8/22/2017

What is client side vs server side?

* Client side is what you would see in everything you do; server side is the background communications that make it run

What is cloud computing?

* When a large network of computers share processes and power

What are client-side languages?

* HTML
* JavaScript
* CSS

What are server-side languages?

* Java, Pyhton
* C++, C#(C sharp)
* Ruby (ruby on rails is ruby on framework)
* PHP
* JavaScript in some cases

What are not programing languages?

* jQuery
* Ruby on Rails
* Wordpress
* BootStrap (front end framework)

When starting a project, what should we consider when deciding on a language?

* Project Functionality
* Budget + Legacy Code
* Security
* Client Needs
* Experience and proficiency in a language
* System optimization and compatibility

Basic development tools

* To build a website you need things
* You need a computer
* Internet (not 100% Necessary)
* Text editor
* Webserver (also not necessary however will help a LOT)
* Text compiler (runs the code you write)
* *DON’T USE - WYSIWYG (what you see is what you get) they’re bad, create broken code and don’t always work correctly (such as Dreamweaver)*
* A team helps
* Graphics Editor (creates something you need to code such as photoshop)
* FTP(File Transfer Protocol) Program, transfers files from server to server or local to live, ect.
* Command Line, pushes code to github, you can get programs through it, Terminal allows you to access your computer completely

Text Editors

* Atom\* (recommended)
* Sublime
* TextEdit
* Notepad++
* *CANT USE MICROSOFT WORD BECAUSE IT’S A WORD PROCESSOR NOT A TEXT EDITOR*

IDEs

* Integrated Development Environment – consists of a text editor, code compiler and debugger
* Visual Studio
* IntelliJ
* Eclipse
* Netbeans

Web Broswers

* Firefox
* Chrome
* Safari
* IE
* Microsoft Edge
* Opera
* Tor
* Chromium (beta chrome)
* Caniuse.com will help with showing what browers support code properties in CSS and HTML

Graphics Editors

* GIMP
* Photoshop

FTP Program

* Cyberduck
* Filezilla
* transfers files from server to server or local to live, ect.
* Some webhosts require SFTP instead of FTP

Version Control GUI (Graphics User Interface)

* SourceTree (PP Preferred)
* Kraken
* GitHub
* Creates an interface to use to keep an eye on version control

Control Line

* Mac uses terminal to visualize the command line
* Terminal controls the OS – Meaning if you mess something up it could really get messed up

Front End Frameworks

* Framework is an easier way to write a language with shortcuts and other helpful resources
* Helps standardize a process, used as a base
* Bootstrap
* Semantic-UI
* Foundation
* Materialize
* Material UI
* Pure
* Skeleton
* UIKit
* When they are no longer supported it can create nightmares
* They help keep things consistent and have component libraries

Documentation

* Usually instructions noted in the code
* Documentation can help out a lot, however it can also be a pain if it isn’t udated
* Good documentation should walk you through every process needed to be done to get the program working

Content Management System (CMS)

* CMS is a content management system, allows a non-technical user to edit the content of their website
* Some are easier to use than others
* Drupal
* Wordpress (easy to start with, simple to use)
* Joomla (may be typically used in education, PP isn’t sure)
* SquareSpace (No budget, can be done themselves)

What was the initial intention of websites?

* Military
* Businesses
* Communications
* It was made to share research papers
* Internet in the 80’s was all text base, white screen with text

What makes up a website?

