[Meta Back-End Developer Professional Certificate](https://www.coursera.org/professional-certificates/meta-back-end-developer)

The Full Stack

Week 1: Introduction to the course 3-2-2024 0725  
intro to the course. Will teach things I have somewhat of a grasp on, but there was one thing mentioned that I have never heard of *N-Tier architecture*.

Week 1: A day in the life of a full stack developer<video> 3-2-2024 0731

Week 1: What do you hope to learn? 3-2-2024 0731

* [The Full Stack - Discussions | Coursera](https://www.coursera.org/learn/the-full-stack/discussions/all/threads/24_l25eBEe6x-ApTkXfOSw/replies/6UvBT9iqEe6muwr_9fW95w) My comment

Week 1: Course syllabus 3-2-2024 0741

Week 1: How to be successful in this course 3-2-2024 0747

* Nothing new!

Week 1: Working with labs and exercises in this course 3-2-2024 0753

* Nothing new! But I read it again for review.

Week 1: What is full stack development? 3-2-2024 0803

* Backend Stack – Python, JavaScript, Django, DRF ect. Builds tools, caching applications, and databases.
* Front end – Web and Mobile interface. HTML, CSS and frameworks. JavaScript, Typescript, React, IOS and Android.
* DataStack – SQL, NoSQL, My SQL, MariaDB, PostgreSQL. Redis for caching.
* Full Stack – Applications back end, Process data, and present data via API’s.

Week 1: N-tier architecture 3-2-2024 0825

* N tier architecture mean separate machines handling a stack such as the Database server or the web server. 3-tier could be data, webserver and client machine. The 4th, if used is a delivery tier.

Week 1: Recap: What you know about client-server architecture 3-2-2024 0850

* Thin -clients : Just displays the information from the server.
* Thick-clients: Does some heavier processing with the data on the client side.

Week 1: Knowledge check: Introduction to the full stack<mini-quiz> 3-2-2024 0901

* 5 of 5

Week 1: Module summary: Introduction to the full stack<Review> 3-2-2024 0905

Week 1: Module Quiz: Introduction to the Full Stack

1 - 70% 3-2-2024 2 90% 3-2-2024 3 87.5% 3-2-2024

4 – 87.5 % 3-3-2024 0640

Week 2: How are HTML and CSS used in the real world? 3-3-2024 0649

Week 2: Semantic tags and why we need them 3-3-2024 0654

Week 2: Semantic HTML cheat sheet 3-3-2024 0745

* [Semantic HTML cheat sheet | Coursera](https://www.coursera.org/learn/the-full-stack/supplement/guED4/semantic-html-cheat-sheet)

Week 2: What is Hyper Text Markup Language? 3-3-2024 0753

Week 2: Semantic tags in action 3-3-2024 0840

* I followed the video, but the output was different. It seems clear the presenter in the video had an actual style.css file somewhere where I did not. The output of the presenter’s html had a log and blue writing. I did not. I also never had an actual logo.png or did I write CSS code.
* This was the first code I wrote since upgrading to copilot and it was impressive.
* I needed to change <h3> tags to <h2> to follow the video exactly. I am trying to automatically change the </h2> closing tag when the opening tag is changed. Gemini told me this is possible by editing emmet in vsc. I clicked on emmet.triggerExpansionOnTab in settings and added {"emmet.includeLanguages": {"html": "html", // or "javascript": "javascriptreact" },"emmet.triggerExpansionOnTab": true} to the settings in VSC. It did not activate the automatic edit of the closing tab as Gemini suggested. Not really sure what it did and likely will never know.

Week 2: Forms and validation 3-3-2024 0905

Week 2: Input types <List types and sample code> 3-3-2024 0910

* Go back to this lesson after I learn more about forms and try each type in a real form.

