESTION.)
SCREW IT.
NO
BUT YOU SEE THE ONES LABELED "NO".
YES
WAIT, WHAT?
NO
LISTEN.
I HATE YOU.

THE DEVELOPMENT PROCESS

Agile Development

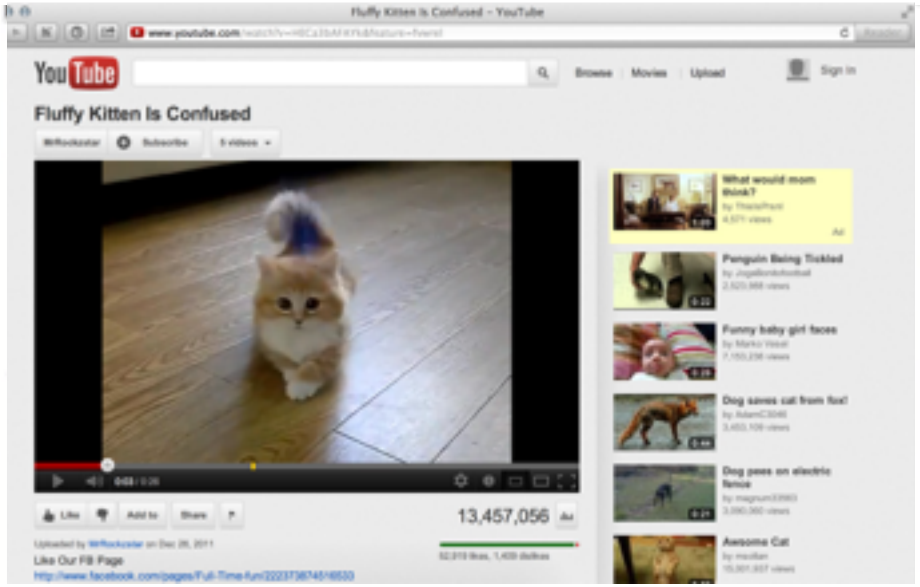

WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.
I'M GLAD IT HAS A NAME.
THAT WAS YOUR TRAINING.
www.dilbert.com scottadams@aol.com
11-26-07 © 2007 Scott Adams, Inc./Dist. by UFS, Inc.

Agile vs. Waterfall



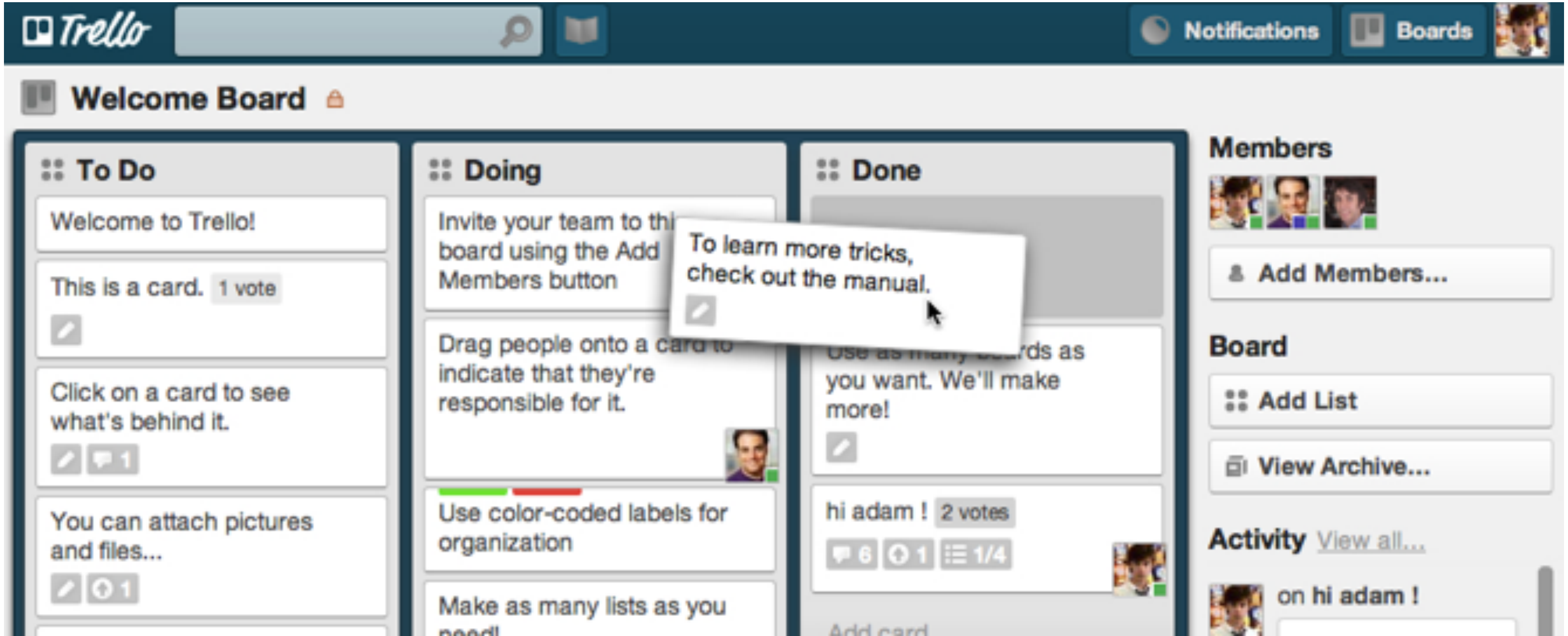
Analyze & Design

- User Research
- Information Architecture
- User Experience Design
- Visual Design



Wireframe vs. Mockup?

Develop



Welcome to Trello!

This is a card. 1 vote

Click on a card to see what's behind it.

You can attach pictures and files...

Invite your team to thi board using the Add Members button

To learn more tricks, check out the manual.

Drag people onto a card to indicate that they're responsible for it.

Use color-coded labels for organization

Make as many lists as you need!

Use as many boards as you want. We'll make more!

hi adam ! 2 votes

on hi adam !

