

# **HTML5 - NexTGen Web**

# **HTML5 - NextGen Web Trainer's Guide**

**© 2014 Aptech Limited**

All rights reserved.

No part of this book may be reproduced or copied in any form or by any means – graphic, electronic or mechanical, including photocopying, recording, taping, or storing in information retrieval system or sent or transferred without the prior written permission of copyright owner Aptech Limited.

All trademarks acknowledged.

**APTECH LIMITED**

Contact E-mail: [ov-support@onlinevarsity.com](mailto:ov-support@onlinevarsity.com)

First Edition - 2014



*Unleash your potential*



**Dear Learner,**

**We congratulate you on your decision to pursue an Aptech course.**

**Aptech Ltd. designs its courses using a sound instructional design model – from conceptualization to execution, incorporating the following key aspects:**

- Scanning the user system and needs assessment

**Needs assessment is carried out to find the educational and training needs of the learner.**

**Technology trends are regularly scanned and tracked by core teams at Aptech Ltd. TAG\* analyzes these on a monthly basis to understand the emerging technology training needs for the Industry.**

**An annual Industry Recruitment Profile Survey<sup>#</sup> is conducted during August - October to understand the technologies that Industries would be adapting in the next 2 to 3 years. An analysis of these trends & recruitment needs is then carried out to understand the skill requirements for different roles & career opportunities.**

**The skill requirements are then mapped with the learner profile (user system) to derive the Learning objectives for the different roles.**

- Needs analysis and design of curriculum

**The Learning objectives are then analyzed and translated into learning tasks. Each learning task or activity is analyzed in terms of knowledge, skills and attitudes that are required to perform that task. Teachers and domain experts do this jointly. These are then grouped in clusters to form the subjects to be covered by the curriculum.**

**In addition, the society, the teachers, and the industry expect certain knowledge and skills that are related to abilities such as *learning-to-learn, thinking, adaptability, problem solving, positive attitude etc.* These competencies would cover both cognitive and affective domains.**

A precedence diagram for the subjects is drawn where the prerequisites for each subject are graphically illustrated. The number of levels in this diagram is determined by the duration of the course in terms of number of semesters etc. Using the precedence diagram and the time duration for each subject, the curriculum is organized.

- Design & development of instructional materials

**The content outlines are developed by including additional topics that are required for the completion of the domain and for the logical development of the competencies identified. Evaluation strategy and scheme is developed for the subject. The topics are arranged/organized in a meaningful sequence.**

**The detailed instructional material – Training aids, Learner material, reference material, project guidelines, etc.- are then developed. Rigorous quality checks are conducted at every stage.**

- Strategies for delivery of instruction

**Careful consideration is given for the integral development of abilities like thinking, problem solving, learning-to-learn etc. by selecting appropriate instructional strategies (training methodology), instructional activities and instructional materials.**

**The area of IT is fast changing and nebulous. Hence, considerable flexibility is provided in the instructional process by specially including creative activities with group interaction between the students and the trainer. The positive aspects of Web based learning –acquiring information, organizing information and acting on the basis of insufficient information are some of the aspects, which are incorporated, in the instructional process.**

- Assessment of learning

**The learning is assessed through different modes – tests, assignments & projects. The assessment system is designed to evaluate the level of knowledge & skills as defined by the learning objectives.**

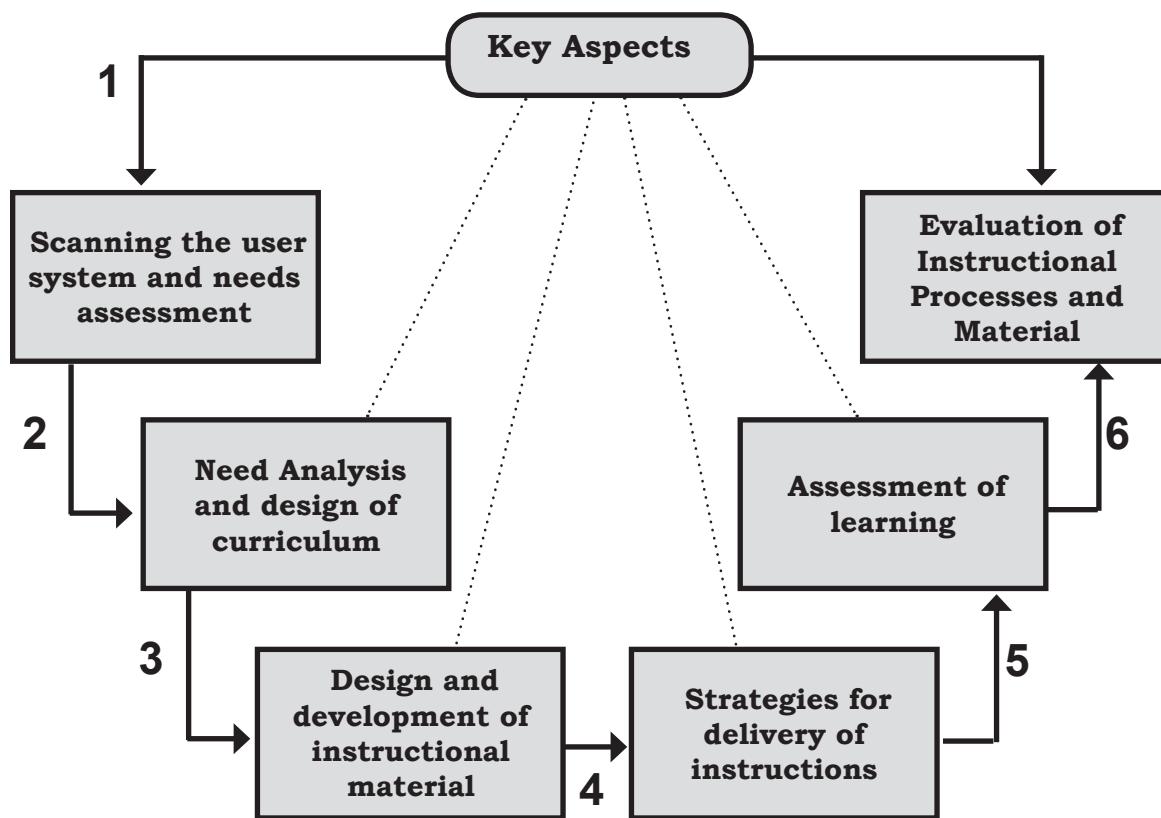
- Evaluation of instructional process and instructional materials

