HTML / CSS : Introduction

HTML is an acronym for Hypertext markup language, it is the language of the web. The first HTML was released in 1991 by Tim Berner-Lee . His creation has seen many updates ever since, including HTML 2.0 in 1995, HTML 3.2 in 1997, HTML 4.01 in 1999, and XHTML in 2000. Currently, the newest HTML version is HTML5 which was released in 2014.

HTML,CSS AND Javascript helps build the website structure. HTML outlines sections of the webpage i.e headers ,paragraphs, quotes,lists,tables e.t.c..

BASIC HTML STRUCTURE.

Opening tag<h1>

content

Closing tag </h1> .

E.g

<h1>Hello Africa!</h1>

Some tags are self closing. Tags like

single line break

<img/>Image tag

<hr/> Adds a stroke

Browsers do not display the tags but uses them to render the content of the web page.

HTML5 TAGS

Just like i mentioned, HTML5 is the latest version of html. The structure of each versions changed over time. Some tags are no longer in use today. Listed below are some of the elements used in HTML5 and all serve difference purposes. Just like the structure of the human body. we have the head, body, foot...

<header></header>

<main></main>

<footer></footer>

<nav></nav>

<video>

<article>

<section>

e.t.c

File extension used is .html, just like we have .docx in a word document and xlsx for excel.

HTML to a webpage is just like the skeleton in the body. If you see a skeleton walking around, what comes to mind?. REPLY AS THREAD ONLY

Cascading style sheet. Unlike HTML, CSS doesn't contain any content rather it is used for styling your existing content outlined in your HTML code. If HTML forms the outline and structure of a house, CSS is the design: the walls, the paint, and the furniture. Same as our body. The designs we choose affect how we structure our house.

when you see a beautiful house, you get attracted to it right?, yes, same as a website with good aesthetics. That's the work of CSS

Without CSS, a website will look as aesthetically pleasing as a word document. With CSS, you can add colours of all kinds,compelling fonts and layouts. CSS can be added to HTML elements in 3 ways:

Inline

- by using the style attribute in HTML elements. It is displayed as show below.

<h1

style="color:blue;">This is a heading</h1>

Internal

- by using a

<style>

element in the

<head>

section

External

- by using an external CSS file

The most common way to add CSS, is to keep the styles in separate CSS files, especially for a large project.

CSS consist of rules. <style> tags are used to tell your browser that you're now using CSS, and not HTML. They use <style> tags within HTML code to demonstrate how CSS works. Every CSS rule, which specifies how content is displayed from the html code, is written on a separate line. This is how CSS works: it selects the element you want to style, declares the property you want to edit, and gives it a value.

For example, below we are selecting a paragraph as the element we want to style, declaring the property of "color," and giving it a value of "pink." This would make the text of this paragraph pink.

<style>

p{

color: pink;

}

</style>

For example, below we are selecting a paragraph as the element we want to style, declaring the property of "color," and giving it a value of "pink." This would make the text of this paragraph pink.

<style>

p{

color: pink;
}
</style>

-

Before we continue with the css, Let me show you the html code so you understand better.
The code displayed above is pure HTML code
and as you can see, it was saved as .html
It shows the various html elements and their purposes, it is but not limited to what is displayed
Can you see the <p> tag there?. Reply as a thread only
Now back to our CSS....
the <p> shows "I am a paragraph" abi
If we want to decide make it more beautiful by add colors, change the font or layout. All we have to do is invite CSS
Then we make use of any of the styling methods, either inline, internal or external
<style>
p{
color: pink;
}
</style>
The above code is called Internal styling. If you were to write another CSS rule underneath the one shown above — let's say a rule to make the text purple — it would overwrite the initial rule. The bottom rule always gets preference with CSS. Below is an example.
<style>
p{
color: pink;
}
p{
color: purple;
}
</style>
Unlike HTML, we can see what CSS rules are being applied to our website with the web inspector.
Web Inspector
You can use the
web inspector
to view, study, and change the code of any web page, and play around with it to learn about coding. Simply right-click a web page in any modern browser and select
Inspect
from the drop-down menu.
Note:
when using the web inspector, make sure you don't refresh the page; refreshing resets everything, and your edits will be lost!
Want to know ow a secret?
HTML and CSS is simple.
This brings us to RESPONSIVE WEB DESIGN
Imagine you want to build a duplex now. and the architect gave a design to the constructor to build. Then the constructor builds a house not so tall, a kind of one a 6'6 person has to bend to enter
And every tall visitor that comes visiting has to struggle....
CSS is what allows
the website to be responsive. Responsive design means that the elements on a web page increase and decrease in size depending on the device being used to view it.
Have you ever wonder how you can view a website from your phone and your laptop as well without seeing layouts sore to the eye
Well, thanks to responsive web design
Responsive websites look good and are easy to read on any device, whether it's a computer or a smartphone. This is really important, because mobile web use is becoming the dominant form of Internet usage. If your website doesn't have responsive design, it will be inaccessible to the majority of visitors.
You can also test your website's responsiveness by changing the size of your browser window (for example, drag your browser window smaller and larger with your mouse).
In the end, all you need to make a beautiful, working website is HTML and CSS!
However, many sites use another language called JavaScript.

PHP: Basics and General Introduction

Hello everyone, i'll be taking the basic intro to PHP. I started my journey as a PHP developer a few years back. I am quite familiar with the language
PHP was initially created by Rasmus Lerdorf, and at the time, it meant : Personal Home Page. However, as the language grew in popularity and "matured" the name changed, and it is now a fully open source language managed and maintained by The PHP Group
PHP now stands for Hypertext Preprocessor, from the name, you can almost tell what it does. Seeing that HTML

stands for hypertext markup language.
PHP is a general-purpose scripting language that is especially suited for web development and can be embedded into HTML.
There are 2 methods of writing PHP (I think this applies to a lot of other languages),
Procedural Programming
OOP - Object Oriented Programming
PHP is a very popular language with a lot of community support.
It is a really good introductory language to the world of programming, however, i would not recommend staying with "only" PHP for too long.
With PHP you can build a fully functional web application.
Procedural Programming is for those just starting out, as it uses small concepts like functions (methods) that act directly on data. On the other hand, OOP allows you to create an entire "structure" that contains it's own methods (functions) which can then be re-used in several ways.
Let me try to explain that in simpler terms,
A car is an object that has different parts... each part with it's own function and operations. If you have a problem in a car, you can identify the part that has the problem, fix that part, or entirely replace the part.
In the same way, parts of a car can be re-used. One part of a car can be taken and moved to another and everything still works well.
Because of the popularity of PHP, there are numerous frameworks out there...
A framework is (in english) - an essential supporting structure of a building, vehicle, or object.
So, someone out there decided to use PHP to build a "scaffolding" (or a skeletal structure) that can then be extended to make more awesome components (or objects).
So, someone out there decided to use PHP to build a "scaffolding" (or a skeletal structure) that can then be extended to make more awesome components (or objects).
One of (if not the most popular) frameworks is Laravel.
In this course, we won't be introducing you to any framework, you just have to know about them.
Other popular frameworks are CakePHP and SlimPHP.
PHP generally can be used to connect your web application to a database, as it works with all sorts of database software including MSSQL (you'll learn more about databases later)
Lets consider a typical example, if you are building a personal portofilio website, and you want to have a contact form on it, so that it's possible for users to send you an email directly from your website, you can use PHP to process the form, so that when they submit it, your PHP code will send you an email that someone tried to reach you via your website.
So you can say, PHP is a backend language
Unlike HTML / CSS (which are not programming languages by the way), PHP code is not exposed via browser.
PHP resides on the server, and except your server is not well configured, then your PHP code is secured.
A PHP file is saved with file extension .php
(just like you save HTML with .html and css with .css)
You can develop PHP locally if you install the write tools...
Unlike HTML / CSS PHP needs special tools to be installed before you can write and test locally...
A professional (like me..) can decide to install all required plugins or components and configure personally (which we normally do on cloud servers), but for this class, i'll be introducing you to a tool that already contains 90% of all you need...
For those on windows, the tool is : WAMP
For those on Mac the tool is : MAMP
for those on Linux, you can use XAMPP (which also works on windows and Mac).
You can download and install these tools.
WAMP Stands for Windows, Apache, MySQL, and PHP
Which means, I am installing: Apache, MySQL and PHP on my compute
you can already tell what MAMP means
and i'm not entire sure what the X in XAMPP stands for (you can google that)
Apache is a web server software...
As i already said, PHP runs on the server, Apache is one of the most popular web server software out there, the alternative is NGINX (which i prefer by the way), but Apache is probably the easiest to use, anyone can use Apache.
so we will be using Apache for this training.
installing Apache on your computer means you are converting your computer from just an ordinary machine to a machine that is capable of 1. receiving requests from 2. and serving out response to external connections (computers), which basically means, you are converting your computer to a "server" (something or someone who serves)
MySQL is an RDBMS (relational database management system)
in simplier terms, MySQL is one of many database softwares available (another example, as i have mentioned is MSSQL, which i think is most suited for C# and .NET framework)
In summary, WAMP (or the others) installs MySQL, PHP and Apache on your computer. Allowing you to begin your journey into PHP development.

Is PHP dying (or dead)?

I just have to address this, it's a new thing that is going around now, where some "geniuses" are spread ing the word that PHP is a dying (or some bold ones even say, dead) language...
This is not true.
being Old does not mean dying or dead... and actually, Old is not the right word to use for PHP, PHP is far younger than Java which is very much useful today.
However, because of the popularity of PHP, it is a very competitive language...
Which is why (i said early) if you start with PHP, make sure you learn other languages, because there are numerous PHP developers in the market.
I experienced this first hand, things you would do for 50k, there is one IT student willing to do it for 2,500
Basic Overview before i close,
you start writing php with <?php and close with?>
you know how you write opening and closing tags in HTML, the same applies to PHP,
anything in between <?php ?> is considered a PHP code.
Variables are declared with a dollar sign, so for example, $lag, is a variable called lag.

You'll lean more about variables in GN class (one of the courses you'll be taking)
Alright, that's basic intro, i'll post a video of how to setup your computer for PHP later tomorrow
Any questions?

