PatternCodding
Tech Academy

Web Development.

{ Full~Stack }

TAUGHT BY	COURSE LENGTH	COURSE FEE
Dr. Chinedum Patterson https://patterncodding.com	3-6 months	500K

SYLLABUS

T H E C O M P L E T E W E B
D E V E L O P M E N T

W E B D E V E L O P M E N T F U N D A M E N T A L S

- Understand how websites work and how HTML, CSS and JavaScript contribute.
- Understand how the internet works and how websites are served. Learn to use git, GitHub and version control. Learn key
- troubleshooting and debugging skills to apply to your projects.
-

H T M L 5

- Learn the anatomy of HTML syntax to structure your websites. Understand the HTML boilerplate and HTML doctypes.
- Understand indentation and nesting in HTML code.
- Learn to use HTML tags to structure headings, paragraphs. How to
- structure HTML lists to create unordered and ordered lists.
- How to insert images using HTML How to create hyperlinks using anchor tags.
- Create multi-page websites. Learn
- HTML best practices.
-
-

C S S 3

- Understand what are cascading style sheets and how you can use it to style your website.
- How to use CSS selectors and properties.
- Learn about how to use inline, internal and external CSS.
- Understand CSS coding best practices.
- Learn about CSS specificity and inheritance.
- Learn and understand the CSS Box Model.
- Learn to use developer tools to inspect and diagnose CSS.
- Learn about CSS positioning and display properties.
- Learn about font styling using CSS and web safe fonts.
- Implement responsiveness using CSS media queries.
- How to use CSS float and clear properties.
- How combine CSS selectors and understand selector priority.

{PC0}PatternC

SYLLABUS

THE COMPLETE WEB
DEVELOPMENT

{ FULL ~ STACK }

FLEXBOX

- Learn about why flexbox was introduced to CSS.
- Understand what happens when display is set as flex. Understand flex
- direction and how it affects layout.
- Learn to use align and justify for layout of child items Understand how child
- items are sized and distributed in a flexbox and use the flex shorthand
  property.

GRID

- Learn about how grid is different to flexbox and their strengths and
  weaknesses.
- Understand what happens when display is set as grid. Learn how grid
- sizing is done and how to create and arrange a grid.
  Learn to position items in a grid and how placement across rows and
- columns is done.
  Learn to use grid in combination with flexbox to create complex layout
- using native CSS.

BOOTSTRAP

- Understand the difference between native CSS tooling and external
  frameworks.
- Learn to install the Bootstrap framework into your website.
- Understand the Bootstrap 12 column layout system. Learn to use
- Bootstrap components such as buttons, carousels, cards and navigation
  bars.
- Accessing and incorporating designer icons from Bootstrap. Find and
- use bootstrap code snippets, examples and pre-built templates to quickly
  created beautiful mobile-first websites.

WEBDESIGN

- Learn the fundamental principles of web design, UI and UX. Understand
- colour theory and how to choose and use modern colour palettes to
  make your website look professional.
- Understand modern typography and choose free-forcommercial use
  typefaces to make your website user friendly and stylish.

{PC0}PatternC

SYLLABUS

{FULL ~ STACK }

- Learn to manage user attention using the key principles of User Interface (UI) design.
- Learn to design user friendly websites by understanding User Experience (UX) design.

DOCUMENTOBJECTMODEL(DOM)

- Learn the tree structure of HTML based websites.
- Traverse through the document using object notation.
- Separation of concerns and coding best practices. Manipulate and
- change the HTML elements using your understanding of the DOM.

JAVASCRIPTES6

Overview of JavaScript and its uses.
Understand how to work with a code editor and IDEs.
Basic syntax and data types in JavaScript, including:
Variables and their scope (let, const)
Data types (string, number, Boolean, null, undefined, symbol) Operators and expressions, including arithmetic, comparison, and logical operators
Control structures like if/else statements and loops, including:
The for loop and while loop
The for...of loop to iterate over arrays the for...in loop to iterate over object properties Functions and their importance in JavaScript, including:
Function declaration and expression
Function scope and closures
Arrow functions and their syntax Higher-order functions, including:
Passing functions as arguments to other functions
Returning functions from other functions
The map, filter, and reduce methods on arrays and their use with higher-order functions
Arrays and their methods, including:
Creating and accessing arrays
Adding and removing elements from arrays
The slice method for slicing arrays
The concat method for concatenating arrays
The indexOf and lastIndexOf methods for finding elements in arrays
The map, filter, and reduce methods for manipulating arrays The spread syntax ... to spread the elements of an array into a new array or function call

