

Core Html Assignment



1. Build a simple webpage that displays text as shown in the below image.

This text will be bolded.

This text will be italic.

This text will be underlined

This text will be highlighted

This is normal text This will be super scripted This is normal again

This is normal text This text will be subscripted

Normal Text Smal Text

~~This text will be deleted~~

2. Build a simple webpage that helps users navigate different web development-related websites. Note: On clicking the hyperlink the web pages should open in a new tab. Below is a reference image.

Navigate Me:

Take me to [PW Skills](#) to buy a course.

Take me to [MDN docs](#) to know more about Web Development.

Take me to [PW Skills Lab](#) to practice live coding.

- 3. Build a simple blog web page with 3 pages home, web development, and web design. Each page must contain hyperlinks to other pages in the top, a heading of the page topic and a paragraph of information. For the home page you can add some information about yourself.**
- 4. Create an ordered list of HTML tags. Each list item must include the tag name and some information about the tag.**
- 5. Create a description list of full stack web development tech stack, using the <dl> tag. Each term should be a tech stack name and each description should be a brief explanation of what the tech stack is used for.**
- 6. Create an ordered list of the full stack web development tech stack HTML, CSS, and JS. For each tech stack, create a table that lists the tech stack name, its primary use cases, and some key features or benefits. Below is a reference image.**

Eg.

1. HTML

Primary Use Cases	Key Features/Benefits
Building the structure of web pages	<ul style="list-style-type: none">Simple and easy to learnCompatible with all web browsersAllows for semantic markup

2. CSS

Primary Use Cases	Key Features/Benefits
Styling and layout of web pages	<ul style="list-style-type: none">Allows for separation of content and presentationEnables responsive designOffers a wide range of styling options

7. Build a complex nested list structure representing a multi-level table of contents. Use unordered lists () and list items () with inline-block styling to create a structured layout. Apply formatting tags to enhance the presentation of list items.

Output should look like this:

Table of Contents

- [Part 1: Introduction](#)
- [Part 2: Getting Started](#)
 - [2.1 Installing the Software](#)
 - [2.2 Creating a New Project](#)
 - [2.2.1 Project Templates](#)
 - [2.2.2 Customizing Settings](#)
 - [2.3 Exploring the Interface](#)
 - [2.3.1 Toolbar Features](#)
 - [2.3.2 Panel Layout](#)
 - [2.3.2.1 Docking Panels](#)
 - [2.3.2.2 Tabbed Interface](#)
- [Part 3: Advanced Topics](#)
 - [3.1 Working with Plugins](#)
 - [3.1.1 Installing Plugins](#)
 - [3.1.2 Plugin Configuration](#)
 - [3.2 Customizing the UI](#)
 - [3.2.1 Changing Themes](#)
 - [3.2.2 Configuring Shortcuts](#)
 - [3.3 Optimizing Performance](#)
 - [3.3.1 Caching Strategies](#)
 - [3.3.2 Resource Minification](#)
- [Part 4: Conclusion](#)

8. Create a table to display a conference schedule. Each row corresponds to a time slot, and each column corresponds to a room. Some time slots might have multiple sessions running simultaneously in different rooms. Utilize rowspan and colspan attributes as necessary to accommodate this complex schedule. (use table attribute "cellpadding" to give extra padding in each table cell).

Output should look like this:

Conference Schedule				
Time	Room 1	Room 2	Room 3	Room 4
9:00 AM - 10:00 AM	Keynote	Session A	Session B	Session C
		Session D	Session E	
	10:30 AM - 11:30 AM	Session F		
12:00 PM - 1:00 PM	Lunch Break			
1:00 PM - 2:00 PM	Session G	Session H	Session I	Session J
	Session K		Session L	Session M