What is GN class - General Class , compulsory classes for all.
Which IDE will you advise for PHP?- I use VSCOde and Sublime. You can even use notepad if you want.
How good with php do have to be before you learn a framework - When you are familar with the basic syntax, and build at least 1 full app without framework, then you should begin learning framework
can you get an entry level job with just php an no framework - Yes, but, quite rare.
Does that mean that we are going to write php code in same page as html - Yes, you can, you can also create it's own file. Or use PHP to output your HTML.
You said later on the php class we going to have a class on database so I was wondering if the database class will be intensive or just basic and move along? - Everything here is basic. The DB or database class is not for PHP, it's for everyone.
Does that mean php and html do the same thing - Ofcourse not.
kindly recommend a good place and resources to learn PHP - yes, i was coming to that, here:
https://www.php.net/
https://www.w3schools.com/php/
Can we say Apache in php is like Node JS in JavaScript?- Not entirely, but yes, more or less.
Should (must) a front-end Dev know and use php to function well ? - No (not must) and not (should), but can.
Can PHP act as a standalone back end for a web app - Yes.
I already have XAMP installed on my system, do you advise I ditch XAMP for WAMP? - No

, if you want to advise a novice on the journey of backend developer, in ascending order, can you please list the types of language l can still with it, to avoid confusion? - PHP , C# , JS (in no other, do anyone that is easiest for you of the 3)

can you use PHP in VS code directly without a WAMP or LAMP server? - yes, but it won't run without a server

Is the php has it own link to other languages as HTML is link to CSS when coding? E.g <link rel="stylesheet" type="text/CSS.....> - Not necessarily, you'll need to wait for more about that in class

From all you have said, can PHP be learned with an android phone? - I am not sure, i've never done it before. But, sure, anything is possible

Please, will this class page be open for us to always refer back to whenever we in need of clarification, or should just duplicate it, because it will be wiped out? - yes, on HNG Board as alwaya
In your opinion, What is the basic reason as to why people refer to PHP as old? - i think because of it's popularity and how you can find a php developer almost anywhere.
Why don't you recommend staying on PHP for too long? - I already answered this in class
So is php a frontend language or backend language. - backend, already said that
So what language is recommended for front end
You said OOP allows you to create an entire "STRUCTURE"
Why do you have structure in quotes? - because OOP might be used to creaate something small or big. Structure makes it sound like it's always something big. So thr quote means "i am loosely using tbis term"
Do you mean MSSQL or MYSQL - they are 2 different things. Already said this...
What happens if your php code is exposed - what happens if a person's underwear (which should be under and not exposed) is exposed? Hackers and other developers will mock you
I have observed that one need to learn different languages to be a good programmer, which of the language would you recommend for a Front End aspiring student - this is like the 100th time this question us being asked just learn HTML/CSS and JS. The rest are additions. Start with those 3

What are cloud servers - a server that is not on your computer or that you do not have physical access to.
What does a framework do exactly? Also, is it a software that one can install on a PC or it comes embedded in a programming langauage? I don't know of any framework that one installs, most of them are just a bunch of code that has been written in a more friendly way so that you don't have to write the raw language every time.
You mean php is a server side language because it cannot work without Apache server also fo you mean laravel php frame work is like wordpress where you dont have to write much code?
This question is not clear... But, wordpress is a CMS: content management system, it's not a framework. And yes, PHP is a server side languagw. With Laravel, you still write plenty code... Depending on what you are building (or it generates code for you)
What do you mean by open source? - software or tools that is created, maintained and used by anyone (everyone) Free too
You mentioned that all the stuff we will be learning here are going to be basic. After we have done with this program, what next? Is there a plan for us to further in those areas we have chosen. We will keep sending you tips and opportunities.

Design: General Overview and Tools

Hello, I am Iheonye Chukwuemeka and I build products.
We are going to start this class by discussing what design is.
So What is Design?
Design, as defined by the dictionary, is a Plan or Drawing produced to show the look and function or workings of a building, garment or other objects before it is made.
Another says that Design is a decorative pattern.
But I can authoritatively tell you that design is everywhere, Design is everything. From the sound, a car horn makes, to the pillows on your bed to the water bottle on your hand. The question is always be “why.” After this class, you would have a rethink of your surroundings and how you perceive design, you will be able to appreciate life and design. As a designer always ask yourself “WHY” Why is the color blue used in this app?

why is a car not round? Why is my shoe pointed? Why is my toothbrush having a handle? And why is it shaped the way it is shaped? Why is the button on this website located in this particular position? In fact without design, everything fades away and there is chaos, even the world was designed. With these points we all can agree that design is everything and so cannot be exhausted.
Not everybody can be a designer, Not everyone should be a designer. In fact, the design is a life and death profession and needs those that have the talent and the grit to pull through it.
Design is so versed that we are going to talk about digital products. In fact, I will love to introduce the concept of user experience (UX). UX goes beyond digital products to physical products. It is as the name is what you feel from using a particular product. How it affects your mood and your actions and inactions. It could be Consciously or unconsciously. UX Design for digital products is also a large field on its own. It is broken down into different parts. They are job titles associated with UX design, they are:
1. UX Designer
2. Product Designer
3. Visual Designer
4. UX Researcher
5. Content Strategist and
6. UX Unicorn
I wouldn't go very deep into explaining different parts of this now but I would use this diagram.
UX Designer

Product Designer

Visual Designer

UX Researcher

Content Strategists

UX Unicorns

Fundamental of Design is the same, The tools doesn't really matter as much as the fundamental of Design because the tools will always evolve but to be a great designer you must understand the fundamentals and be timeless. You may be great at using Photoshop, Figma, XD and be a very poor designer. There are popular design methodologies you should know about.
They are;
Design Thinking: This is a very popular methodology because you design solutions by first emphasizing with people, defining their problems, then coming up with ideas, prototyping solutions and testing them with the users as you can see from the diagrams above.
Agile Design: This is when you design continuous, iterative Improvements while getting frequent feedback from users. Normally we use the Waterfall approach, where you hand over fixed designs to the developers, in the agile method you will work with the developers closely and constantly change your designs.
Most of the methods are often used together there is no hard and fast rule on how to use them.
Now you heard we will use Figma as the main tool in this training. YES, you heard right. The reason is that I love Figma. *Smiles* Scratch that.
Figma Provides us with certain Collaborative power we can't get on another app just yet and all your designs are saved on the cloud, so nothing like lost files and all those hard disk stories, also its Free. I know we have money to pay but People like us are managing it.
During the course of this program, I will introduce you to the core of design and show you how to use Figma to achieve great stuff.

JavaScript Introduction

Hi everyone, I'm Jeff and I'll be introducing you to JavaScript. I'll mostly focus on explaining some concepts and using code snippets to illustrate later. For now, don't pay too much attention to the code, this intro is meant to show you some things you can do with JavaScript and to help you build an initial understanding of basic JS.
JavaScript often abbreviated to JS is one of the core technologies of the web (alongside HTML and CSS). It is a cross-platform, object-oriented scripting language used to make webpages interactive (e.g., having complex animations, clickable buttons, popup menus, etc.). It does this by adding behavior to your web pages without a need to reload them.
For example. you can:
Add new HTML to the page, change the existing content, modify styles.
Reacting to user actions, run on mouse clicks, pointer movements, key presses.
Send requests over the network to remote servers, download and upload files.
Get and set cookies, ask questions to the visitor, show messages.
Store the data on the client-side ("local storage").
Fun fact: Asides "Java" and some common syntax, JavaScript and Java have almost nothing in common. Supposedly, JavaScript was named so because Java was really popular at the time and someone thought it'd be a good idea to ride on the fame as a marketing strategy.
JavaScript is almost everywhere. It's one of the most popular languages are you can find/use it almost everywhere.
On Frontend you have popular frameworks like React, Vue, Angular
On Backend you have Node and Express
Database there's Mongo
Mobile there's React Native
Machine Learning there's Tensorflow.
You can't really go wrong in learning it

JavaScript follows the ECMAScript specification. The names are used interchangeably, so you might see JavaScript being referred to as ECMAScript or ES (an abbreviation). Sometimes there's a year or number attached eg ES6 to indicate which version is being used.
JavaScript comes built into all modern browsers. This means you don't have to install anything to write JavaScript

and you can get started right away using your web browser. To experiment, you can open your browser's Developer Tools (DevTools). Usually, this is located in browser menu under "Tools". Alternatively, most browsers have Ctrl+Shift+I as a shortcut to open DevTools.
DevTools are a set of web developer tools built directly into your browser. You can use this to play around and edit HTML/CSS and JS on the fly. There are a lot more things you can do with DevTools but that's a topic for another day.
With DevTools open, click on the console tab. This is where you write JavaScript code while experimenting. In Chrome, you can use Shift+Enter to go to the next line instead of executing the command. In Firefox, you can use Ctrl+B for multi-line mode.
If you have DevTools open and you're in console tab, type console.log("JavaScript is awesome") and press enter. Let's do one better, type alert("Life is Good"). Did you see the popup? If so, good. You have gained mastery and there's nothing more for me to do. JK JK, but if this is your first experience with JS then that probably felt cool.
I mentioned DevTools is used for experimenting, so how do you write JS code? There are two ways. If you were in the HTML/CSS class, then you already know about tags. There's a "script" tag for JS. For the first way, you can reproduce the above by putting something like this
<script>alert("I'm awesome")</script> in your HTML code. You can actually put this anywhere between your html tag and it would work, but it's good practice to put your JS just before the closing body tag </body>.
For the second way, you can write your JavaScript code in it's own file, save it with a .js extension and then import in your html like this
<script src="index.js"></script>. The src is the path to the file, if the file is in the same directory (folder) as your html, then you can just use the name like so. If it's another folder, you would use a relative path eg js/index.js.

So finally, before I close, here's a code snippet highlighting some basic JS code. I'll explain how they work briefly.

```
let name = "Jeff";
console.log("My name is: " + name);
const data = [];
for (let i=0; i<=100; i++) {
  if (i%2===0) {
    data.push(i);
  }
}
console.log(data);
```

The code above stores and displays my name, then finds and displays all even numbers from 1 to 100.
First two lines have let and const. These are keywords for declaring variables. Variables are named memory locations for holding data. You can think of it as having a box with a label on it. The rules for naming variables are:
cannot contain spaces.
must begin with a letter, an underscore (_) or a dollar sign ($).
can only contain letters, numbers, underscores, or dollar signs.
are case-sensitive.
must not be a reserved word.
When naming variables, it is good practice to use descriptive names. For example, using studentName is much better than saying value. The former let's you know at a glance what data it's stores or is expected to store. There are 3 ways of declaring variables in JS: var, const, let. I won't go into much details for the first two but you can think of var as the older way of declaring variables. It has some specific rules that might give you unexpected results if you're not familiar with how it works. Const is for declaring variables whose reference is not intended to change. It works from what you might traditionally expect with constants because for arrays and objects, you can still modify the values themselves as long as the reference doesn't change. In the snippet above, I was still able to modify the content of data even though I used const.
You can think of it this way, when you use const, you point to a "person". It's like saying, okay, this is the driver. This driver is constant i.e, he won't change (re-assign), however, his "values" can still be changed. For example, he can change his clothes, or his hair, shoes etc. He can wear a completely different attire but still remain the same driver. If it's const, trying to change the driver will throw an error because he cannot be reassigned. This is more or less how const works.
Let is the regular way of declaring variables. With let, you can change both values and reference. So in the above example with driver, we would be able to change both the driver or his "values".
There are seven data types that are primitives:
Boolean. true and false.
null. A special keyword denoting a null value. (Because JavaScript is case-sensitive, null is not the same as Null, NULL, or any other variant.)
undefined. A top-level property whose value is not defined.
Number. An integer or floating point number. For example: 42 or 3.14159.
BigInt. An integer with arbitrary precision. For example: 9007199254740992n.
String. A sequence of characters that represent a text value. For example: "Howdy"
Symbol (new in ECMAScript 2015). A data type whose instances are unique and immutable.
Object

JavaScript is loosely typed, so you don't have to specify which data type you are using when declaring variables, it will figure it out automatically for you. When I typed let name = "Jeff", it knows that name holds a Stirng.
I used an array in the snippet. An array is a list of data separated by commas and enclosed in square brackets []. Arrays can hold different data types including other arrays. You may have noticed that Array isn't listed among the data types. That's because Array is a special type of object. You'll learn more of that later.
So next there's my data variable with an empty array. I intentionally used const here to illustrate my previous explanation about being able to modify values even though you used const as long as you aren't changing the reference.
Next is a loop. A loop is a repetition of code until a certain condition is met. Sometimes, you might have to do something repeatedly with few changes. For example, assuming I want to display from 1 to 100. I could type them all out if I was jobless and had a lot of time to spare. And in cases where I have to display maybe a million numbers, I could also type it all out if I I'm...well you get the point.
Repeating such instructions is time-consuming and also prone to error, and loops come in to simplify all that. I

mentioned earlier I won't go into much code-specific details, but for (let i=0; i<=100; i++) controls the loop. It starts my "counter", updates it and checks to be sure the loop runs for the intended number of time.
if (i%2===0) This is a conditional statement. if means an expression will be evaluated. It either returns true or false and then executes a code block depending on the result. The % is modulo. If you recall math from school, this is known as remainder after division. So this expression is saying if the remainder after dividing the current number by 2 is 0, then do what it is the code block
and finally, data.push(i); is a way of adding values to an array. Arrays can supercharged with a lot of methods (abilities) to do various things. push is one of such, where you can quite literally "push" a value into the array. As mentioned earlier, even though my array is a const, I can still change/update the values because I am not "reassigning"
Our data array now holds all the even numbers.