Test & Deploy

- Automated vs. Manual
- Test First & TDD
- Continuous Integration

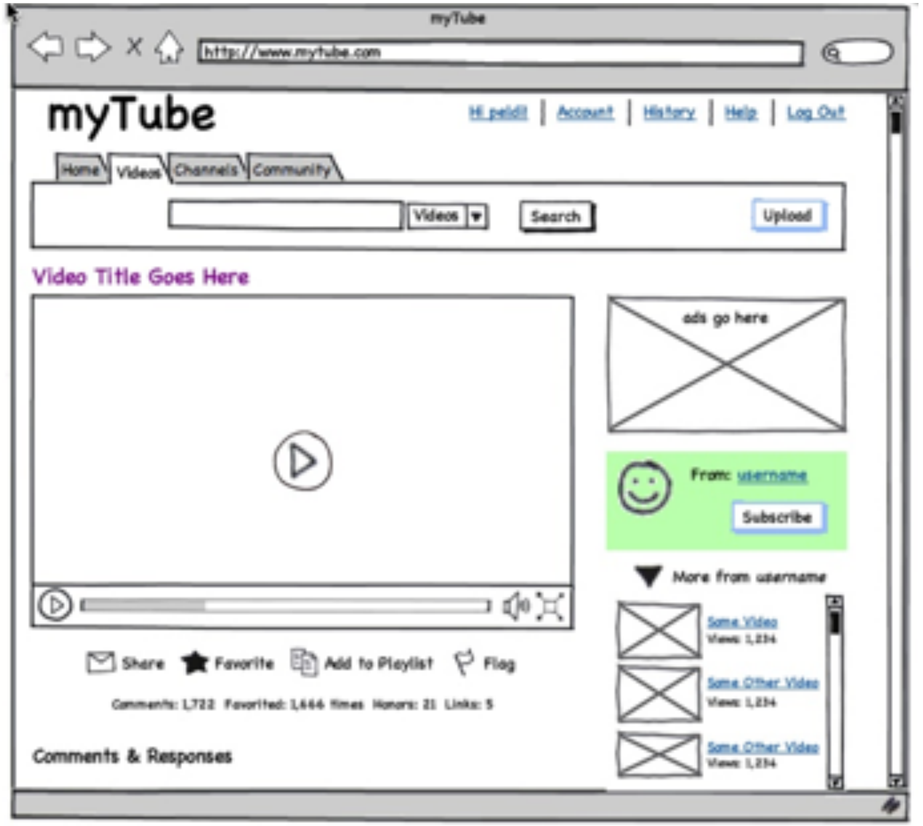
LAB

WEBSITE DESIGN

Website Design Lab

Using wireframe.cc design a 2 page biography website.

- Home Page
  - Person's name
  - Profile Picture
  - Description
  - Key Achievements
- Projects/Publications Page
  - Each: Name, Description, Image & External Link



PROGRAMMING FOR NON PROGRAMMERS

PART 2

BUILDING WEBSITES WITH HTML & CSS

Learning Objectives

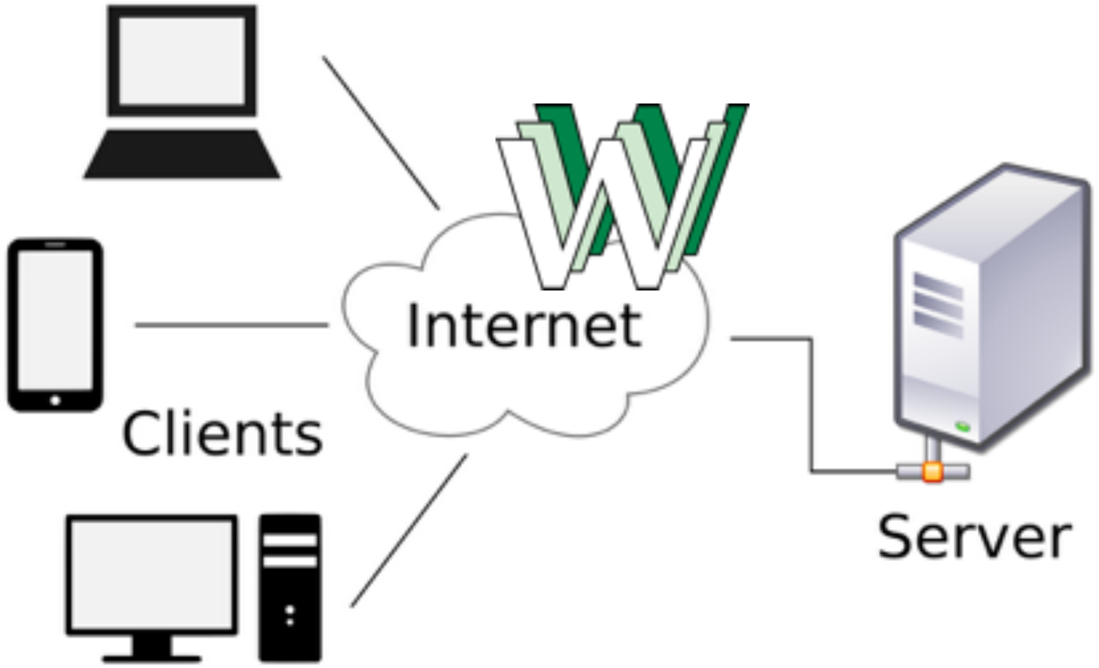
- Explain the purpose of HTML, CSS & JS
- Develop a Basic Website using HTML & CSS

RECAP
PROGRAMMING THE WEB

What is Programming?

Receiving Input	Text, Clicks, Speech, Location, Big Data
Remembering	Data Stores: Variables, Cookies, Files, Databases
Making Decisions	Conditionals: if, else, then
Repeating Processes	Iteration/Loops: for, each, while
Producing Output	Text, Image, Sound, Movement

Client vs. Server



Front-end / Client-side Development

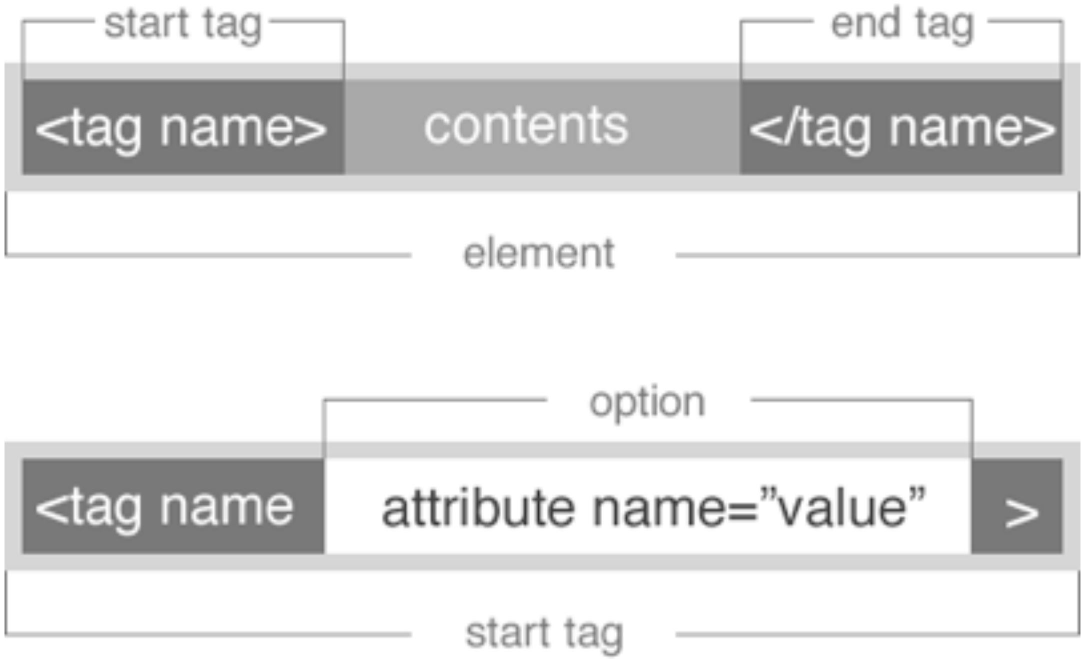
HTML — structure

CSS — look

JS — dynamism

HYPERTEXT MARKUP LANGUAGE

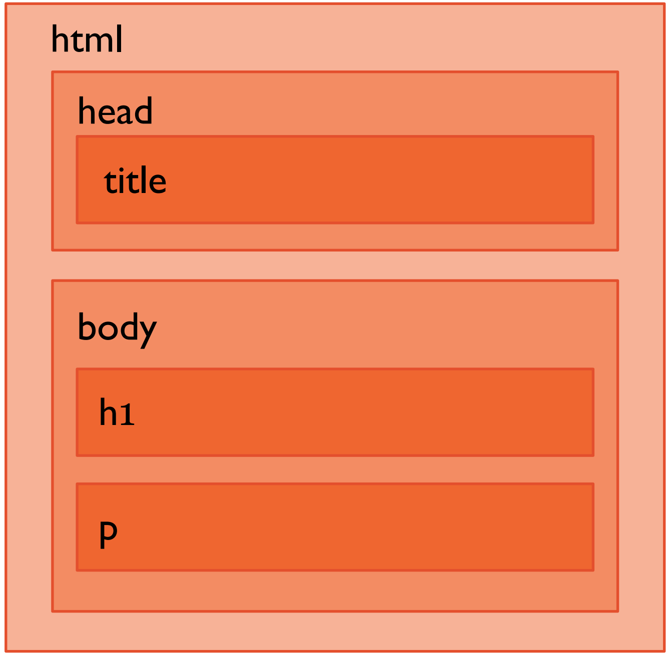
HTML


start tag
end tag
<tag name>
contents
</tag name>
element
option
<tag name
attribute name="value"
>
start tag

