ROAD MAP Front-end Web Développement

***Les fondements du développement Web :***

* Comprendre le fonctionnement des sites web et la contribution du HTML / CSS & Javascript
* Internet et le protocole « http »
* Comprendre comment faire du debbuging et le troubleshooting (Best Practices), en utilisant Git/Gihub et le versioning

***HTML 5 :***

* L’anatomie Syntaxe HTML5
* Comprendre HTML « boilerplate » & HTML « Doctypes »
* Structuration du Texte en HTML
* Liste ordonnée et non ordonnée
* Insertion des images dans un document HTML
* Hyperlinks and Anchor Tags
* HTML Tables and layout
* HTML best Practices et HTML sémantique
* HTML Divs et la séparation pour le CSS.

***CSS 3 :***

* Comprendre le Cascading et le CSS « behind the Scene » (CSSOM)
* Les sélecteurs et les propriétés
* Inline, internal and External CSS?
* CSS sizing method ( Grid ou Flexbox)
* CSS structure, spécificité et la hiérarchie du Code CSS
* Class VS ids et l’ancrage
* CSS display et layout
* Fonts styling et les web fonts.
* Centrage des éléments en CSS
* CSS float and Clear
* Combinaison des sélecteurs CSS (sélecteurs et pseudo sélecteurs).
* CSS Animations and transitions.

***Trainning first part: Voir Repo Github (ateliers CSS)***

***Bootstrap 4 :*** <https://getbootstrap.com/>

* Les fondements du responsive web design
* UX/UI et web design (déjà vu)
* Installation du Framework bootstrap
* Comprendre Bootstrap Grid Layout
* Bootstrap containers
* Bootstrap components (Buttons, Slider, forms etc.)
* Symblos & font awesome
* Bootstrap Carousel
* Bootstrap Cards and bootstrap navigation bars.

***BONUS CSS :***

***SASS (pre-processor CSS)***

Variables

Partials

Mixins

Imports

Media queries

Operators

Functions

Inheritance

Loops

SASS Frameworks

***Grunt or gulp: automatisation***

Installation de environment

Creating and configuring tasks

Templating

Tasks workflow

Working with git

Building Grunt plugin

***JavaScript ES5/ES6:***

* ***JavaScript fundamentals :***
* Using the console
* Variables : let var and const
* Data Types
* Type conversion
* Numbers et Math objects ( math.floor () & math.random() )
* String and method of string
* Arrays and Arrays method
* Dates & time
* Switches
* Functions ( declaration , Expressions , Callback)
* Loops and loops for arrays
* Window objects
* ***DOM :***
* What’s the DOM
* Examinate the Document object Model
* DOM selector for single HTML Element
* DOM selector for multiple HTML elements
* Creating HTML elements
* Removing replacing Elements
* Events and Event-listener
* Mouse events
* KeyBoard events
* Local session storage
* ***OOP: Oriented Object programming.***

In this section, I invite you to see the concept in general after using a programming language, I mean the oriented programming object in details and after that attacking JavaScript oriented object programming (Classes, Object, Constructors, Inheritance, Encapsulation, Calling objects etc…)

Hint: to be master Oriented Programming language, I invite you to manipulate this in JAVA Language by installing ‘NETBEANS IDE’ and practicing some algorithm like (FIZZ BUZZ, SORTING, SEARCHING etc…).

* ***JavaScript OOP :***
* Constructors, keyword ‘this’
* Protoypes
* Prototype inheritance
* Using Object.create
* ES6 classes
* Using Sub Classes
* ***Asynchronous JavaScript :***
* What’s asynchronous JavaScript
* AJAX & XHR
* XHR Object Methods Working With Text
* Working With Ajax JSON
* Data From an External API
* REST APIs HTTP Requests
* Callback Functions
* Custom HTTP Library (Ajax With Callbacks)
* The Fetch API
* Arrow Functions
* Custom HTTP Library (Fetch With Promises)
* Async Await
* Custom HTTP Library (Fetch With Async Await)
* ***Error handling :***
* Error Handling with Try...Catch
* Regular Expressions
* ***Go Far in JavaScript :***
* Iteratos Generatos
* Symbols
* Destructing
* ES6 maps
* ES6 Sets
* Javascript Pattern