

PROGRAMMA COMPLETO CORSO WEB DEVELOPER FULL STACK

FRONT END DEVELOPER 115 ORE

PROGRAMMING USING HTML AND CSS 40 ORE

1.0 CONOSCERE LE BASI DI HTML (10-15%)

- Costruire marcatori che usano elementi metadati
 - Script; noscript; stile, collegamento; met a tags; inclusa codifica, parole chiavi, viewport e traduci
- Costruisci dei marcatori ben formati che si conformano alle migliori procedure d'impresa
 - Dichiarazione DOCTYPE; HTML; testa, corpo, sintassi propria, incluso chiudere tags e simboli comunemente usati; commenti

1.1 CONOSCERE LE BASI DI CSS (15-20%)

- Analizzare l'impatto di utilizzo degli stili allineati, fogli con stile interno e con stile esterno
 - Quando usare stili allineati; quando usare fogli con stile interno; quando usare fogli con stile esterno; precedenza quando si usa una combinazione di stili allineati e fogli di stile
- Costruire e analizzare serie di regole
 - Sintassi valida per serie di regole CSS; selettori, incluso classe, id, elementi e pseudo-classe
- Costruisci dei fogli con stile ben formattato che si conformano alle migliori procedure d'impresa
 - Riutilizzo di regole e serie di regole; commentare; test su browser multipli, caratteri sicuri per il web

1.2 STRUTTURA DEI DOCUMENTI CON HTML (30-35%)

- Costruisci e analizza marcatori per struttura dei contenuti e organizza i dati
 - Etichette Tabelle; h1-h6; p; br; hr; div; span; ul; ol; li
- Costruisci e analizza marcatori che utilizzano gli elementi semantici di HTML5
 - Etichette semantiche; titolo; nav; sezione; articolo; lato; titolo in basso; dettagli; sommario; figura; intitolazione

- Costruisci e analizza dei marcatori che implementano la navigazione
 - Collegamenti immagine; a; obiettivo; segnalibro; collegamenti relativi contro assoluti; navigare su gerarchie di cartelle semplici
- Costruisci e analizza marcatori che utilizzano elementi di forma
 - Attributi di forma; azione; metodo; invio metodi; accessibilità; input tipologia e restrizioni; seleziona; area testo; bottone, output; scelta, datalist, fieldset

1.3 _PRESENTARE MULTIMEDIA CON HTML (10-15%)

- Costruisci e analizza dei marcatori che mostrano immagini
 - Img e elementi immagine e loro attributi
- Descrivi l'uso appropriato di img, svg, ed elementi canvas
- Costruisci e analizza marcatori che riproducono video e audio
 - Video; audio; traccia; fonte; semplici implementazioni iframe

1.4 CREA DEGLI STILI PAGINE WEB CON CSS (20-25%)

- Costruisci e analizza stili che posizionano contenuti
 - Posizionamento, incluso galleggiamento, relativo; assoluto; ampiezza massima, sovrappiù, altezza, ampiezza e allineamento; allineamento contro blocco; visibilità; modello scatola, incluso margini e riempimento
- Costruisci e analizza stili che formattano testo
 - Famiglia di caratteri; colore; stile carattere; grandezza carattere, ampiezza carattere, collegamento colori; formattazione testo, incluso allineamento e decorazione testo e indentazione
- Costruisci e analizza stili che formattano sfondi e bordi
 - Colore bordo; stile bordo; ampiezza bordo; sfondi; divs; colori
- Analizza stili che implementano una disposizione dinamica semplice
 - Unità di misura; effetti dinamici con CSS, incluso viewport e richiesta media; percentuali contro pixel; cornici e modelli; ampiezza massima