Introduction to Go (golang)

What is Go?
Go is an open source programming language that makes it easy to build simple, reliable, and efficient software
That probably may not make sense to you, but hold on let me it down a little more
Go was built at google to solve very many challenges that a big tech company like google would generally face. They figured that the current languages they use are good but lacked some features that makes it a strife to use productively, efficiently.
So Go was born out of "we need to scratch a giant itch" thingy.
Go is a multi purpose language, initially designed to serve as a systems programming language (Operating systems, network code, drivers, database servers etc.).
Currently, you can use Go for Web development, OS development, embedded systems, build large scale infrastructure, microservices etc
Go is widely known for it's speed. It has a peculiar inbuilt concurrency support.
Don't worry about the concurrency thingy for now, it'll all be clear eventually.Go was designed to be simple and concise, readable and less stressful to become productive with.
There's a mantra in Go that suggests that THERE'S ONLY one way to do something in Go, ONE.
2:52
This is 95% true, and the benefits of that is that it makes the language much simpler to learn
The complexities you'll find in Java, C++ or C# you won't see in Go. In conclusion, Go is like a Rocket engine that can be driven like a simple car on a simple road, and still maintains it's ability to shoot you to the moon if you want it to.
Below are few known projects written in Go:
Docker
Kubernetes
Ethereum Client
etc etc
Go is open source so you can look up the code here: https://github.com/golang/go

Introduction to NodeJS

Hi, I'm Solomon Eseme, I build Business Logic (Backend Dev).
i will be introducing you to Node.js. It's going to be more like getting started with Node.js.
Previous today @Jeff Introduced us to JavaScript.
JavaScript is a language that has been doing well at the Front end or Dom manipulation and thousand of developers loved it (OOOIN to others, I say OOIN to them).
Until an Engine was developed to interpret/run that same JavaScript on the Server (Backend).
This Engine is what we call Node.js.
What is NODE.JS
Node.js is an open-source, cross-platform runtime environment for developing Server-side web applications using Javascript.
You can read a clean introduction to Node.js here www.nodejs.dev
Installation of Node.js
Installation of Node.js is straightforward using the installer package available at the Node.js Official website.
1. Download the installer from https://nodejs.org/en/
2. Run the installer.
3. Follow the Installer steps and agree to the license and click Next.
4. Restart your System.
You can test by typing `node -v` into your terminal.
The Scary Part
As a beginner, it's hard to get to a point where you are confident enough in your programming abilities.
While learning to code, you might also be confused at where does JavaScript end and where Node.js begins and vice versa.
I would recommend you to have a good grasp of the main JavaScript concepts before diving into Node.js.
1. Lexical Structure
2. Expressions
3. Types
4. Variables
5. Functions
6. this keyword
7. Arrow Functions
8. Loops
9. Scopes
10. Arrays

11. Template literals
12. Semicolons
13. Strict Mode
14. ECMAScript 6, 2016, 2017.
With those concepts in mind, you are well on your road to become a proficient JavaScript developer, in both the browser and in Node.js
I believe your JavaScript class should introduce you to these concepts, if not google the hell out of it or ask questions.
BREAK TIME
Drop your Questions and try to install Node.js on your system if you don't have it already.
THE FUN PART.
In this introductory section we're going to create our first Node.js application. Don't get scared yet, it's the traditional `Helloworld1 program.
1. Create a JavaScript file called helloworld.js
2. Open the File with any text editor and type in the following codes.
`console.log("Helloworld");
3. Save the file.
4. Open your terminal / command prompt.
5. Navigate to the folder where you saved the helloworld.js file (mine is in Desktop)
`cd Desktop`
6. Run the command
`node helloworld.js`
If you see 'Helloworld' as the output
Congratulations
If not, OOIN ask questions
You can be creative, type in the code from your JavaScript class and run it..
So we are going to create our second Node Application, this one will be a little complex
Don't worry, if you don't get it now..
Create another file inside that same directory as before and call it `server.js
Paste in the following codes

```
const http = require('http');const hostname = '127.0.0.1';
const port = 3000;const server = http.createServer(function(req, res) {
 res.statusCode = 200;
 res.setHeader('Content-Type', 'text/plain');
 res.end('Hello World\n');
});server.listen(port, hostname, function() {
 console.log('Server running at http://'+ hostname + ':' + port + '/');
});
```

Open your Terminal/Command prompt again and Navigate to where the file is saved
Type the following command node server.js
Open your Browser and type the following address localhost:3000
If you see helloword Congratulations
Now let me explain what is going on
The first line const http = require('http');
Imports Node.js ModuleModules are just reusable codes that You and I can create....
So the Node.js developers create the http module for use to interact with HTTP (servers of course)
The second line create a variable and assign a web address to it, same with the third assign Port number to it
We have the next `const server http.createServer(function (req, res) { ..... `
That line simply use our http module we imported earlier to create a new server and pass on the response data to it...In our case we pass on status Code 200, Content Type = Plain text and then we render the text with the res.end method
After setting up this server with the information, we simply create the server and listen to incoming request to the port and address we provide...If it's a success, we log the status
Alright guys. We are done with the introductory class. I will answer some questions now
I do a lot of talk about Build and Developing Business Logic (backend) on Social Media.. Lets connect better on twitter https://twitter.com/kaperskyguru

Python: Introduction to Python

I am Davis. I'm a Data analyst.
this is going to be just a high level overview of what python is, its features and what you can do with it. Very little code.
I'd share a pdf and walk you through the slides.
so what is python?. I believe we can read so i will skip the first 2 bullet points and explain the last.
procedural, object oriented and functional programming are 3 different paradigms and python supports all 3 of these. You will get more clarity on this when you start delving deeper. Just take them as 3 different approaches to programming for now.
Python is easy to learn in my opinion (easy english like syntax) and very beginner friendly. I started with python so i might be bias buts its true.
it is cross platform. it runs on windows, mac and linux operating systems. Some OS now ship with python inbuilt
It is readable. Like i said, easy english like syntax. Makes use of indentation to define scope and control structures. To programmers that are used to curly braces, this can be a pain in the butt but its cool in my opinion. Less cluttering.
Large community : Python has a large community of users so theres no error you'd get that someone hasnt seen. There are also a lot of community developed libraries that makes your work easier. Python wasnt built solely for data analytics. R was. but python has outshone R for data. Why?? Community power.
Python is fairly easy to setup compared to some other languages. It is interpreted so you don't need a compiler. (Except you're me with 3 different versions of python and getting around them can be tedious sometimes.)

so... What can you do with python?. A lot of cool stuff apparently. From web to machine learning, embedded systems programming and even desktop apps.
Frameworks such as django and flask can be used for the web apps.
Scikitlearn library helps you do machine learning with python. A ton of others too. Frameworks such as tensorflow and keras for deep learning. A whole lot can be done with python.
Most top companies use python
my personal favourite is Netflix. Netflix's recommender system is state of the art and thats one of the things possible with python.
Installation
6:26
You can get python for your computer from python.org
6:26
it installs like any other application
6:26
windows users make sure to click add to path when installing
once youre done with installation. go to your terminal and type python --version
should tell you the version of python youre running
after that, type python and hit enter
6:28
you should be in a python environment now
6:28
type print('Hello world')
6:28
you should see hello world as output
python also ships with an IDE called IDLE
open it up, you should see a command prompt waiting for your code. you can do the hello print there too
6:30
Thats it for today.
Questions??
What's the scope of our python here...
Backend right
Django or flask or both?
Thanks : Backend
what aspects of python are we focusing on? thanks : Backend web development.
Is python good for beginner who has no idea of any programming ? : YES
Are you going to teach us in relation to data science? And how do we get to connect it with vs code or we are using pycham? : No DS track. yes you can use any IDE you want later on. We will get there
So once u download python it comes with it's IDE? How do u access the IDE to start coding? : On windows, search IDLE
What do you mean by indention ? : Indentation. It will become clearer soon.
Why do we need to download python... can't we just use an IDE to code in python language and save with the corresponding extension? : You need an interpreter to run python codes. You are basically downloading the interpreter. Just like to download the jdk compiler for java.
is there going to be data analysis class : NO
Which IDE is preferable for Python Anaconda or pycharm : Anaconda is not an IDE. it is more than that. It ships with IDEs jupyter notebook and spyder. I use jupyter notebooks. You should use pycharm for web dev.
What are the major differences between the three versions of python and which do you recommend : There are only two versions of python. Python 2 and python 3. I mentioned 3 versions because my mac came with 2.7, My anaconda has 3.7.4 and i downloaded 3.7.6 for something else. You should download the latest version 3.
What's the difference between Machine Learning and Data Scientist : Machine learning is an aspect of data science
What other major companies use Python : Most you can think of. Google, Microsoft. You name it.
For the internship, can I focus on just Python tho I am in the back end track ? : Yes you can.
Is the use of R and python interchangeable : For data analysis yes though python is smoother in my opinion. R doesnt do any other thing outside data. It is a statistical tool.
i just installed the latest python from python.org but its for 32bit but my system is 64bit can you share a link where i can download that of 64bits so we can move forward easily or caan i still use 32bit : There should be a 32 bit link on the website. Check well.
As a beginner who's interested in front end and back end which would you advice I learn JS or Python? : JS.
Can you please differentiate between anaconda, Spyder and jupyter : So anaconda is basically a distribution. A collection of tools to make your life easier as a data scientist. Jupyter notebook and spyder are IDEs. Are tools in the distribution
i actually need the one for 64bit : All links are on the website.
What do you mean by Django and flask : Python frameworks for web development
The 64bit links are confusing. which one exactly are we to download? : Windows x86-64 executable installer
So thats it guys. Thank you.

Java: The Introduction

I am Mba and I'm excited to be here.
I do Java and Android development, I love challenges and trying out new things. Fun fact: I successfully went through the 1st StartNG program last year and went on to become a finalist of the preceding HNG Internship
So yes, it's possible!
Okay...
Since I'm handling the first class in the Mobile space, I'd talk a bit about Mobile development, then Java and finally JAVA for Android development
So Mobile development....

