2025

LEARNER GUIDE 2025

OCC: Software Developer

NQF 5

WEB DEVELOPMENT

WEB512

2025

CTU training solutions | Creative & Technology Universitas

Table of Contents

Occupational Certificate:

Software Developer

(NQF 5)
(SAQA ID: 118707)

LEVEL	SAQA ID
NQF 5	118707

How to use the guide

The guide will provide an overview off the syllabus and will provide the learning outcomes of the module. It will indicate each major topic that will be covered, as well as the learning outcomes of each topic.
The study guide is NOT a replacement of textbooks and should be studied in conjunction with the required textbooks.
At the end of each study unit there will be a summary, followed by several self-assessment questions. These questions will assist you to prepare for the tests and exams.

The following icons will be used in the study guide:

	Sections in the prescribed textbook that the student needs to study
	Additional reading that the student needs to study
	A video that the student needs to watch
	Class activities to be completed
	Activities to be uploaded to CampusOnline
	Exercises to be completed
	Indicates exercises to be uploaded to CampusOnline
	Flipped Classroom

	Group activities to be completed
	Group activities to be uploaded to CampusOnline
	Relevant points for the student to consider
	Projects to be completed
	Tests to be completed
	Revision questions to be completed
	Webinar to be attended
	A field trip to be attended

Introduction

This Learner Guide provides a comprehensive overview of the module. It is designed to improve the skills and knowledge of learners and thus enabling them to effectively and efficiently complete specific tasks.

Purpose of module

The main focus of the learning in this knowledge module is to build an understanding of these topics: Core Programming, Object-Oriented Programming, General Software Development, Web Applications, Desktop Applications, Databases, Manage the Application Life Cycle, Build the User Interface by Using HTML5, Format the User Interface by Using CSS, and Code by Using JavaScript

Duration

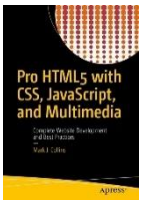
The total notional hours will be allocated according to the table below:

Proposed Roll Out Strategy	01/10/2025 - 07/11/2025
Credits	40
Total Notional Hours:	400
Theory	160
Practical	235
Contact Sessions	16
Formative Assessments	2
Summative Assessments	1

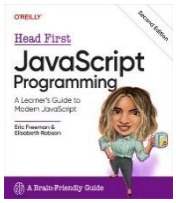
Learning outcomes / Unit standards

- KM-06-KT01: Core programming
- KM-06-KT02: Object-Oriented Programming
- KM-06-KT03: General Software Development
- KM-06-KT04: Web applications
- KM-06-KT05: Applications development
- KM-06-KT06: HTML5
- KM-06-KT07: CSS
- KM-06-KT08: JavaScript
- PM-08-PS01: Creating and Styling HTML5 Pages
- PM-08-PS02: Display data and handle events by using JavaScript
- PM-08-PS03: Create forms to collect and validate user input
- PM-08-PS04: Communicate with a remote data source
- PM-08-PS05: Style text and block elements
- PM-08-PS06: Refine code for maintainability and extensibility
- PM-08-PS07: Create interactive pages by using HTML5 APIs
- PM-08-PS08: Add offline support to web applications
- PM-08-PS09: Implementing an adaptive user interface
- PM-08-PS10: Creating advanced graphics
- PM-08-PS11: Animate the user interface
- PM-08-PS12: Implementing real-time communication by using web sockets
- PM-08-PS13: Create a web worker process
- PM-08-PS14: Package JavaScript for production deployment
- PM-09-PS01: Plan a project to build a solution which effectively solve the customer's business problems (project design phase)
- PM-09-PS02: Configure middlewares and services
- PM-09-PS03: Develop controllers for processing web requests
- PM-09-PS04: Develop views to define the user interface for web applications
- PM-09-PS05: Create code to develop MVC models to interact and model various types of data or objects
- PM-09-PS06: Connect an application to a database to access and store data using an objectdatabase mapper to build a database-driven website in MVC
- PM-09-PS07: Build web applications apply a consistent look and feel to the application
- PM-09-PS08: Develop the client-side of a web application using applicable tools
- PM-09-PS09: Test and troubleshoot for bugs that results in exceptions or unexpected behaviour
- PM-09-PS10: Manage security aspects of a web application
- PM-09-PS11: Increase the speed of data access and communication to improve performance
- PM-09-PS12: Implement web APIs to enable and promote application interaction with external systems
- PM-09-PS13: Host and deploy and web application ensuring it is accessible by clients on a wide
- variety of machines

Module Resources

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/	

Recommended additional reading

Head First JavaScript Programming, 2nd Edition by Eric Freeman, Elisabeth Robson. Available on O'Reilly Books Online at: https://learning.oreilly.com/library/view/head-first-javascript/9781098147938/	

Formative assessment breakdown

Formative 1	Formative 2
Project 1	Project 2
50%	50%

Summative assessment:

Exam
100%

Formative assessments (50%) + **Summative** assessment (50%) = **Final mark**

Practical assessment breakdown

Practical (PM-08)	Practical (PM-09)
Project	Project
100%	100%

Assessment Preparation Guidelines

Formative assessment 1:	**Format of the Assessment:** Students will receive a research project that will require them to study a scenario and develop a solution in the form of a website. **Learning outcomes covered:** • HTML5 **Resources required:** • MS Word • VS Code
Formative assessment 2:	**Format of the Assessment:** Students will receive a research project that will require them to study a scenario and develop a solution in the form of a website. **Learning outcomes covered:** • HTML5 • CSS **Resources required:** • MS Word • VS Code
Exam:	**Format of the Assessment:** Students will write a written theory and practical exam on campus under invigilation. This will be a closed book exam. **Learning outcomes covered:** • Core programming • Object-Oriented Programming • General Software Development • Web applications • Applications development • HTML5 • CSS • JavaScript **Resources required:** • MS Word • VS Code

Practical Preparation Guidelines

Practical (PM-08):

Format of the Assessment:

Students will receive a research project that will require them to study a scenario and develop a solution in the form of a website.

Learning outcomes covered:

- Creating and Styling HTML5 Pages
- Display data and handle events by using JavaScript
- Create forms to collect and validate user input
- Communicate with a remote data source
- Style text and block elements
- Refine code for maintainability and extensibility
- Create interactive pages by using HTML5 APIs
- Add offline support to web applications
- Implementing an adaptive user interface
- Creating advanced graphics
- Animate the user interface
- Implementing real-time communication by using web sockets
- Create a web worker process
- Package JavaScript for production deployment

Resources required:

- MS Word
- VS Code

Practical (PM-09):

Format of the Assessment:

Students will receive a research project that will require them to study a scenario and develop a solution in the form of a website.

Learning outcomes covered:

- Plan a project to build a solution which effectively solve the customer's business problems (project design phase)
- Configure middlewares and services
- Develop controllers for processing web requests
- Develop views to define the user interface for web applications
- Create code to develop MVC models to interact and model various types of data or objects
- Connect an application to a database to access and store data using an objectdatabase mapper to build a database-driven website in MVC
- Build web applications apply a consistent look and feel to the application
- Develop the client-side of a web application using applicable tools
- Test and troubleshoot for bugs that results in exceptions or unexpected behaviour
- Manage security aspects of a web application
- Increase the speed of data access and communication to improve performance
- Implement web APIs to enable and promote application interaction with external systems
- Host and deploy a web application ensuring it is accessible by clients on a wide variety of machines

Resources required:

- MS Word
- VS Code

Formative Assessment Submissions:

Formative assessment:	Assessment description:	Submission:
FA 1	Research (Project 1)	Week 2
FA 2	Research (Project 2)	Week 3
Practical (PM-08)	Project	Week 5
Practical (PM-09)	Project	Week 5

Please note – There are two (2) steps in the submission process.

- Step 1: Required evidence in the specified formats are submitted on Campus Online to the designated assignment description.
- Step 2: Complete and submit document of authenticity for every formative and summative assessment submitted.

Weekly planner 2025

Campus Week	Module Week	Dates	Holidays and exams
Week 1			
Week 2			
Week 3			
Week 4			
Week 5			
Week 6			
Week 7			
Week 8			
Week 9			
Week 10			
Week 11	Week 1	01 Oct – 03 Oct	FA 1, 2 and PM-08, 09 Hand out
Student Holiday			
Week 12	Week 2	13 Oct – 17 Oct	FA 1 Hand in
Week 13	Week 3	20 Oct – 24 Oct	FA 2 Hand in
Week 14	Week 4	27 Oct – 31 Oct	
Week 15	Week 5	03 Nov – 07 Nov	PM-08, 09 Hand in Summative Practical and Theory
Week 16	Week 6	**WIL Simulation**	
Week 17	Week 7		
Week 18	Week 8		
December Holidays			

Lesson Plan:

Week 1:	**Theme:** **HTML 5** • Chapter 1 - 6 **Class activity:** Facilitator to demonstrate practical steps where applicable. **Material requirements:** • Learner Guide • Textbook: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/
Week 2:	**Theme:** **HTML 5** • Chapter 7 - 12 **CSS** • Chapter 13 - 14 **Class activity:** Facilitator to demonstrate practical steps where applicable. **Material requirements:** • Learner Guide • Textbook: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/
Week 3:	**Theme:** **CSS** • Chapter 16 – 20 **Class activity:** Facilitator to demonstrate practical steps where applicable. **Material requirements:** • Learner Guide • Textbook: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Week 4:	**Theme:** **JavaScript** • Chapter 21 - 24 **Class activity:** Facilitator to demonstrate practical steps where applicable. **Material requirements:** • Learner Guide • Textbook: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/
Week 5:	**Theme:** **JavaScript** • Chapter 25 - 26 **Class activity:** Facilitator to demonstrate practical steps where applicable. **Material requirements:** • Learner Guide • Textbook: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/
Week 5:	• **Summative Exams** • **International Exams (ITS- HTML&CSS and ITS-JavaScript)**

Week 1

Day 1 – Learning Unit 1

Introduction

This module will introduce the basic syntax for HTML, CSS and JavaScript. Further more, this module will explain the head and body element of HTML. The facilitator will cover slide 1 and is linked Chapters 1 to 4 in the Textbook.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcomes

- HTML5
- CSS
- JavaScript

Additional Resources

HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web.
HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

The new standard incorporates features like video playback and drag-and-drop that have been previously dependent on third-party browser plug-ins such as Adobe Flash, Microsoft Silverlight, and Google Gears.

Browser Support

The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features and Internet Explorer 9.0 will also have support for some HTML5 functionality.

The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all have excellent support for HTML5.

New Features

HTML5 introduces a number of new elements and attributes that can help you in building modern websites. Here is a set of some of the most prominent features introduced in HTML5.

- **New Semantic Elements** – These are like <header>, <footer>, and <section>.
- **Forms 2.0** – Improvements to HTML web forms where new attributes have been introduced for <input> tag.
- **Persistent Local Storage** – To achieve without resorting to third-party plugins.
- **WebSocket** – A next-generation bidirectional communication technology for web applications.
- **Server-Sent Events** – HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).
- **Canvas** – This supports a two-dimensional drawing surface that you can program with JavaScript.
- **Audio & Video** – You can embed audio or video on your webpages without resorting to third-party plugins.
- **Geolocation** – Now visitors can choose to share their physical location with your web application.
- **Microdata** – This lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics.
- **Drag and drop** – Drag and drop the items from one location to another location on the same webpage.