1.0 PROGRAMMA CON OPERATORI JAVASCRIPT, METODI E PAROLE CHIAVE (20-25%)

- Completa o effettua Debug di codici che usano compiti e operatori matematici
 - Assegna; incrementa; decrementa; somma; sottrazione; divisione; moltiplicazione; modulo; operatori di compiti composti
- Applica migliori pratiche JavaScript
 - Commenti; indentazioni; convenzioni nomi; noscript; costanti; parole chiave riservate; parola chiave debugger; punti di rottura di impostazioni, console.log
- Valuta l'uso delle linee interne e degli script esterni
 - Quando usare, come usare e cosa succede quando sono usati entrambi
- Implementare la gestione delle eccezioni
 - Prova; prendi; finalmente
- Completa il codice Debug che interagisce con il Browser Object Model (BOM)
 - Gestisce stato; visualizza dialoghi; determina la grandezza schermo

1.1 PROGRAMMARE UTILIZZANDO DATA TYPES E OPERATORS (25-30%)

- Dichiarare e usare le variabili di tipi di dati primitivi
 - Numero; Booleani; Stringa, Nulla; indefinito; tipo di operatore; tipo di funzioni di controllo; usare stringa; conversione tra tipologie di dati; formattazione numeri; operazioni di stringa; citazione singola contro citazione doppia (annidamento); inizializzazione
- Dichiarare e usare array
 - Array unidimensionali; array multidimensionali; iterazione; inizializzazione; definire un array; gestire e cercare un array; usare push, pop, metodi di shift e unshift; usare le proprietà di lunghezza, accedere a un elemento di array;
- Completare ed effettuare il debug del codice che utilizza oggetti
 - Proprietà; metodi; istanza; oggetto data; recuperare data e parti di tempo; localizzare formato data (MM/DD contro DD/MM); aggiungere e sottrarre le date
- Completare ed effettuare il debug del codice che utilizza funzioni somma built-in
 - Casuale; circolare; abs; pavimento; soffitto; min; max; pow; sqrt
- Completare ed effettuare il debug di una funzione che accetta parametri e consegna un valore
 - Codice riutilizzabile; ambito locale contro globale, ridefinisci variabili, passa parametri, valore contro referenza, consegna valori

1.2 IMPLEMENTA E ANALIZZA DECISIONI E CICLI (20-25%)

- Valuta espressioni che usano degli operatori di paragone e logici
 - ==; !=; <; >; <=; >=; !; &&; ||
- Completa ed effettua debug di comandi decisionali
 - Se; anche se; cambia; annida se
- Completa ed effettua debug di cicli
 - Per; mentre; fai; pausa; continua

1.3 INTERAGISCI CON IL DOCUMENTO OGGETTO MODELLO (15-20%)

- Identifica e costruisci l'albero Documento Oggetto Modello (DOM)
 - Finestra; documento; corpo; altri elementi HTML
- Identifica e gestisci eventi HTML
 - Con cambiamento; al passaggio del mouse; su caricamento; con click; con allontanamento del mouse, con pressione del pulsante
- Completa ed effettua il debug di codici che rimandano a un documento HTML
 - innerHTML; document.write
- Completa ed effettua il debug del codice che colloca, modifica e aggiunge elementi HTML e attributi
 - getElementById; getElementsByTagName; getElementsByClassName; setAttribute; createElement

1.4 INTERAGISCI CON HTML, MODELLI (5-10%)

- Completa ed effettua il debug del codice che recupera input dai modelli e imposta valori campi
 - Recupera valori modello; identifica il comando DOM; ottiene i valori da diversi tipi di elementi; prepopola valori; maschera valori
- Completa ed effettua il debug di codici che eseguono validazione input
 - Caso; confronto stringhe; Not-A-Number (NaN)
- Descrivi il processo di invio del modello
 - Con invio; dopo contro ottieni; obiettivi potenziali per l'invio

BACK END DEVELOPER 130 ORE

PHP DEVELOPER 20 ORE

1.5 AMBIENTE DI SVILUPPO

1.6 WEB SERVER E DATABASE MYSQL

- Comprendere la definizione dei processi che si occupano di fornire i servizi web su richiesta di un browser: I server web Apache e IIS
- Effettuare le installazioni e le configurazioni dei server web Apache e IIS
- Comprendere l'utilizzo di un gestore di basi dati come servizio di un server web: MySQL
- Effettuare l'installazione e la configurazione di un DBMS MySQL in un server Comprendere l'utilizzo del linguaggio di scripting interpretato PHP Effettuare l'installazione e la configurazione dell'ambiente di sviluppo di PHP