Have you noticed Apple's iOS and Google's Android has squeezed all other mobile devices out of business? Developing applications for use on these devices is basically all Mobile development is
Let's be clear... Mobile development in the context of StartNG is not development of Mobile responsive web applications or even the development of hybrid applications that can be uploaded to the app stores using Ionic and Cordova in addition to HTML and her siblings
It refers to the development of native applications for specific mobile OSs. By native, I mean platform specific development I.e developing specifically for Android or iOS. It's basic advantage over hybrid applications is better performance, localization and better ability to manipulate the devices' hardware.
There's also cross platform development - Writing one code for both Android and iOS.
Cool I know... but you see, using a generic measurement to sew clothes for 3 12year olds can never fit any as well as taking individual measurements. It's definitely faster and cheaper though.
A quick summary
Android development
Java - Oracle or Kotlin - JetBrains
iOS
SWIFT
Cross Platform
ReactNative - Facebook or Flutter- Google
JAVA is a very powerful general purpose programming language.
Developed by Oracle in 1995, JAVA has for 25 years
Java runs on about 3 billion devices making it one of the most popular programming languages and it is designed with the WORA intention - meaning the developer only has to write his code once and then run it on different Java supporting devices without having to recompile the code. Cool right?
Java finds application in Mobile development (Android specifically), Web development(BE), Desktop application development, Games development, Database connection, Web servers & application servers and much much more... so when we say "general purpose", we mean general purpose. It is Class based and Object Oriented, similar to C++ and C#
Some stats:
It's placed as the 2nd most popular
According to the Stack Overflow survey, Java's popularity stands at 45.3%, which is the second-best
TIOBE community places Java as the most popular language, as of July 2018.
A 16.038% rating places Java as "most popular" programming language in 2019 according to Github
Many of the world's biggest enterprises choose JAVA for its portability & scalability
Java is a platform-independent language. It's because when you write Java code, it's ultimately written for JVM but not your physical machine (computer). The implication of this is that you can write and run Java code easily on Any OS, Raspberry PI, virtually any device
It is open source, easy to learn, simple to use, and has a huge community for support which means you never have to solve a problem alone.
Let me just stop here...
As developers, we should always be open to new technologies and embrace them where necessary, don't be a die hard fan of a language so that when it's time for growth, you'd see
That said, I'd like to say that Kotlin is a very cool language for Android development, even easier and more efficient than Java for Mobile development. So cool that it's part of my learning goals for 2020.
If Android development is all you're interested in right now and you'd love to explore it deeply, please skip JAVA□. However, if you want the openness and versatility, JAVA is your plug
Okay now let's get to the main dish, Java for Android development
JAVA is the basis of the Android SDK and for generations, the best native Android applications were built using JAVA
A typical Android developer is basically both a Front end and Backend developer.
We develop our own interfaces using XML, write the backend code to bring it to life with JAVA, do database integrations, Build and sign apks, Test app and Analyze Apks all in one IDE.... Android Studio□
It's easy to setup and lightweight on your machine.
Android Studio and Android development as a whole is known for crazy bugs and tiring errors but not to worry, there's a ton of resources and help is easily accessible online
To develop Android applications, you require
An IDE - Android Studio (for this program)
Java Development Kit
Android NDK
Android SDK
Not to worry, it's easy to setup + there'd be resources to guide you
Thanks for your time, I'd take questions now
As a thread here

NODE JS Class 1: Introduction to Node Js

Okay. I'll start.
My name is Abasifreke Ekwere and I'm a javascript developer.
I also do PHP/Laravel, flutter and pretty much anything that interests me.
I will like this quick class to be as interactive as possible.
I love JavaScript.
I hate JavaScript
But the obvious question you have or have had since joining this channel is;
What is node
Js?
I could simply quote a google definition and drop here and well, move ahead but let me tell you a story
In the late 90's, Brendan Eich invented JavaScript
Guess how long it took him?
10 days
Many argue that this is the reason you have quirky behaviors like ```2 == '2'``` evaluating to

true where the first 2 is a number and the other 2 is a string.
in a language like java (which is no way related to JavaScript), that will throw an error.
Why is this the case? Java is a static-typed language. JavaScript is dynamic.
static typed ? dynamic? How does this fit in the context of a beginner ?
in a static typed language, you need to declare the type of a variable before even going on to do anything with it:
e.g
Number char = 2
this simply says whatever you give to char (a variable) must be a number
if i did something like this;
Number char = '2';
it would throw an error
here's what a static typed language does.
you declare a variable say
char
guys, think of variables as containers that hold information.
those information are values
so for an expression like
num = 1
num is a container that holds 1
so to retrieve and reuse 1, you simple call on its container
num
it will always give you 1.
@here
do you get this analogy? good.
java does this
you declare num = 3
it looks for the type declaration.
here's a type declaration;
Number num
you explicitly state that it's a number;
it does not find a type declaration so throws an error that you should declare a type
you declare num = 3
it's simply saying; better tell me what you want your container to contain before hand so i can look out for that in an orderly manner
okay so you proceed to do this:
.
.
Number num = '2';
java looks for a type declaration, gives you a thumbs up
tries to store the value in
'2'
in that container
lol there's a problem
what the problem ?
@here
you can't store a string in a number container
this is very important and helpful when you grow into programming
what will javascript do ?
you do:
var num;
var is just a syntax you use to declare variables in JS
there are others like; let and const:
very helpful but have deep underlying reasons as to why you could use them
we will stick with var for now for the sake of simplicity
you could also do;
num;
that will become a global variable;
in javascript, you have the concept of environments
on the browser, you have
window
on the server, you have
global
we, node js developers are mostly only concerned with global
this is just a side note: let's go back to what javascript would do;
it will simply just set
num
to undefined and let your application go on gracefully
so if you tried to access
num
you get undefined;
what will happen if you did;
var num = '2';
?
do you get an error or not if you tried to access num? you don't get an error
javascript looks at the value about to be stored in num and internally determines its type and stores it in its container
num
(edited)
does it gracefully without errors.

This is a very powerful feature of dynamic typed languages like javascript.
you can use it to your advantage or you can let it give you nightmares
i assume you all understood that concept well enough
the difference between loosely typed (dynamic) and static typed languages.
okay i don't want to bore you so much tonight. i just wanted to introduce an easy yet very important concept generally found across programming languages
In 2009, javascript was brought to the server side invented by Ryan Dahl
We call it Node JS.
it offers a way to communicate with a database, build a server that can communicate with a client and even more
when you visit HNG board and try to log in, there's something that helps identify you
a server and a database
that server could be built with Node JS
when you click that log in button, the browser sends your details to the server
the server already has implementations set in place to check your details
your details are stored in a database like MySQL, Mongo DB etc
so we write scripts to check your details against a database to confirm it really is you and we let you in or just tell you you can't go in
that's simply what node js is in a nutshell.
what resources would i recommend for you to learn node js as beginner ?
okay. how did i even learn node js ?
i wrote my first line of code on April, 2nd, 2019
it was a simple "Hello World"
i used an app called "solo learn"
a few months later, i joined the pioneer set of start ng
i had just meddled into javascript slightly
that's where it all got serious
just like it is just getting serious for a number of you here
i had a burning passion and desire to learn so i literally wrote code everyday day and moment i had
i started at
w3schools.com
officially.
they have a very nice javascript section
also,
freecodecamp.org
is worth your time too.
they both have very nice node js sections.
here's what you should know, you own your personal progress.
A tutor can give you tips, directions and impart knowledge as they can
but your progress is determined mostly by you
visit those two sites and start getting acquainted with the basics of node js
so you can be ahead of the class and also discover things you may not be told in class
very soon, we will be pouring out snippets and code.
i'm here for the beginners. if you're an expert, sorry for painlessly watching me explain concepts you knew since 1960
here's a resource to get you started:
https://m.youtube.com/watch?v=fBNz5xF-Kx4
YouTube
|
Traversy Media
Node.js Crash Course
here's a reference material:
https://www.w3schools.com/nodejs/
it is important you go through the video especially and use the reference material to clear a blocker
your tasks will revolve around them
Before we get started, does everyone have node install on their computer ?
Open up your terminalSearch for terminal on your computer, make sure it is openand type node -v
Today we will be diving straight into code.
3
You should see something like this when you run that command
if you no not see that, visit here:
https://nodejs.org/en/download/
download node js and type it afterwards..
you will see it
A simple challenge;
Create a folder and name it after your slack username:
Create a file within that folder and name it:
app.js
inside your
app.js
file, write
console.log('My name is *insert your name* and I love Node JS')
open your terminal under the directory you are currently at,to make it easy, you should be using VScode.
in vscode, you can press
CTRL + J

to open/ close a terminal below your workspace
once your terminal is open, type
node app.js
I will move on
We will be creating a simple server.
Clear the console.log from your app.js file
make sure you have an empty file;

```
var http = require("http");http.createServer(function(req, res) {
res.writeHead(200, { "Content-Type": "text/plain" });
res.end("Hello World!");
})
.listen(8080);
```

Paste the above snippet in that empty app.js file.
Do not worry, i will go explain what it does
run
node app.js
visit
http://localhost:8080/
on your browser after running that
Ladies and gentlemen, you have successfully created a server and you should find that fun
okay.what does it do ? how can i open that on a browser and see what I just wrote on my editor ?
is it magic ?
oluwa wetin dey occur ?
Think no further.
Yesterday I talked about environments
i presume your server is still running
type
CTRL + C
on your terminal Make sure to click on the terminal first
that
terminates the server
okay do this:
type
node
it takes you to the next line with a symbol like this:
>
@here
do you see that ?
excellent.
type
console.log(global)
awesome.
do this next;
console.log(http)
okay great
the http module is part of the global environment
to use it, you need to import it to your desired file location
there are two major ways to import files and modules in javascript
1.
The ES6 way
example:
import http from 'http'
2) The ES5 way
var
http = require('http');
The first one is called an ES6 import
The second is called the common JS import
The second method is native to the node global environment and is heavily recommended
The first mostly requires transpiling by a tool like babel to be better understood by systems and browsers
so as node js developers, we stick to the second way of importing.
@here
do you follow ?
http is an object found inside the global object
in javascript, you can nest objects within objects and within objects
lemme visualize this object for you

```
var global = {
http : {
createServer: function(){}
listen: function(){}
}
}
```

do you understand what is going on ?
@here
global is an object that has properties and methods

```
var intern = {
track: "Backend", // property.
submitTask: function() {
console.log('just submitted my task')
```