Basic Document Structure

```
<html>
    <head>
        Meta Data goes here
    </head>

    <body>
        Document Content goes here
    </body>
</html>
```



Headings

```
<h1>Most Important Title</h1>
<h2>Second Most Important</h2>
<h3>A Less Important Title</h3>
<h4>Even Less Important Title</h4>
<h5>No One Really Uses This Title</h5>
<h6>Title Too Deep, Time To Stop</h6>
```

Text

```
<p>A paragraph of text</p>
<em>Stress this point</em>
<strong>Pay Attention Here</strong>
<hr>
<br>
<pre>Do        not Change
My Formatting browser!</pre>
```

Lists

```
<ol>
	<li>1st item</li>
	<li>2nd item</li>
</ol>
<ul>
	<li>an item</li>
	<li>another item</li>
</ul>
<dl>
	<dt>definition term</dt>
	<dd>description</dd>
</dl>
```

Containers

```
<span>simple grouping</span>
<div>simple container</div>

<header>container for header content</header>
<nav>container for navigation links</nav>
<main>container for main content</main>
<article>Self standing content group</article>
<section>a section of content</section>
<footer>container for footer content</footer>
```

Images

```
<img src="path/to/image.png" alt="image desc">
```

I SPENT ALL WEEK TWEAKING HTML FOR MY INTRANET WEB PAGE. YOU SHOULD SEE IT, MOM.
I CONVERTED THE VIDEO OF MY BIRTH INTO AN MPEG FILE. ANYONE BEHIND THE FIRE WALL CAN VIEW IT.
YOU SHOULD HEAR THE NICKNAME THEY HAVE FOR YOU AT WORK!
YOU SHOULD HEAR THE ONE I HAVE FOR YOU RIGHT NOW.
www.unitedmedia.com
S. Adams
© 1997 United Feature Syndicate, Inc.
2/4/97

Links

```
<a href=“path/to/page.html”>Internal Link</a>

<a href=“http://google.com”>External Link</a>
```

url goes here

display text goes here

LAB

BIOGRAPHY SITE

PART 1: HTML STRUCTURE

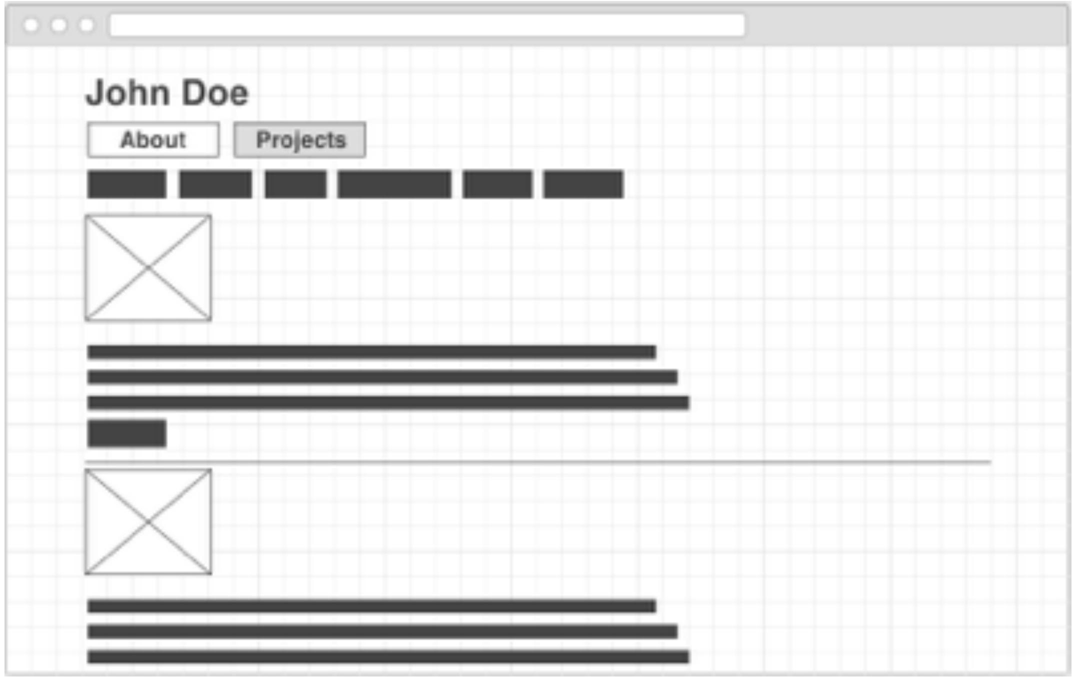
Biography Site Part 1: HTML

Make 2 HTML pages (NO CSS):
index.html & projects.html

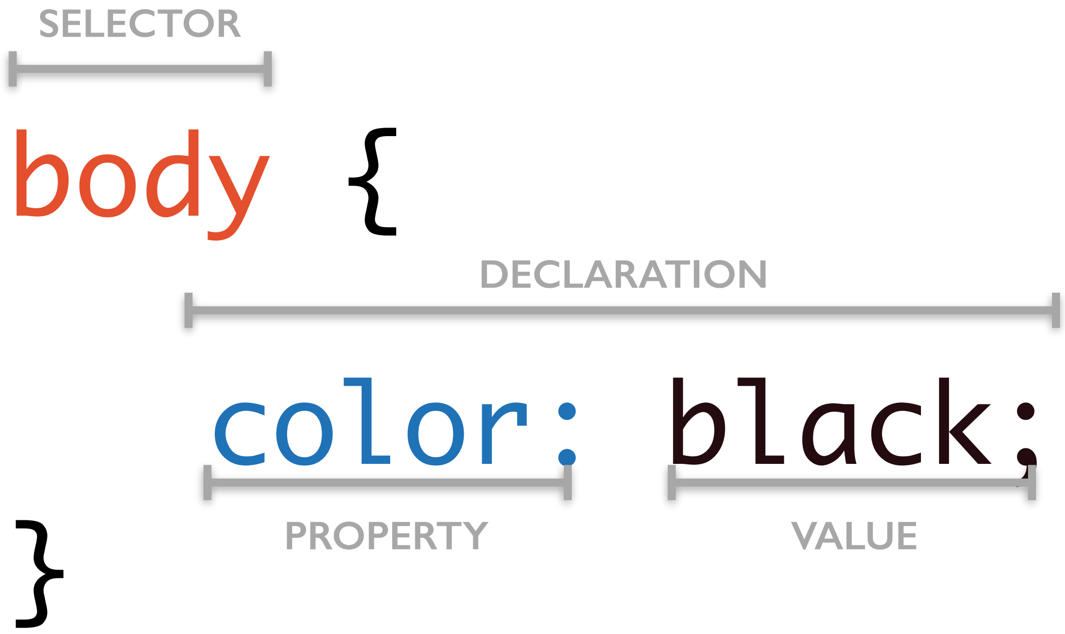
- Home Page
- Person's name
- Profile Picture
- Short Biography
- List of Key Achievements