* UI (user interface)
* Code
* Content

HTML

* HyperText Markup Language
* Text file containing tags that differentiate content based on hierarchy layout and structure
* Tags tell the browser how to display a webpage
* Basically an outline

Exploring HTML

* Open up any webpage
* Right click > View Page Source
* Right click > Inspect

History of HTML

* First version was created in the 80s by Berners-Lee.
  + The purpose? To share simple research among colleagues on the newfangled internet.
* The first “official” edition was published in ’93. Until then, we all hacked our way around
* It was simple = beginner friendly
* This also created the World Wide Web (w3) consortium
  + W3 schools
* In 2014, html\_ was standardized and recommended for general use

Anatomy of an HTML element

* <header> (start)
* </header> (end)
* Each element has a start & end tag, often content in between and perhaps a few attributes to make it special. Tags can be seen as commands, instructing the browser to display a particular element, or close it and look for the next

HTML Basics

* Add the following elements and their corresponding closing elements to your index.html file
* <!DOCTYPE html>, <html>, <head>, <body>
* doctype doesn’t have a close, it tells the browser what kind of file to read

Head vs. Body Content

* The browser reads tags and content in the head but displays the content within the body tags

Meta Content

* Meta elements are often used to define page descriptions, keywords, and other metadata
* Metadata is often used by search engines (keywords and SEO), browsers (how and when to display content or reload a page), and other web services
* Add <meta charset=”utf-8”> and a title tag to the head element <title> title here </title>

Body

* The body of HTML includes all the content and images that you will see on your website
* Whatever you include in the body will be on the page
* Take your notes from the history of HTML and add them to the <body> tag.
* Browser doesn’t recognize breaks in coding, it reads it as one long line

Headings

* <h1> <h2> <h3> <h4> <h5> <h6> designate levels of hierarchy within headings
* think of headings like an outline

Paragraphs

* <p> designates a paragraph
* put all text laying around in here, all text needs a tag

Line Breaks

* <br>
  + doesn’t need to be closed

Lists

* <ul> (bullets) vs <ol>(numbers)
* <li>

HOMEWORK

1. research the history of HTML another program language OR how the internet works
2. A Subject of your chosing (or I will sugguest one for you) where technology and another industry overlap

BOTH MINIPAPERS SHOULD BE BETWEEN 300 AND 400 WORDS EACH AND CONSIST OF A FEW HEADINGS FOLLOWED BY A FEW PARAGRAPHS

8/23/2017

Let’s review

* HTML
* Development Tools
* Clients and Search engines

Why was HTML created? And when?

* It was created for communication between people and universities, in the 1980’s (1989)
* It was standardized by W3
* Some basic tags we learned were headings, line breaks, paragraphs, lists (OL, and UL), and character set
* Very top of HTML document is just <head>
* All content belongs in the <body> section
* <p> is paragraph

Structural vs. Semantic Markup

* headings and paragraph tags add structure to the document
* bolding, block quotes, and citations, acronym meanings are all Semantic Markup

Coursework

* Chose ONE of your essays and create a NEW HTML document in a NEW folder on your desktop
* Using yesterday’s classwork as a guide, and your peers, begin to layout your new HTML file.
* You should have 2-3 headlines, and paragraph tags wrapping any content not in a headline. Remember, headlines do NOT require paragraph tags.
* If you do not include a list in your essay, create one for fun and use what you learned.

Semantic Markup

* Will provide extra information, adding quotations, citations, bolding
* Doesn’t effect structure in any way just helps portray things differently
* Bold and italic are the most popular ways
* <strong> this is a sentence of importance. </strong> (**BOLD**)
* <em> this should be emphasized. </em> (*ITALICS*)
* <b></b> is bold <i></i> is italics, however it is not the best practice

Super and Subscript

* <sup> and <sub>
* <sup> puts number above, <sub> puts it below
* Superscript contains characters that should be suffixes like raising something to the something power in math, or like H2O, the 2 should be subscript = H<sub>2</sub>O

Blockquotes

* <blockquote> </blockquote>
* It tells the browser that there is a quote, could auto indent (PP isn’t sure)

Other Semantic Markup

* <abbr> abbreviation
* <acronym> tells browser there is an acronym
* <cite> for citing sources
* <dfn> definition
* <address> address

Links!