Week 2: Form submission 3-3-2024 0921

Week 2: Submit 3-3-2024 0924

* It seems I can put a form straight into the html file. <form></form>, <form action=”/login”></form>, <form method=”get”></form>, <form method=”post”>

Week 2: Knowledge check: HTML 3-3-2024 0949

* I had already answered 5 of 5 correctly but retook the test on the above date/time. Initial retry I got 4 of 5 and missed “what is the benefit of adding the required attribute to an input element?” I got it on my second try, but obviously need to hit this quiz in the future.

Week 2: Additional resources 3-3-2024 0945

* [HTML meta tags](https://www.dofactory.com/html/metatags)
* [Semantic elements](https://www.freecodecamp.org/news/semantic-html5-elements/)
* [Client-side validation of forms with HTML5](https://www.sitepoint.com/client-side-form-validation-html5/)
* [<input> tag in HTML](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input)
* [Form validation examples](https://www.the-art-of-web.com/html/html5-form-validation/)

Week 2: CSS web layout 3-3-2024 0956

* *CSS web layouts control how webpage elements are positioned and displayed. Modern layouts use flexbox (flexible rows/columns) or grid (2D grid system) for responsive designs.*
* **Block Layout (Traditional):** Basic structure using stacked block-level elements.
* **Flexbox:** Arranges elements in rows or columns with control over alignment and spacing.
* **Grid:** Creates complex, table-like layouts with precise control over rows and columns.

Week 2: Widely used selectors 3-3-2024 1015

* *CSS selectors are patterns used to target specific HTML elements for styling. They provide precise control over which elements on your webpage receive your custom styles.*
* **Universal selector (\*):** Targets all elements on the page.
* **Type selector (e.g., p, div):** Targets elements based directly on their HTML tag name.
* **Class selector (.classname):** Targets elements that have been assigned a specific class.
* **ID selector (#idname):** Targets a single unique element based on its ID attribute.
* **nth-of-type(n): Selects an element based on its position among siblings of the same type.**
* **nth-child(n): Selects an element based on its position among all sibling elements.**

Week 2: CSS units of measurement 3-3-2024 1025

Week 2: Document flow - block vs. inline 3-3-2024 1043

* *I followed along with the following files.*
* C:\Users\Ron\coursera\meta-backend-developer-professional-certificate\week2-semantic-tags-in-action\index.html
* C:\Users\Ron\coursera\meta-backend-developer-professional-certificate\week2-semantic-tags-in-action\style.css

Week 2: Basic flexbox 3-4-2024 0845

* This took longer than it should have, not because it was any harder, it was easy stuff, but I branched off learning more git version control, VSC commands and added copilot. Copilot is the best thing since sliced bread or at least SSD drives.
* CSS Playground Time: Experimenting with various border options and color combinations to explore design possibilities.
* Because I was trying to follow the lesson plan from the video, and it used jpg files I had to capture the images with Snipit and make 6 jpg files. This took a bit but was fun. It also got me some practice moving files in windows.
* I pushed the lesson to GitHub. This is also something that I am learning to do. I want to retake the earlier Version Contol module because I was not interested in version control at the time and wanted to get to the meat of this course, stupid! If you’re about to start that module or just completed it, please make sure you fully understand it and then use it religiously.

Week 2: CSS grids 3-4-2024 0955

* *CSS Grid layouts provide a powerful, two-dimensional system for webpage structure. This presentation explored how grids use rows, columns, gaps, and functions like repeat() and minmax() for flexible and responsive designs.*
* Interesting concepts. Learned a lot. Theres so much to learn and so many options for every command such as using selectors, the changing colors, borders, grids, background, auto or specific values.

Week 2: Grids and flexbox cheat sheet 3-4-2024 1037

* Created a word document for cheatsheet and saved to GitHub and my cheatsheet folder.
* [Grids and flexbox cheat sheet.docx](https://1drv.ms/w/s!AmfcmSwl5svMxEbWI7LTFfOoooS1?e=0xmtkT)

Week 2

Create a grid layout<lab> 3-4-2024 1132

* I followed the lab instructions and got a slightly different layout. I wonder if im missing a typo or if the instructions are not 100% in line with the video. Not a big deal, I used the exemplar code to make it work.