- Projects/Publications Page
- List: Name, Description, Image & Link For Each

http://www.w3schools.com/tags



CASCADING STYLESHEETS



```
body {
     color: black;
}
```

Linking To A Stylesheet

```
<html>
  <head>
     <link rel=“stylesheet” href=“styles.css”>
  </head>
</html>
```

Selectors: Type

```
body {
    background-color: pink;
}
div {
    border: 1px solid black;
}
```

Selectors: Class

```
<p>Normal text</p>
<p class=“special”>Special text</p>
<p class=“summary special”>Multiple classes</p>
```

```
.special {
    font-weight: bold;
}
```

Selectors: ID

```
<h1 id=“main-title”>Only 1 of These Ever</h1>
```

```
#main-title {
    color: #ff0000;
}
```

Selectors: Advanced

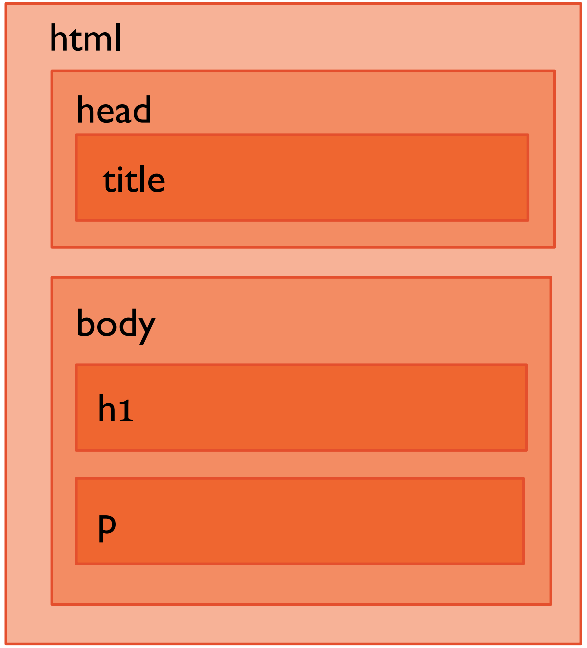
```
<p>First paragraph</p>
<p>Second paragraph</p>
```

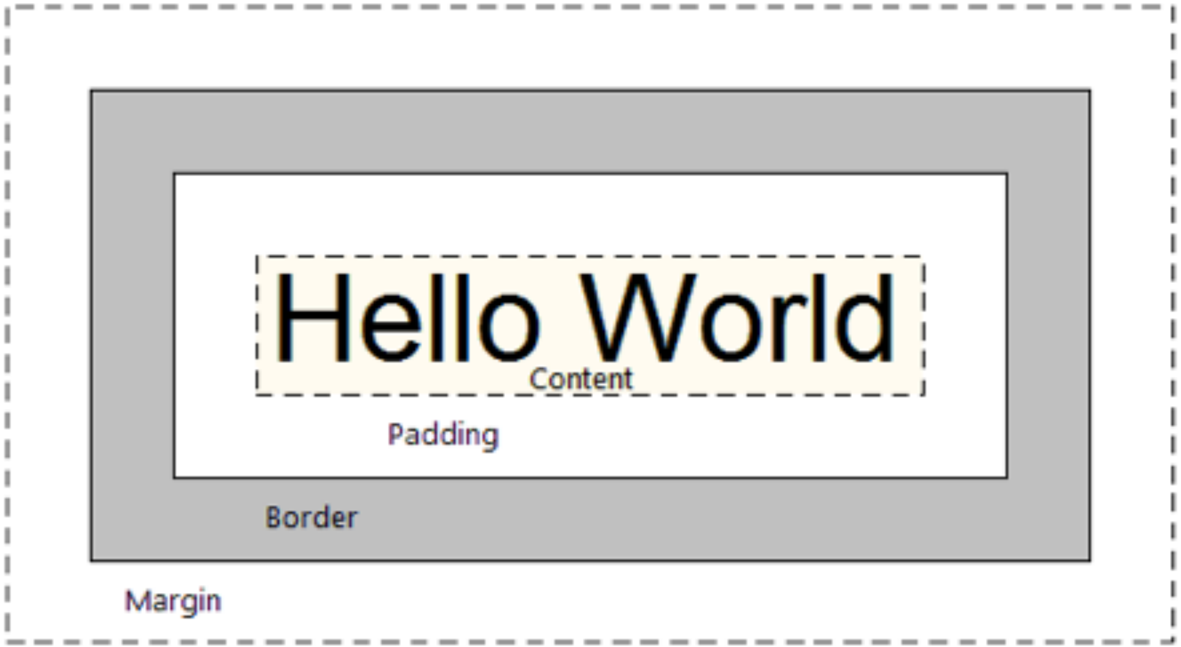
```
p :first-of-type {
    font-weight: bold;
}
p:hover {
    text-decoration: underline;
}
```

Common Properties

```
body {
   border: 1px solid black; /*thickness style color*/
   background-color: pink;
   background-image: url(backdrop.png);
   font: italic 1.5em Georgia;
   padding: 0 5px 5px 0; /*top left right bottom*/
   width: 800px;
}
```

Box Model


html
head
title
body
h1
p


Hello World
Content
Padding
Border
Margin

Positioning

```
.nav {
  display: block; /*inline inline-block flex none*/
  position: static; /*absolute fixed relative*/
  top: 0;
  right: 0;
}
```