* Links are important, you can do a lot with them to get around the internet

Writing Links

* <a href=”link here”> description here</a>
* theres is absolute links and relative
* absolute take a user to a website, contains the entire URL
* relative URL is a link that will take you somewhere to the page you’re on

More URL anatomy

* <a href=”https:google.com” target=”\_blank”title=”Search the web with google!”> </a>

Email Links

* <a href=”<mailto:piccari@stevenscollege.edu>” title=”Email Bri Piccari”> Email Me! </a>

Linking to a specific part of the same page

* <a href=”#about”> About Us! </a>
* <div id=”about”> how the browser knows where to go, will get further into it later
* # means ID
* <a href=”/about#about”> some text that’s a link that you click on </a>

Coursework

* In your HTML file, add an ordered or unordered list of links with the sources you’ve used to research your topic.
* Also include a link to contact the author (YOU!) via email

Images

* Image tag is a rule breaker
* <img src=”images/fern.png” alt=”” height=”” width=””>

Important Considerations

* File Size
* File Type
* Intended visual aesthetic (transparent v. static v. colors v. animated)
* File compression
* File quality

Layout: Images and Content

* Image next to content is “In line”
* Image on top of content is called “block” they are not together
* To have it show up as block write <img----> then <p> content </p>
* To have it show up in line write <p><img src=””> content </p>

Figure and Figure Caption

* <figure> wraps an image and its caption.
* Captions should be wrapped in <figcaption>
* <figure>
  + <img src=”/”>
  + <figcaption> insert caption here </figcaption>
* </figure>

Homework

* Apply what we’ve learned today to your second essay.
* Include headings,images, and paragraph tags to your website
  + Don’t forget to include a list of links to your sources, as well as the ability to email the author (again, you.)
  + Can you figure out how to preset a subject line?
  + Upload this project you’ve been working on in class by midnight, Monday, August 28th to the PORTAL. Make sure you are in the web development 1 course space.
  + If you have a folder be sure to compress it.

8/24/2017

What is the directory structure and why does it matter?

* It organizes files for you, the biggest reason it matters is that if you can get in the habit now to do it, it will make your life easier in the future.

What tags did we go over?

* <a>
* <sub><sup>
* <figure> <figcaption>
* emphasis
* email link tags

What are some text tags?

* <p>
* <ul ol>

What are some semantic text tags?

* Dfn
* Address
* Abbreviation
* Site
* Block quote
* Acronym

GET SOME BROWSERS AND TEXT EDITORS WITH ADMIN ACCESS

What is inline vs. block

* Block, image and content is separate in a new line
* Inline image and content become in line with eachother

What is important to know about copy and pasting into your code?

* Format issues that cause errors

What is Project Management?

* The project manager interfaces with the client more than you do
* They oversee everything
* They set launch dates
* They understand technology, devotement, and that things go wrong
* A good one will advocate for team while keeping client’s needs in mind
* They will QA your code (quality assurance)
* Project management is planning, monitoring, implementation, and control of a projects path and the outcome
* Project manager sets goals and makes sure they can do most of those goals, if not all of them “which is awesome, but like, not so awesome” – PP
* The scope is the outline of the project, helps keep things on budget, on time “and like keeping people happy it saves peoples butts” – PP
* 4 P’s
  + Planning
  + Process
  + People – best and worst part
  + Power

Software project management

* SP managers Have a technical background, interested in everything tech
* They are the middle man between development team and client

Internships

* Go to the Lancaster tech community for internships, that’s how PP got her job

Software development process decisions

* It focuses on the production of your project over what technical tools your using.
* Technology is constantly evolving, If you don’t have new tools (which are constantly changing) you will have no change in your project.
* The process should almost always be the same, even though the tools are different
* Process exists to support the management of development as well as the business’s goals
* Some things to consider when determining what processes to chose
  + People to work with
  + Conflict management and resolution
    - “I have seen people rip out lines of code because they didn’t write it” – PP
  + Risk and opportunity management
  + Requirement management
  + Change management
  + Software configuration management
  + Release management

The six fazes (Software development life cycle)