} // method.
}
in the context of above example, what would you consider http, createServer and listen ?A method or a
property
so let us proceed
http. createServer() calls the createServer method
the create server does what it implies it creates a server
when we move to express in the long run, you will see a very elegant way of doing this
not the express that egungun enters o
Express JS - The Node JS Framework
then we listen on a port
listen(8080)
you can listen on any port of your choice but it is nice to use ports like
3000, 8000, 8080
makes more sense sematically imagine visiting
localhost:1000
so we set headers
var http = require('http');http.createServer(function (req, res) {
res.writeHead(200, {'Content-Type': 'text/plain'});
res.end('Hello World!');
}).listen(8080);
the createServer method takes in a callback
what is a callback ?
In javascript, functions are first class citizens
Just keep that at the back if your mind
I will explain that much much later
So you can pass functions as an argument
just like we pass a function to createServer
that function that is passed to createServer, takes in two arguments too
req, res
req and res are objects
how do I know this ? it is part of the node js implementation
var http = require("http");http
.createServer(function(req, res) {
console.log(req, res);
res.writeHead(200, { "Content-Type": "text/plain" });
res.end("Hello World!");
})
.listen(8080);
paste the above guys and visit
localhost:8080
then check your terminal.
a screenshot fo what you see in your terminal after visiting that link
these are the response and request objects
they do no make sense surely but they have methods and properties so they are objects
var http = require("http");http
.createServer(function(req, res) {
console.log(req, res);
res.writeHead(200, { "Content-Type": "text/plain" });
res.end("Hello World!");
})
.listen(8080);
so res.writeHead() is a method inside the response object
that method sets a header for what will be sent to the browser200 is an html status code.the content type here we say should be a plain text
we also need to end the session with res.end() so our browser does not keep spinning
do not believe ?
try this.
var http = require("http");http
.createServer(function(req, res) {
console.log(req, res);
res.writeHead(200, { "Content-Type": "text/plain" });
res.write("Hello World!");
})
.listen(8080);
notice that i used
res.write('hello world')
do CTRL + C in your terminal
do
node app.js
again
visit
localhost:8080
does it stop loading ?
try this
var http = require("http");http
.createServer(function(req, res) {
console.log(req, res);
res.writeHead(200, { "Content-Type": "text/plain" });

res.write("Hello World!");
res.end();
})
.listen(8080);
and follow the steps i highlighted
it will show
Hello World
this time
you can see
res.end()
(edited)
that terminates the session
we have successfully understood how to create a simple server
@channel
I will be releasing your task shortly.
It will build upon what i just taught and will require you to do a bit of research.
it will be very simple and only needs you to google which is what developers do.
That will be all for now guys. We will be holding our classes regularly.
Almost every day.
At the end of this program, you will leave with a very good understanding of node JS.
you will also play around with express and write an API.

NODE JS Class 3: Introduction to Databases

Today, we will be talking about databases and do operations on one ourselves
kingabesh
@here

what is your definition of a database?
Your first task was designed to show you how to write data to somewhere, albeit, a file
You will make use of the fs module as you proceed and when you start writing CLIs and certain packages

(edited)
The database

is a systematic collection of data databases support storage and manipulation of data.

Databases

make data management easy
The main operations you carry out on a database are:
Insert, Read, Update and Delete
We have two major types of databases
Relational databases and non-relational databases
Relational databases are well suited for

creating relations

and make use of tables, rows,
and columns
Non-relational databases make use of documents, collections, and fields
If we are to create a comparison
Relational Non-relational
Tables === Documents
Rows === Collections
Columns === Fields
Examples of relational databases are but not limited to

MySQL, Microsoft SQL, Oracle
Database, PostgreSQL
Example of non-relational databases are:

MongoDB

, DocumentDB,

Cassandra

, Coachbase,
HBase

,

Redis

, and

Neo4j

I highlighted the words

creating relations
In the world of databases, you are better off associating items with their owners.
Let me use you guys as examples
Every intern here must have a slack username:
You are related to your slack username in the world of databases
This is helpful so we can do stuff like

getting all slack details for a particular intern
That relation enables us to do this
This is just what you need to know
Should in case anyone decides to ask you what relations are
Non-relational databases do not prioritize creating relations where you can relate items with their owners of course
The key difference to note is, relational databases make use of tables but non-relational databases do not
That is easily the difference.
Our main focus today is on a database that goes well with node
Mongo DB
You can use any database with node but this is the most popular and for reasons, you might find interesting
-
It is fast
-
Collections are just like plain javascript objects so you have an easier time wrapping your head around them
A document is made of many collections
Matter of factly speaking, data is stored like objects in non-relational databases and javascript engineers love objects
a collection has fields just like an object would have a field
{
name: "Node JS",
email: "nodejs@gmail.com",
phone_no: 07012345678
address: "start ng road, node, HNG"
}
the above is what a collection looks like
the

name

,

email

,

phone_no,

and

address

are fields on the collection
the above could be a user collection. That could be how you are stored in our database
Why don't we create our own database?
@here

you will all be creating databases now
1). Install Mongo DB
- Open up your terminal
- Run npm install mongodb
React here once you have done so:
5 replies
create a folder or use the same folder you used for the previous class:
- Create a file called mongo_db.js
- Write var mongo = require('mongodb') at the top of this file;
React here once you do so:
@here
2) Now we create a database.
- To create a database in MongoDB, start by creating a MongoClient object, then specify a connection URL with the correct IP address and the name of the database you want to create.
MongoDB will create the database if it does not exist, and make a connection to it.
The above may seem like too much talk:
Let me show you in code:
Paste this in the file you just created.
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/mydb";

MongoClient.connect(url, function(err, db) {
if (err) throw err;
console.log("Database created!");
db.close();
});
What does this do?
The MongoClient comes with MongoDB
It is an object.
I guess at this point, we all understand that that connect is a method on the MongoClient object

@here
right?
connect does what it says, connect your app to MongoDB
connect takes in an error object and a callback
we talked about callbacks in the last class, it is a function that is passed into a function.
Run node mongo_db.js
What do you see on your terminal?
v
6 replies
View thread
@class run

mongo --versio
n

and send a screenshot
+2
9 replies
View thread
@class:
navigate here:
C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
on your terminal
go to program files
click on Mongo DB
click on Server
click on 4.2
Click on bin
click on mongo
Do this and send a screenshot

@here
+2
8 replies
View thread
@here

did you run npm install MongoDB as instructed?
3 replies
View thread
Capture.PNG
@here

to get it running like that you need to have your local mongo db server running
Those who navigated to that path, can run

node mongo_db.js

and send a screenshot again
When you do not have your server running
Capture.PNG
@here

send a screenshot if you have successfully run

node mong
o_db.js
See where to find your server and start it for windows users
Capture.PNG
@here

you should get this
mongodb_1.png
@JamilUmar

Please explain how you got that for the class
The floor is open for you
JamilUmar
First, install MongoDB:

npm install MongoDB
3:51
once install is complete put down the following code

```
var mongoClient = require('mongodb').MongoClient;
var url = 'mongodb://localhost:27017/mydb';
mongoClient.connect(url, function (err, db) {
if (err) throw err;
console.log('Database created!');
db.close();
});
```

just run node mongo_db.js which is the name of your file
if everything is successful you should get the message Database Created
else an error is thrown
done sir
kingabesh
@here

you can install MongoDB server here
https://www.mongodb.com/download-center/community
hit install the MongoDB server above first
then you can run npm install MongoDB in your project's directory
navigate to the mongod.exe and start it
then, you can optionally start mongo.exe
Capture.PNG
those are your local servers...
MongoDB is the server
mongo is just a terminal you can use
we will play there next class
The class is adjourned for now.
Please reach out to your fellow classmates for help on how to install the MongoDB server
Next class, we will be running database commands

Principles of Software Development

Don't overthink things:

Problems are just that, problems... You should not approach problems with a "problematic" mindset. Approach problems with a solution mindset. If you try to solve a problem by "attacking" with more problems it won't work. For example, if you get a question, and after you read it the first time, you felt so lost and maybe immediately started having a headache... Somewhere in your mind "problem has started", and if you do not learn to control and feel, take a step back, calm yourself down... and read it again, you might not be able to solve the problem. Even if you do, you might end up with a solution that is not full-proof.
So, here is what you should do when you get a chance to fix something...
Have a first read, just go over it. don't think much about it.
Then read it again, this time, point out things using bullet points on a paper or on your computer.
Look at each point and see if you have correctly captured everything in the problem.
Each point you have, is a percentage of the problem. Each of the points you are able to resolve shows how many percent of the problem you have been able to solve.
After going through each point, getting solutions for each, look at the question again, and see what you might have missed.
In summary, do not overthink things, learn to calm yourself, and make it simple for yourself. No matter how complex it appears, break it down into simpler units. Doing these, I strongly believe the problem is already half-way solved; and then coding becomes easier.

Identify what works for you:
In school, I'm sure a lot of you have experienced this, there are people who never attend class, and some serious minded people (those who sleep in class) look at them as "failures", and oftentimes, when results come out, they do better than the rest... This is because some of them have been able to identify what works for them... In fact, it is a big problem in Nigeria... Do not be that person, look at yourself, try things out, and stick with what works for you. If working at night works for you, do it...Stop wasting the night candle because of a stereotype. Don't be a statistic. Be intentional.
No matter what you know, you can still learn more: This is easy to explain, but I'll still say it anyway... You can learn from anyone! I have been doing this for less than 5 years, I have met those who have been in the field for more than 20 years... and one thing they all have in common, "eagerness to learn from anyone and everyone". Do not look at people as "below you", no matter how many complex infrastructures you have built, a beginner that started 2 weeks ago could still teach you how to center a div (which by the way, is the terrible thing).

Always write human-readable codes:
Don't write unnecessary complex looking codes.... Don't have the mindset of "whoever is going to take this job next will die!" (I actually heard someone say this).
I took on a project (for a radio station), and after 2 weeks of trying to understand the code, we decided to just rebuild the entire thing... the former developer wanted to remain needed and wanted, he expects the company to call him back to come and fix things... This is bad work ethics! Your relevance should not be hidden in "deceit" or broken code. Write clear and well-structured codes...

Add comments to your code:
There is a school of thinking that says if your code is simple enough, then you don't need comments. But I still say, add comments. It might be impossible to write very simple codes every time, somethings are just like that, so, still include comments when you write codes.

Use descriptive names for functions/methods/variables/classes etc:
Do not name things as though you are building a treasure hunt. The next developer should not have to read 4 pages of codes to understand what a variable does. For example, if you are using a variable to collect the email address of users, just use var email or var user_email Don't do things like var eU or var Ue. It makes more sense to me, the developer coming after you, to see user_email than eu.

When not sure what to do, brute-force first, then optimize later:
Brute force simply means, "do it anyhow, make sure it works" then make it better later... It's not something you should do every time, but, during live coding interviews or pair coding, or even real work environments, users (product owners) care more about results, then optimized code (at first anyway).
I worked with a man who has no idea the meaning of HTML, but he has a lot of things he wants built, and he likes to see result... if I tell him "I am working on a brilliant algorithm to solve this thing and permanently fix it", he'll ask "where is it"... and so, I learned to make things work first, then optimize later... because he just wants to see it ... and this applies in a lot of ways...
However, I forgot to mention this earlier... "Shorter code is not always better, write readable and easy to understand code". For example, using the normal "IF ELSE" statement is better than using the ternary operator sometimes...