**The instructional process is backed by an elaborate monitoring system to evaluate - on-time delivery, understanding of a subject module, ability of the instructor to impart learning. As an integral part of this process, we request you to kindly send us your feedback in the reply pre-paid form appended at the end of each module.**

\*TAG – Technology & Academics Group comprises of members from Aptech Ltd., professors from reputed Academic Institutions, Senior Managers from Industry, Technical gurus from Software Majors & representatives from regulatory organizations/forums.

Technology heads of Aptech Ltd. meet on a monthly basis to share and evaluate the technology trends. The group interfaces with the representatives of the TAG thrice a year to review and validate the technology and academic directions and endeavors of Aptech Ltd.

### Aptech New Products Design Model



“

A little learning is a dangerous thing,  
but a lot of ignorance is just as bad

”

---

## Preface

---

The book 'HTML5 - NextGen Web' Trainer's Guide aims to teach the students the HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript. HTML is the markup language used by developers to design Web sites for the World Wide Web. CSS helps Web designers to create style sheets for the Web sites. JavaScript is a scripting language used for adding interactivity to Web pages. The book also covers the new features of HTML5.

The faculty/trainer should teach the concepts in the theory class using the slides. This Trainer's Guide will provide guidance on the flow of the session and also provide tips and additional examples wherever necessary. The trainer can ask questions to make the session interactive and also to test the understanding of the students.

This book is the result of a concentrated effort of the Design Team, which is continuously striving to bring you the best and the latest in Information Technology. The process of design has been a part of the ISO 9001 Certification for Aptech-IT Division, Education Support Services. As part of Aptech's quality drive, this team does intensive research and curriculum enrichment to keep it in line with industry trends.

We will be glad to receive your suggestions.

Design Team

“ Practice is the best of  
all instructors. ”



## Table of Contents

### Sessions

1. Introduction to the Web
2. Introduction to HTML5
3. Formatting Text Using Tags
4. Creating Hyperlinks and Anchors
5. Introduction to CSS3
6. Formatting Using Style Sheets
7. Displaying Graphics and CSS3 Animation
8. Creating Navigational Aids and Division-Based Layout
9. Creating Tables
10. HTML Forms
11. HTML5 Ausio and Video
12. Introduction to JavaScript
13. Operators and Statements
14. Loops and Arrays
15. Functions and Objects
16. Building a Mobile Web Application
17. Canvas and JavaScript
18. HTML5 Web Storage
19. HTML5 Geolocation and APIs

“ The future depends on what  
we do in the present. ”

# Session 1 – Introduction to the Web

---

## 1.1 Pre-Class Activities

Before you commence the session, you should familiarize yourself with the topics of this session in-depth.

### 1.1.1 Objectives

By the end of this session, the learners will be able to:

- Explain the evolution of HTML
- Explain the page structure used by HTML
- List the drawbacks in HTML 4 and XHTML
- List the new features of HTML 5
- Explain CSS
- Explain JavaScript
- Explain jQuery
- Explain browser support for HTML 5

### 1.1.2 Teaching Skills

To teach this session, you should be well-versed with concept Web, HTML, and its evolution. You should also be aware with drawbacks of HTML 4 and XHTML. You should explore the Websites to get the understanding on the new features of HTML5, CSS, JavaScript, and jQuery.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

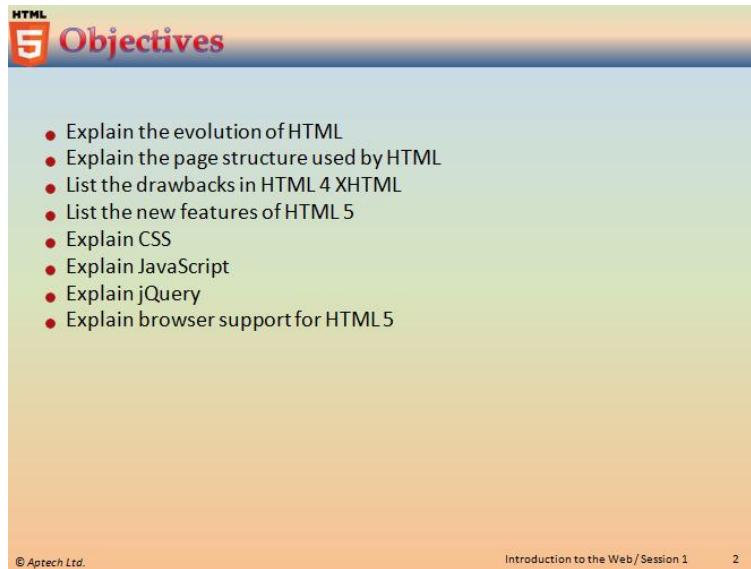
It is recommended that you test the understanding of the students by asking questions in between the class.