Revision Questions Day 1

1. Which of the following best describes metadata elements in HTML5?
    A. They define the main content of the page
    B. They provide information about the HTML document to browsers and applications
    C. They are used to insert multimedia such as images and videos
    D. They define user interaction elements such as buttons

2. Which HTML5 element is most appropriate for grouping content that can stand on its own, such as a blog post?
    A. <section>
    B. <article>
    C. <aside>
    D.

3. The <aside> element is typically used for:
    A. The main flow of document content
    B. Grouping contact information about the author
    C. Supporting or tangential content like sidebars or ads
    D. Reusable standalone content

4. Which of the following elements creates a new section in the document outline?
    A. <section>
    B. <article>
    C. <aside>
    D. <nav>
    E. All of the above

5. What is the main difference between <section> and <article>?
    A. Section is for navigation, while article is for main content
    B. Section groups topical content, while article is for independent, reusable content
    C. Section is for sidebars, while article is for footnotes
    D. Section has no semantic meaning, while article always has semantic meaning

6. Which element is intended specifically to provide contact information for a document or article?
    A. <footer>
    B. <header>
    C. <address>
    D. <aside>

7. Which of the following is NOT a sectioning root in HTML5?
    A. <blockquote>
    B. <details>
    C. <figure>
    D. <main>

8. Which element is used to define collapsible sections of content, often with a <summary> element?
    A. <figure>
    B. <details>
    C. <aside>
    D. <nav>

9. Which of the following grouping elements has been deprecated in HTML5?
    A.
    B. <dir>
    C.
    D. <dl>

10. The <main> element in HTML5:
    A. Can be used multiple times in a document
    B. Represents the primary content unique to the document
    C. Can be placed inside <header> or <footer>
    D. Provides metadata about the document

Class activity 1

Class Activity: Starting with HTML
Objective:

To help students become familiar with the basics of HTML.

Task

Complete the following tasks in VS Code or other text editor

Create an HTML document with the title Hello World and the text in red "My first HTML Document", using the proper layout.

Homework: Facilitator to give additional homework activity

Formative 1 Project

FORMATIVE 1 – (PROJECT 1) TO BE HANDED OUT TO STUDENTS

Project 1

Read through the briefing document with the students and emphasize evidence requirements in the project including submission format.

Brief description

Students will receive a research project that will require them to study a scenario and develop a solution in the form of a Webpage.

Learning outcomes

- **HTML5**
- **CSS**

Formative 2 Project

FORMATIVE 2 – (PROJECT 2) TO BE HANDED OUT TO STUDENTS

Project 2

Read through the briefing document with the students and emphasize evidence requirements in the project including submission format.

Brief description

Students will receive a research project that will require them to study a scenario and develop a solution in the form of a Webpage/Website.

Learning outcomes

- Core programming
- Object-Oriented Programming
- General Software Development
- Web applications
- Applications development
- HTML5
- CSS
- JavaScript

PM-08 Project

PRACTICAL PROJECT – PM-08 TO BE HANDED OUT TO STUDENTS

Project

Read through the briefing document with the students and emphasize evidence requirements in the project including submission format.

Brief description

Students will receive a research project that will require them to study a scenario and develop a solution in the form of a Website.

Learning outcomes

- Creating and Styling HTML5 Pages
- Display data and handle events by using JavaScript
- Create forms to collect and validate user input
- Communicate with a remote data source
- Style text and block elements
- Refine code for maintainability and extensibility
- Create interactive pages by using HTML5 APIs
- Add offline support to web applications
- Implementing an adaptive user interface
- Creating advanced graphics
- Animate the user interface
- Implementing real-time communication by using web sockets
- Create a web worker process
- Package JavaScript for production deployment

PM-09 Project

PRACTICAL PROJECT – PM-09 TO BE HANDED OUT TO STUDENTS

Project

Read through the briefing document with the students and emphasize evidence requirements in the project including submission format.

Brief description

Students will receive a research project that will require them to study a scenario and develop a solution in the form of a Website.

Learning outcomes

- Plan a project to build a solution which effectively solve the customer's business problems (project design phase)
- Configure middlewares and services
- Develop controllers for processing web requests
- Develop views to define the user interface for web applications
- Create code to develop MVC models to interact and model various types of data or objects
- Connect an application to a database to access and store data using an objectdatabase mapper to build a database-driven website in MVC
- Build web applications apply a consistent look and feel to the application
- Develop the client-side of a web application using applicable tools
- Test and troubleshoot for bugs that results in exceptions or unexpected behaviour
- Manage security aspects of a web application
- Increase the speed of data access and communication to improve performance
- Implement web APIs to enable and promote application interaction with external systems
- Host and deploy a web application ensuring it is accessible by clients on a wide variety of machines

Day 2 - Learning Unit 2

Introduction

In this learning unit, we look at HTML elements that are used to mark up text content. This learning unit, we will also look at working with lists and tables in HTML. The facilitator will cover slide 2 and is linked to Chapters 5, 6 in the Textbook.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcomes

- HTML

Additional Resources

HTML Lists
Unordered List (<ul>): Creates a bulleted list.

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>
```

Ordered List (<ol>): Creates a numbered or lettered list.

```
<ol>
  <li>First item</li>
  <li>Second item</li>
</ol>
```

List Item (<li>): Defines an item within an ordered or unordered list.
Description List (<dl>): Creates a list of terms and their descriptions.

```
<dl>
  <dt>Term</dt>
  <dd>Description of the term.</dd>
</dl>
```

Description Term (<dt>): Defines a term in a description list.
Description Details (<dd>): Defines the description of a term in a description list.

HTML Tables
Table (<table>): Defines an HTML table.

```
<table>
  <!-- Table content goes here -->
</table>
```

- Table Row (<tr>): Defines a row in a table.
- Table Header (<th>): Defines a header cell in a table. Typically rendered in bold and centered.
- Table Data (<td>): Defines a standard data cell in a table.
- Table Head (<thead>): Groups the header content in a table.
- Table Body (<tbody>): Groups the body content in a table.
- Table Foot (<tfoot>): Groups the footer content in a table.
- Caption (<caption>): Provides a title or description for a table.