* Requirement gathering and analysis
  + Teams talk to the client and make a plan and ask why they’re doing it
  + The client and team will work together and come up with requirements such as technical requirements or user based requirements
  + This is where you determine if you can meet the requirements, or if you should even take the project
  + THERES ALWAYS FACTORS
  + Functionality is key
  + Have to figure out if all skillsets are met
* Software design
  + After you have gathered requirements you need to have a thing to make!
  + Developers, information architects designers, will figure out what info needs to exist and where
    - The visual design of it
    - Architectural designs of it
    - User flow
  + All of that goes through revisions until perfected (or at least close enough)
* Coding and implementation
  + Going to take time, you have to learn languages and adjust
* Testing
  + Tests for bugs, makes sure code goes through QA standards
  + Does automated testing (run one code against another to make sure it works in other environments)
  + All of that done to make sure it doesn’t fail
  + In some teams testing and coding is done at the same time (which is called test driven development)
* Deployment
  + Website and product go live
* Maintenance and post deployment
  + MUST launch at an optimal time where no one is using it
  + Stuff will always go wrong
  + You make a living off of solving problems
  + User errors are a thing, so that’s where post deployment comes in
    - Check analytics, user reports, and feedback
  + You get to fix bugs and fix other issues
  + This time is used for brainstorming for the next project
  + Once it is launched you get a break
  + It then repeats, you work in a circle

Just make sure you upload your websites to portal

8/28/2017

Review:

* 5 things we learned last week
  + What HTML stood for
  + How to code basic HTML
  + What HTML tags do
  + How to set an image as the background in HTML
  + How to fully create a fully functional HTML document

Server side languages

* Java
* PHP
* Python
* C++
* C#
* Ruby
* JavaScript (full stack language)

Front End Languages

* CSS (cascading style sheets)
* HTML
* JavaScript

What is a full stack developer?

* Typically, pretty knowledgeable on the full front and back end stack, can write most languages but is really good at one usually.
* Most PP has worked with are really good at JavaScript

What is semantic vs structural makeup?

* Semantic reinforces meaning, uses emphasis and strong, not all comes through visually.

What are some tags we’ve learned so far?

* <p>
* <body>
* <head>
* <h1>-<h6>
* <br>
* <strong>
* <img>
* <ol><ul>

How many phases in the software development life cycle?

* 6 phases are typical

List some phases

* Requirements
* Software design
* Coding and implementation
* Testing
* Deployment
* Maintenance and post deployment

Which phase is the longest?

* Coding and implementation, maintenance could take longer however

What is project management?

* Planning monitoring and implementing a projects path
* The 4 Ps
  + Planning
  + Process
  + People
  + Power

What do we have to be careful with when copying and pasting code?

* Quotes auto format, have to fix it

Presentation Dates for me

* September 29th
* November th

We have some kind of ETS testing on the 8th and PP has no idea what it is

Review

* What tools did we go over?
  + FTP
  + Text editors
  + Graphic editors
  + Web Browsers
  + IDs
  + Command line
  + Version control

Additional Tools

* Inspect Element
* Web based code editors
  + Codepen.io
  + Jsfiddle.net
  + Jsbin.com
* Tutorial Sites
  + Kaan academy
  + Code academy
  + Lynda (paid)
* StackOverflow
  + QNA website, it helps a ton

Project Management Tools

* Jira (most common)
* BaseCamp

The Parking Lot

* Have posted notes and sharpies set up in a corner and have the client write down every though they have and that makes it easier for PP to answer questions

Why did you choose CSET? What interests you about it?

* DISCUSSION –

Project Workflow Methodologies

* What is a project workflow? Why does it matter?
  + The way you get through your project
  + A software project can have a lot of moving pieces; a project workflow can set the tone if new people have to join in or if anything else happens
* Why can’t we jump into coding?
  + You would only be working on one small piece of a giant project, everyone has an assignment to work on at all times.