LAB

BIOGRAPHY PAGE

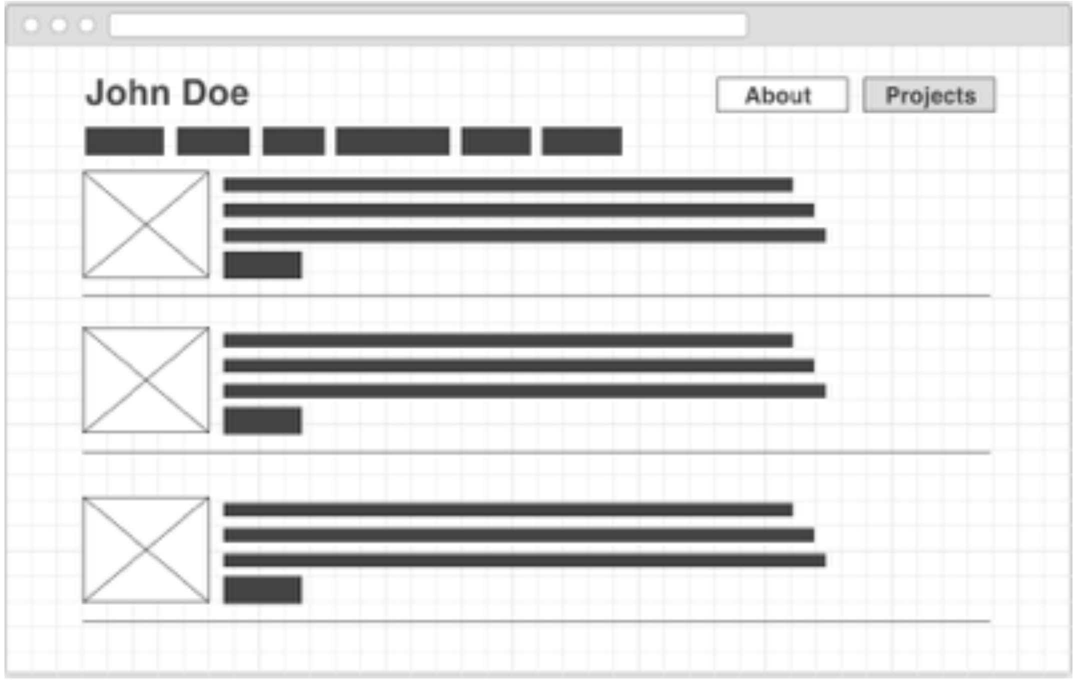
PART 2: CSS DESIGN

Biography Site Part 2: CSS

Make 1 CSS file (styles.css) and import it in your html.
Styles

- Heading
- Paragraphs
- Lists
- Links
- Sections
- Images

http://www.w3schools.com/cssref/



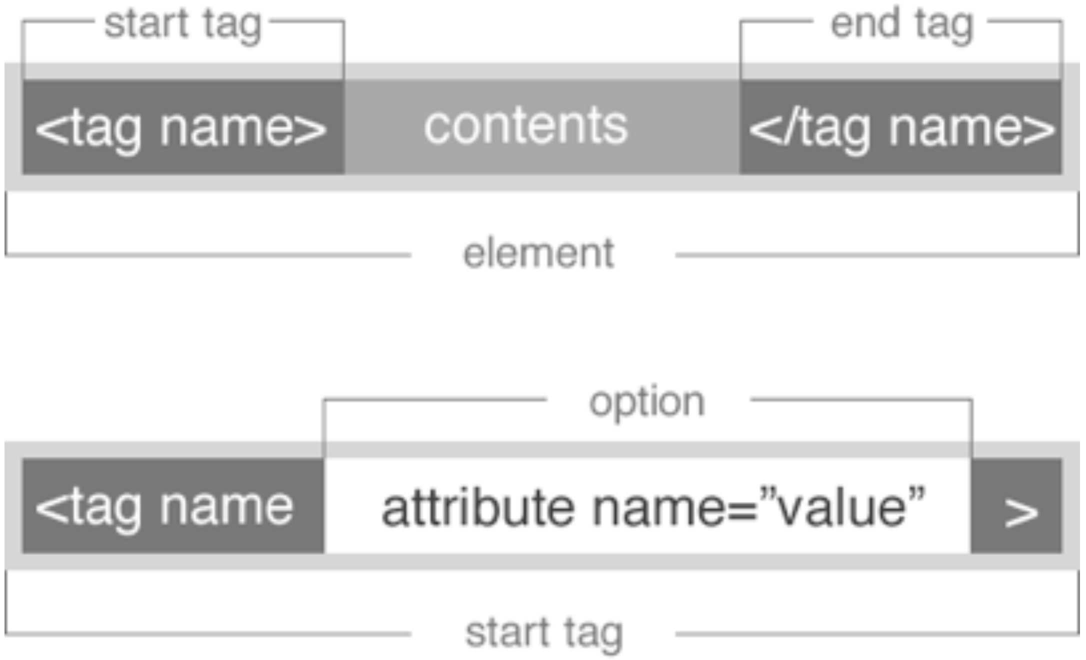
PART 3
ADDING INTERACTIVITY WITH JAVASCRIPT

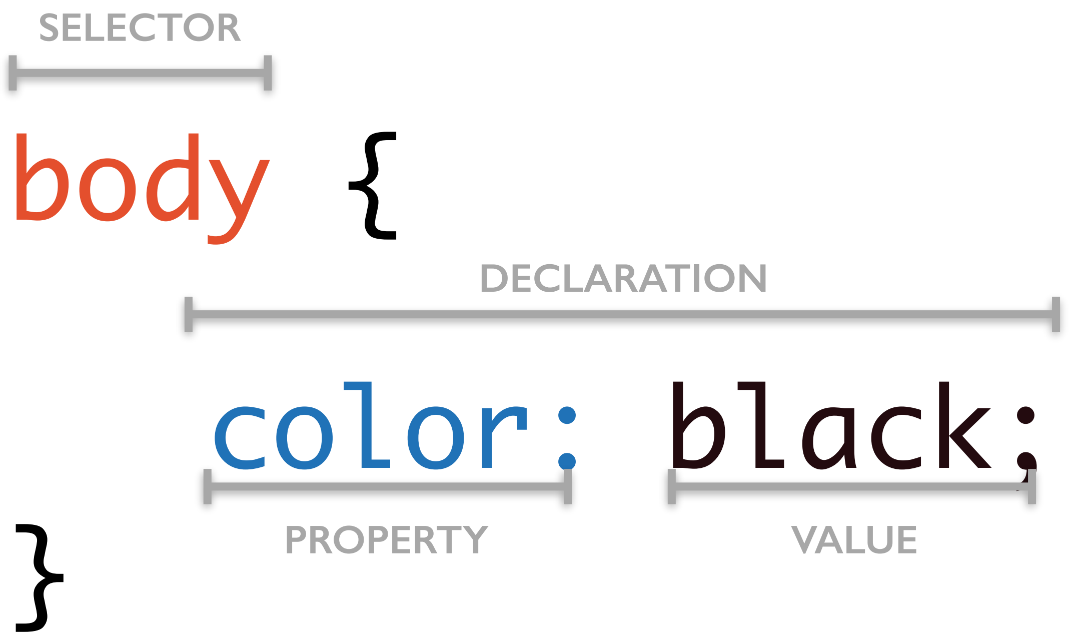
Learning Objectives

- Explain basic JS concepts including Browser Events & DOM manipulation
- Add basic interactivity to a Website using JS

RECAP

BUILDING WEBSITES WITH HTML & CSS


start tag
end tag
<tag name>
contents
</tag name>
element
option
<tag name
attribute name="value"
>
start tag



```
body {
  color: black;
}
```

PROGRAMMING FOR NON PROGRAMMERS

JAVASCRIPT FUNDAMENTALS

Receiving Input	Browser Events
Remembering	Variables
Making Decisions	Conditionals
Repeating Processes	Loops
Producing Output	DOM Manipulation

Including jQuery & Our Javascript

```
<html>
 <body>
  <!-- content goes here -->
  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
  <script src="js/main.js"></script>
 </body>
</html>
```

Receiving Input: Browser Events

```
$('.some-class').on('click', function() {
  console.log('User just clicked')
})

// click, mouseenter, mouseover, mouseleave,
etc..
```

Remembering: Variables

```
var counter = 0

$('.some-class').on('click', function() {
  counter = counter + 1
  console.log('Clicks: ' + counter)
})

// 1, 0.5, true, false, 'hello world'
```

Making Decisions: Conditionals

```
var counter = 0

$('.some-class').on('click', function() {
  counter = counter + 1
  if (counter > 5) {
    counter = 0
  }
  console.log('Clicks: ' + counter)
})
```