{PC0}PatternC

SYLLABUS

{FULL ˜ STACK }

Object-oriented programming in JavaScript, including: Creating objects with object literals

Creating objects with constructor functions and the new keyword

Creating objects with classes and the class keyword

Adding and accessing properties and methods on objects

The this keyword and how it works in object methods The super keyword for accessing parent class methods and properties

Manipulating objects and arrays using methods and iteration, including:

The Object.keys and Object.values methods for getting the keys and values of an object as arrays

The for...in loop for iterating over object properties The Object.assign method for copying properties from one object to another

R E A C T . J S

- Learn front-end development with React.
- Understand when and how to use React Components.
- Learn to pass Props and work with them.
- Learn to write JSX and understand JSX syntax.
- Learn about the React DOM.
- Learn State Management in React.
- Learn about React Hooks.
- Learn about conditional rendering in React.
- Understand the difference between class and functional components.

T H E U N I X C O M M A N D L I N E

- How to use basic bash commands in a Unix/Linux Terminal. How to
- manipulate files and folders without needing a graphical user interface.
- How to download and install to your computer using command line.

{PC0}PatternC

https://patterncodding.com

SYLLABUS

THE COMPLETE WEB DEVELOPMENT

{FULL ~ STACK }

GIT GITHUB AND VERSION CONTROL

- Using git for version control and collaboration
- Git forking, branching and cloning
- Using GitHub as a remote repository
- Checkout and rolling back changes with git
- Using git and GitHub with Xcode

Understand and use middleware for Node applications

APPLICATION PROGRAM INTERFACES (APIS)

- Understand what APIs are and how they work.
- HTTP in Depth
- Calling APIs
- Reading API documentation
- Basic API Authentication
- Server to server communication JSON vs. XML,
- sending data over the wire.

EJS

- Understand what EJS does and how to use it with Node and Express. Templating with EJS
- Running code in EJS templates
- Passing data from server to template and vice versa Creating
- layouts/partials with EJS
-

SYLLABUS

THECOMPLETEWEB
DEVELOPMENT

BOOTCAMP

DATABASEFUNDAMENTALS

- Data Relationships
- Designing a Data Model
- Relational Databases
- Alternative Databases
- Entity Relationship Modelling (ERM) and Object Relational Mapping (ORM)

SQLDATABASES

- Working with Database Schemas
- Create-Read-Update-Destroy (CRUD)
- Database Joins
- Querying SQL databases

POSTGRESDATABASES

- PostgreSQL queries
- Create-Read-Update-Destroy (CRUD) Postgres Best
- Practices
- Installing Postgres
- Relational Databases
- Implementing One to Many, One to One and Many to Many Relationships.
- Working with Postgres in Node with the pg Package.

PYTHON

Python Basics

- Introduction to Python
- Installing Python and Setting Up Environment
- Python Syntax and Variables
- Data Types and Type Casting
- Operators in Python
- Input and Output Handling

-

Control Flow

- Conditional Statements (if, elif, else)
- Loop (for while)
- Iterators and Generators
- List Comprehensions

Functions and Modulus

- Defining Functions
- Function Arguments and Return Values
- Lambda Functions
- Modules and Packages
- Importing and Using Packages

Object-Oriented Programming (OOP)

- Classes and Objects
- Attributes and Methods
- Inheritance and Polymorphism
- Encapsulation and Abstraction
- Magic Methods and Dunder Methods

SYLLABUS

THE COMPLETE WEB DEVELOPMENT

Exceptional Handling

- Try, Except, Finally Blocks
- Raising Exceptions
- Custom Exceptions

File Handling

- Reading and Writing Files
- Working with CSV and Json Files
- Handling File Path

Data Structure in Python

- List and Tuples
- Dictionaries
- Stacks and Queues
- Linked Lists and Trees

Advances Python Concepts

- Decorators and Generators
- Threading and Multiprocessing
- Regular Expressions
- Memory Management in Python
- Working with Dates and Times