Coursework

* On Thursday, September 7th, you will present research on a methodology or workflow, as assigned, in a group

Methodologies

* Agile
* Waterfall – Bobby, Jessie, Brandon
* Scrum – Zach, Cameron, Jerry
* KanBan – Collin C, Anthony, Eric
* Extreme Programming – Caleb, Collin A, Rafael
* Lean Software – Cole, Puspa, Shane, Jeffery
* Spiral – Jeffery, Edgar, Jose
* Dynamic System – Darvian, Chris, Justin
* Feature Driven – Stefan, David, Danny
  + ^most common

Agile

* Main framework for how development teams work, we should be familiar with it
* The term “agile” is used to describe a number of iterative software development methodologies
* Some of the most popular methodologies include Extreme Programming (XP), Scrum, Crystal, Dynamic Systems Development Method (DSDM), Lean Development, and Feature-Driven Development (FDD)
* To be agile is to have “the ability to create and respond to change in order to succeed in an uncertain turbulent environment.”

Agile Values

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer Collaboration Over Contract Negotiation
4. Responding to Change Over Following a Plan

Agile Principles

* *We follow these principles:*
  + Our highest priority is to satisfy the customer through early and continuous delivery  
    of valuable software.
  + Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  + Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  + Business people and developers must work together daily throughout the project.
  + Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  + The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  + Working software is the primary measure of progress.
  + Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  + Continuous attention to technical excellence and good design enhances agility.
  + Simplicity--the art of maximizing the amount of work not done--is essential.
  + The best architectures, requirements, and designs emerge from self-organizing teams.
  + At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile in practice

* The intention of agile is to align development with business needs, and the success of agile is apparent. Agile projects are customer focused and encouraged customer guidance.

Agile Resources

* <https://www.atlassian.com/agile>
* <https://www.agilealliance.org/>

Reminders

* Off on Monday for Labor Day.
* TechTalk topics due tomorrow by 11:50pm EST – first four will be on Friday.
* Web Development homework is due tonight at 11:50pm EST – I will not take late submissions.
* Workflow Methodology presentations are 9/7.

8/29/2017

Tables and Layout

* Tables
  + Still used but isn’t what websites are made of completely anymore
  + <table>
    - <tr> - table row
      * <td> - table data
      * </td>
    - </tr>
    - <tr>
      * <td>
      * </td>
      * <td>
      * </td>
    - </tr>
  + </table>
* Table headings
  + Can represent the heading for column or row
    - <th scope=”col”>content</th> - for colums heading
    - <th scope”row”>content</th> - for row headings
* Spanning columns
  + <td colspan=”2”>content</td> makes content span 2 columns

Commenting

* It is a way to leave comments that the browser wont read, used for other developers
  + You can explain how you fixed or did something for another development
  + You could leave to-do lists
  + You can use them to remind yourself on something so you don’t forget
* Comments are written <!-- comment -->
  + You can highlight and “command + slash” to also comment something out

Divs sections & containing elements

* <div> </div>
  + contains information
  + they help us create content
  + helps us style things easier
  + <div>
  + <h1> content </h1>
  + <p> content </p>
  + </div>
* <span></span>
  + display content in line
  + used to target something very specific
  + divs used more often
* Let’s group some content, using the pages we made last week.
  + Add in span and div tags to group content, images and more.

HTML 4 vs HTML 5

* HTML 4
  + <div>
  + <span>
* HTML 5
  + <article>
  + <aside>
  + <figure>
  + <footer>
  + <header>
  + <main>
  + <nav>
  + <section>