Documentation is important:
Developers hate documenting, but it is a sign of professionalism. As we go along in the course, you might still be taught how to do that, but, for now, see the documentation as a user manual for your code. If the things we buy don't come with manuals, then, it will be misused.
Finally, for today, the last point I have is,

Test as you go:
There is something called TDD, Test Driven Development, you can look it up, it's a style of programming that states that you should write tests before writing the actual code, but, this is not what I mean by the point... What I mean is, don't write 3 features, and 3000 lines of code to prove that you "sabi" work... It's kind of childish to do that... Except when you are having a good time with your friends, it is not acceptable to keep writing untested code over untested code
You can read more here: https://docs.google.com/document/d/10CK1dRx005gbqRrGyGbfDWrZQgfpPb-0Cph6n8y1JZo/edit?usp=sharing

NODE JS Class 4: Basic CRUD operations (Part I)

The challenging part of the assignment tasked you guys to write code to create an intern collection in the database. The assignment specifically asked that you create a separate file called interns.js and call it from your db.js. But most of you did something else. A lot of you guys were repeating the code you used to connect your database, in the interns.js file
const assert = require('assert'); this line imports the assertmodule that provides some sort of testing service for our code. So it ensures(by means of assertion) that we are doing the right thing.
I saw some of you use it in your submissions, that's supercool.
I then created the insertDocumentsfunction which takes in the logic to insert the Docs in our database

```
const insertDocuments = function(db, callback) {
// Get the documents collection
const collection = db.collection('documents');
// Insert some documents
collection.insertMany([
{a : 1}, {a : 2}, {a : 3}
], function(err, result) {
assert.equal(err, null);
assert.equal(3, result.result.n);
assert.equal(3, result.ops.length);
console.log("Inserted 3 documents into the collection");
callback(result);
});
};
```

Different database types, perform CRUD operations in their own unique way so the methods you will learn here for mongoDB will be different from SQL databases like mySQL and postgreSQL
In order to CREATE in mongodb, we use the insertOne method, which obviously is used in inserting just one document into the database, and the insertManymethod for documents more than one.
As you can see from the code above I'm inserting 3 documents, in the form of objects in an array
[{a : 1}, {a : 2}, {a : 3}]
We can also notice the methods also take a callback function, where you can deal with any errors and results generated.
result argument in the callback function is an object. logging it in the console returns the following:

```
{ result: { ok: 1, n: 3 },
ops:
[ { a: 1, _id: 5e8cccb43940620947778247 },
{ a: 2, _id: 5e8cccb43940620947778248 },
{ a: 3, _id: 5e8cccb43940620947778249 } ],
```

insertedCount: 3,
insertedIds:
{ '0': 5e8cccb43940620947778247,
'1': 5e8cccb43940620947778248,
'2': 5e8cccb43940620947778249 } }
it's from the result object, I was able to pass result.result.ninto the second assert.equal method and result.ops.length into the 3rd assert.equal method
To make the insertDocument function available throughout our work folder, we export it using module.exports = insertDocument
After exporting the insertDocument function in my insertdocs.js file, then I can go back to my index.js file to import it and call it.
In our index.js file, the first pic I uploaded, you can see where I imported it, the line const insertDocuments = require('./insertdocs');
Then it was called inside the client.connect() method as follows
insertDocuments(db, function() {
findDocuments(db, function() {
client.close();
});
});
The line with findDocuments was from another file for reading documents from the database, which is what we'll consider next
Let's take a short break and use the next 10 minutes for Questions. Kindly post them under this post as a thread.
@murewaashiru asks if the insertDocument function can be written as an async/await function. It definitely can be written as one
@Haywhy mongodb doesn't log them on it's own. They are logged because I'm logging the result parameter in the function. It's the result parameter that contains result, ops, insertedCount and insertedIds.
@vipul-aggarwal-099 my mongod server is running, and it's through that I can connect to my mongodb database. I'm sure you're referring to http servers, I'll touch on that in the next class.
Let's continue @here. How do you READ data from the database.
You use the find() and findOne() methods
In my findDocuments.js file, I have the following code:
const assert = require('assert');
const findDocuments = function(db, callback) {
// Get the documents collection
const collection = db.collection('documents');
// Find some documents
collection.find({}).toArray(function(err, docs) {
assert.equal(err, null);
console.log("Found the following records");
console.log(docs)
callback(docs);
});
};
module.exports = findDocuments;
The findOne() method returns the first occurrence in the selection.
To return specific data based on some query, we can pass in 2 parameters - the first parameter it takes is a query object, the second is a callback function that handles any errors and the results of our query.
The find() method returns all occurrences in the selection
It also takes 2 parameters as the findOne() method, and queries can be passed in to make it return some data(instead of all available data)
the toArray() method returns data found based on the find({}) query as an array of objects.
To return some data, rather than all the data, we pass in a projection object into the find() method as find({}, { projection: { _id: 0, a: 1} })
The data in your projection object is determined by what sort of data you have in the collection you're querying. You can see from earlier that my database have key: values data as [{a : 1}, {a : 2}, {a : 3}], and the _id is automatically ascribed to each data in your database as they are being created. So my projection is querying the database, asking it to exclude the _id's but include the a's. The result of my query is seen below:
[{ a: 1 },
{ a: 2 },
{ a: 3 },
{ a: 1 },
{ a: 2 },
{ a: 3 },
{ a: 1 },
{ a: 2 },
{ a: 3 },
{ a: 1 },
{ a: 2 },
{ a: 3 },
{ a: 1 },
{ a: 2 },
{ a: 3 },
{ a: 1 },
{ a: 2 },

{ a: 3 },
{ a: 1 },
{ a: 2 },
{ a: 3 },
{ a: 1 },
{ a: 2 },
{ a: 3 }]
I promised the class was going to be for 2 hours, I've been interrupted so many times by external factors most of which I can't control, that's why the class has been going kinda slow. I couldn't be apologising for each and every delay. Let's agree on a time we can complete the class tomorrow please, as I have to go now.

Node JS Class 4: Basic CRUD Operations (Part II)

Yesterday, we stopped at READ operations. I mentioned 2 methods to achieve this from the database - find() and findOne().
I also mentioned how find() returns all data in the collection, and findOne() returns the first data in the collection.
To return some of the data, we can pass in an empty query object {} as the first parameter in the find() method, with a projection object as a second parameter.
To find specific data however, the query object will be populated with key:values data, depending on the specific data we want.
Imagine I have the following data in my collection:
[{first_name: "Daniel", last_name: "Craig", email: "danielcraig007@mi6.com"},
{first_name: "Roger", last_name: "Moore", email: "rogermoore@mi6.com"},
{first_name: "Pierce", last_name: "Brosnan", email: "piercebrosnan@mi6.com"}]
@channel what do you think would be the result of
collection.find({last_name}).toArray(function(err, docs) {
assert.equal(err, null);
console.log("Found the following records");
console.log(docs)
callback(docs);
});
Nice attempts... But it would actually return an error. The query is an object so it expects key:value pairs.
So you must pass a value to last_name otherwise it gives an undefined variable error
So I'll modify my query to be something like
collection.find({last_name: "Craig"}).toArray(function(err, docs) {
assert.equal(err, null);
console.log("Found the following records");
console.log(docs)
callback(docs);
});
Which will return all last_names that are Craig
To get all last_names we'll use the projection object as I explained yesterday.
You can also query your database using regex(regular expressions). If you don't know what that means, you can read up on that from many available web resources.
I'll post resources I find useful later
The query object will be formatted as before, except that the value will be the regex you want to pass in, to filter your database.
So if I want first_names starting with a D, I can write my query as
collection.find({first_name: /^D/})
To delete items in your database, there are deleteOne() and deleteMany() methods.
As their name implies deleteOne() deletes a single record, or document as it's called in mongoDB. It accepts a query parameter that specifies which document to delete. If there are many documents that matches the query, it deletes the first document it encounters.
The deleteMany() method deletes more than one documents, depending on the occurrences returned by your query.
If you want to delete an entire collection, then the drop() method is used.
You can also use the dB.dropCollection() method, which takes in 2 parameters, the name of the collection and a callback function.
To modify values in your collection, the methods updateOne() and updateMany() are used. They take in 3 parameters: first, a query object, second, an object containing the new values to update to and then the callback function.
The second object looks like this {$set: {first_name: "Timothy", last_name: "Dalton"}}
Pay attention to the $set operator. It'll tell the database what new data to update the queried data to.
As you must have known by now updateOne() only updates the first occurrence of the query, even if the query returns many data.
Those are all required methods we can use to perform basic CRUD

operations in mongoDB with Nodejs.
Let's take questions at this juncture. Please ask as a thread on this post.
Let's try to drop our questions within 10 minutes. The class has officially ended for today, but I'll be available to answer questions till 9.30pm.
Thanks (edited)
Q: What's the difference between drop() and dropCollection() methods?
A: Because I'm not with a system, I'll take a code from an example I can find online to explain. I'll get back to you before 9.30pm on that
Q: Hello Sir. Nice class. A lot to take in. My question is concerning the previous task. Unlike what you said, the instructions actually said we should name the file that handles the collection creation anyhow we want and not interns.js . Did this affect the scores?
Secondly, I observe that instead of creating collections like I did... you used insertions to create the collection. Is this a recommendation and did it affect my score?
A: The naming didn't affect your scores. The instructions asked you to create a separate file, your documents, and call it from your index.js or db.js or whatever your main database connection logic was. But most of you didn't do exactly that. There were a lot of cases where the separate file contained all the logic required to create its own connection, and log the "interns collection created" to the console, whereas it was meant to contain only the documents to be created, then called from your main database logic file, like what I did earlier.
Q: Thanks a lot sir. Please can you throw more light on the projection object. Secondly can creation of collection, update, find -the operation and methods, can they be written on a single file and used or we need separate files.
A: Projection is an optional second parameter you pass into the find() method, if you need the database to return many documents that match a particular projection. The first, and required parameter is the query object, which can be just an empty object {}, if you want all documents in a collection, or contain key:value pairs to return specific documents matching your query.
Also all the logic can be in one file, but it'll make your work clumsy. It's usually advisable they're all in separate files and exported, so you can only call them when needed in any particular file, where they're required.
Have a great night rest y'all. The next task will be given tomorrow, based on our classes yesterday and today, definitely with something that challenges you further.

WE WOULD BE TALKING ABOUT GIT & GITFLOW

Objective of this Class
After this class you all should have a better understanding of the following major concepts
GIT COMMIT
COMMIT MESSAGE
GIT FLOW

1.GIT COMMIT
Commiting in git means you are ready to save the changes you have made so far while working in your repository.
Ohk, committing your changes might be confused with saving your files on any system.
But the simple differences is that
When you use ctl+ s on windows or cmd + s on mac or any other methods you use for saving your files, it simply means you are saving the files on your local device, such changes is not saved in your repository. if you use git push a thousand times, such changes will not be saved to your online repository. I'm sure many of use are not new to the command git push
When you commit your changes, it means that when you push from your local repo, the changes you have made will be saved to your online repository.
Now, for the purpose of this class,
I made the following assumptions:
You have logged in your account into git,
You have created a repo online and
You have cloned the repo into your device
The purpose of these assumptions is to know you have connected your local repo with your online repo. So whatever we do in this class these 3 points are expected to have been carried out.
For a better understanding,
I will use a case study, so you can all understand the concept I'm to talk today.
For those in the frontend family, your piggyvest task will be used as case study
In your repo,
you have created an index.html and main.css
You have also written all the codes you want to get the page up and running.
git commit and commit messages ; these 2 go together and will explain both together.
It is best to commit changes stage by stage.
So let's say you have written all the html codes from beginning to end and you want to commit.