#### In-Class Activities:

Follow the order given here during In-Class activities.

## Overview of the Session:

Give the students a brief overview of the current session in the form of session objectives.  
Show the students slide 2 of the presentation.



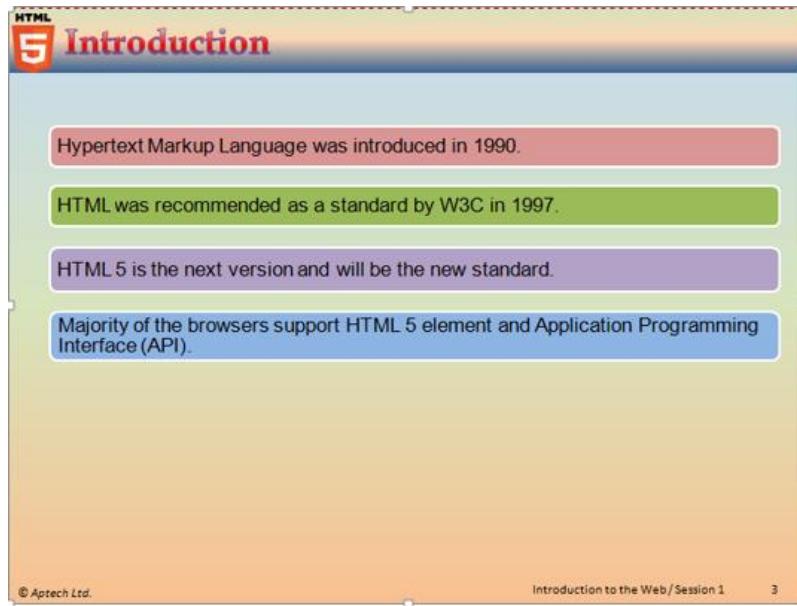
Tell the students that this session will introduce them to the basic concepts of Web, HTML, and its evolution to be selected as the Web language for design the Web pages for your Web sites.

They will learn about the new features of HTML 5 and its predecessor versions drawbacks. Apart from HTML5, they will also understand the features of other Web technologies such as CSS, JavaScript, and jQuery in this session.

## 1.2 In-Class Explanations

### Slide 3

Let us understand the HyperText Markup Language (HTML).



Using slide 3, explain the introduction of Web and HTML. Explain the student about the evolution of HTML language.

Tim Berners-Lee is the inventor of the World Wide Web (W3C). In 1989, Tim was working in a computing services section of CERN when he came up with the concept of sharing the data globally through a hypermedia. Since then, the use of Internet came into existence. There has been constant evolution of the technologies to the W3C.

HyperText Markup Language (HTML) was introduced as Web language in the year 1990. It is the standard language used for creating Web pages that are accessible over the Internet. Mention HTML 4 was recommended as a standard by W3C in 1997. HTML5 is the next version of HTML and will be the new standard. Today, majority of the browsers support HTML5 elements and Application Programming Interfaces (APIs) supported by HTML5.

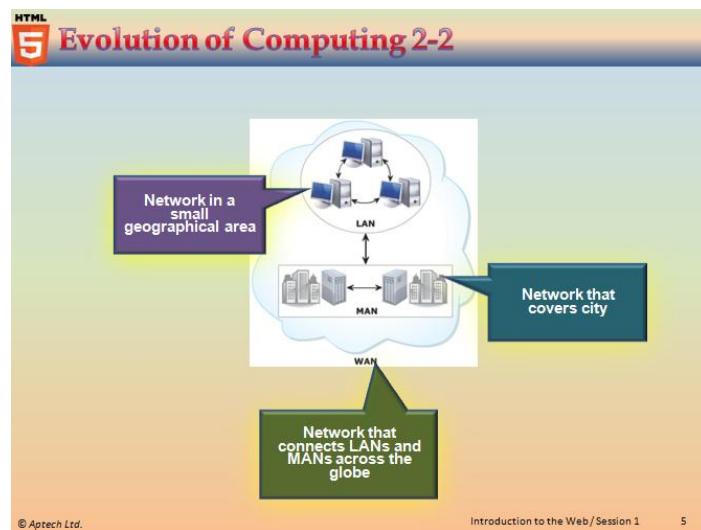
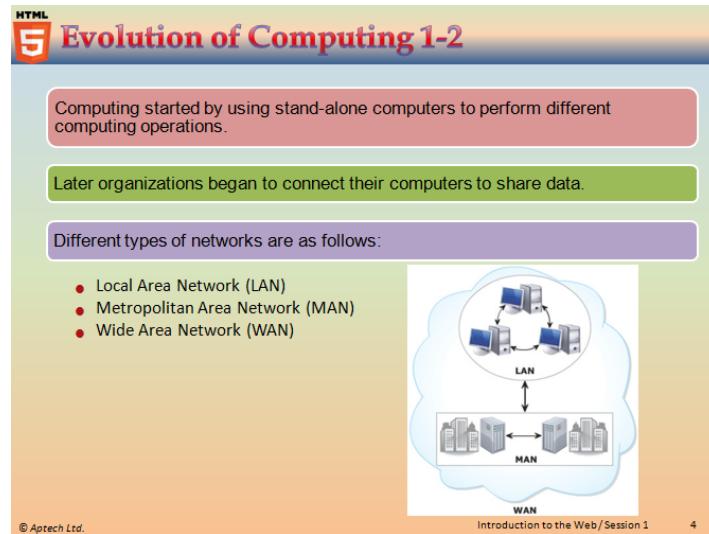
#### Tips:

World Wide Web is a network of information in the form of the resources. To make these resources available to the people across the globe, it has three mechanisms:

- A Uniform Resource Locator (URL).
- A protocol to access the resource, such as HyperText Transfer Protocol (HTTP).
- Hypertext, such as HTML for accessing resources by navigation.