Week 2: Solution to Create a grid layout 3-4-2024 1143

Week 2: Self-review: Create a grid layout 3-4-2024 1154

* OMG sometimes I wonder about these cheaper courses. The lesson said this *“In this lab, you will be creating a grid layout commonly called the Holy Grail layout.”* Ok that’s cool. But then I get a question on the quiz *“Were you able to get the desired Holy Grail layout?”*. WTH how would I know if I got that. Nowhere was The Holy Grail Layout defined. I did a search on Gemini and then understood and was able to honestly answer the question.

**Gemini**Absolutely! The "Holy Grail Layout" is a classic and sought-after web design pattern. Here's what it's about:

**Features of the Holy Grail Layout:**

* **Three Columns:** A header, a footer, and a main content area with fixed-width sidebars (left and right) flanking the flexible center column.
* **Equal Height Columns:** All columns stretch to the same height, regardless of which one has the most content. This creates visual consistency.
* **No Tables:** Originally, this layout was difficult to achieve without resorting to HTML tables. True Holy Grail solutions use pure CSS for flexibility.
* **Responsiveness:** Modern interpretations of the Holy Grail often include responsiveness to adapt to different screen sizes.

A diagram of a website

Description automatically generatedA black and white rectangular frame

Description automatically generated

Week 2: All selectors and their specificity 3-4-2024 1214

Week 2: Pseudo-classes 3-4-2024 1318

* This video provides a comprehensive guide on using pseudo-class selectors in CSS to enhance web page interactivity and styling based on element states, covering user action states, form states, and specific position-based pseudo-classes.
* **Pseudo-classes**: State-based selectors that allow styling of elements in specific states, like hover or focus.
* A screenshot of a computer

  Description automatically generated

Week 2: Pseudo-elements 3-4-2024 1341

* Pseudo-elements are used in CSS to style specified parts of an element. They allow you to add content or styling that's not directly part of the HTML structure, like the first letter of a paragraph or an icon before a list item.
* Below is some css code
* A screenshot of a computer code

  Description automatically generatedA white background with black text

  Description automatically generatedA white background with black text

  Description automatically generated

Week 2: Knowledge check: CSS 3-4-2024 1421

* Missed one partial and one full questions in this quiz. I did not recall that ‘ch’ is a font sizing nomenclature. Or that flex-wrap is not part of the shorthand flex property.
* The flex shorthand is specifically for setting the flex-grow, flex-shrink, and flex-basis.
* flex: [flex-grow] [flex-shrink] [flex-basis];
* flex-wrap: nowrap | wrap | wrap-reverse;
* flex-flow: [flex-direction] [flex-wrap];
* [Broad overview of layouts in CSS](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout)
* [Detailed overview of flexboxes](https://css-tricks.com/snippets/css/a-guide-to-flexbox/)
* [Detailed overview of grids (1)](https://learncssgrid.com/)
* [Detailed overview of grids (2)](https://web.dev/learn/css/grid/)
* [Commonly used selectors](https://www.geeksforgeeks.org/10-css-selectors-every-developer-should-know/)
* [Combinator selectors](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors/Combinators)
* [Comprehensive list of selectors](https://www.w3schools.com/cssref/css_selectors.asp)
* [Comprehensive list of pseudo-classes](https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes)
* [Comprehensive list of pseudo-elements](https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements)

Week 2: Why JavaScript? 3-4-2024 1439

Week 2: Programming in JavaScript 3-3-2024 1444

* React(2011) is a framework built on top of JavaScript
* Missed the in video quiz. I guessed JS could not interact with the DB. Guess I was wrong.

Week 2: Variables 3-4-2023 1452

* Very much a beginner lesson on JS.