2. COMMIT MESSAGE

These are the following steps to take to commit your messages.
Step1:Save Your File

You can do this locally by pressing ctrl + s on windows and cmd + s on mac.
Step 2: Do git status
note: whenever you see this message DO ===> it means write that command in red or orange on the command line(TERMINAL) and press enter.
The common app you use, on Windows is Git Bash and on mac there is a terminal that comes with your laptop or you can also install Git bash.
or finally, there is a terminal you can use in your vscode - my favourite actually :heart:
This is a sample of what your will see when you do git status the first time.
https://startng2020.slack.com/files/UUSGXF70U/F011X8MJLG1/screenshot_2020-04-10_at_19.38.02.png
Untracked files:
  (use "git add <file>..." to include in what will be committed)
This message you see here means there is still another step to do before you finally commit your changes and that step is the next one I will talk about now.
Step 3:Do git add index.html
While working in your repo, there are 2 phases, the working tree and the staging.
when you initially save your file locally, it is stored in the working tree and that is why you see it in red, telling you that you have made some changes and it is in working tree.
when you do git add index.html it tells git, move this file from working tree to staging.
After doing git add index.html now do git status this time the color of your file will change from red to green meaning the file is in staging and is ready to be committed.
This is what I have after doing git add index.html.
https://startng2020.slack.com/files/UUSGXF70U/F011X98BSDP/screenshot_2020-04-10_at_19.45.25.png.
BUT YOU MIGHT BE WONDERING, WHAT IF I HAVE MORE THAN index.html file and I want to add everything to staging for committing,
fear not git has you covered. at this point instead of doing
git add index.html
Do git add .
note the dot
git add . - will add all the files in your working tree to staging and at this point, all the files will be green.
https://startng2020.slack.com/files/UUSGXF70U/F011NNJA7UM/screenshot_2020-04-10_at_19.50.00.png
at this stage your changes are ready....for the next step.
Step 4: Do git commit -m"built the html structure of piggyvest task"
here is where commit message comes into play.
commit message is that text that will be displayed alongside your changes.
and to add a commit message to your changes
git commit -m "your commit message goes here"
is your best bet
Note the -m and your message must be in a quote, else, git will throw error of too many arguments.
After using the command, the following messages tell you your changes have been committed successfully.
https://startng2020.slack.com/files/UUSGXF70U/F011UL3AA6Q/screenshot_2020-04-10_at_19.58.19.png
And at this point, you have successful saved/committed your changes and added a proper commit message your project.
And if your are here, you can easily push your files and the file will be saved in your online repo.
Step 5:Do git push
git push means you are ready to push and save your files on the online repository.

You can view more about this class here:
https://docs.google.com/document/d/1EMc9obV6r9BTEC06gmo9Kw5xlh9NQeqyr1u_yxCMqxg/edit

Software Testing

In preparation for your project, and the final stages, you are required to be able to carry out several types of test on software. Therefore, we recommend you begin preparing for that by studying the following tutorials:

https://www.youtube.com/watch?v=T3q6QcCQZQg
https://www.guru99.com/software-testing-life-cycle.html
http://softwaretestingfundamentals.com/unit-testing/#google_vignette

We expect you be able to answer questions in the following aspects:
Types of testing
Bug reporting
Testing automation tools
QA practices
Test Cases (Building and Validation)

DOMAIN NAMES, HOSTING AND SSL

I have this illustration I use for explaining what domain name stands for, and it goes this way.
Imagine your current address where you live, you have the following (probably): A house, the house apartment, the apartments have numbers, each apartment belongs to someone, each person gives address that points to their specific apartment, not just the building.
In some parts of Nigeria, you get apartment number, house name, street name, and other details, just so that whoever is trying to reach (for example) apartment number 5 won't go to apartment number 9.

A domain name serves as the specific address that points to your specific content on a server that might be shared by many people.
The internet is a big place, there are different things on it, see the domain name as a way to send people to the

part of the internet that belongs to you, or that you want them to see when they try to reach you which means; your domain name, just like your house address should be specific, descriptive, easy to remember and easy to find.

A domain name has two (2)main parts, the name and the extension. For example, xyluz.com the name is xyluz the extension is .com
According to Wikipedia:
A domain name is an identification string that defines a realm of administrative autonomy, authority or control within the Internet.
(You can do more reading on your own, I don't want to repeat what you can easily find on the internet).
In a domain, extensions are quite important, the extension basically gives whoever is trying to visit your website an idea of what they might find.
.org stands for organization, which means this website belongs to an organization, it could be a commercial organization, NGO, etc…
.gov stands for government, means it belongs to a government body, in most countries. .gov domains are reserved only for government bodies.
.com stands for commercial, which means, this website is definitely aimed at selling something to you.
There are other extensions, you can find out more about them.
A domain name is valued based on demand. If the demand for a domain extension is high, then the domain would be expensive.
For example, there was a time .io was quite cheap because people didn't want to buy it. but, when tech companies started buying .io (input output), the price started rising, and now, .io is probably one of the most expensive extensions out there.
Another factor that decides the value of domain is the serviceability of the name.
Names that are more like every day search names are quite expensive.
Things like sugar.com would probably be super expensive (if it is available).
A final factor that can determine the price of domain is, if there is a regulatory body in charge. For example, .ng domains start from minimum 12k per year in Naira. This is because there is a government body in Nigeria (NIRA or so) that sets the base price for domain merchants.

Domain Merchants

People that buy and sell domains. Some of them are companies (that also do hosting) and some are just individuals who buy domains and keep for someone to come and buy at a higher price.
Mark told me he had to pay for hotels.ng, I think he paid $1k for it (I can't remember), that is because it was bought by someone (who was not using it at the time) and the person just waited till someone needed it.
So, it is also possible to do that as a person, search for domain names that would likely be needed by someone later, buy them, and keep. Maybe google will need your domain one day.
Domain values can also increase over time...
xyluz.com used to belong to an Indian company,
I first attempted buying the domain 2011 (I think), I realized it already belonged to him.
I then set up a domain tracking alert, so that each time anything changes about the domain I get an alert.
So, one year he forgot to renew, I renewed only a few hours after his subscription expired. So I just paid for 3 years straight and that's how I got it.
Although, before then, I tried buying from him, he didn't respond....
However, while I was waiting, I bought https://www.xyluzdevelopers.com/ so that I'll have something online to represent my brand.
Your name, as a developer, should be your brand. It should represent you, and I am one of those that believe every developer should have a domain.
To check the value of a domain, you can go to any domain merchant website and see what it is worth.
You can also check whois.net to see who owns the domain (if they are displaying their details.
Apart from the things I have mentioned above, you should also note the following about domain names:
Always buy from reliable merchants
Check the available options, if a merchant does not offer DNS settings, you should not buy from them.
Compare price across different platforms. Sometimes, prices differ depending on the merchant.
Don't leave any domain parked…

-

Parked Domain

A parked domain is a domain bought and left unused. i.e. it is not linked to anything.
This, in my opinion, is a waste of money. Most domain merchant would use your parked domain to display advert.
So, I always believe in at least putting some content on the domain, no matter how little, even a simple coming soon.
If you buy a domain to resell, then, put something on it that says "Available for sale contact blah blah".
Which brings me to hosting…

HOSTING

Remember our housing illustration? How each person has an apartment that belongs to them?
Imagine hosting company as the real estate company, that owns different apartment complex, and rent out each building to different people...
The hosting company basically has managed servers, on this server, they then "rent" out a small portion so that you can move your things (website content) in there.
(this is a general explanation, there are different cases).
So, for example, if Company Pako has 50 servers, each with 1 TB space and you want to host a static website that only requires less than 1GB space; you can go to companypako.com and say, I need 1 GB for my website, then they tell you, I GB is $2 per month.
Then, after you pay, they allocated 1 GB to you from one of their 1 TB servers.
In that 1 GB you can put your content, and finally, point your domain name to that your own content on that server.
I'll read this again when I'm done, just saying it as it's in my head... if any part doesn't make sense to you, ask questions later...
DNS Stands for Domain Name System.
You can read about it. I mean DNS not DNA
DNS is what allows your domain name to be mapped to a server. Servers are identified by IP addresses.
A server can be configured to hold multiple applications, in the configuration, you can decide to point different names to different parts of the server...

For example, start.ng server has different applications in it, and different domain names pointing to the same IP address, each address is configured to only display a specific part of the server which is why: dev.start.ng points to the board, start.ng points to a simple page etc…

Sub-Domains

Subdomain as the name suggests, is a domain under another domain.
start.ng is the domain, dev.start.ng, discuss.start.ng are all subdomains...
Subdomains have all the properties of a domain, except that their names are always associated with a main domain.

SSL

Is an extra layer of security for domains. SSL stands for Secure Sockets Layer
It is an encryption later applied to domain names as they communicate with the server.
There are different types of SSL Certificates, some are free, most are paid.
You can get free SSL certificate from letsencrypt. You can get paid certificates from most hosting platforms. For some applications, letsencrypt SSL might not be enough.
There are SSL Certificates that only says "we are not sure about this company, or the person operating this website, but, we are certain that this domain name is encrypted" which means a hacker can use that SSL. There are SSL certificates that says "We know this company, we have can verify that this company is authentic, and we can verify that this company won't steal from you intentionally".
If you are building a financial app, you definitely want to go for the second type of SSL Certificate. Those also come with labels on your website "Secured by something".
You can find out more here; https://www.liquidweb.com/blog/ssl-certificates/
Ensure you gain trust with your website visitors right away with an SSL certificate. Find out the different SSL certificate types and their features.

Node JS Class 5: Introduction to Express

Please have your laptops and editors open. We will be progressively building an e-commerce API.
Please react if you are ready for the class
@channel. First of all, what have
you learned so far about node is? Please comment as a thread here.
I believe we are now known what node is, but we have seen so far is nothing compared to what you can do with node js. Node JS has a lot of use cases but henceforth, we will be focusing on just building APIs.
What is an API ?
It simply stands for Application Programming Interface. How do you explain API in plain English?
When I think about the Web, I imagine a large network of connected servers. Every page on the internet is stored somewhere on a remote server. A remote server is not so mystical after all — it's just a part of a remotely located computer that is optimized to process requests. To put things in perspective, you can spin up a server on your laptop capable of serving an entire website to the Web (in fact, a local server is what engineers use to develop websites before releasing them to the public).When you type www.facebook.com
into your browser, a request goes out to Facebook's
remote server. Once your browser receives the response, it interprets the code and displays the page. To the browser, also known as the client, Facebook's server is an API. This means that every time you visit a page on the Web, you interact with some remote server's API. An API isn't the same as the remote server — rather it is the part of the server that receives requests and sends responses. - source:
freecodecamp.org
the API basically receives and sends responses. The interface there implies that...
What is an interface ?
In plain english, a point where two systems, subjects, organizations, etc. meet and interact. the request-response cycle is
basically
communication
between the server and a client. It is basically why
you call it
endpoints, as you expose an interface for a client to interact with. I believe you get this easily already. Understanding this is important. Let me give you real life scenarios Let us imagine you are a stadium,
I will use the
santiago bernabeu because
hala madrid
Barca fans will be angry You do not want everybody gaining access to your stadium, But if there was not stadium, how do you even get to watch Real Madrid play at home ?
@here
please someone should be compiling the
class. Imagine that the stadium is an API Remember real madrid has lots of infrastructures and what have you. The fans only meet with Real Madrid at the stadium
The stadium is an API where the fans(clients) can interact with. Real Madrid is the server. So you see, an API is not synonymous to a server. Just part of