## Slides 4 and 5

Let us understand the evolution of computing



Using slides 4 and 5, explain the evolution of computing to the students.

Explain about the isolated systems used in different fields such as research and military. These systems were connected after years for processing large amount of data and to get faster results. Thus, sharing of resources was required. However, as devices were very far from each other resulted in networking of computers.

Networks provide interconnecting between the systems that permits distributed processing of information. Organizations began to connect their computers and share data amongst their people.

Mention the types of networks:

- Local Area Network (LAN)
- Metropolitan Area Network (MAN)

- Wide Area Network (WAN)

Explain LAN, MAN, and WAN in detail using figure in slide 5.

## Slides 6 and 7

Let us understand Web and Internet.

**HTML 5 Web and Internet 1-2**

WAN raised the need to share data across the globe rather than within an organization.

This resulted in the evolution of Web also known as World Wide Web (WWW).

Internet is known as the largest WAN.

Web is a way to access information using Internet.

Multiple computers are connected to each other irrespective of geographical locations.

© Aptech Ltd. Introduction to the Web / Session 1 6

**HTML 5 Web and Internet 2-2**

Information is available in the form of Web pages

Web page is a file containing information and instruction

Information is displayed to the user

© Aptech Ltd. Introduction to the Web / Session 1 7

Using slides 6 and 7, explain the concepts of Web and Internet to the students.

Mention advent of WANs raised a strong need to share data across the globe rather than just sharing the data within the organization. This is because organizations can share their problems, solutions, experiences, and updates along with other organizations and customers. This would facilitate faster analysis and decision-making process. This resulted in

the evolution of the Web, also referred as World Wide Web or WWW. Internet is known as the largest WAN.

The Web is a way to access information using the Internet that is referred to as a network of networks. Here, multiple computers are connected to each other irrespective of their geographical locations. Information is made available across the globe in the form of Web pages. The Web pages are created as a part of Web applications or Websites that are hosted on the Web servers.

The Web page request is sent to a server which can be located anywhere by the user using browser and the server response back to the user by showing a Web page. The URL entered in the address bar of a browser is the request sent to the server and the Web page received is the response from the server.

**Tips:**

The browsers are also referred to as Web clients.

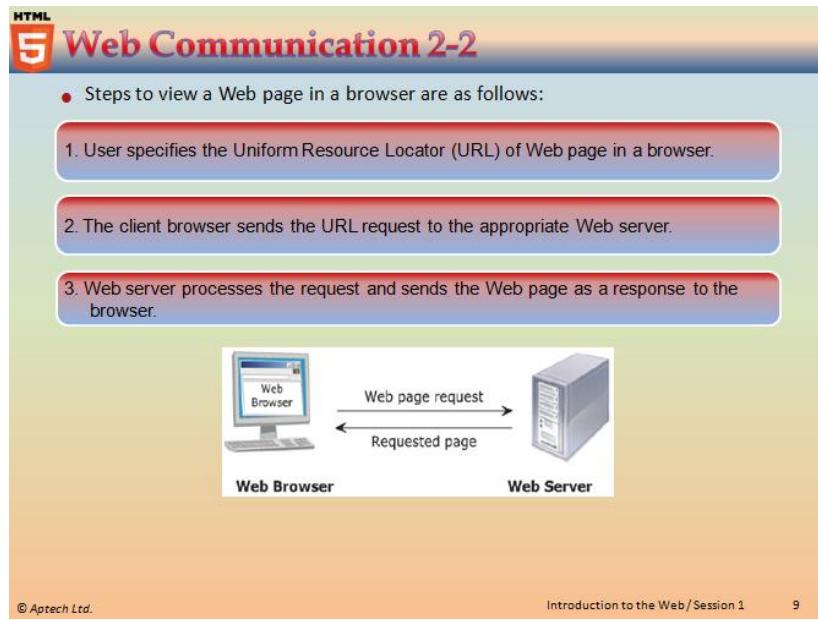
## Slides 8 and 9

Let us understand Web communication.

**HTML 5 Web Communication 1-2**

- Web pages are stored on a Web server to make them available on the Internet for the users.
- Web server is a computer with high processing speed and connected to the Internet.
- Web server is used to host and display the Web pages on a Web browser.
- Web browser displays the Web pages using the HTTP protocol.
- HTTP is a protocol that specifies how a Web page will be retrieved from the Web server.

© Aptech Ltd. Introduction to the Web / Session 1 8



Using slides 8 and 9, explain the need of Web for communication.

Mention, Web pages are stored on a Web server to make them available on the Internet so that users can view them. A Web server is a computer with high processing speed connected to the Internet and is used to host Web pages. Web browsers such as Microsoft Internet Explorer or Netscape Navigator are used to interpret and display the Web pages using a protocol (set of rules). The most popular protocol used to view Web pages is Hypertext Transfer Protocol (HTTP). It is a protocol that specifies how a Web page will be retrieved from the Web Server.

Using slide 9, explain the process of Web communication with the help of the figure displayed on the slide.