2.0 FONDAMENTI DI PHP

2.1 CONCETTI GENERALI

- Riconoscere la sintassi corretta per scrivere codice PHP all'interno di una pagina HTML
- Includere porzioni di linguaggio in varie sezioni della pagina web utilizzando i relativi tag
- Riconoscere le varie tecniche di visualizzazione di una pagina dinamica attraverso l'utilizzo di un server web

2.2 STILE DI PROGRAMMAZIONE

- Riconoscere e saper utilizzare le diverse tipologie di commenti per rendere il codice più comprensibile
- Utilizzare una struttura ad albero per una corretta leggibilità del codice

2.3 STRUTTURE DI CONTENIMENTO DEI DATI

- Comprendere come identificare una porzione di memoria destinata al contenimento di dati
- Riconoscere le diverse tipologie di contenimento dei dati per la gestione degli script PHP, come ad esempio tipi booleani, interi, virgola mobile, stringhe Analizzare i diversi operatori di assegnazione per la gestione dei dati

- Comprendere l'utilizzo del concatenamento di stringhe per la gestione dei testi Gestire le operazioni aritmetiche tra variabili utilizzando i relativi operatori Riconoscere come incrementare e decrementare una variabile
- Analizzare i diversi operatori booleani per la gestione dei dati
- Analizzare gli operatori di confronto utili per la gestione delle condizioni

2.4 GESTIRE L'OUTPUT DEGLI SCRIPT PHP

- Riconoscere la sintassi corretta del costrutto echo per restituire l'output di uno script PHP
- Analizzare il costrutto echo per la stampa di variabili, testo e funzioni
- Comprendere l'utilizzo della funzione var_dump per fornire il valore ed il tipo di una espressione.
- Comprendere la sintassi corretta del costrutto print per restituire l'output di un script PHP.
- Riconoscere i vantaggi e gli svantaggi dei costrutti echo e print.

3.0 LE ISTRUZIONI CONDIZIONALI, DI CONTROLLO E ICICLI

3.1 LA LOGICA CONDIZIONALE

- Riconoscere la logica condizionale mediante l'utilizzo dell'operatore ternario Comprendere come utilizzare i controlli di flusso attraverso l'istruzione IF Riconoscere le differenze tra le istruzioni Else ed Elseif
- Gestire i controlli multipli per il flusso dei dati attraverso l'istruzione Switch

3.2 LE OPERAZIONI RIPETITIVE

- Comprendere come eseguire operazioni ripetitive attraverso l'utilizzo del ciclo For
- Gestire operazioni ripetitive attraverso la logica condizionale: While Comprendere le differenze tra il ciclo While e Do While Comprendere come evitare cicli "infiniti"
- Utilizzare le istruzioni fondamentali per interrompere una routine

4.0 STRUTTURE DATI COMPLESSE

4.1 ARRAY

- Definizione di una struttura complessa vettoriale per la gestione dei dati Comprendere l'utilizzo di una struttura di dati complessa e dei relativi indici Modificare una struttura di dati ed eliminarne i relativi elementi

- Effettuare operazioni di interazione tra gli elementi di una o più strutture complesse di dati
- Utilizzare il costrutto FOREACH per la gestione di strutture di dati complesse Saper effettuare azioni su ogni elemento di un vettore tramite l'uso dei puntatori
- Effettuare ricerche all'interno di una struttura vettoriale

5.0 GESTIONE DEL CODICE PHP

5.1 RIUTILIZZARE IL CODICE PHP

- Comprendere come ottenere il codice di programmazione da un altro file
Analizzare le diverse sintassi per includere unfile