HTML 5 Containing Elements

* Article
  + Article refers to any content that could stand alone.
  + Articles can be used more than once, depending upon the content being presented and may even be nested within one another.
  + Typically used more than once, can be nested in one another
  + Used in blog notes or news article tags
  + <article>
  + <h1>Headline</h1>
  + <p>Content</p>
  + </article>
* Aside
  + Aside refers to either content related to an article if used within an article tag, or refers to content related to the entire page, if used outside or without an article tag.
  + Asides are typically used once on a page, or per article.
  + <article>
  + <aside>
  + <h3>Related links</h3>
  + <a href=”/”>Link 1</a>
  + <a href=”/”>Link 2</a>
* Figure
* Footer
  + <footer>
  + footer content!
  + </footer>
* Header
  + <header>
  + Header!
  + </header>
* Main
  + Typically added by Content Management systems, not needed
* Nav
  + Represents page navigation
  + Can be used more than once
  + <nav>
  + <a href=”/home”>Home</a>
  + <a href=”/work”>Work</a>
  + <a href=”/about”>About</a>
  + <a href=”/contact”>Contact</a>
  + </nav>
* Section
  + Section is the most generic of the new HTML5 semantic tags 0 it groups together related content, typically containing at least one heading.
  + Can be used multiple times within other tags, or as or as the main element
  + <section>
  + <h1> The life changing magic of front end development </h1>
  + <h2> intro to HTML, CSS & JavaScript </h2>
  + <p> content </p>
  + </section>

But what about divs?

* When none of the previous elements are suitable, div is the way to go.
* Dive can be used as a general structural element – wrapping an entire page or an entire area of content that is being styled using CSS

Sectioning Content

* Use as many of the HTML 5 tags listed above and make a new website.

8/30/2017

HTML Pt. 4

How do you create a table layout>

* Table tags, you need <tr> and <td>
* Sports stats, schedules, anything like this is where you would use a table

What is the purpose of commenting?

* So you can leave reminders for yourself or to tell others what you were doing

What is a div?

* A grouping tag, groups information that is related

What are some HTML5 layout elements? Why were these created?

* Article
* Header
* Nav
* Main
* Footer
* Aside
* Section
* They are good for semantics, which helps website ranks

Video

* We won’t use flash
  + Its not supported anymore
* Video Fundamentals
  + Online video formats include, AVI, MPEG, WebM, and more. Certain browsers support certain formats, creating a need for fallbacks.
  + Broswers did not support video until 2010.
  + Now we can use <video>, an HTML5 element, to utilize videos on your website.

Hosted videos

* 2 ways to utilize videos on your website – self-hosted and hosted through a third party site, such as YouTube or Vimeo

<video>

* <video src=”/” poster=”/” preload controls loop>
  + <P>descriptive content</p>
* </video>
* the poster is the backup if the video can’t play for whatever reason
* cope iframe (embedded code) for youtube videos

Forms

* Forms are typically a fullstack action, beginning on the front end, sending information to a server and displaying a result on the front end.

Basic Form Structure

* <form action=<http://example.com/subscription.php> method=”get”>
* </form>

Typed of Form Controls

* Text
* Password
* Text Area
* Checkbox
* Dropdown
* Multiple Select Box
* File Input
* Submit
* Radio Button
* Date
* Email & URL
* Search

Escape Characters

* Less than &lt;
* Greaterthan &rt;
* Ampersand &amp;
* Quote Marks &quot;
* Copyright &copy;
* Trademark &trade;

Project Due the 8th on portal

* 3 headlines check
* 5 paragraphs of real content
* 4 images check
* 2 videos check
* proper usage of section and article tags
* a form 1 field and a submit button check
* a header check
* a footer check
* aside (reference links)
* navigation check
* a table with at least 3 rows and 4 columns check

Things to consider when planning a project:

-Size of project

-Timeline

-Budget

-Technical requirement

-Client requirement

-Scope

Fitbit steps

Gather the research and content for website

Add structure and semantics like p, header

Add info to the head/body

Add anchor links

Complete table

Complete form username/ empty box

Check indenting and unclosed tags

1. Header beside it Navigation= Product, Why Fitbit, Get motivated, Help Form=email, password, login<value
2. Gather info, 5 paragraphs, with 2 images beside it, and table, one video
3. Pros and cons =Headlines,
4. Structure div, sections, article, footer, aside
5. Sources, aside
6. footer