LAB

BIOGRAPHY PAGE

PART 3: ADD INPUT

Repeating Processes: Loops

```
var happiness = 5
var loop = 0

console.log('I am')

while (loop < happiness) {
  loop = loop + 1
  console.log('so')
}

console.log('Happy')
```

Processes: Timers

```
setInterval( function() {
  console.log('5 seconds have passed')
}, 5000)
```

Producing Output: DOM Manipulation

```
console.log('simplest form of output')

$('#some-id').hide()
$('#some-id').show()
$(‘.some-div').html('<h1>title</h1>')
$('img').attr('src', 'new-image.jpg')
```

LAB

BIOGRAPHY PAGE

PART 4: ADD OUPUT

PART 4
VCS & DEPLOYMENT WITH GIT & GITHUB

Learning Objectives

- Explain what a VCS is and how it is used in development
- Store files in a .git repository and backup on Github
- Configure Github Pages to serve Website
- (bonus) Configure a custom domain name

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

RECAP

WEB DEVELOPMENT

Front-end / Client-side Development

HTML

CSS

JS

structure

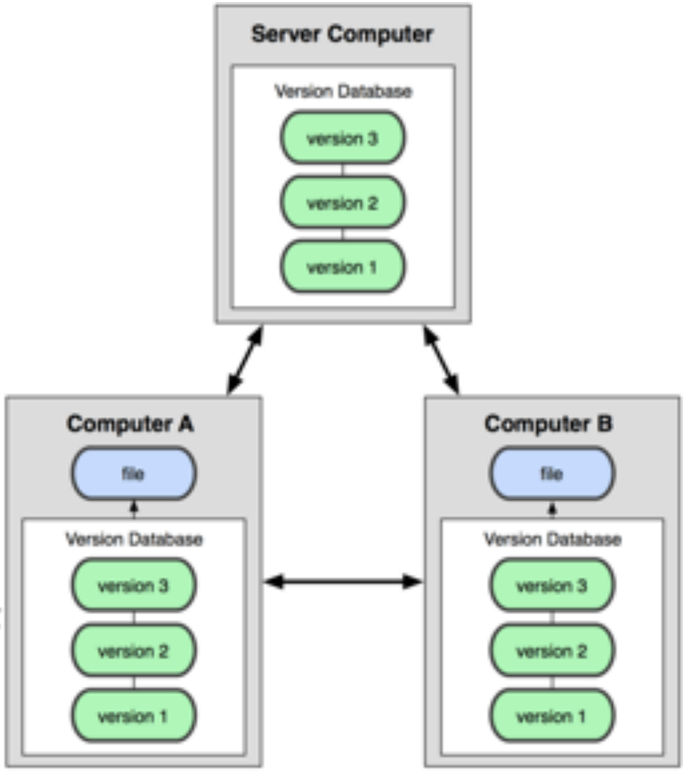
look

dynamism

VERSION CONTROL

What is GIT? Why use it?

- keep track of changes
- collaborate with others
- manage variants and releases
- share on Github



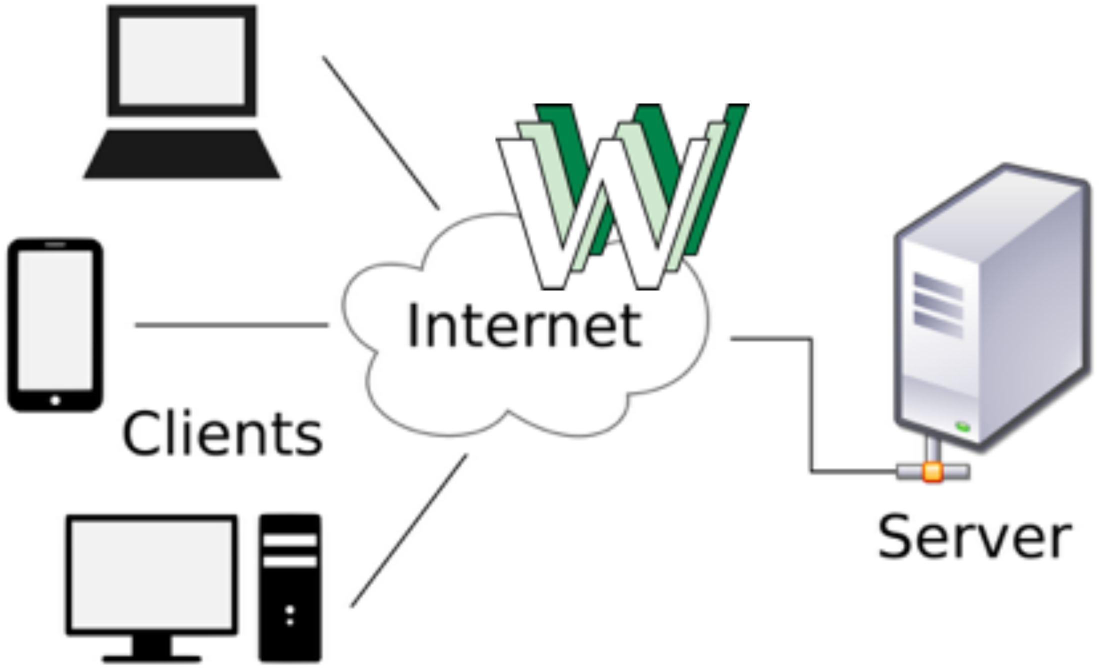
GIT Basics

`git clone [url]`	Create a local copy of Github Repo
`git add [filename]`	Stage file/s, folder/s for next commit
`git commit -m [message]`	Save a commit (i.e a version)
`git push origin master`	Send those changes to Github

http://bit.ly/1LHKj5a

HOSTING OUR WEBSITE

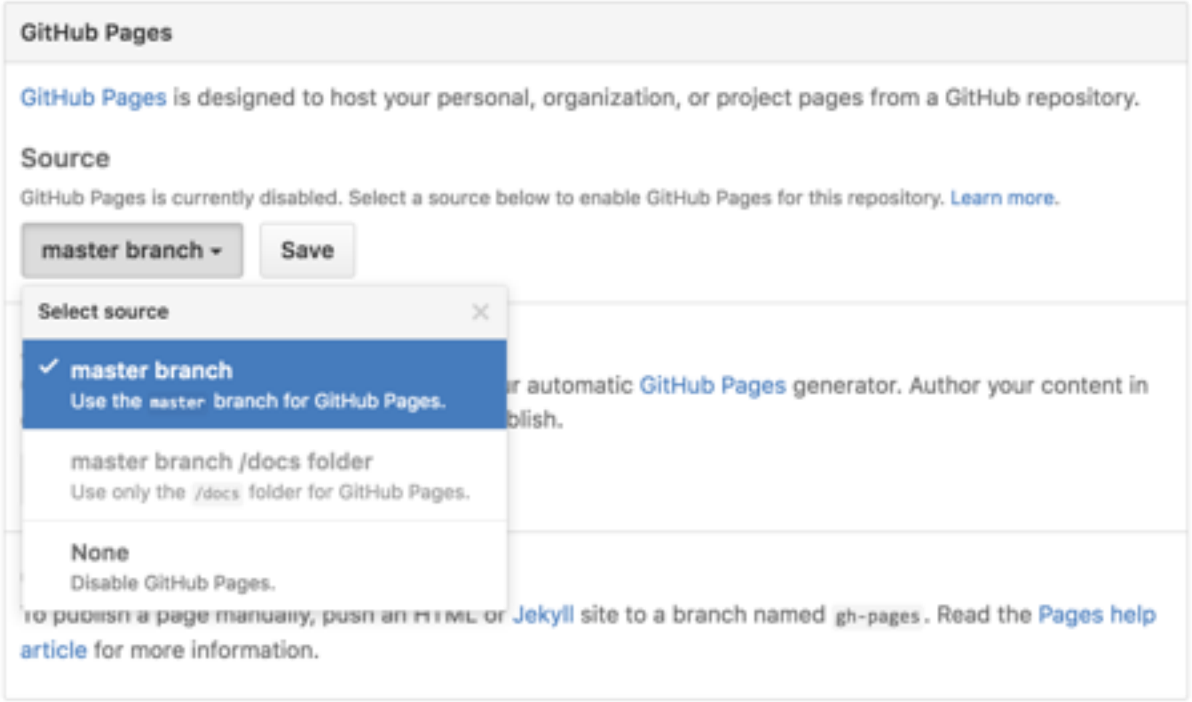
The Internet & The WWW



Github Pages

> Configure Settings to use master branch for Github Pages

view site at: http://username.github.io/repository-name



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. Learn more.