### Tips:

HTML is a language for describing web pages. The features of HTML language are as follows:

- HTML stands for Hyper Text Markup Language.
- HTML documents are also called web pages.
- HTML is a markup language which contains a set of markup tags. Tags are basically the angular bracket <> with keywords.
- The tags describe document content. For example, <p> Content </p> adds the text as paragraph on the Web page.
- HTML documents contain HTML tags and plain text

The purpose of a web browser such as Google Chrome, Internet Explorer, Firefox, Safari, and so on is to read HTML documents and display them as web pages.

**In-Class Question:**

After you finish explaining Web communication, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



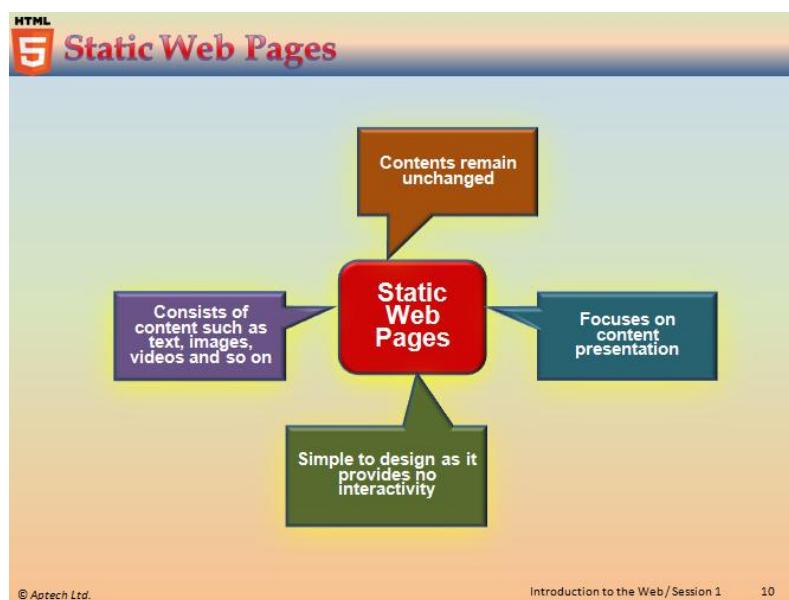
What types of contents are observed on a Web page?

**Answer:**

Text, Images, Audio, Video, Forms, Hyperlinks, and so on.

**Slide 10**

Let us understand the static Web pages.



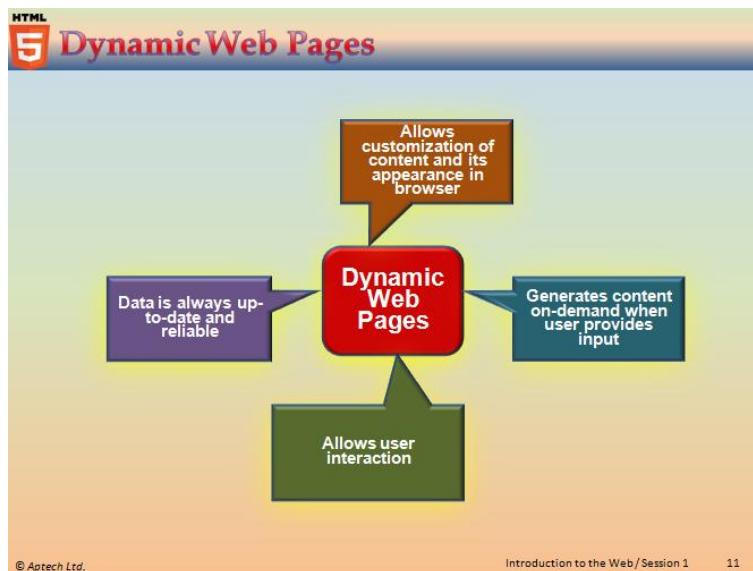
Using slide 10, explain the static Web pages to the students.

Static Web pages contain simple content, not interactive and does not involve any dynamic behavior on the Web pages. Therefore, such a Web page is called a static Web page, as the contents of the Web page remain unchanged. The only way to update a static Web page is to change the content manually.

Static Web pages include contents developed using HTML tags. The static content includes: text, images, audio, video, hyperlink, and so on.

## Slide 11

Let us understand dynamic Web pages.



Using slide 11, explain the dynamic Web pages to the students.

Explain the limitation of static Web page which resulted in development or evolution of dynamic Web page.

A dynamic Web page generates content 'on-demand' when user provides certain inputs. It accepts the inputs from the user based on which it displays the content in the browser. Consider an example of an online store where the users can buy different products by selecting them online. Based on the selected products (input), a page with the total cost is displayed to the user.

The page interacts with the user based on the action performed by the user.

## Slide 12

Let us understand the technologies for creating dynamic Web pages

**HTML Technologies**

- Technologies used for creating dynamic Web sites are as follows:

JavaScript, a scripting language, is used for creating dynamic Web pages.

CSS specifies the formatting of a Web page for both static and dynamic Web pages.

Extensible HTML when used with JavaScript, displays the required user-defined data each time the Web page is loaded in the browser.

Dynamic HTML uses JavaScript and CSS to make dynamic Web pages and transform the look and feel of the Web pages.

© Aptech Ltd. Introduction to the Web / Session 1 12

Using slide 12 explain the technologies used for developing a Website to the students. Explain requirements for a Website such as UI interface, database, and also the logical tier for processing data.

Mention use of Cascading Style Sheet (CSS). CSS are style sheets that specify the formatting of a Web page for both static and dynamic Web pages. The formatting options include font, color, background, spacing, positioning, and borders. It is used in combination with JavaScript to format Web pages dynamically.