Week 2: Exercise: Declaring variables 3-4-2024 1500

Week 2: Solution: Declaring variables 3-4-2024 1508

Week 2: Solution: Declaring variables<quiz> 3-4-2024 1511

Ugg another oops in this course. I don’t understand why it doesn’t get fixed. Or other errors. The correct answer is Rex, which I originally tried and got wrong. Even the answer says the value of petDog is the printout, which is Rex.

A screenshot of a computer code

Description automatically generated

Week 2: Data types 3-4-2024 1522

Week 2: Operators 3-4-2024 1620

Week 2: Numbers 3-5-2024 0615

Week 2: Strings 3-5-2024 0621

Week 2: Booleans 3-5-2024 0626

Week 2: JavaScript interactivity 3-5-2024 0951

Week 2: JavaScript selectors 3-5-2024 1030

* JavaScript allows you to manipulate HTML pages through the Document Object Model (DOM), changing their appearance and content. Selectors provide a way to target specific HTML elements within the DOM using methods like querySelector, querySelectorAll, getElementById, and getElementsByClassName.
* DOM(Document Object Model): The DOM is a tree-like model of a web page that lets programs change its content, structure, and style.
* Selector: A pattern used to identify specific elements within the DOM for manipulation.
* How to search the DOM by using JavaScript to find specific objects.
* Kewords= ["JavaScript", "DOM", "HTML", "selectors", "querySelector", "querySelectorAll", "getElementById", "getElementsByClassName"]

Week 2: Scoping with var, let and const 3-5-2024 1047

* ECMAScript is the official name of the scripting language standard that JavaScript(JS) is based on. ES5 & ES6 refer to versions of JS.
* JavaScript has different types of scope: global, local (function-level), and block-level (introduced in ES6). Block-level scoping, achieved using let and const, limits a variable's accessibility to the specific code block ({}) where it's declared.
* keywords = ["JavaScript", "scope", "ES5", "ES6"

Week 2: Functions <extremely basic stuff> 3-5-2024 1058

* Functions provide a way to group reusable blocks of code in JavaScript. Using parameters makes functions more flexible, allowing you to pass different values (arguments) each time you call the function.

Week 2: JavaScript DOM manipulation 3-5-2024 1109

* The DOM (Document Object Model) is a tree-like representation of a webpage stored in the browser's memory. JavaScript can be used to interact with the DOM, allowing you to modify the structure, content, and appearance of the webpage.
* keywords = ["DOM", "Document Object Model", "JavaScript"

Week 2: Event handling 3-5-2024 1226

JavaScript can respond to events like clicks on a webpage. You can use the addEventListener method or HTML event attributes to write code that executes when specific events occur.

* There is some cool stuff in this lesson. I need to go over this a couple more times and then use the lessons on some of my html.
* keywords = ["JavaScript", "events", "click", "addEventListener", "HTML", "event handler", ”.console”, “example.com”,”onclick”, ”handleClick”]
* I was able to do the first onClick function on the Coursera video page and got the output, but I tried doing the second on the actual video and did not get a response. Some error saying the video did not allow it. Seems like its possible to prevent modification of certain html code and Coursera has done this on the lesson page. I’ll come back and try this again on the example.com page.

Week 2: Exercise: Web page content update 3-5-2024 1243

* *This lesson teaches how to capture user input on a webpage using JavaScript. You can use the prompt() method for basic input, or create HTML forms with input fields and use event listeners to dynamically update web page content based on what the user types.*
* keywords = ["JavaScript", "HTML", "DOM", "forms", "input", "prompt", "event listener", "event handling"]
* Some interesting stuff. I tried the code on example.com as suggested.