master branch
Save
Select source
✓ master branch
Use the master branch for GitHub Pages.
master branch /docs folder
Use only the /docs folder for GitHub Pages.
None
Disable GitHub Pages.

r automatic GitHub Pages generator. Author your content in

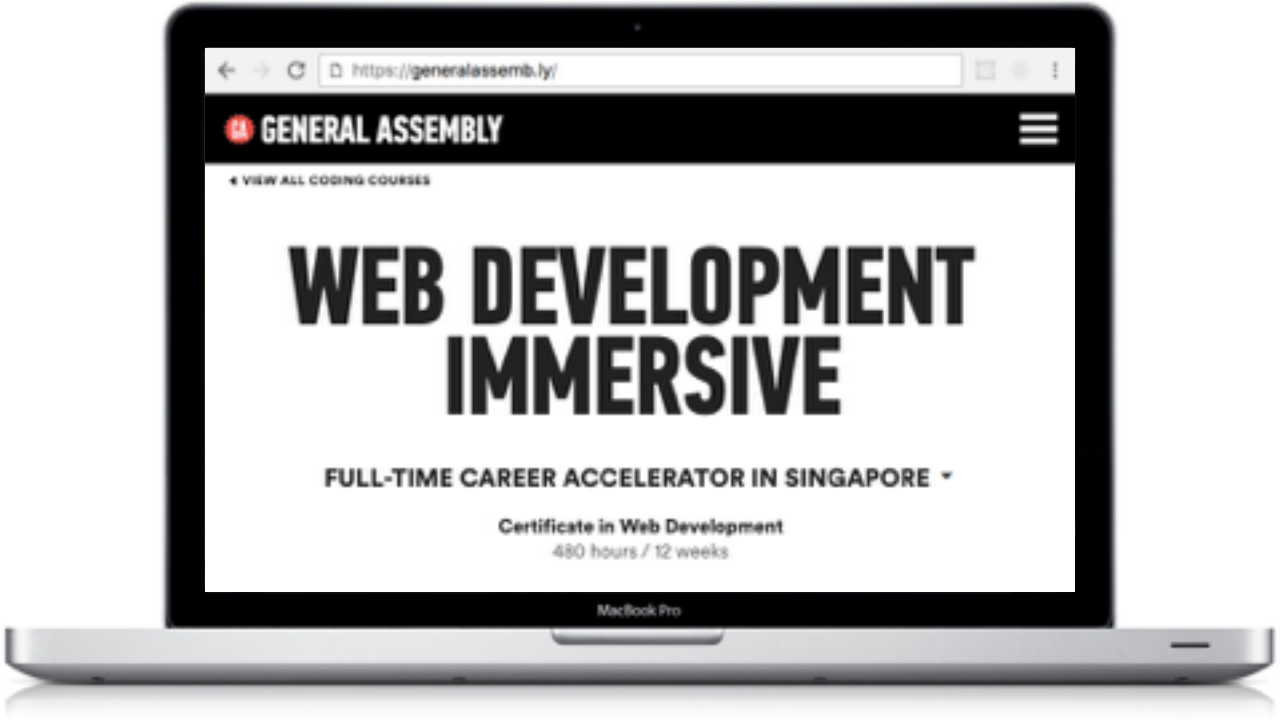
blish.

To publish a page manually, push an HTML or Jekyll site to a branch named gh-pages. Read the Pages help article for more information.

CUSTOM DOMAIN NAMES

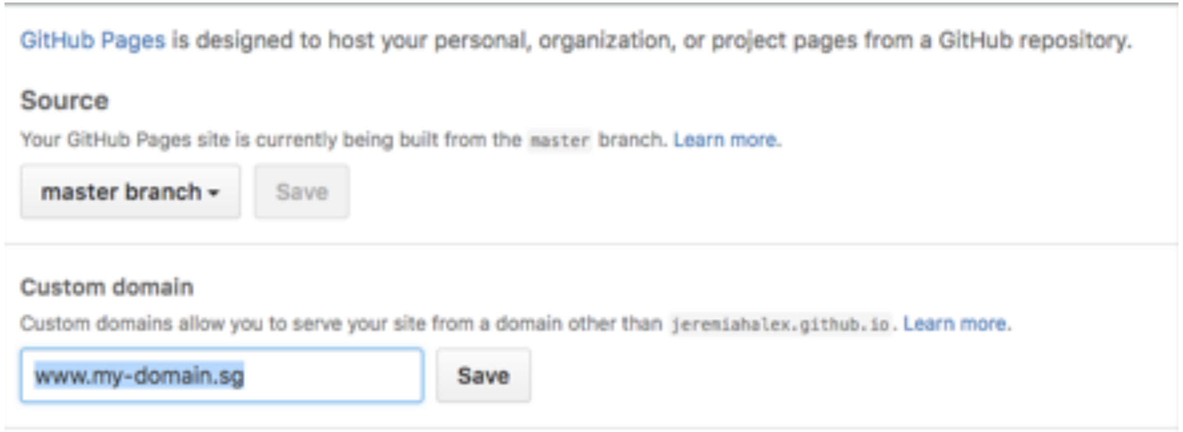
(bonus)

IP Address vs. Domain Name


https://generalassemb.ly/
GENERAL ASSEMBLY
VIEW ALL CODING COURSES
WEB DEVELOPMENT IMMERSIVE
FULL-TIME CAREER ACCELERATOR IN SINGAPORE
Certificate in Web Development
480 hours / 12 weeks
MacBook Pro

Github Pages: Custom Domain

> Configure Settings to use a custom domain



GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source

Your GitHub Pages site is currently being built from the `master` branch. Learn more.