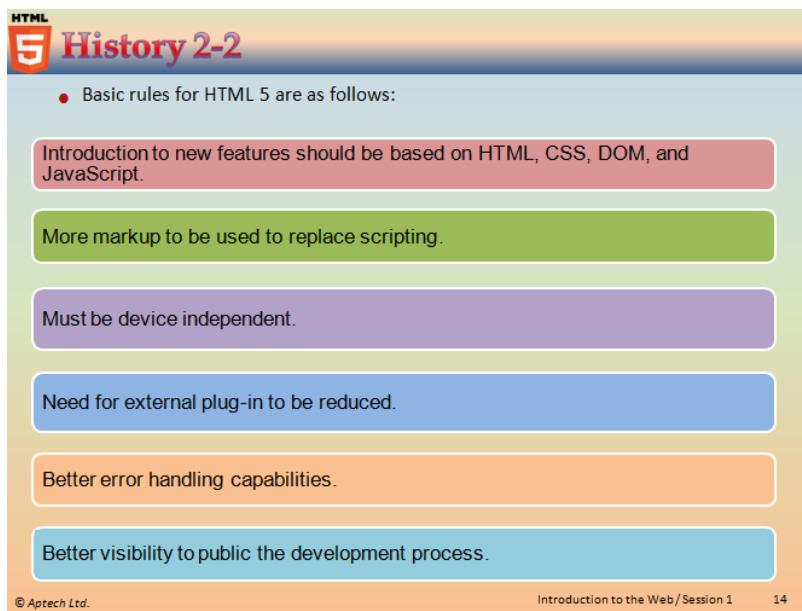
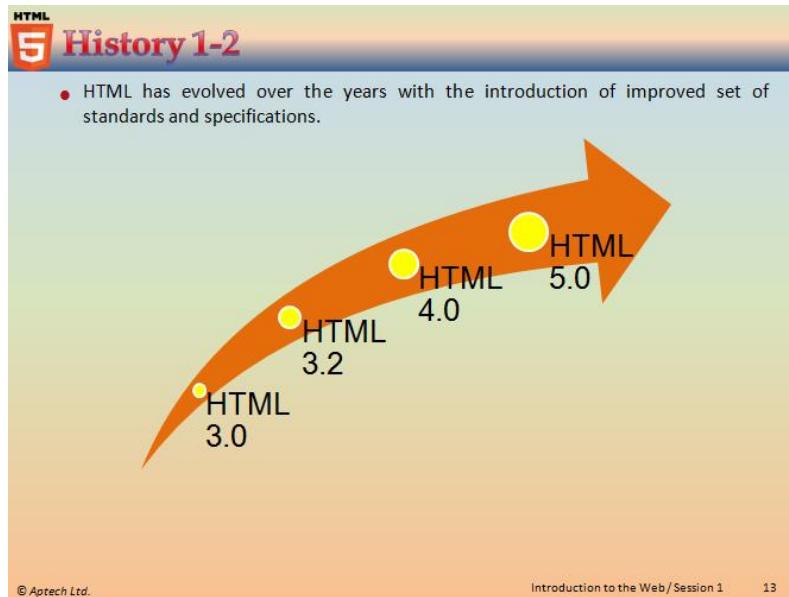
Explain them the use of JavaScript on Web pages. JavaScript is used to develop interactive Web pages by adding programming on the page.

Similarly, explain Extensible HTML (XHTML) and Dynamic HTML (DHTML). XHTML is a language that combines HTML with Extensible Markup Language (XML). XML allows defining your own data in a structured format, which can be displayed in any browser. When you use XHTML with JavaScript, the required user-defined data is displayed each time the Web page is loaded in the browser.

DHTML uses JavaScript and CSS to make dynamic Web pages. It allows you to transform the look and feel of Web pages. It allows Web pages to respond to the user's actions and enables focus on the content changes in the browser.

## Slides 13 and 14

Let us understand about HTML language and its versions.



Using slides 13 and 14, explain the history of HTML.

Mention, HTML is derived from Standard Generalized Markup language (SGML). SGML is a markup language that defines the structure of other markup languages. HTML has evolved over the years with the introduction of improved set of standards and specifications.

HTML 1.0 was the first version of HTML introduced in 1993. At that time, there were very less people involved in designing Web sites. HTML 2.0 was introduced in 1995 and included the complete HTML 1.0 specifications with additional features.

The other versions are as follows:

- **HTML 3.0:** HTML 3.0 specifications included new features for the Netscape Navigator browser as it became very popular. The new improvements did not work on any other browsers such as Internet Explorer. Therefore, this specification was abandoned.
- **HTML 3.2:** Additional browser-specific features revolutionized the need for standardization of HTML. Therefore, the World Wide Consortium (W3C) organization was formed to specify and maintain the HTML standards. HTML 3.2 was the first specification introduced by W3C in January 1997 and was fully supported by all the Web browsers.
- **HTML 4.0:** W3C introduced HTML 4.0 in December 1997 with the motive for facilitating support for CSS, DHTML, and JavaScript. However, HTML 4.0 prevailed for a short period and was revised, which led to HTML 4.01 specification in 1999.

**Tips:**

SGML was originally designed to enable the sharing of machine-readable large-project documents in government, law, and industry.

**In-Class Question:**

After you finish explaining HTML, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which language is mother of HTML?

**Answer:**

SGML is the mother language of HTML.

Then, mention HTML5 is cooperative project between the W3C and the Web Hypertext Application Technology Working Group (WHATWG).

W3C was busy working with XHTML 2.0 and WHATWG was working with Web forms, new HTML features, and applications. In 2006, the two groups decided to work together and develop a new version of HTML. Then, explain the basic rules of HTML5 listed on the slide.