5.2 LE FUNZIONI PER LA GESTIONE DELLE VARIABILI

- Comprendere il concetto di visibilità, o scope, di una variabile Valutare il contenuto di una variabile attraverso la funzione empty()
- Comprendere come verificare se una variabile è definita mediante la funzione isset()
- Eliminare una variabile attiva utilizzando la funzione unset()
- La funzione is_string() per verificare che una variabile sia una stringa
- La funzione is_numeric() per verificare che una variabile sia una numerica

5.3 LE FUNZIONI PER LA GESTIONE DELLE STRINGHE

- Gestione delle stringhe all'interno di raggruppamenti di codice Eliminare tutti gli spazi da una stringa mediante la funzione trim().
- Verificare se una stringa contiene degli spazi iniziali ed eliminarli attraverso la funzione ltrim().
- Verificare se una stringa contiene degli spazi finali ed eliminarli attraverso la funzione rtrim().
- Verificare la lunghezza di una stringa utilizzando la funzione strlen(). Gestire i caratteri maiuscoli e minuscoli di una stringa.
- Comprendere come gestire file di testo con campi delimitati.
Comprendere come creare ed accedere ai file di dati

5.4 LE FUNZIONI PER LA GESTIONE DELLE STRUTTURE COMPLESSE DI DATI

- Comprendere i concetti relativi all'ordinamento dei dati
- Ordinare gli elementi di un vettore in ordine crescente aggiornando le chiavi Ordinare gli elementi di un vettore in ordine decrescente

aggiornando le chiavi Ordinare gli elementi di un vettore in ordine crescente mantenendo le chiavi di origine

- Ordinare gli elementi di un vettore in ordine decrescente mantenendo le chiavi di origine

5.5 LE FUNZIONI PER LA GESTIONE DELLE DATE

- Comprendere il significato di timestamp
- Comprendere come restituire il timestamp di un preciso istante Restituire la data del timestamp in un formato definito dall'utente Verificare se una determinata data risulta corretta

5.6 LE FUNZIONI PERSONALIZZATE

- Comprendere come poter personalizzare una funzione per la creazione di nuove funzionalità
- Gestione delle variabili all'interno di raggruppamenti di codice personalizzato Comprendere come inviare valori ad una funzione
- Gestire i valori restituiti da una funzione

5.7 INVIO EMAIL IN PHP

- Comprendere come inviare email attraverso PHP
- Inserire le intestazioni in un'email per verificarne i dettagli Comprendere come inviare un'email a più destinatari

6.0 RECUPERO E VERIFICA DEI DATI INSERITI IN UN FORM HTML

6.1 RECUPERO DATI DA UN MODULO

- Comprendere con quali metodi sono inviati i dati di un form ad uno script Accodamento dei dati ad una URL di una pagina attraverso l'utilizzo dell'array
- \$_GET
- Invio dei dati di un form in modo invisibile all'utente utilizzando l'array \$_POST

6.2 CONTROLLO DEI DATI DI UN MODULO

- Comprendere come monitorare la compilazione dei campi di un form
- Verificare la lunghezza ed il tipo di un campo di un form per la registrazione di un utente
- Convalidare il campo email di un form Convertire caratteri particolari in entità HTML

7.0 IL MANTENIMENTO DI INFORMAZIONI IN PHP

7.1 LE SESSIONI

- Comprendere come mantenere attivi, durante la navigazione, dati significativi per la gestione delle procedure
- Avviare e memorizzare i dati in una sessione Gestire i dati di una sessione
- Comprendere come è possibile eliminare una sessione

8.0 IL DATABASE MYSQL

8.1 COME AMMINISTRARE IL DATABASE MYSQL

- Comprendere come accedere al database MySQL attraverso la riga di comando
- Effettuare l'avvio e l'arresto dei servizi MySQL
- Effettuare una visione globale dei database e delle tabelle di MySQL Comprendere come creare un nuovo account MySQL
- Aggiungere ed eliminare i privilegi ad un utente
- Modificare le proprietà di un account per l'accesso e la gestione di MySQL Eliminare un account MySQL
- Gestire il backup dei dati MySQL
- Amministrare MySQL attraverso software specifici