SYLLABUS

Database Integration

- Introduction to Database
- SQLite, MySQL, and PostgreSQL with Python
- ORM (Object-Relational Mapping)
- CRUD Operations in Database

INTRODUCTION TO DJANGO

- Overview of Django Framework
- Installing Django and Setting Up Environment
- Understanding MVT (Model-View-Template) Architecture
- Creating a Django Project and App

Django Model and ORM

- Defining Models and Files
- Database Migrations
- Querying the Database with Django ORM
- Relationships: One-to-One, One-to-Many, Many-to-Many
- Custom Model Managers and QuerySets

Django Views and Templates

- Understanding Views and URL Routing
- Function-Base View (FBVs) vs Class-Based Views (CBVs)
- Template Syntax and Template Inheritance
- Template Filters and Tags
- Context and Static Files Management

Django Forms

- Working with Django Forms
- Model Forms Vs Regular Forms
- Validating Form
- Handling Forms Submissions
- Customizing Form Fields and Widgets

Django Authentication and Authorization

- User Authentication Syntax
- User Registration, Login, Logout
- Password Reset and Change Functionality
- Group Permission
- Custom User Model

Django Middleware

- Introduction to Middleware
- Built-in Middleware
- Creating Custom Middleware

Django REST Framework (DRF)

- Introduction to REST APIs
- Serializers and ModelSerializers
- API Views: Function-Based vs Class-Based
- Authentication and Permissions in APIs
- ViewsSets and Routers
- Paginations and Filtering

SYLLABUS

Working With Databses

- Connecting Django to PostgreSQL, MySQL, SQLite
- Raw SQL Queries
- Database Indexing and Optimization
- Using Django Signals

Advamced Django Topics

- Caching in Django

DEPLOYMENT

- Understand hosting and deployment.
- Hosting static websites with GitHub Pages.
- Deploying server based applications with Heroku. Deploying
- Databases with Mongo Atlas.

BUILDING RESTful APIs

- Understand REST and guiding principles behind API design.
- Learn to work with a MongoDB GUI Robo 3T
- Implementing GET, POST, PUT, PATCH and DELETE by creating a public API from scratch.
- Understand and use chained route handlers from Express.

SYLLABUS

B O O T C A M P

A U T H E N T I C A T I O N a n d S E C U R I T Y

- Understand the need for authentication and keeping user details secure.
- Learn about Encryption and use encryption to keep your database secure.
- Learn and implement Hashing and Salting with bcrypt Using
- Sessions and Cookies to persist user log in sessions.
- Setting up local authentication from scratch.
- Implementing Passport to authenticate users quickly and effectively. Understand and use environment variables to keep secret keys secure.
- Understand and use OAuth 2.0 to log in users using Google and Facebook.

W E B 3 B A S I C S A N D B L O C K C H A I N D E V E L O P M E N T

- Understand Web3 and blockchain development.
- Understand a Canister
- Understand a Motoko actor

D E P L O Y Y O U R C V W E B S I T E O N T O A B L O C K C H A I N

- Set up a cycles wallet
- Obtain free cycles from Dfinity
- Understand how to manage the cycles balance of a canister
- Build and deploy a static, personal CV website to the Internet Computer
- Check how many cycles were used

{PC0}PatternC

SYLLABUS

WORKWITHAREACTFRONTENDANDBL OCKCHAINBACKEND

- Use a React frontend in combination with the Internet
Computer
- Allow users to create their own notes and persist these in a canister
Learn to use CRUD style operations on a canister from the front end
- Learn to use the Candid interface to test the app

-

CREATEYOUROWNCRYPTOTOKEN

- Understand the role of tokens in dapps and open internet services
Create a token modelled on Ethereum's ERC-20 and deploy it on the
- Internet Computer
Create a faucet contract that dispenses small amounts of the token.
- Understand and work with principal IDs to identify users and smart
contracts
- Use the Plug wallet to track token balances
Learn how to transfer the token between user accounts.
-
-

CREATEANNFTANDMARKETPLACE

- Create a non-fungible token modelled on ERC-721 and deploy it to the
Internet Computer
- Build a marketplace website like OpenSea to allow users to mint, buy
and sell NFTs.
- The owner of the NFT can interact with the canister to transfer the
ownership of the .jpeg to another account

{PC0}PatternC