the server. Also, you do not want everybody gaining access to the stadium. Real madrid does not allow that. They sell tickets. This is similar to making sure your API can only be accessed if an API key is sent.
NOTE: You do not always need to issue API keys unless it is a project that needs that layer of security. Also, real madrid needs to make sure only people who have passed security checks can gain access to the stadium.
This is similar to authorization. This means that you have been identified as being able to go into the stadium before being allowed access. Most times, you do not want people carrying out some actions with your API unless authorized.
Can this client send delete requests ? Are they authorized to ? Authentication has to do with registration and logging in.
We have real madrid ultras. Registered fans who influence decisions at the stadium. You may want some of your users to be authenticated before accessing some features on your API. This may be useful in tracking their identity and what have you.
Enough story for now.
The big question. What is express ? Some of you are already thinking of Egungun
Express JS is what we want to talk about NOTE: Every mention of Express is synonymous to express js. Forget about egungun for now
Express JS is a fast, minimalist, unopinionated framework for Node JS I just lifted it off here:
https://expressjs.com/
.
But let me break it down. For a start, node js is super duperly fast and express is a wrapper for node js so you expect it to be super fast. Minimalist implies that although being a framework, it does not really add anything new to node js. It is just node js packaged in a presentable way
unopinionated means that despite being a framework, it lets you do what you want.
This is different from many frameworks. Many frameworks do too much and give you little to moderate room for flexibility.
With express, you can do anything and how you want,
Here is how express implements
server.listen(3000)
app.listen = function listen() {
var server = http.createServer(this);
return server.listen.apply(server, arguments);
};
so you can do this:
app.listen(3000) and if you look at the function above,
app.listen(3000) will create a server and listen on port 3000. so you do not even need to call http.createServer yourself. Express will do that for you when you call app.listen() isn't that cool guys ?
Shall we write some code ?
Create a project. Call it
my-express-app Open up a terminal on your project directory - run
npm init -y
run
git init - run
npm install express - React once you are done.
create a file and name it
app.js
const express = require("express");
const app = express();
Paste that snippet that the start of your life
The snippet imports and executes express
give some spaceMaybe 10 lines, the write that magic line
app.listen(3000)
app.use((req, res) => {
res.send("<h1>Welcome to my app</h1>");
});
paste that just some lines below
const app = express()
open your terminal in the same directory and run
node app.js
visit your broswer at
http://localhost:3000/
What do you see ? Post a screenshot as a thread
so did you see how easy it was to get that set up ?
That is the benefit of a framework
but visit here:
http://localhost:3000/jshdhdhdhfhfhffhf
what do you see ?
same thing...
http://localhost:3000/xyluz
http://localhost:3000/kingabesh
http://localhost:3000/egungun
they will all give you the same response. that is not ideal typically, you only

want to catch random routes as an error.
and send a nice error message like:
oops, this route does not exist.
Awesome, so you place generic routes like the above at the bottom of all your declared routes, so
express does a check for all your routes, only send that error message when someone tries to access a non-existent route, good.So now..Let us connect to a database
before that, I hate that we always have to run
node app.js, why not run
npm
install nodemon
go to your
package.json
"scripts": {
"test": "echo \"Error: no test specified\" && exit 1",
"start": "nodemon app.js"
},
post the above, you might see a start script alreadyjust replace it with the above, done ?
Run
nodemon app.js. . . so it works and you never have to do
node app.js
nodemon has taken over.
we will be using mongoose, so please run
npm install mongoose.
visit here:`
https://www.mongodb.com/
` sign in if you do not have an account, please create an account.I will give five minutes for that.You can decide to log in with google all choice, done ?
Okay, Click on
build a new cluster, you need to be logged in to see that. then
you click on
create cluster, Click on
connect, you should see
connect metrics
collections, do you see that
@here
if you already have a cluster, please do not create a new one, you will have to pay if you try to, clicking on connect gives you this: select the second option.Click on
connect your application, you should see
connection
string only.
Click on add your current IP address
once that is done, supply a username and password
note, that username and password is any of your choice and is used to create a user for the cluster
click on
create MongoDB User once that this is done, you will proceed to connect
are you here yet ?
@here
const mongoose = require("mongoose")
import that at the top of your file Note: imports should be done at the top of your files.
mongoose
.connect(
"mongodb+srv://<username>:<password>@cluster0-g8lvt.mongodb.net/shop?retryWrites=true&w=majority",
{ useNewUrlParser: true, useUnifiedTopology: true }
)
.then(result => {
console.log("Database connected");
app.listen(3000);
})
.catch(err => console.log(err));
delete `app.listen(3000)` and paste the above in its position
replace
"mongodb+srv://<username>:<password>@cluster0-g8lvt.mongodb.net/shop?retryWrites=true&w=majority"
with the link you
copied on clicking
copy
at the connection string page
that user you created..put the password you created at the place where you see password, do you see database created in your console ? nice, we have created our database.
We now have a full server running in a standard wayThis is how I create my local nodejs servers

@here
shall we do sign up/sign in ?
My classes tend to be slow but this is to make you understand how core
concepts work
Create a
routes
folder - inside that folder, create a file
auth.js - create
another folder, call it
models, inside this folder, create a file
called
user.js
(edited)
- create another folder, call it
controllers, inside this
folder, create a file called
auth.js
go to the auth.js file in the routes file or
routes/auth.jsGet used to that
convention as we will be having files with similar names so i will always
prepend their folders as seen above Paste this:
const
router
=
require
("express")
.Router
();
so by now, i know you can tell that Router() is a method found on the express
object.
It returns a router object too. See how beautiful and powerful objects are ?
the router object also has methods on them post, get, put, delete, patch.
all are methods on the router object
@here
are you with me ?
router.get("/", (res, res) => {
res.send("This is the express app. You have now entered express");
});
post that in the
routes/auth.js
file go to app.js and import that file. I need you
to import that yourself as you should be used to that already by now initialize
the import to
authRoutes okay I will just show you how to
const authRoutes = require("./routes/auth");
Now, just above that your generic route that shows
welcome to my app Paste
this.
app.use
(authRoutes);
then visit
http://localhost:3000/
Thank you
@kareeba
You cannot import what you did not export This is the
solution.
at the bottom of every file you want to importYou need to add an export Add
this at the bottom of your
routes/auth.js
file - module
.
exports
=
router;
router.get("/",
(res, req) => {
res.send("This is the express app. You have now entered express");
});
the above is the correct version.
We have our app set up now and we can do a lot lot , ut we will adjourn, From
next class, we will not be visiting URLs, we will be using a rest client why ?
Browsers can only do get requests when you visit a link, Registration is
always a post request, I will need you to download a rest client
It is called postman.
In the next class, we will register and sign up a user
I will take you through the whole process and handle all edge cases
Be ready to download more packages.
We have a few days left for lectures so we will have to move very very fast.
That would be all for tonight.
Please ask questions if you have
@here

and remember to download postman
Okay. No questions. Glad that you all understood. - See you next time.
@here
never reveal your cluster password. So I should not be seeing your password if you ever decide to post your Mongo URI for any reason whatsoever

SOFTWARE TESTING Class

What is software testing? It is the act of carrying out activities that ensure the quality and standards of an application are met. The person who carries out these activities is a Software Tester.

As a software tester, you are responsible for ensuring software is free of any defects and that the minimum requirements for the software to be launched are met.
Now you may say, you've seen the phrase Software Tester, Quality Assurance Software Tester, Software Testing Engineer and similar.... They are all Software Testers with slightly different responsibilities or skills.

Software testing can be divided in two main types: Manual and Automated. From the name, they should be somewhat obvious. Manual testing involves the tester going through the process of testing step by step without the use of any special tools used in speeding up the process of the test. Automated testing involves the use of special tools to set a test process on autopilot (the computer does all the testing work by itself, you just wait for results).
STLC (Software Testing Life Cycle): This simply is the start and end of the test of a software in a Software Development Life Cycle. It involves creating necessary documentation needed for the tests, testing the product and creating test reports. It's more detailed than you think.

Software Testing is still a developing career especially in Africa. In other parts of the world testers are employed. As a matter of fact, automation test engineers are very much required. Salary expectations in Africa may not be encouraging because most companies haven't started investing in Testers just yet. So if Software Testers aren't employed in most companies who then ensures the software released is bug free or meets expectations? Some companies utilise in-house staff, Project Managers and even the software developers to perform tests.
Software testing is a fun career path, but even if you don't take it up as a career you must still understand some concepts involved in testing software. A lot of use will end up as developers of some sort or designers. Even as a developer or designer, you must know how to test anything you are doing.

As a developer, would you just churn out six thousand lines of code and bundle into an app or website and discover nothing works when you deliver your project to your client? And even if it runs, how about the checkout button? What happens if someone enters an invalid email address, what should happen next? Should a user be able to open their browser console, change the price of a product in the browser and checkout with the altered price?
How?
As a designer, does your design reduce the number of steps a user would take to register an account to the minimum? If a frontend developer sees your design, will they be able to interpret your implementation without explanation?
Testing…

A little underrated but can reduce bad user experience significantly if done thoroughly…
What are the types of software testing?
I'll call Manual and Automated the modes of testing... The types of testing are Functional and Non-functional testing. And these can be further broken down into sub-types.

Functional testing can be broken down into:
• Unit Testing
• Integration Testing
• System Testing
• Sanity Testing
• Smoke Testing
• Interface Testing
• Regression Testing
• Beta/Acceptance Testing
Non-functional tests can be broken down into:
• Performance Testing
• Load Testing
• Stress Testing
• Volume Testing
• Security Testing
• Compatibility Testing
• Install Testing
• Recovery Testing
• Reliability Testing
• Usability Testing
• Compliance Testing
• Localization Testing

Some of you might be wondering, how will you test a software?
Take for example, a client approaches you to build an app like Truecaller for example. it has the following features:
- Ability to call one number at a time
- Ability to add no more than 4 people to an existing call
- Ability to minimize the app during a call

- App must be themed with blue and white colours.

These are the expectations of the app. When you build, you're supposed to test each feature to be sure it works.

• If you can't call one number at a time, that's an issue

• If you can't add up to 4 more people to an existing call, that's another issue

• If you can't minimise the app during a call or you minimise the app and each time the app crashes or disconnects all calls, that's an issue

• If you do not use the specified colours according to the design guide, that is an issue.

But the test doesn't end there... Software Testing involves going beyond the obvious checks. Can someone tell me something else to look out for? Now you have an idea of testing...Now you have an idea of testing...And will save you from bad reviews. Before we go…

There are facts and myths in testing

Fact: everyone on this planet is a software tester.

Myth: you need to be technical or learn programming to test software.

You wonder why I said everyone is a software tester? Imagine Facebook releases a new WhatsApp update and when Iya Oloja downloads the update, the app starts crashing each time she tries to view someone's status? If she knows how to, she'll send an in-app bug report to Facebook. That makes her a tester... A Black Box Tester (story for another day). She has no idea of the internals of the app (whether it was written using Kotlin or Java, whether it's for Android 50 or iOS 25). She's only seeing and reporting as is... It's a functional bug.