```
<table>
  <caption>Sales Data</caption>
  <!-- Table content -->
</table>
```

- Colgroup (<colgroup>): Specifies a group of one or more columns in a table for formatting.
- Column (<col>): Specifies column properties within a <colgroup>.

Common Attributes

- colspan (for <th> or <td>): Specifies how many columns a cell should span.
- rowspan (for <th> or <td>): Specifies how many rows a cell should span.
- type (for <ol>): Specifies the numbering type for ordered lists (e.g., 1, a, A, i, I).
- start (for <ol>): Specifies the starting value for an ordered list.

Revision Questions Day 2

1. Which element is used to define a table in HTML5?
    A. <tr>
    B. <table>
    C. <td>
    D. <th>

2. In HTML5, which elements are used to define the cells inside a table row?
    A. <caption> and <colgroup>
    B. <td> and <th>
    C. <thead> and <tfoot>
    D. and

3. Why should tables be used in HTML5?
    A. To define the layout of a web page
    B. To organize tabular data such as lists or scores
    C. To replace CSS styling
    D. To create navigation menus

4. Which attribute is used with the <th> element to specify whether the header applies to a row or column?
    A. align
    B. rowspan
    C. scope
    D. colspan

5. What is the purpose of the <caption> element in a table?
    A. To display the table's footer
    B. To merge cells together
    C. To provide a title or description for the table
    D. To define row headings

6. Which group of elements is used to semantically organize a table into sections?
    A. <thead>, <tbody>, <tfoot>
    B. <section>, <article>, <aside>
    C. , , <p>
    D. , <ol>,

7. What is the difference between colgroup and col elements?
    A. colgroup defines rows, while col defines columns
    B. colgroup groups columns, while col defines individual columns within a group
    C. colgroup is deprecated, but col is valid
    D. colgroup creates new cells, while col merges them

8. Which attribute is used to make a table cell span across multiple columns?
    A. rowspan
    B. colspan
    C. span
    D. width

9. In HTML5 tables, multiple <tbody> elements can be used, but only one of which element is allowed?
    A. <thead>
    B. <tfoot>
    C. <caption>
    D. <colgroup>

10. If a cell should span both multiple rows and columns, which attributes can be combined?
    A. scope and span
    B. rowspan and colspan
    C. caption and align
    D. colgroup and thead

Class activity 2

Title: Periodic Table Site
Objective:

Students will create a website with the periodic table to illustrate they're ability to work with tables in HTML.

Instructions:

Complete the following tasks using VSCode or any text editor of your choice.

Create a website with the following characteristics:

- Title: Periodic Table
- Heading: Your Periodic Table (In the color of your choice)
- Using the table element, create a periodic table.

Periodic Table of the Elements

I	II											III	IV	V	VI	VII	VIII
1	2											13	14	15	16	17	18
H																	He
Li	Be											B	C	N	O	F	Ne
Na	Mg	3	4	5	6	7	8	9	10	11	12	Al	Si	P	S	Cl	Ar
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe
Cs	Ba		Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn
Fr	Ra		Rf	Db	Sg	Bh	Hs	Mt	Ds	Rg	Cn	Uut	Fl	Uup	Lv	Uus	Uuo

Homework: Facilitator to give additional homework activity

Week 2

Day 3 - Learning Unit 3

Introduction

In this chapter, we will look at embedded elements, such as video and audio, in HTML. This unit also covers using HTML form elements to gather user input. The facilitator will cover slide 3 and is linked to Chapter 7 and 8 in the Textbook.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcomes

- HTML

Additional Resources

Basic Form Structure:

- <form>: Defines the entire form.
- action: Specifies the URL where the form data is sent upon submission.
- method: Specifies the HTTP method (e.g., GET, POST) for sending data.
- target: Specifies where to display the response after submission (e.g., _blank for a new tab).
- autocomplete: on or off for browser autofill.
- novalidate: Prevents default HTML5 validation.

Input Elements (<input>):

- type="text": Single-line text input.
- type="password": Password input (characters are masked).
- type="email": Email input (includes basic email format validation).
- type="number": Numeric input.
- type="checkbox": Checkbox for multiple selections.
- type="radio": Radio button for single selection within a group (requires same name attribute for grouping).
- type="submit": Button to submit the form.
- type="reset": Button to reset form fields to their initial values.
- type="file": File upload input.
- type="date", type="time", type="datetime-local": Date and time inputs.
- type="color": Color picker.
- type="range": Slider for a range of values.

Common Input Attributes:

- name: Identifies the input field for submission.
- value: Initial value of the input.
- placeholder: Hint text displayed when the input is empty.
- required: Makes the field mandatory.
- disabled: Disables the input, preventing user interaction.
- readonly: Makes the input non-editable but still submittable.
- autofocus: Automatically focuses on the input when the page loads.
- min, max: Minimum and maximum values for number/range inputs.
- maxlength: Maximum number of characters for text/password inputs.

Other Form Elements:

- <textarea>: Multi-line text input.
- rows, cols: Define the visible dimensions.
- <select>: Drop-down list.
- <option>: Defines an option within a select list.
- value: The value sent upon submission.
- selected: Makes the option pre-selected.
- <label>: Associates text labels with form controls for accessibility.
- for: Links the label to an input's id.
- <fieldset>: Groups related form elements.
- <legend>: Provides a caption for a <fieldset>.
- <button>: Generic button element.
- type="submit", type="reset", type="button".

Class activity 3

Title: Form Elements

Objective:

Students will need to demonstrate their understanding of form elements in HTML.

Instructions:

Students will have to use VSCode or any text editor of their choice.

Create a website with a form element that will help a company to gather information from the user.

The following information should be gathered:

- Name
- Surname
- Age – Drop Down List
- Birth Date – Date and Time picker
- Email
- Phone number
- TextArea for writing a short bio
- Favorite music – checkbox
- Submit Button

Homework: Facilitator to give additional homework activity

Revision Questions Day 3

1. What is the primary purpose of the <form> element in HTML5?
   A. To display formatted text
   B. To group input elements for submitting data to a server
   C. To validate JavaScript code
   D. To style web page layout

2. Which attribute of the <form> element specifies the URL where form data is sent?
   A. method
   B. action
   C. enctype
   D. target

3. Which form method sends input data as query parameters in the URL?
   A. GET
   B. POST
   C. PUT
   D. DELETE

4. When using method="post", the form data is:
   A. Added to the URL as query parameters
   B. Sent in the body of the HTTP request
   C. Ignored by the server
   D. Encrypted by default

5. Which attribute is required when uploading files through a form?
   A. autocomplete="on"
   B. enctype="multipart/form-data"
   C. pattern="[A-Za-z]"
   D. inputmode="latin"

6. What is the main difference between the disabled and readonly attributes on input fields?
   A. Disabled fields are submitted, readonly fields are not
   B. Disabled fields cannot be interacted with, readonly fields can be copied but not changed
   C. Both prevent input and behave identically
   D. Readonly applies only to checkboxes and radios

7. Which attribute provides placeholder text inside an input field?
   A. title
   B. label
   C. placeholder
   D. hint

8. Which input type is best suited for selecting a single option from a group where only one value is allowed?
    A. checkbox
    B. radio
    C. select multiple
    D. datalist

9. Which element is used to visually group related form fields with a border and caption?
    A. <section>
    B. <fieldset>
    C. <legend>
    D.

10. Which HTML5 element is used to display progress of a task, such as form completion?
    A. <meter>
    B. <output>
    C. <progress>
    D. <status>

Day 4 – Learning Unit 4

Introduction

This chapter introduces students to applying styles using CSS. We will cover CSS Selectors and Positioning content using CSS. The facilitator will cover slide 4 and is linked to Chapter 9 and 10 in the Textbook.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcomes

- HTML
- CSS

Additional Resources

Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page. It describes how a webpage should look: it prescribes colors, fonts, spacing, and much more. In short, you can make your website look however you want. CSS lets developers and designers define how it behaves, including how elements are positioned in the browser.

While html uses tags, css uses rulesets. CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.

Why CSS?

- CSS saves time: You can write CSS once and reuse the same sheet in multiple HTML pages.
- Easy Maintenance: To make a global change simply change the style, and all elements in all the webpages will be updated automatically.
- Search Engines: CSS is considered a clean coding technique, which means search engines won't have to struggle to "read" its content.
- Superior styles to HTML: CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- Offline Browsing: CSS can store web applications locally with the help of an offline cache. Using this we can view offline websites.

CSS Syntax:

CSS comprises style rules that are interpreted by the browser and then applied to the corresponding elements in your document.
A style rule set consists of a selector and declaration block.

Selector – h1

Declaration -- {color:blue;font size:12px;}

- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- For Example:
- color is property and blue is value.
- font-size is property and 12px is value.
- CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

Class activity 4

Class Activity: CSS positioning

Objective:

To demonstrate understanding Positioning using CSS.

Instructions:

Students will have to use VSCode or any text editor of their choice.

Use the previous 3 class activities and apply CSS styling to the activities to properly position and style the pages.

Homework: Facilitator to give additional homework activity

Revision Questions Day 4

1. Which of the following CSS selectors selects all <p> elements in a document?
   A. .p { }
   B. #p { }
   C. p { }
   D. *p { }

2. What is the correct way to select all elements with the class name featured?
   A. #featured { }
   B. .featured { }
   C. featured { }
   D. *featured { }

3. Which pseudo-class is used to select an element when the mouse pointer is over it?
   A. :active
   B. :hover
   C. :focus
   D. :visited

4. In CSS, what does the selector header + p select?
   A. All <p> elements inside <header>
   B. The first <p> element inside <header>
   C. The <p> element immediately following a <header>
   D. All <p> elements after <header>

5. Which of the following CSS rules applies styles only when the screen width is 600px or less?
   A. @media screen and (min-width: 600px)
   B. @media (width: 600px)
   C. @media screen and (max-width: 600px)
   D. @media screen (<=600px)

6. Which CSS property determines whether an element behaves as a block, inline, or other type?
   A. position
   B. display
   C. float
   D. z-index

7. What is the main difference between display: none; and visibility: hidden;?
   A. Both hide the element but visibility: hidden; removes it from the DOM
   B. display: none; hides the element and removes it from the layout, while visibility: hidden; hides it but leaves its space reserved
   C. display: none; only works for inline elements
   D. They are exactly the same

8. Which CSS property ensures that padding and borders are included in the element's specified width and height?
    A. overflow
    B. box-sizing: border-box;
    C. position: relative;
    D. max-content

9. If an element has position: fixed;, it is positioned relative to:
    A. Its nearest positioned ancestor
    B. Its immediate parent container
    C. The entire HTML document
    D. The viewport (browser window)

10. Which CSS property controls the stacking order of overlapping elements?
    A. float
    B. z-index
    C. position
    D. overflow

Day 5 - Learning Unit 5

Introduction

In this chapter, we will look at applying text styles, borders and backgrounds using CSS. The facilitator will cover slide 5 and is linked to Chapters 11 and 12 in the Textbook.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcomes

- HTML
- CSS

Additional Content

Text Styles

```
/* Font properties */
font-family: Arial, sans-serif; /* Specifies font type(s) */
font-size: 16px; /* Sets font size */
font-weight: bold; /* Sets font boldness (e.g., normal, bold, 100-900) */
font-style: italic; /* Sets font style (e.g., normal, italic, oblique) */
line-height: 1.5; /* Sets the height of each line of text */

/* Text alignment and decoration */
text-align: center; /* Aligns text (e.g., left, right, center, justify) */
text-decoration: underline; /* Adds text decoration (e.g., none, underline, overline, line-through) */
text-transform: uppercase; /* Transforms text (e.g., none, uppercase, lowercase, capitalize) */
letter-spacing: 1px; /* Adjusts spacing between characters */
word-spacing: 2px; /* Adjusts spacing between words */
color: #333; /* Sets text color */
text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); /* Adds a text shadow */
```

Borders

```
/* Shorthand for all border properties */
border: 1px solid #000; /* Sets border width, style, and color */

/* Individual border properties */
border-width: 2px; /* Sets border width */
border-style: dashed; /* Sets border style (e.g., none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset) */
border-color: blue; /* Sets border color */

/* Individual side borders */
border-top: 1px solid red;
border-right: 2px dotted green;
border-bottom: 3px double blue;
border-left: 4px solid purple;

/* Border radius for rounded corners */
border-radius: 8px; /* Applies to all corners */
border-top-left-radius: 5px; /* Applies to specific corners */
```

Backgrounds

```
/* Shorthand for all background properties */
background: #f0f0f0 url('image.png') no-repeat center top; /* Sets color, image, repeat, position */

/* Individual background properties */
background-color: #eee; /* Sets background color */
background-image: url('pattern.gif'); /* Sets background image */
background-repeat: repeat-x; /* Sets image repetition (e.g., repeat, repeat-x, repeat-y, no-repeat) */
background-position: center center; /* Sets image position */
background-size: cover; /* Sets background image size (e.g., auto, contain, cover, % values, length units) */
background-attachment: fixed; /* Sets if background scrolls with content (e.g., scroll, fixed, local) */
```

Class activity 5

Class Activity: Text Styles, Borders and backgrounds

Objective:

To illustrate understanding of how to implement Text Styles, Borders and Backgrounds using CSS.

Instructions:

Students will have to use VSCode or any text editor of their choice.

Use the class activities altered in the previous class activity and apply the following:

- Text styles
- Borders around elements
- Backgrounds

Homework: Facilitator to give additional homework activity

Revision Questions Day 5

1. Which of the following is considered a web-safe font?
    A. Comic Sans MS
    B. Verdana
    C. Roboto
    D. Cabin

2. Which CSS rule is used to embed a custom font stored on your server?
    A. @import
    B. @font-face
    C. font-family
    D. @custom-font

3. In CSS, which property controls the boldness or thickness of a font?
    A. font-style
    B. font-weight
    C. font-variant
    D. font-stretch

4. What does the CSS property text-overflow: ellipsis; do?
    A. Cuts off text and hides it completely
    B. Wraps text automatically into the next line
    C. Displays "..." when text overflows its container
    D. Shrinks the font size to fit the container

5. Which CSS property is used to add shadows to text?
    A. box-shadow
    B. text-decoration
    C. text-shadow
    D. font-effect

6. Which of the following is NOT one of the eight basic border styles in CSS?
    A. solid
    B. ridge
    C. groove
    D. shadow

7. What does the border-radius: 50% / 50%; declaration create?
    A. Square corners
    B. Circular or elliptical corners
    C. Only rounded top corners
    D. A border that becomes transparent

8. Which attribute defines how a border image is divided into regions for scaling and repetition?
    A. border-image-source
    B. border-image-slice
    C. border-image-repeat
    D. border-image-outset

9. What is the key difference between a border and an outline in CSS?
    A. Outlines are always dotted, borders can vary
    B. Borders take up space in the layout, outlines do not
    C. Borders are only applied to text, outlines are for images
    D. Outlines require background images, borders do not

10. If you want a background image to remain fixed in place even when scrolling, which value should you use for background-attachment?
    A. scroll
    B. local
    C. fixed
    D. cover

Day 6 - Learning Unit 6

Introduction:

In this chapter, we look at styling tables and arranging elements using CSS. We will also cover Flexboxes, a display property, used to arrange elements in HTML using CSS. Lastly, we will also dive into Animation, using CSS. The facilitator will cover slide 6 and is linked to Chapters 13, 14 and 15 in the Textbook.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcomes:

- HTML
- CSS

Additional Resources

1. @keyframes Rule:

This rule defines the animation sequence by specifying styles at various points (keyframes) within the animation.

```
@keyframes animationName {
  0% { /* styles at the beginning of the animation */ }
  50% { /* styles at the midpoint */ }
  100% { /* styles at the end of the animation */ }
}
```

2. animation Shorthand Property:
This property combines multiple animation-related properties into a single declaration.