8.2 GESTIONE DEL DATABASE MYSQL

- Comprendere come creare, utilizzare ed eliminare una database Riconoscere le diverse tipologie di dati esprimibili in MySQL: i domini
- Gestire le operazioni di creazione, aggiornamento, ed eliminazione della struttura di una tabella
- Gestire le operazioni di creazione, aggiornamento, ed eliminazione di una riga di una tabella
- Estrarre i dati da una tabella attraverso dei criteri prestabiliti: Select. Riconoscere le funzioni di supporto alle query: COUNT(), MAX(), SUM(), NOW(), Like
- Effettuare relazioni tra tabelle: Join
- Riconoscere le classificazioni delle diverse join in una query

9.0 INTERAZIONE CON IL DATABASE MYSQL

9.1 PHP E MYSQL

- Riconoscere il codice PHP per effettuare la connessione e la disconnessione ad un database MySQL
- Implementare uno script PHP per la creazione di una tabella MySQL Inserire i dati in una tabella MySQL attraverso un form
- Visualizzare i dati di una tabella MySQL e formattarli per il web
- Effettuare l'aggiornamento e l'eliminazione dei dati di una tabella MySQL attraverso uno script PHP
- Ottenere il dettaglio di una riga MySQL attraverso uno script PHP
-
- Comprendere come realizzare un motore di ricerca interno ad un database MySQL
- Protezione delle pagine web e gestione degli accessi: login e logout utente

MYSQL DEVELOPER 35 ORE

1. MYSQL ARCHITECTURE

- Use MySQL client programs to interface with the MySQL Server interactively and in batch
- Describe SQL Modes and their impact on behavior of MySQL
- Identify characteristics which have session scope

2. CREATION AND DESIGN OF MYSQL SCHEMA OBJECTS

- Design and create normalized databases
- Create and modify tables using appropriate data types and indexing
- Describe and create table constraints enforcing data integrity
- Creating and modifying views
- Identify and use various methods to obtain metadata for MySQL database objects

3. QUERYING FOR DATA

- Execute a basic SELECT statement
- Limit rows returned by a SELECT statement
- Limit columns returned by a SELECT statement
- Apply sorting to SELECT statement results

- Execute SELECT statements which aggregate and group data

4. JOINS, SUBQUERIES AND UNION

- Identify, describe and use JOINS in MySQL commands
- Describe and utilize subqueries in MySQL commands
- Perform operations using UNION clause

5. BASIC OPTIMIZATIONS

- Identify statements requiring optimization
- Recognize and create optimal indexes for query optimization
- Recognize and fix sub-optimal SQL commands
- Identify appropriate optimization strategies for InnoDB usage
- Optimize performance through data normalization

6. GENERAL MYSQL SYNTAX

- Explain MySQL implementation of identifiers including case sensitivity, qualified names, aliases and use of reserved words
- Identify MySQL data type properties and appropriate usage
- Recognize and use common functions and expressions for all MySQL data types
- Identify and use comment syntax
- Describe and utilize prepared statements
- Describe transactions and transaction isolation levels and the impact they have on database behavior

7. CREATION, DESIGN AND USE OF MYSQL STORED PROGRAMS

- Describe and use triggers
- Create and execute stored procedures
- Create and use stored functions
- Implement error handling within stored procedures

8. MODIFYING DATA

- Describe and execute INSERT statements
- Describe and execute REPLACE statements

- Describe and execute UPDATE statements
- Describe and execute TRUNCATE statements
- Describe and execute LOAD DATA statements
- Describe and execute DELETE statements

9. MYSQL APPLICATION DEVELOPMENT

- Identify key characteristics, features and options for PHP, Java and .NET development using MySQL standard drivers
- Write a basic Java application that uses MySQL
- Write a basic PHP application that uses MySQL
- Write a basic .NET application that uses MySQL
- Interpret MySQL error messages
- Collect available diagnostic information
- Describe and use NoSQL and memcached API