Week 2: Frameworks and libraries 3-5-2024 1321

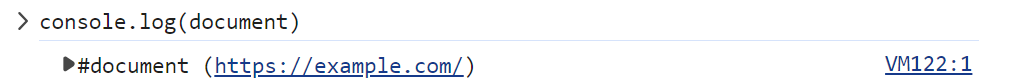
* Frameworks provide a structured foundation for building applications, handling common tasks and imposing design patterns. Libraries offer reusable code for specific functions, giving developers flexibility and saving time.
* keywords = ["framework", "library", "software development", "open source", "structure", "functionality", "opinionated"]

Week 2: Knowledge check: JavaScript 3-5-2024 1338

* Short quiz. 4 or 5 correct. I had no clue about Q2 “What does the document variable return in JavaScript?” console.log(document); I guessed 3 then 1 then obviously 2.Still don’t fully grasp that.

1. The HTML code of the downloaded webpage, as a JavaScript string.
2. The entire web page in the browser's memory, as a JavaScript object.
3. The entire body tag of the web page in the browser's memory, as a JavaScript object.

I ran the command in the console and saw the result: It seems



A screenshot of a computer code

Description automatically generated

* Retry quiz until its sunk in [Knowledge check: JavaScript | Coursera](https://www.coursera.org/learn/the-full-stack/quiz/pOos1/knowledge-check-javascript)

Week 2: Module summary: Front-end technologies<summary video> 3-5-2024 1440

Week 2: Module Quiz: Front-End Technologies 3-5-2024 1359

* First try on module test was 92.5%
* second try on module test was 90% Rewatched [Form submission | Coursera](https://www.coursera.org/learn/the-full-stack/lecture/Gn9ZI/form-submission)

1. I missed Question 2
2. Your web browser is currently at the URL <https://meta.com/hello>. What address will the following HTML form submit to? <form action="login">
3. A group of web address bars

   Description automatically generated with medium confidence
4. You see my guess. The feedback said to use that link I listed above, but that said nothing about this question. Its actually [Submit | Coursera](https://www.coursera.org/learn/the-full-stack/supplement/NXnpe/submit). Oh well I am getting used to it.
5. It is important to note that action can be a full URL address such as https://meta.com, an absolute path such as /login, or a relative path such as login.
6. The absolute path, which starts with a forward slash, will use the base address of the current website, such as https://meta.com and combine it with the absolute path. For example, if /login is the absolute path, the form will be submitted to https://meta.com/login. If the address is https://meta.com/company-info/ and /login is the absolute path, the submission address will still be https://meta.com/login.

* On third try I got 100%. I will retry in a week.

Week 3:Recap: What you know about Django 3-5-2024 1614

* This was somewhat frustrating. I expected it to go over some concepts, but the whole video was the instructor repeating what I knew. Then saying if I missed anything in the recap then go back and review. Almost useless.

Week 3: Recap: What you know about APIs 3-5-2024 1633

* Same as previous video.
* Why it falls short: No Depth: It simply mentions concepts without explaining them. There's no actionable knowledge to be gained from this alone.
* Assumes Prior Knowledge: It's clear this transcript is intended for someone who has already completed an API course and needs a refresher, not for a true beginner.
* Lacks Examples: Real-world examples or code snippets are essential for understanding how API concepts are actually used. This transcript lacks those.
* Could it still be useful? Quick Reference: If someone has prior knowledge, this could be a very fast way to scan for terms they might want to brush up on, prompting them to revisit the relevant course sections.
* "Spark" for Exploration: For a complete beginner, it might give them just enough to be curious to learn more and seek out a proper course.

Week 3: Environment check 3-5-2024 1638

* Just a quick overview of how your programming environment should be set up. Several valuable links.
* [Postman API platform for building and using APIs](https://www.postman.com/)
* [Postman Echo service to test REST clients and make sample API calls](https://postman-echo.com/)
* [Insomnia homepage](https://insomnia.rest/)
* [Getting started with Insomnia](https://docs.insomnia.rest/insomnia/get-started)
* [Httpbin HTTP request and response service](https://httpbin.org/)
* [VS Code](https://code.visualstudio.com/)
* [Pipenv website.](https://pipenv.pypa.io/en/latest/)
* [Git](https://git-scm.com/)
* [Node.js](https://nodejs.org/en/)

Week 3: Optional: Creating a Django project (steps and code)