```
.element {
  animation: animation-name animation-duration animation-timing-function animation-delay animation-iteration-count animation-direction animation-fill-mode animation-play-state;
}
```

3. Individual Animation Properties:

- animation-name: Specifies the name of the @keyframes rule to use.
- animation-duration: Sets the time an animation takes to complete (e.g., 1s, 500ms).
- animation-timing-function: Defines the speed curve of the animation (e.g., ease, linear, ease-in, ease-out, ease-in-out, cubic-bezier(), steps()).
- animation-delay: Specifies a delay before the animation starts (e.g., 0.5s).
- animation-iteration-count: Determines how many times the animation should repeat (e.g., 1, infinite, 3).
- animation-direction: Sets whether the animation should play forwards, backwards, or alternate (e.g., normal, reverse, alternate, alternate-reverse).
- animation-fill-mode: Specifies how styles are applied before and after the animation (e.g., none, forwards, backwards, both).
- animation-play-state: Controls whether the animation is running or paused (e.g., running, paused).

Class activity 6

Class Activity: Styling Tables, flexboxes, animations and Transformations

Objective:

Students will practice creating and styling HTML tables using CSS properties such as border, padding, border-collapse, caption, nth-of-type, and background styling techniques (gradients, zebra striping, highlighting).

Instructions:

Students will have to use VSCode or any text editor of their choice.

1. Create the Base Table
   - Write an HTML table that lists Chess pieces (Name, Quantity, Points, Symbol, Movement).
   - Use <caption>, <thead>, <tbody>, and <tfoot> correctly.
   - Insert placeholder images (e.g., king.png, queen.png, etc.) or use text instead if images are unavailable.

2. Apply Basic Styling
   - Add a thick border around the table and a thin border around each cell.
   - Experiment with border-spacing: 0; and border-collapse: collapse;.
   - Add padding (5px) to all cells and captions.

3. Improve Readability
   - Right-align the Qty and Points columns.
   - Center-align the Symbol column (images).
   - Style the caption so it looks like part of the table (bordered, larger font size).

4. Add Visual Enhancements
   - Apply a gradient background to header cells.
   - Use zebra striping to alternate row colors.
   - Highlight one row and one column (e.g., highlight the Knight row and Points column).

5. Extension (Optional for Advanced Students)
   - Rebuild the same table layout using only and CSS display: table;, display: table-row;, display: table-cell;.
   - Add responsiveness: When the screen width is below 700px, remove the table layout and stack items vertically using media queries.

Homework: Facilitator to give additional homework activity

Revision Questions Day 6

1. Which CSS property removes the spacing between table cell borders and merges them into a single border?
    A. border-spacing
    B. border-collapse
    C. border-style
    D. border-merge

2. Which pseudo-class selector is most often used to create “zebra striping” in table rows?
    A. tr:first-child
    B. tr:last-child
    C. tr:nth-of-type(even)
    D. tr:odd

3. In the default table display model, which CSS display value is assigned to the <tr> element?
    A. table-row
    B. table
    C. block
    D. inline-table

4. Which CSS declaration turns an element into a flex container?
    A. display: block;
    B. display: inline;
    C. display: flex;
    D. display: grid;

5. Which property controls whether flex items wrap onto multiple lines?
    A. align-items
    B. flex-wrap
    C. justify-content
    D. flex-flow

6. Which property allows you to override the cross-axis alignment of an individual flex item?
    A. align-items
    B. align-content
    C. align-self
    D. justify-content

7. Which animation property controls how many times an animation repeats?
    A. animation-delay
    B. animation-iteration-count
    C. animation-direction
    D. animation-play-state

8. Which CSS rule is used to define keyframes for an animation?
    A. @frames
    B. @animate
    C. @keyframes
    D. @transition

9. Which of the following is NOT a transform function?
    A. translate()
    B. rotate()
    C. scale()
    D. gradient()

10. Which CSS property hides the backside of an element when performing a 3D rotation?
    A. transform-style
    B. perspective
    C. backface-visibility
    D. visibility

Week 3

Day 7 - Learning Unit 7

Introduction

In this chapter, we take a look at JavaScript. We will start by looking at the operating environment of JavaScript, the browser environment, and we will also explore the Window Object. The facilitator will cover slide 7 and is linked to Chapters 16 and 17 in the Textbook.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcomes

- HMTL
- CSS
- JavaScript

Additional Resources

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

The ECMA-262 Specification defined a standard version of the core JavaScript language.

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform

Client-Side JavaScript

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field. The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

Advantages of JavaScript

The merits of using JavaScript are –

- Less server interaction − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- Immediate feedback to the visitors − They don't have to wait for a page reload to see if they have forgotten to enter something.
- Increased interactivity − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- Richer interfaces − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features –

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any multi-threading or multiprocessor capabilities.

Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

JavaScript Development Tools

One of major strengths of JavaScript is that it does not require expensive development tools. You can start with a simple text editor such as Notepad. Since it is an interpreted language inside the context of a web browser, you don't even need to buy a compiler.

To make our life simpler, various vendors have come up with very nice JavaScript editing tools. Some of them are listed here –

- Microsoft FrontPage – Microsoft has developed a popular HTML editor called FrontPage. FrontPage also provides web developers with a number of JavaScript tools to assist in the creation of interactive websites.
- Macromedia Dreamweaver MX – Macromedia Dreamweaver MX is a very popular HTML and JavaScript editor in the professional web development crowd. It provides several handy prebuilt JavaScript components, integrates well with databases, and conforms to new standards such as XHTML and XML.
- Macromedia HomeSite 5 – HomeSite 5 is a well-liked HTML and JavaScript editor from Macromedia that can be used to manage personal websites effectively.

Class activity 7

Class Activity: Browser Object Model

Objective:

Students will learn to interact with the window object and its child objects (screen, location, history, navigator) using JavaScript, while applying HTML and CSS for structure and styling.

Instructions:

Students will have to use VSCode or any text editor of their choice.

1. Create the HTML Structure
    - Make a simple webpage with:
        - A header (<h1>Browser Object Model Demo).
        - A styled section to display screen info (resolution, color depth).
        - A section to display URL info (hostname, pathname, protocol).
        - A section for navigator info (browser name, platform, user agent).
        - Buttons:
            - Reload Page (uses location.reload()).
            - Go Back & Go Forward (uses history.back() and history.forward()).
            - Open Popup (uses window.open() with specified size).
            - Show Modal Dialog (simulated with a CSS overlay).

2. Style the Page with CSS
    - Center the content with flexbox.
    - Add card-like boxes for each info section with borders, padding, and shadows.
    - Style buttons with hover effects.
    - Style the modal dialog (position: fixed, semi-transparent overlay, centered box).

3. Add JavaScript Functionality
    - On page load, display:
        - Screen width, height, and color depth using window.screen.
        - Current URL using location.
        - Browser/platform/userAgent using navigator.
    - Implement button functionality:
        - Reload Page → refresh page from server.
        - Go Back/Forward → navigate browser history.
        - Open Popup → open a 400x300 window with custom message.

- Show Modal Dialog → overlay with input asking "How many pets do you have?" (close dialog and log result).

Revision Questions Day 7

1. Which of the following is true about the window object in JavaScript?
    A. It represents the HTML document being rendered in the browser.
    B. It is the starting point for accessing the browser's facilities and exists in the global namespace.
    C. It only exists when a web page has multiple tabs open.
    D. It is only accessible through the document object.

2. Which location method does NOT save the current page in the browser history when navigating to a new URL?
    A. assign()
    B. replace()
    C. reload()
    D. Direct assignment like location = "url"

3. The history object in JavaScript allows you to:
    A. Access the complete browsing history from all windows and tabs.
    B. Navigate backward and forward in the history of a single window or tab.
    C. Modify the browser's bookmarks.
    D. Access the server's log of visited pages.

4. Which storage mechanism persists data even after the browser is closed?
    A. sessionStorage
    B. localStorage
    C. Cookies with no expiration date
    D. Both B and C

5. Which of the following console methods allows you to group multiple log entries together in a collapsible section?
    A. console.time()
    B. console.group()
    C. console.profile()
    D. console.clear()

6. Which of the following statements about the window.open() method is true?
    A. It can manipulate any browser window, even those not created by JavaScript.
    B. It requires two parameters: the URL of the document and the name of the window.
    C. It always creates a new browser tab, regardless of the parameters.
    D. It cannot be assigned to a variable for later manipulation.

7. What happens if you use the same name for a window in multiple calls to window.open()?
    A. A new window is created every time.
    B. The script throws an error.
    C. The existing window is reused and its contents are replaced with the new document.
    D. The window is ignored and nothing happens.

8. Which of the following methods is used to move a window to an absolute screen position?
    A. moveBy()
    B. moveTo()
    C. resizeTo()
    D. scrollTo()

9. Which of the following is not one of the three standard modal dialog boxes provided by JavaScript?
    A. alert()
    B. confirm()
    C. prompt()
    D. dialog()

10. What is the purpose of the sandbox attribute in an <iframe> element?
    A. It specifies the visual style of the frame border.
    B. It enables resizing of the frame.
    C. It applies restrictions to the embedded content for security reasons.
    D. It prevents the parent document from accessing the frame's content.

Day 8 - Learning Unit 8

Introduction

In this chapter, we look at DOM (Document Object Model). We will be exploring the DOM, looking at the different parent and child nodes. We will also delve into element inheritance. The facilitator will cover slide 8 and is linked to chapter 18 in the Textbook.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcomes:

- HTML
- CSS
- JavaScript

Additional Resources

1. Accessing Elements:

- document.getElementById(id): Selects a single element by its unique ID.
- document.getElementsByClassName(className): Returns a live HTMLCollection of elements with a specific class name.
- document.getElementsByTagName(tagName): Returns a live HTMLCollection of elements with a specific tag name.
- document.querySelector(selector): Returns the first element that matches a CSS selector.
- document.querySelectorAll(selector): Returns a static NodeList of all elements that match a CSS selector.

2. Manipulating Content and Attributes:

- element.innerHTML: Gets or sets the HTML content inside an element.
- element.textContent: Gets or sets the plain text content inside an element.
- element.setAttribute(attribute, value): Adds or changes an attribute on an element.
- element.getAttribute(attribute): Retrieves the value of an attribute.
- element.removeAttribute(attribute): Removes an attribute from an element.
- element.classList.add(className): Adds a class to an element's class list.
- element.classList.remove(className): Removes a class from an element's class list.
- element.classList.toggle(className): Toggles a class on an element's class list.

3. Styling Elements:

- element.style.property: Modifies inline CSS properties (e.g., element.style.color = 'red';).

4. Creating and Modifying Elements:

- document.createElement(tagName): Creates a new HTML element.
- parentNode.appendChild(childNode): Adds a child node to a parent node.
- parentNode.removeChild(childNode): Removes a child node from a parent node.
- parentNode.insertBefore(newNode, referenceNode): Inserts a new node before a reference node.

5. Event Handling:

- element.addEventListener(event, handlerFunction): Attaches an event listener to an element.
- element.removeEventListener(event, handlerFunction): Removes an event listener from an element.

6. Traversing the DOM:

- element.parentNode: Returns the parent node of an element.
- element.children: Returns a live HTMLCollection of the child elements.
- element.firstElementChild: Returns the first child element.
- element.lastElementChild: Returns the last child element.
- element.nextElementSibling: Returns the next sibling element.
- element.previousElementSibling: Returns the previous sibling element.

Class activity 8

Class Activity: DOM Manipulation

Objective:

To demonstrate an understanding of working with files and directories in Python.

Instructions:

Students will have to use VSCode or any other text editor.

1. Create the HTML Structure
    - Start with a container that has a heading (<h2>My List) and an unordered list (<ul>) with three items: Apple, Banana, Cherry.
    - Add buttons below the list:
        - Add Item
        - Remove Last Item
        - Highlight First Item
        - Replace Second Item
        - Clear List

2. Style the Page with CSS
    - Center the container on the page.
    - Style the list with borders, padding, and background color.
    - Add hover effects to buttons.

3. Write JavaScript to Manipulate the DOM
    - Use document.getElementById() and document.querySelector() to access elements.
    - Implement button functionality:
        - Add Item → creates a new <li> with text "New Item" and appends it.
        - Remove Last Item → removes the last <li>.
        - Highlight First Item → changes the first 's background color.
        - Replace Second Item → replaces the second <li> with "Replaced Item".
        - Clear List → removes all <li> elements.

Homework: Facilitator to give additional homework activity

Revision Questions Day 8

1. In the Document Object Model (DOM), each HTML element is represented as a:
   A. Attribute
   B. Node
   C. Style rule
   D. Function

2. Which property of the window object gives you access to the HTML document?
   A. window.body
   B. window.page
   C. window.document
   D. window.node

3. Which of the following methods returns only one element or null if no match is found?
   A. getElementsByTagName()
   B. getElementsByClassName()
   C. querySelectorAll()
   D. getElementById()

4. What is the main difference between textContent and innerHTML?
   A. innerHTML can include child elements, while textContent only sets text
   B. textContent replaces attributes, while innerHTML does not
   C. textContent allows JavaScript execution, while innerHTML does not
   D. They are identical in all cases

5. Which method is used to add a new child element to the end of a parent node?
   A. insertBefore()
   B. appendChild()
   C. replaceChild()
   D. addNode()

6. What happens if you call insertBefore(childNode, null) on a parent node?
   A. The child is not inserted
   B. The child is inserted as the first child
   C. The child is inserted as the last child
   D. An error occurs

7. Which JavaScript property can be used to access all attributes of an element?
   A. element.children
   B. element.attributes
   C. element.properties
   D. element.nodes

8. Why is the property className used instead of class in JavaScript?
    A. class is not supported in HTML
    B. class is a reserved word in JavaScript
    C. className is faster than class
    D. className is only used with CSS selectors

9. Which of the following is NOT a valid sibling or parent navigation property in the DOM?
    A. parentNode
    B. previousSibling
    C. nextSibling
    D. siblingNode

10. In jQuery, what does the following syntax do?
    $("p").wrapInner("");

    A. Removes the <p> element and replaces it with
    B. Wraps the <p> element itself with
    C. Wraps only the content of <p> with
    D. Creates a new <p> with a inside it

Day 9 - Learning Unit 9

Introduction

In this chapter, we look at applying dynamic styling using JavaScript. This includes replacing, changing, modifying and adjusting inline styles. The facilitator will cover slide 9 and is linked to chapter 19 in the Textbook.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcomes:

- HTML
- CSS
- JavaScript

Additional Resources

Direct Style Manipulation:
Accessing the style property.