LARAVEL DEVELOPER 75 ORE

1. PHP

- Version 7.1+
- Composer
- Autoloading Standards

2. ARCHITECTURE

- Request Lifecycle
- Service Container Binding and Resolution
- Service Providers
- Facades
- HTTP Verbs

3. ROUTING

- Redirects
- Route Parameters
- Named Routes

- Route Groups
- Route Model Binding
- Rate Limiting

4. MIDDLEWARE

- Defining / Registering Middleware
- Middleware Parameters

5. CONTROLLERS

- Defining Controllers
- Controller Namespacing
- Single Action Controllers
- Middleware
- Resource Controllers
- Dependency Injection
- Route Caching

6. REQUESTS

- PSR-7 Requests
- Request Path and Method
- Retrieving Input
- Uploaded Files
- Proxy Configuration

7. RESPONSES

- Attaching Headers / Cookies
- Encryption
- Redirects
- Flash Data
- Response Types
- Response Macros

8. VIEWS

- Creating Views
- Passing Data to Views
- View Composer

9. URL GENERATION

- Named Routes
- Controller Actions
- Default Values

10. SESSIONS

- Configuration
- Storing Data
- Retrieving Data
- Deleting Data
- Flash Data
- Custom Drivers

11. VALIDATION

- Form Requests
- Manually Creating Validators
- Error Messages
- Validation Rules
- Custom Rules

12. LOGGING

- Configuration
- Writing to Specific Channels
- Creating Custom Channels

13. FRONTEND

- Blade Templating
- Localization
- Asset Compilation

14. SECURITY

- Authentication
- Authorization
- Encryption / Hashing
- CSRF Protection
- XSS Protection

15. ARTISAN CONSOLE

- Generating Commands
- Command I/O
- Registering Commands
- Executing Commands

16. WEBSOCKETS

- Broadcasting Events
- Receiving Events
- Broadcasting Channels
- Presence Channels
- Client Events

17. CACHING

- Drivers / Configuration
- Storing Items
- Retrieving Items
- Cache Tags
- Creating Custom Drivers

18. COLLECTIONS

- Creating / Extending Collections
- Collection Methods
- Higher-order Messages

19. EVENTS

- Registering Events / Listeners
- Queued Listeners
- Dispatching Events
- Subscribing to Events

20. FILE STORAGE

- Configuration / Drivers
- Storing / Retrieving Files
- Custom Filesystems

21. HELPER METHODS

- Arrays / Objects
- Paths
- Strings
- URLs
- Misc

22. MAIL

- Drivers / Configuration
- Generating Mailables
- Writing Mail
- Sending Mail
- Markdown
- Local Development

23. NOTIFICATIONS

- Creating Notifications
- Sending Notifications
- Mail Notifications
- Markdown
- Database Notifications
- Broadcast Notifications
- SMS Notifications

- Slack Notifications
- Custom Channels

24. PACKAGE DEVELOPMENT

- Discovery
- Service Providers
- Resources
- Commands
- Assets
- Publishing File Groups

25. QUEUES

- Drivers / Configurations
- Creating / Dispatching Jobs
- Running Queue Workers
- Supervisor
- Handling Failed Jobs

26. TASK SCHEDULING

- Scheduling Artisan Commands
- Scheduling Queue Jobs
- Scheduling Shell Commands
- Time Zones
- Preventing Task Overlaps
- Maintenance Mode

27. DATABASE

- Query Builder
- Pagination
- Migrations
- Seeding

28. ELOQUENT ORM

- Conventions
- Relationships
- Eloquent Collections
- Mutators / Accessors
- API Resources
- Serialization
- Scopes

29. TESTING

- Creating / Running Tests
- HTTP Tests
- Session / Authentication
- Testing File Uploads
- Available Assertions
- Browser Tests / Dusk
- Data Factories
- Fakes / Mocking