Custom domain

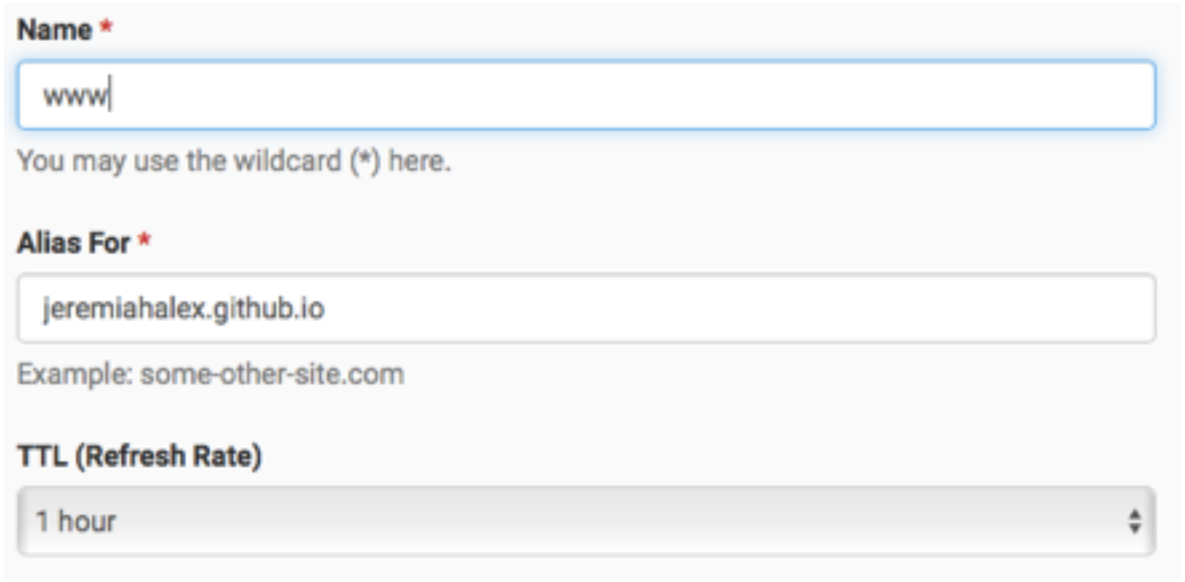
Custom domains allow you to serve your site from a domain other than `jeremiahalex.github.io`. Learn more.

www.my-domain.sg

Configure DNS

> Configure your DNS with a CNAME record that points to your Github.io site:

http://username.github.io



Name *

www

You may use the wildcard (*) here.

Alias For *

jeremiahalex.github.io

Example: some-other-site.com

TTL (Refresh Rate)

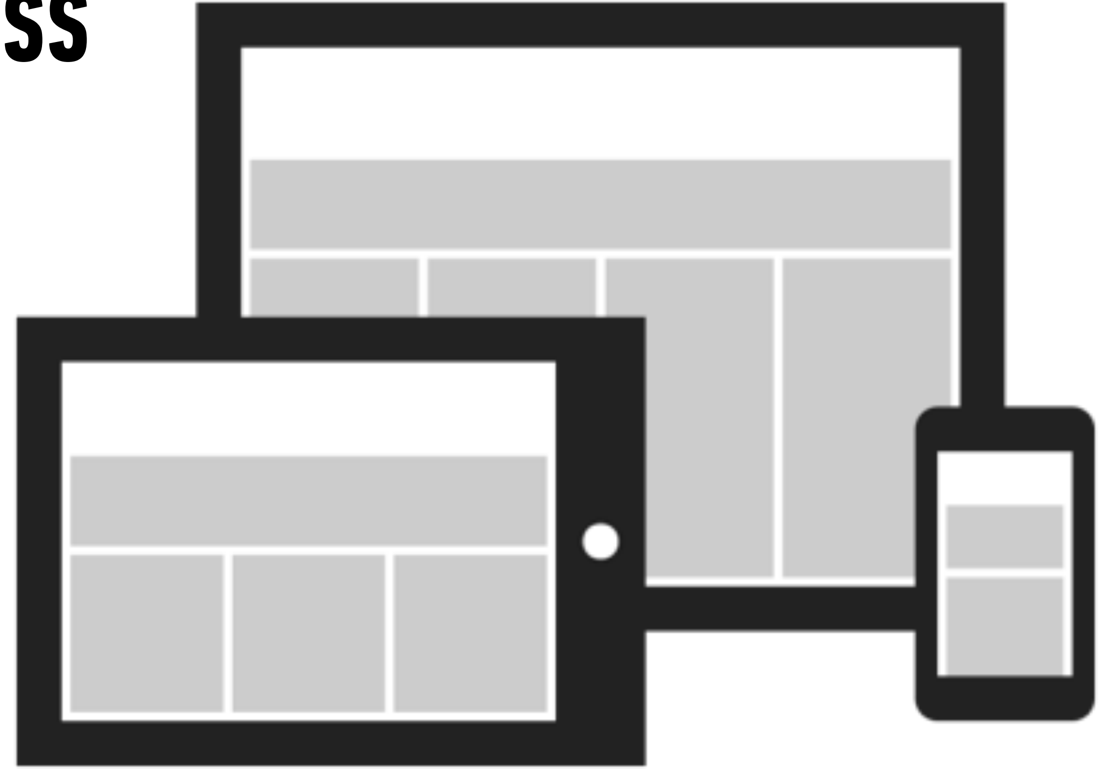
1 hour

PROGRAMMING FOR NON PROGRAMMERS

WHERE TO GO FROM HERE

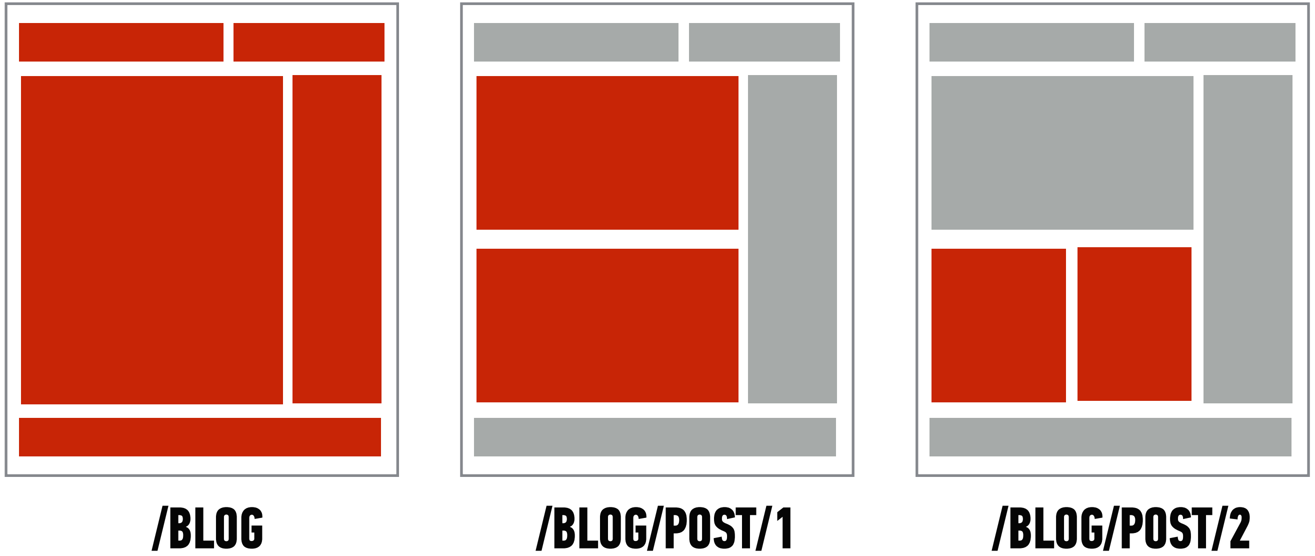
Advanced HTML & CSS

- Responsive Design
- CSS Transitions
- CSS Animations



Advanced Javascript

- AJAX
- Frameworks
- nodeJS
- testing


/BLOG
/BLOG/POST/1
/BLOG/POST/2