```
element.style.propertyName = "value"; // e.g., element.style.color = "red";
```

Note: CSS properties with hyphens (e.g., background-color) become camelCase in JavaScript (e.g., backgroundColor).

Setting multiple styles:

```
Object.assign(element.style, {
    color: "blue",
    fontSize: "16px",
    border: "1px solid black"
});
```

Class Manipulation:
Adding a class.

```
element.classList.add("className");
```

Removing a class.

```
element.classList.remove("className");
```

Toggling a class.

```
element.classList.toggle("className");
```

Checking for a class.

```
element.classList.contains("className"); // Returns true or false
```

Replacing a class.

```
element.classList.replace("oldClass", "newClass");
```

Working with CSS Variables (Custom Properties):
Setting a CSS variable.

```
element.style.setProperty("--variable-name", "value"); // e.g., element.style.setProperty("--primary-color", "green");
```

Getting a CSS variable.

```
const value = getComputedStyle(element).getPropertyValue("--variable-name");
```

Manipulating Stylesheets (Advanced):
Adding a new CSS rule to a stylesheet:

```
document.styleSheets[0].insertRule("selector { property: value; }", index);
```

Modifying an existing rule.

```
document.styleSheets[0].cssRules[index].style.propertyName = "newValue";
```

Class activity 9

Class Activity: Dynamic styling with JavaScript

Objective:

Students will learn how to use JavaScript to dynamically change styles of HTML elements by interacting with the DOM

Instructions:

Students will have to use VSCode or any text editor of their choice.

1. Create an HTML page with:
    - A heading (<h1>) that says "Dynamic Styling Activity".
    - A paragraph (<p>) with some sample text.
    - Three buttons:
        - Change Color
        - Change Font Size
        - Toggle Dark Mode
2. Use CSS to style:
    - The default page (light theme).
    - A .dark-mode class for dark theme styling.
3. Use JavaScript to:
    - Change the paragraph text color to a random color when Change Color is clicked.
    - Increase the paragraph font size step by step when Change Font Size is clicked.
    - Toggle the dark mode on and off when Toggle Dark Mode is clicked.

Homework: Facilitator to give additional homework activity

Revision Questions Day 9

1. Which of the following is NOT one of the four main techniques for dynamically changing styles in JavaScript?
    A. Replacing style sheets
    B. Modifying CSS classes
    C. Adding pseudo-classes directly
    D. Adjusting inline styles

2. The document.styleSheets property returns:
    A. A list of all CSS rules in the document
    B. A collection of all style sheets loaded in the document
    C. A string containing inline styles
    D. Only the external style sheets

3. Which property of a style sheet can be toggled (true/false) to enable or disable it?
    A. enabled
    B. disabled
    C. active
    D. visible

4. In the example where different color stylesheets (Red.css, Green.css, Blue.css) are used, the enable() function identifies the correct style sheet using which property?
    A. title
    B. className
    C. href
    D. id

5. What is the difference between a preferred style sheet and an alternate style sheet?
    A. Preferred is always disabled; alternate is always enabled
    B. Preferred is enabled by default; alternate is disabled by default
    C. Preferred must not have a title; alternate must have one
    D. Preferred works in Chrome; alternate works only in Firefox.

6. Which browsers do NOT support alternate style sheets?
    A. Chrome and Opera
    B. Firefox and Safari
    C. Edge and Firefox
    D. Internet Explorer only

7. Which method is used to add a new CSS rule to a style sheet through JavaScript?
    A. appendRule()
    B. addCSSRule()
    C. insertRule()
    D. pushRule()

8. Which of the following methods is NOT part of the classList API?
    A. contains()
    B. add()
    C. toggle()
    D. switch()

9. Which object represents the collection of name/value pairs for inline styles on an element?
    A. CSSRule
    B. CSSStyleDeclaration
    C. CSSInline
    D. StyleAttribute

10. If you want to see the final computed styles applied to an element (including external, internal, and inline rules), which method should you use?
    A. element.style
    B. getPropertyValue()
    C. getComputedStyle()
    D. styleSheets[0].cssRules

Day 10 - Learning Unit 10

Introduction

In this chapter, we look at adding events in HTML, using JavaScript. We, will look at creating and executing events. The facilitator will cover slide 10 and is linked to Chapter 20 in the Textbook.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcomes

- HTML
- CSS
- JavaScript

Additional Resources

The primary method for attaching event listeners is addEventListener().

element.addEventListener(eventType, handlerFunction, options);

- eventType: A string representing the event (e.g., 'click', 'mouseover', 'submit').
- handlerFunction: The function to be executed when the event occurs.
- options: An optional object for advanced settings (e.g., capture, once, passive).

Common Event Types:
Mouse Events:

- click: When an element is clicked.
- dblclick: When an element is double-clicked.
- mousedown: When a mouse button is pressed down on an element.
- mouseup: When a mouse button is released on an element.
- mousemove: When the mouse pointer moves over an element.
- mouseover: When the mouse pointer enters an element.
- mouseout: When the mouse pointer leaves an element.
- contextmenu: When the right-click context menu is requested.

Keyboard Events:

- keydown: When a key is pressed down.
- keyup: When a key is released.
- keypress: When a key is pressed and held down (deprecated in modern usage, keydown and keyup are preferred).

Form Events:

- submit: When a form is submitted.
- input: When the value of an input element changes.
- change: When the value of an input element changes and the element loses focus.
- focus: When an element gains focus.
- blur: When an element loses focus.

Window Events:

- load: When the entire page (including images, scripts, etc.) has loaded.
- DOMContentLoaded: When the HTML document has been fully loaded and parsed.
- resize: When the browser window is resized.
- scroll: When the element's scroll position changes.

Event Object:

The handlerFunction typically receives an Event object as its first argument. This object contains information about the event, such as:

- event.target: The element that triggered the event.
- event.currentTarget: The element to which the event listener is attached.
- event.type: The type of event.
- event.preventDefault(): Prevents the default action associated with the event.
- event.stopPropagation(): Stops the event from propagating up the DOM tree (event bubbling).

Example:

```
const myButton = document.getElementById('myButton');

function handleClick(event) {
  console.log('Button clicked!', event.target.id);
}