## Slides 15 and 16

Let us understand the layout of a page in HTML.

**HTML 5 Layout of a Page in HTML 5 1-2**

HTML 5 contains a head section containing the unseen elements and the body section containing the visible elements of the document.

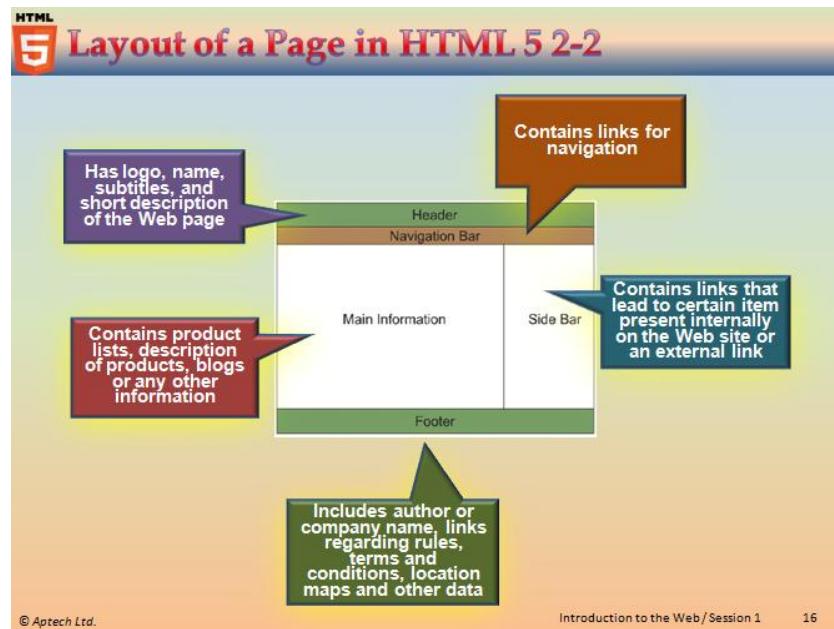
Earlier HTML provided different tags to build and organize the content in the body of the document.

The <table> tag was an element often used to present the data in an organized manner.

The <div> tag was another element used to display contents such as images, links, text, menus, forms, and so on.

HTML 5 includes new elements that identify and organize each part of the document body.

© Aptech Ltd. Introduction to the Web / Session 1 15



Using slides 15 and 16, explain the layout of a page in HTML 5.

Explain that the layout of page in HTML 5 is same like that of HTML appended with the support for new tags and Application Programming Interfaces (APIs).

Explain them the <table> and <div> tags used to layout the page in HTML. However, in HTML5 the main structure does not depend on <div> or <table> tags.

Explain the different sections on the HTML5 page used to layout the Web page. The header on the top usually has the logo, name, subtitles, and short descriptions of the Website or Web page.

Similarly, the navigation bar includes a menu that contains links for navigation. Web users can navigate to different pages or documents using the navigation bar.

The most relevant content is generally shown in the middle of the page. The content presented in the main information part of the layout usually has a top priority. It can have a list of products, description of products, blogs, or any other important information.

The side bar shows a list of links that lead to certain items that may be present internally on the Web site or on an external link. For example, in a blog, the last column offers a list of links that can lead to the blog entries, information about the author, and so on. These two sections are extremely flexible. Web designers can perform variety of actions, such as inserting more rows or splitting the columns, to edit the Web page as required.

The footer at the bottom is used to represent general information about the Website. This can include the author or the company name, links regarding rules, terms and conditions, location maps, and any other additional data.

**In-Class Question:**

After you finish explaining HTML5, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What are the different parts of a Web page in HTML5?

**Answer:**

Header, navigation bar, main content, sidebar, and footer.

## Slide 17

Let us understand the drawbacks of HTML4 and XHTML.

The slide has a header 'HTML' with a logo and 'Drawbacks in HTML 4 and XHTML'. It contains five colored callouts:

- Red: HTML 4 was a universally accepted standard for developing Web sites and is a stable coding language.
- Green: HTML 4 is compatible with all important browsers.
- Purple: HTML 5 adds new capabilities and provides improvements through better interactivity, multimedia services, and application handling.
- Blue: XHTML created with XML is a rigid, standard-based language.
- Orange: XHTML was supposed to be the next version of XML but took a backseat because of interoperability problem.

At the bottom left is '© Aptech Ltd.', at the bottom center is 'Introduction to the Web / Session 1', and at the bottom right is '17'.

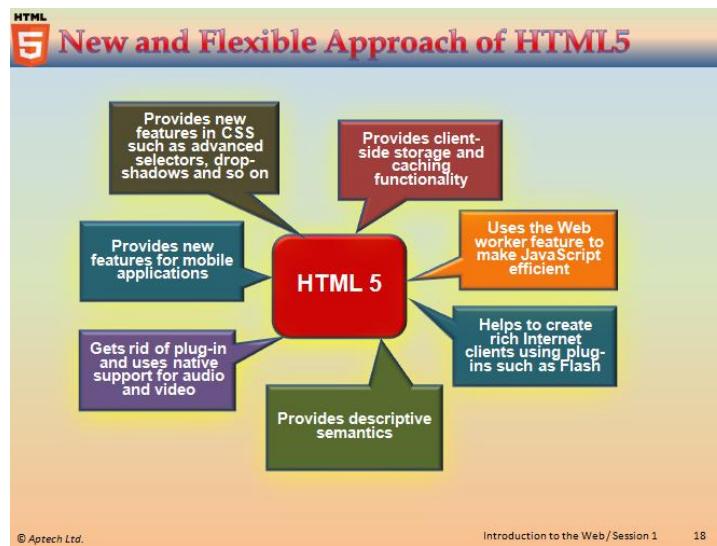
Using slide 17, explain drawbacks of the HTML4 and XHTML in details.