myButton.addEventListener('click', handleClick);
```

Class activity 10

Class Activity: Event Handlers

Objective:

Students will learn how to use JavaScript events to interact with HTML elements and respond to user actions like clicks, mouse movements, keyboard input, and form events.

Instructions:

Students will have to use VSCode or any other text editor of their choice.

1. Create an HTML page with the following elements:
   - A heading (<h1>) that says "JavaScript Events Activity".
   - A button labeled "Click Me".
   - An input field for typing text.
   - A area for displaying messages.
   - A paragraph (<p>) that changes when the mouse hovers over it.

2. Use CSS to style the page and elements.

3. Use JavaScript to handle events:
   - Display a message when the button is clicked.
   - Update the dynamically as the user types in the input field.
   - Change the paragraph color when the mouse enters and leaves.
   - Show an alert when the input loses focus.

Homework: Facilitator to give additional homework activity

Revision Questions Day 10

1. What is the role of an event handler in JavaScript?
    A. It defines the structure of the DOM.
    B. It is a CSS rule applied dynamically.
    C. It is a function executed when an event occurs.
    D. It propagates events between elements.

2. Which method is the preferred way to register an event handler?
    A. Inline event attributes in HTML
    B. Assigning to an event property (e.g., onclick)
    C. addEventListener()
    D. attachEvent()

3. What is the main limitation of assigning an event handler directly to a property like element.onclick?
    A. It only works in Chrome.
    B. Only one handler can be assigned per event.
    C. It does not work for click events.
    D. It cannot access the event object.

4. What are the two phases of event propagation?
    A. Forward and backward
    B. Capturing and bubbling
    C. Sending and receiving
    D. Binding and unbinding

5. When using addEventListener(), what does passing true as the third argument mean?
    A. Listen during the bubbling phase
    B. Listen during the capturing phase
    C. Register multiple handlers
    D. Disable default browser actions

6. Which method removes an event handler previously registered with addEventListener()?
    A. deleteEventListener()
    B. detachEvent()
    C. removeHandler()
    D. removeEventListener()

7. In an event object, what is the difference between target and currentTarget?
    A. target is always the window; currentTarget is the document
    B. target is the element clicked; currentTarget is the element the handler was attached to
    C. They are identical in all cases
    D. target refers to parent; currentTarget refers to child

8. Which method stops the event from continuing to propagate but allows other handlers on the same element to run?
    A. stopImmediatePropagation()
    B. stopPropagation()
    C. preventDefault()
    D. cancelBubble()

9. Which event object method prevents the browser's default action for an event?
    A. preventDefault()
    B. stopPropagation()
    C. cancelAction()
    D. stopDefault()

10. If you want multiple event handlers on the same element for the same event, which registration method should you use?
    A. Inline registration
    B. Traditional onclick assignment
    C. addEventListener()
    D. attachEvent()

Week 4

Day 11 - Learning Unit 11

Introduction

In this chapter, you'll use your own controls that are wired up to the audio and video elements through JavaScript. All the DOM elements and events are available in JavaScript, so it's a fairly straightforward process to create your own controls to work with the audio or video element. The facilitator will cover slide 11 and is linked to Chapter 21 in the Textbook.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcomes

- HTML
- CSS
- JavaScript

Additional Resources

HTML Audio/Video Methods

Method	Description
addTextTrack()	Adds a new text track to the audio/video
canPlayType()	Checks if the browser can play the specified audio/video type
load()	Re-loads the audio/video element
play()	Starts playing the audio/video
pause()	Pauses the currently playing audio/video

HTML Audio/Video Properties

Property	Description
audioTracks	Returns an AudioTrackList object representing available audio tracks
autoplay	Sets or returns whether the audio/video should start playing as soon as it is loaded
buffered	Returns a TimeRanges object representing the buffered parts of the audio/video
controller	Returns the MediaController object representing the current media controller of the audio/video
controls	Sets or returns whether the audio/video should display controls (like play/pause etc.)
crossOrigin	Sets or returns the CORS settings of the audio/video
currentSrc	Returns the URL of the current audio/video
currentTime	Sets or returns the current playback position in the audio/video (in seconds)
defaultMuted	Sets or returns whether the audio/video should be muted by default
defaultPlaybackRate	Sets or returns the default speed of the audio/video playback
duration	Returns the length of the current audio/video (in seconds)
ended	Returns whether the playback of the audio/video has ended or not
error	Returns a MediaError object representing the error state of the audio/video
loop	Sets or returns whether the audio/video should start over again when finished
mediaGroup	Sets or returns the group the audio/video belongs to (used to link multiple audio/video elements)
muted	Sets or returns whether the audio/video is muted or not
networkState	Returns the current network state of the audio/video
paused	Returns whether the audio/video is paused or not
playbackRate	Sets or returns the speed of the audio/video playback

played	Returns a TimeRanges object representing the played parts of the audio/video
preload	Sets or returns whether the audio/video should be loaded when the page loads
readyState	Returns the current ready state of the audio/video
seekable	Returns a TimeRanges object representing the seekable parts of the audio/video
seeking	Returns whether the user is currently seeking in the audio/video
src	Sets or returns the current source of the audio/video element
startDate	Returns a Date object representing the current time offset
textTracks	Returns a TextTrackList object representing the available text tracks
videoTracks	Deprecated. Do not use it.
volume	Sets or returns the volume of the audio/video

HTML Audio/Video Events

Event	Description
abort	Fires when the loading of an audio/video is aborted
canplay	Fires when the browser can start playing the audio/video
canplaythrough	Fires when the browser can play through the audio/video without stopping for buffering
durationchange	Fires when the duration of the audio/video is changed
emptied	Fires when the current playlist is empty
ended	Fires when the current playlist is ended
error	Fires when an error occurred during the loading of an audio/video
loadeddata	Fires when the browser has loaded the current frame of the audio/video
loadedmetadata	Fires when the browser has loaded meta data for the audio/video
loadstart	Fires when the browser starts looking for the audio/video
pause	Fires when the audio/video has been paused
play	Fires when the audio/video has been started or is no longer paused
playing	Fires when the audio/video is playing after having been paused or stopped for buffering
progress	Fires when the browser is downloading the audio/video
ratechange	Fires when the playing speed of the audio/video is changed
seeked	Fires when the user is finished moving/skipping to a new position in the audio/video
seeking	Fires when the user starts moving/skipping to a new position in the audio/video

stalled	Fires when the browser is trying to get media data, but data is not available
suspend	Fires when the browser is intentionally not getting media data
timeupdate	Fires when the current playback position has changed
volumechange	Fires when the volume has been changed
waiting	Fires when the video stops because it needs to buffer the next frame

Class Activity 11

Class Activity: Custom Video Controls using JavaScript

Objective:

To demonstrate an understanding of implementing custom video controls using JavaScript.

Instructions:

Students will have to use VSCode or any other text editor of their choice.

- Create a website and embed a video into the webpage
- Use JavaScript to create custom video controls

Homework: Facilitator to give additional homework activity

Revision Questions Day 11

1. Which attribute is used on the <audio> element to display the browser's built-in controls?
   A. autoplay
   B. controls
   C. native
   D. show

2. In custom audio controls, the Play button text should toggle between "Play" and "Pause". Which events are most relevant for updating the button label?
   A. start and stop
   B. play and pause
   C. timeupdate and ended
   D. durationchange and mute

3. The durationchange event on an audio element is raised when:
   A. The audio has finished playing.
   B. The metadata is loaded and the clip length is known.
   C. The volume is adjusted.
   D. The file is deleted.

4. Which property of the audio element represents the current playback position in seconds?
   A. seek
   B. currentTime
   C. position
   D. elapsed

5. Which event fires periodically during playback to update progress in custom controls?
   A. timeupdate
   B. progress
   C. tick
   D. loop

6. To mute or unmute audio, which property is toggled?
   A. audio.volume
   B. audio.mute
   C. audio.muted
   D. audio.silent

7. What is the valid range of values for the volume property on an audio or video element?
   A. 1 to 100
   B. 0 to 255
   C. 0 to 1
   D. -1 to 1

8. Which method must be called if you change multiple <source> elements of an audio element before playing a new track?
    A. reload()
    B. reset()
    C. load()
    D. update()

9. The JavaScript objects representing <audio> and <video> elements are derived from:
    A. HTMLAudioElement and HTMLVideoElement → HTMLMediaElement
    B. HTMLElement → HTMLSourceElement
    C. HTMLTrackElement → HTMLObjectElement
    D. HTMLStreamElement → HTMLControlElement

10. Which property should be updated in custom controls to synchronize the progress slider with playback?
    A. value of the range input
    B. innerHTML of the span
    C. id of the audio element
    D. src of the media file

Day 12 - Learning Unit 12

Introduction

In this chapter, we'll look at how to use Scalable Vector Graphics (SVG) in an HTML5 web application. There are a lot of really cool things that you can do with SVG. The facilitator will cover slide 12 and is linked to Chapter 22.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcomes

- HTML
- CSS
- JavaScript

Additional Resources

Shapes

```
<!-- Rectangle -->
<rect x="10" y="10" width="100" height="50" fill="blue" stroke="black" stroke-width="2" />

<!-- Circle -->
<circle cx="75" cy="75" r="50" fill="red" stroke="black" stroke-width="2" />

<!-- Ellipse -->
<ellipse cx="150" cy="80" rx="100" ry="50" fill="green" />

<!-- Line -->
<line x1="10" y1="10" x2="200" y2="100" stroke="black" stroke-width="2" />

<!-- Polygon -->
<polygon points="50,10 90,80 10,80" fill="orange" stroke="black" />

<!-- Polyline -->
<polyline points="0,40 40,40 40,80 80,80" fill="none" stroke="purple" stroke-width="2" />
```

- Paths

```
<path d="M10 80 C 40 10, 65 10, 95 80 S 150 150, 180 80"
    fill="transparent" stroke="brown" stroke-width="2"/>
```

Commands:

M = Move to (x, y)

L = Line to

H / V = Horizontal / Vertical line

C = Cubic Bezier curve

Q = Quadratic Bezier curve

Z = Close path

Text

```
<text x="50" y="50" font-family="Arial" font-size="20" fill="black">
  Hello SVG
</text>

<!-- Text along a path -->
<path id="curve" d="M10 80 Q 95 10 180 80" fill="none"/>
<text>
  <textPath href="#curve">Text on a curve</textPath>
</text>
```

Styling

```
fill="color"          <!-- inside color -->
stroke="color"        <!-- border color -->
stroke-width="2"      <!-- border thickness -->
opacity="0.5"         <!-- transparency -->
fill-opacity="0.3"    <!-- fill only -->
stroke-opacity="0.8"  <!-- stroke only -->
```

Gradients & Patterns

```
<defs>
  <!-- Linear Gradient -->
  <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
    <stop offset="0%" style="stop-color:blue; stop-opacity:1" />
    <stop offset="100%" style="stop-color:lightblue; stop-opacity:1" />
  </linearGradient>

  <!-- Radial Gradient -->
  <radialGradient id="grad2" cx="50%" cy="50%" r="50%">
    <stop offset="0%" style="stop-color:yellow;" />
    <stop offset="100%" style="stop-color:red;" />
  </radialGradient>
</defs>

<circle cx="80" cy="80" r="70" fill="url(#grad1)" />
```

Transformations

```
transform="translate(50,50)"
transform="rotate(45)"
transform="scale(1.5)"
transform="skewX(20)"
transform="skewY(20)"
```

Groups

```
<g fill="blue" stroke="black" stroke-width="2">
  <circle cx="40" cy="40" r="30"/>
  <rect x="80" y="20" width="60" height="40"/>
</g>
```

Animation

```
<!-- Animate position -->
<circle cx="50" cy="50" r="20" fill="red">
  <animate attributeName="cx" from="50" to="200" dur="3s" repeatCount="indefinite"/>
</circle>

<!-- Animate color -->
<rect x="10" y="10" width="50" height="50">
  <animate attributeName="fill" values="red;blue;green;red" dur="4s" repeatCount="indefinite"/>
</rect>
```

Class activity 12

Class Activity: Scalable Vector Graphics

Objective:

Students will learn how to create, style, and manipulate SVG elements using HTML, CSS, and JavaScript, including shapes, colors, and basic animations.

Instructions:

Students will have to use VSCode or any other text editor of their choice.

1. Create an HTML page with an SVG canvas (e.g., 500x400 pixels).

2. Add the following shapes inside the SVG:
   - A circle
   - A rectangle
   - A line

3. Add buttons to:
   - Change the circle's color randomly.
   - Move the rectangle to a new position.
   - Animate the line length dynamically.

4. Use CSS to style the SVG and buttons.

5. Use JavaScript to manipulate SVG attributes dynamically when buttons are clicked.

Homework: Facilitator to give additional homework activity

Revision Questions Day 12

1. What is the main difference between vector graphics (SVG) and bitmap graphics?
    A. Vector graphics use pixels, bitmap graphics use formulas
    B. Vector graphics are formula-based, bitmap graphics use pixels
    C. Vector graphics cannot be scaled, bitmap graphics can
    D. Vector graphics only support circles and lines

2. In SVG, which attributes define a circle?
    A. x, y, r
    B. cx, cy, r
    C. x1, y1, x2, y2
    D. width, height, r

3. How can you save an SVG image to use it like a normal image file?
    A. Save it as .jpg
    B. Save it as .png
    C. Save it as .svg
    D. Save it as .gif

4. Which SVG element is used to create a complex shape with "move to," "line to," and "curve to" commands?
    A. <circle>
    B. <rect>
    C. <polygon>
    D. <path>

5. How do you apply a gradient fill to an SVG element?
    A. Using CSS background-color property
    B. Defining a <linearGradient> in <defs> and referencing it via URL
    C. Using the fill-opacity attribute
    D. Setting the fill attribute to a color name

6. Which method is required to create an SVG element in JavaScript?
    A. document.createElement()
    B. document.getElementById()
    C. document.createElementNS()
    D. document.appendChild()

7. What is the purpose of the cloneNode() method in the animation example?
    A. To delete the selected SVG element
    B. To create a copy of the selected element for animation
    C. To highlight the element with CSS
    D. To change the fill color

8. How can you make an SVG element respond to mouseover and mouseout events?
    A. Use the hover pseudo-class in CSS
    B. Add event listeners with addEventListener() in JavaScript
    C. Set the fill attribute to "hover"
    D. Use the onclick attribute

9. Which SVG element allows embedding another image (like a flag) as a background?
    A. <circle>
    B. <image> inside <pattern>
    C. <rect>
    D. <polygon>

10. What does setting transform-style: preserve-3d on an SVG element enable?
    A. Two-dimensional transformations only
    B. Three-dimensional transformations for animation
    C. Automatic gradient fills
    D. Dynamic color changes on hover

Day 13 - Learning Unit 13

Introduction:

In this chapter, We will look at how to use the canvas element in HTML5 to create some fun graphics. As you'll see, it is very different from SVG, which you explored in the previous chapter. The facilitator will cover slide 13 and is linked to Chapter 23.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcome:

- HTML
- CSS
- JavaScript

Additional Resources

Basic Setup:

```
<canvas id="myCanvas" width="500" height="300"></canvas>
<script>
  const canvas = document.getElementById('myCanvas');
  const ctx = canvas.getContext('2d'); // Get the 2D rendering context
</script>
```

Drawing Shapes:
Rectangles:

ctx.fillRect(x, y, width, height); (filled)
ctx.strokeRect(x, y, width, height); (outline)
ctx.clearRect(x, y, width, height); (clear a rectangular area)
Paths (Lines, Arcs, etc.):
ctx.beginPath(); (start a new path)
ctx.moveTo(x, y); (move the pen to a point)
ctx.lineTo(x, y); (draw a line to a point)
ctx.arc(x, y, radius, startAngle, endAngle, anticlockwise); (draw an arc/circle)
ctx.closePath(); (close the current path)
ctx.stroke(); (draw the outline of the path)
ctx.fill(); (fill the path)

Text:
ctx.font = "30px Arial"; (set font style)
ctx.fillText("Hello Canvas!", x, y); (filled text)
ctx.strokeText("Hello Canvas!", x, y); (outlined text)
ctx.measureText("Text").width; (get text width)

Styles and Properties:
Colors & Fills:
ctx.fillStyle = "color"; (for filling shapes and text)
ctx.strokeStyle = "color"; (for stroking shapes and text outlines)
Colors can be named ("red"), hex ("#FF0000"), or RGBA ("rgba(255,0,0,0.5)").

Line Styles:
ctx.lineWidth = number;
ctx.lineCap = "butt" | "round" | "square";
ctx.lineJoin = "bevel" | "round" | "miter";

Shadows:
ctx.shadowOffsetX = number;
ctx.shadowOffsetY = number;
ctx.shadowBlur = number;
ctx.shadowColor = "color";

Transformations:
ctx.translate(x, y);
ctx.rotate(angle);

ctx.scale(x, y);
ctx.save(); (save current transformation state)
ctx.restore(); (restore saved transformation state)

Images:
ctx.drawImage(image, sx, sy, sWidth, sHeight, dx, dy, dWidth, dHeight); (draw an image)

Compositing:
ctx.globalAlpha = number; (set transparency)
ctx.globalCompositeOperation = "source-over" | "destination-over" | ...; (control how new shapes are drawn over existing ones)

Class activity 13

Class Activity: Canvas

Objective:

To demonstrate an understanding of using the Canvas HTML Element.

Instructions:

Students will have to use VSCode or any other text editor of their choice.

1. Create an HTML page with a <canvas> element (e.g., 500x400 pixels).

2. Add buttons to:
   - Draw a rectangle.
   - Draw a circle.
   - Draw a line.
   - Clear the canvas.

3. Use JavaScript to draw shapes dynamically on the canvas when buttons are clicked.

4. Optional: Add a color picker input to allow students to draw in different colors.

Homework: Facilitator to give additional homework activity

Revision Questions Day 13

1. What is the primary purpose of the <canvas> element in HTML5?
    A. To define a fixed image like SVG
    B. To provide a blank area that can be drawn on using JavaScript
    C. To replace HTML forms
    D. To embed video and audio only

2. Which method is used to get the 2D drawing context of a canvas element?
    A. getElementById()
    B. getContext('2d')
    C. querySelector()
    D. createCanvas()

3. Which of the following is NOT a method for drawing rectangles on a canvas?
    A. fillRect()
    B. strokeRect()
    C. clearRect()
    D. drawRect()

4. In the drawBoard() function, how are alternating squares drawn on a chessboard?
    A. Using only the x-coordinate
    B. Using only the y-coordinate
    C. Using the condition (x + y) % 2
    D. By filling all squares the same color

5. How can you apply a gradient fill to a shape on a canvas?
    A. By using CSS background-gradient
    B. By calling createLinearGradient() and setting fillStyle
    C. By drawing SVG elements inside the canvas
    D. By using strokeStyle only

6. Which of the following correctly creates an image object in JavaScript for canvas?
    A. var myImage = document.createElement("image");
    B. var myImage = new Image(); myImage.src = "file.png";
    C. var myImage = canvas.drawImage("file.png");
    D. var myImage = getContext("2d");

7. What does the translate() method do in canvas transformations?
    A. Rotates the drawing context
    B. Scales the shapes proportionally
    C. Moves the origin of the drawing context
    D. Changes the fill color of shapes

8. In the solar system example, how is the earth rotated around the sun?
    A. By moving the x and y coordinates manually every frame
    B. By using rotate() and translate() transformations
    C. By scaling the canvas
    D. By using CSS animations

9. What is the purpose of save() and restore() in canvas?
    A. To save the entire canvas as an image
    B. To save and restore the drawing context state including transformations
    C. To save a variable for later use in JavaScript
    D. To pause and resume animations

10. What does the globalCompositeOperation property control?
    A. The fill color of rectangles
    B. The order and method by which shapes are drawn over each other
    C. The speed of animations
    D. The size of the canvas

Day 14 - Learning Unit 14

Introduction:

In this chapter, we will look at the ability to drag and drop elements on a web page. We will look at the fundamental concepts and explore the code required to lend us the ability to Drag and Drop elements. The facilitator will cover slide 14 and is linked to Chapter 24 in the Textbook.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcome:

- HTML
- CSS
- JavaScript

Additional Resources

1. Making an Element Draggable:

Attribute: Add draggable="true" to the HTML element you want to make draggable.

```
<div draggable="true">Drag Me!</div>
```

2. Drag Events (on the Draggable Element):

dragstart: Fired when the user starts dragging the element.
Use event.dataTransfer.setData(format, data) to store data (e.g., ID, text) that can be retrieved by the drop target.
Use event.dataTransfer.effectAllowed to specify the allowed drag effects (e.g., copy, move, link).
drag: Fired continuously while the element is being dragged.
dragend: Fired when the drag operation ends (either by dropping or canceling).

3. Drop Events (on the Drop Target Element):

dragenter: Fired when the dragged element enters the drop target's boundaries.
dragover: Fired continuously while the dragged element is over the drop target.
Crucial: Call event.preventDefault() inside this event listener to allow a drop. Without this, the drop event will not fire.
dragleave: Fired when the dragged element leaves the drop target's boundaries.
drop: Fired when the dragged element is dropped onto the drop target.
Crucial: Call event.preventDefault() to prevent default browser behavior (e.g., opening dropped links).
Use event.dataTransfer.getData(format) to retrieve the data set during dragstart.
Use event.dataTransfer.files to access files if files are being dragged.

4. Data Transfer Object (dataTransfer):

setData(format, data): Stores data associated with the drag operation.
getData(format): Retrieves data stored during setData.
effectAllowed: Specifies allowed drag effects.
dropEffect: Specifies the actual effect that occurred after a drop.
files: A FileList object containing files if a file is being dragged.

Example Snippets:
JavaScript

```
// Draggable element
const draggableItem = document.getElementById('myDraggable');
draggableItem.addEventListener('dragstart', (event) => {
    event.dataTransfer.setData('text/plain', event.target.id);
    event.dataTransfer.effectAllowed = 'move';
});
```

```
// Drop target
const dropZone = document.getElementById('myDropZone');
dropZone.addEventListener('dragover', (event) => {
    event.preventDefault(); // Allow drop
});

dropZone.addEventListener('drop', (event) => {
    event.preventDefault(); // Prevent default browser behavior
    const data = event.dataTransfer.getData('text/plain');
    console.log('Dropped item ID:', data);
    // Perform actions based on the dropped item
});
```

Class activity 14

Class Activity: Drag 'n Drop

Objective:

To demonstrate an understanding of how to drag and drop elements in HTML.

Instructions:

Students will have to use VSCode or any other text editor of their choice.

1. Create an HTML page with:
    - Three draggable colored boxes (divs).
    - Two drop zones (divs) where boxes can be dropped.

2. Style the boxes and drop zones using CSS.

3. Use JavaScript to:
    - Allow boxes to be dragged.
    - Allow drop zones to accept boxes.
    - Update the drop zone when a box is dropped.

Homework: Facilitator to give additional homework activity

Revision Questions Day 14

1. What is the primary purpose of the HTML5 Drag and Drop (DnD) API?
    A. To create animated graphics on a web page
    B. To allow elements to be selected and moved to another location
    C. To store user login credentials
    D. To validate form input

2. Which object is used to store and transfer data during a drag-and-drop operation?
    A. event
    B. dataTransfer
    C. dragObject
    D. transferData

3. Which of the following events is fired on the element being dragged when the drag operation starts?
    A. dragenter
    B. dragstart
    C. drop
    D. dragleave

4. In a DnD operation, which event is triggered on the target element when the dragged item is released over it?
    A. dragover
    B. dragend
    C. drop
    D. drag

5. How can you make an HTML element draggable?
    A. <element draggable="true">
    B. element.draggable = "false"
    C. element.enableDrag()
    D. <element draggable="false">

6. What does the effectAllowed property of the dataTransfer object specify?
    A. The allowed types of data that can be dragged
    B. The actions that are allowed when an element is dropped (e.g., copy, move, link)
    C. The browser compatibility for the drag operation
    D. The speed of the drag animation

7. What is the purpose of the dragOver() event handler in the checkers application?
    A. To initialize the game board
    B. To cancel the browser's default action and indicate if a drop is allowed
    C. To remove the dragged piece from the board
    D. To promote a piece to a king

8. Which method is used to access the data stored in the dataTransfer object during a drop event?
    A. getData()
    B. setData()
    C. transferData()
    D. dragData()

9. How is a checker promoted to a king in the application?
    A. When it moves backward
    B. When it reaches the last row on the opponent's side
    C. After jumping over an opponent's piece
    D. When all pieces of the opponent are captured

10. How does the application ensure that players alternate turns?
    A. By using a timer to control moves
    B. By setting the draggable attribute to false for the current player's pieces after a move
    C. By removing pieces from the board after each turn
    D. By restricting moves to black squares only

Week 5

Day 15 - Learning Unit 15

Introduction:

In this chapter, we will look at IndexedDBs, what they are and how they work will be covered in this learning unit. The facilitator will cover slide 15 and is linked to Chapter 25 in the Textbook.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcome:

- HTML
- CSS
- JavaScript

Additional Resources

IndexedDB is an asynchronous, transactional, event-driven database system built into browsers for client-side storage of large amounts of structured data.

Key Concepts:

- Database: A collection of object stores.
- Object Store: A collection of records, similar to a table in a relational database. Each record is a JavaScript object.
- Key: A unique identifier for each record in an object store. Can be a keyPath (a property within the object) or autoIncrement.
- Index: A way to quickly look up records based on properties other than the key.
- Transaction: A set of operations that are treated as a single, atomic unit. Ensures data integrity.
- Request: An object representing an asynchronous database operation. Emits success or error events.

Basic Operations:
Opening/Creating a Database.

```
const request = indexedDB.open('myDatabase', 1); // databaseName, version

request.onupgradeneeded = function(event) {
    const db = event.target.result;
    // Create object stores and indexes here
    const objectStore = db.createObjectStore('myObjectStore', { keyPath: 'id', autoIncrement: true });
    objectStore.createIndex('nameIndex', 'name', { unique: false });
};

request.onsuccess = function(event) {
    const db = event.target.result;
    // Database opened successfully
};

request.onerror = function(event) {
    // Handle error
};
```

Adding Data.

```
const transaction = db.transaction(['myObjectStore'], 'readwrite');
const objectStore = transaction.objectStore('myObjectStore');
const addRequest = objectStore.add({ id: 1, name: 'John Doe', age: 30 });

addRequest.onsuccess = function() {
    console.log('Data added successfully');
};
```

Retrieving Data.

```
const transaction = db.transaction(['myObjectStore'], 'readonly');
const objectStore = transaction.objectStore('myObjectStore');
const getRequest = objectStore.get(1); // Get by key

getRequest.onsuccess = function(event) {
    const data = event.target.result;
    console.log('Retrieved data:', data);
};
```

Updating Data.

```
const transaction = db.transaction(['myObjectStore'], 'readwrite');
const objectStore = transaction.objectStore('myObjectStore');
const putRequest = objectStore.put({ id: 1, name: 'Jane Doe', age: 31 }); // Update existing record

putRequest.onsuccess = function() {
    console.log('Data updated successfully');
};
```

Deleting Data.

```
const transaction = db.transaction(['myObjectStore'], 'readwrite');
const objectStore = transaction.objectStore('myObjectStore');
const deleteRequest = objectStore.delete(1); // Delete by key

deleteRequest.onsuccess = function() {
    console.log('Data deleted successfully');
};
```

Using Cursors (Iterating).

```
const transaction = db.transaction(['myObjectStore'], 'readonly');
const objectStore = transaction.objectStore('myObjectStore');
const cursorRequest = objectStore.openCursor();

cursorRequest.onsuccess = function(event) {
    const cursor = event.target.result;
    if (cursor) {
        console.log(cursor.value);
        cursor.continue(); // Move to the next record
    } else {
        console.log('No more records');
    }
};
```

Class activity 15

Class Activity: Indexed DB

Objective:

To demonstrate an understanding of Indexed DB.

Instructions:

Students will have to use VSCode or any other text editor of their choice.

1. Create an HTML page with:
    - Input fields for name and email.
    - Buttons for Add Contact, View Contacts, and Clear All.
    - A or <ul> to display the list of contacts.

2. Use IndexedDB to store the contact information.

3. Use JavaScript to handle:
    - Adding new contacts.
    - Displaying all stored contacts.
    - Clearing all contacts from the database.

Homework: Facilitator to give additional homework activity

Revision Questions Day 15

1. What is the primary storage unit in Indexed DB?
    A. Table
    B. Object store
    C. Index
    D. Transaction

2. How can an object store assign unique keys to objects automatically?
    A. Inline key
    B. Out-of-line key
    C. Key generator
    D. Transaction scope

3. Which of the following is true about Indexed DB transactions?
    A. They are optional for reading and writing data
    B. Data is immediately committed as soon as you call put()
    C. They must define a scope and mode (read-only or read-write)
    D. They allow read-write transactions with overlapping object stores

4. Which event handler is fired only when creating or upgrading a database?
    A. onsuccess
    B. onerror
    C. onupgradeneeded
    D. oncomplete

5. How are Indexed DB operations typically processed?
    A. Synchronously
    B. Asynchronously
    C. Sequentially without callbacks
    D. Inline with HTML events

6. What does the second parameter in createIndex("lastName", "id") specify?
    A. Name of the object store
    B. The key path used to extract the key from the object
    C. Mode of the transaction
    D. Cursor direction

7. Can Indexed DB enforce foreign key relationships or perform joins between object stores?
    A. Yes, automatically
    B. Yes, but only in read-only transactions
    C. No, the database does not enforce relationships
    D. Only with a key generator

8. What happens if you try to use a transaction after it has completed?
    A. The database automatically creates a new transaction
    B. TRANSACTION_INACTIVE_ERR error is raised
    C. The operation succeeds normally
    D. Indexed DB queues the request for later

9. Which method is used to add or update an object in an object store?
    A. get()
    B. add()
    C. put()
    D. delete()

10. Why might multiple database requests require nesting of onsuccess event handlers?
    A. Because transactions are synchronous
    B. Because operations are asynchronous and order matters
    C. Because Indexed DB does not support multiple requests
    D. Because indices can only be used sequentially

Day 16 - Learning Unit 16

Introduction:

This chapter will demonstrate two technologies that provide powerful features that enable you to easily create some useful web sites. Geolocation provides a standardized API that is used to determine the client's location. Mapping technology adds the ability to display this location on a map along with other points of interest. Together, these form a platform that has many useful applications. The facilitator will cover slide 16 and is linked to Chapter 26 in the Textbook.

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/

Learning outcome:

- HTML
- CSS
- JavaScript

Additional Resources

Basic Geolocation API

```
// Check if supported
if (navigator.geolocation) {
  navigator.geolocation.getCurrentPosition(success, error);
} else {
  alert("Geolocation not supported");
}

// Success callback
function success(position) {
  console.log("Latitude:", position.coords.latitude);
  console.log("Longitude:", position.coords.longitude);
  console.log("Accuracy (m):", position.coords.accuracy);
}

// Error callback
function error(err) {
  console.warn("ERROR(" + err.code + "): " + err.message);
}
```

Watch Position (Track movement)

```
let watchId = navigator.geolocation.watchPosition(success, error, {
  enableHighAccuracy: true,
  timeout: 5000,
  maximumAge: 0
});

// Stop watching
navigator.geolocation.clearWatch(watchId);
```

Position Object

```
position.coords.latitude       // Decimal degrees
position.coords.longitude      // Decimal degrees
position.coords.altitude       // In meters (may be null)
position.coords.accuracy       // Meters
position.coords.altitudeAccuracy // Meters (may be null)
position.coords.heading        // Degrees (0 = north)
position.coords.speed          // m/s (may be null)
position.timestamp             // Time (ms since epoch)
```

Options

```
{
  enableHighAccuracy: true,  // GPS if available
  timeout: 10000,            // Max time (ms) to wait
  maximumAge: 0              // Accept cached position (ms)
}
```

Simple Google Maps Embed (no API key needed)

```
<iframe
  width="600" height="450" style="border:0"
  loading="lazy" allowfullscreen

src="https://www.google.com/maps/embed/v1/view?key=YOUR_API_KEY&center=LAT,LNG&zoom=14">
</iframe>
```

Google Maps JavaScript API (basic)

```
<div id="map" style="height:400px; width:100%"></div>
<script src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY"></script>
<script>
function initMap() {
  const location = { lat: -25.7479, lng: 28.2293 }; // Example: Pretoria
  const map = new google.maps.Map(document.getElementById("map"), {
    zoom: 12,
    center: location
  });
  new google.maps.Marker({ position: location, map: map });
}
window.onload = initMap;
</script>
```

Leaflet.js (Open-source Mapping)

```
<link rel="stylesheet" href="https://unpkg.com/leaflet/dist/leaflet.css"/>
<script src="https://unpkg.com/leaflet/dist/leaflet.js"></script>

<div id="map" style="height:400px;"></div>
<script>
  const map = L.map('map').setView([-25.7479, 28.2293], 13);
  L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
    attribution: '&copy; OpenStreetMap contributors'
  }).addTo(map);
  L.marker([-25.7479, 28.2293]).addTo(map)
    .bindPopup("Hello from Pretoria!")
    .openPopup();
</script>
```

Combining Geolocation + Map

```
navigator.geolocation.getCurrentPosition((pos) => {
  const lat = pos.coords.latitude;
  const lng = pos.coords.longitude;
  map.setView([lat, lng], 13);
  L.marker([lat, lng]).addTo(map)
    .bindPopup("You are here!")
    .openPopup();
});
```

Class activity 16

Class Activity: Geolocation and Mapping

Objective:

To demonstrate an understanding of Geolocation and Mapping.

Instructions:

Students will have to use VSCode or any other text editor of their choice.

1. Create an HTML page with:
    - A button to get the current location.
    - A to display the map.

2. Use JavaScript to:
    - Get the user's latitude and longitude.
    - Display the coordinates on the page.
    - Show a map with a marker at the user's location.

3. Use Leaflet.js for mapping (a simple open-source mapping library).

Homework: Facilitator to give additional homework activity

Revision Questions Day 16

1. Which object provides access to the Geolocation API in a browser?
    A. window.geolocation
    B. navigator.geolocation
    C. document.geolocation
    D. location.geolocation

2. Which geolocation technology provides the most accurate location in rural areas?
    A. IP block
    B. Wi-Fi positioning
    C. Cell tower triangulation
    D. GPS

3. What is the main purpose of the getCurrentPosition() function?
    A. Continuously monitor the device location
    B. Retrieve the current location once
    C. Display a map
    D. Create pushpins

4. What does the enableHighAccuracy option in getCurrentPosition() do?
    A. Forces the browser to use GPS only
    B. Provides a hint to the browser to improve location accuracy
    C. Increases the timeout value
    D. Uses cached positions only

5. Which of the following error codes indicates that the user denied permission to access their location?
    A. 1: PERMISSION_DENIED
    B. 2: POSITION_UNAVAILABLE
    C. 3: TIMEOUT
    D. 4: LOCATION_BLOCKED

6. Which method allows you to continuously monitor a device's location?
    A. getCurrentPosition()
    B. watchPosition()
    C. clearWatch()
    D. setInterval()

7. When using Bing Maps, which object is used to add a marker for a location on the map?
    A. MapMarker
    B. Pushpin
    C. LocationMarker
    D. MapPoint

8. How do you stop monitoring a position that is being tracked with watchPosition()?
    A. stopWatch(handle)
    B. cancelWatch(handle)
    C. clearWatch(handle)
    D. deleteWatch(handle)

9. Why is knowing latitude and longitude alone often insufficient for web applications?
    A. It requires a Bing Maps API key
    B. Users cannot understand raw coordinates easily
    C. Coordinates are always inaccurate
    D. Only GPS provides coordinates

10. Which property of the Position object specifies the accuracy of the reported location?
    A. coords.altitude
    B. coords.accuracy
    C. coords.heading
    D. coords.speed

Day 17 to 19 – Summative Revision

Introduction:

Revision will be conducted to prepare students for their summative exams.

Day 17 – Summative Exam

Introduction:

Summative exams will be conducted on campus under invigilation. Strict exam rules will be enforced during the summative exam.

Practical for the practical modules will have to be handed in.

Bibliography

Textbook: Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices by Mark J. Collins. Available on O'Reilly Books online at: https://learning.oreilly.com/library/view/pro-html5-with/9781484224632/