Explain that HTML 4 was a standard that was universally accepted for developing Web sites. It is a very stable coding language which ignores small coding errors. Also, HTML 4 is majorly compatible with all important browsers.

XHTML was a version of HTML 4 that was created along with XML. It was a rigid, standards-based language that allowed no room for errors. XHTML was supposed to be the next version of HTML 4 but due to interoperability problems, it took a backseat and HTML5 would be the next standard for Web site development.

## Slide 18

Let us understand new approach of HTML5.



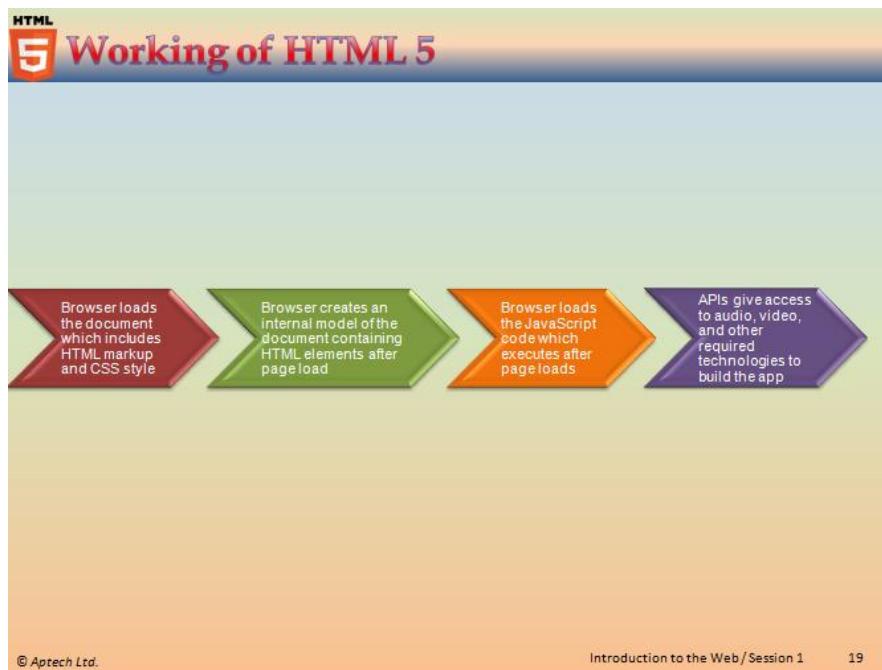
Using slide 18, explain the new and flexible approach in HTML 5 in detail.

Explain the different aspects of HTML 5 by comparing them with the HTML:

- For a multimedia person, HTML5 gets rid of plug-ins and uses new native support for audio and video.
- For a Web designer, HTML5 provides descriptive semantics.
- For a programmer, HTML5 helps to create rich Internet clients. These clients can be built without using plug-ins such as Flash. For this, you can use canvas and JavaScript to create better interfaces and animations. Canvas is a rectangular area on the Web page that uses JavaScript. A developer can control every single pixel in the area. The canvas element has several ways to draw paths, rectangles, filled rectangles, circles, images, and so on.
- For a client-side programmer, the Web workers is one of the features provided that can make JavaScript more efficient. Web workers is a JavaScript based API that is used to run background scripts in a Web application. This helps to mitigate the effect of the background script affecting the main process that is being executed.
- For database administrator, HTML5 has client-side storage and caching functionality.
- For a design expert, CSS in HTML5 has been improved by added features such as advanced selectors, animations, drop-shadows, and so on.
- For a mobile programmer, a lot of features are included for mobile applications. HTML5 is a family of technologies that gives whole new options for building Web pages and applications.

## Slide 19

Let us understand working of HTML5.

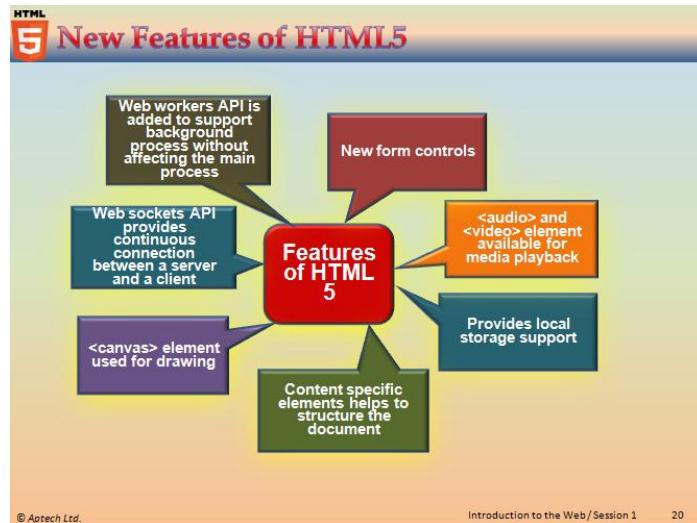


Using slide 19, explain the working of HTML 5 to the students. HTML5 is made up of a family of technologies. HTML consists of markups, improved CSS with CSS3 that provides added options to style your pages. There is also JavaScript and a new set of JavaScript APIs that are available in HTML5.

Then, explain the process followed by browser to interpret the HTML5 page as mentioned in on the slide.

## Slide 20

Let us understand the new features of HTML5.



Using slide 20, explain the new features of HTML 5. Some of the new features introduced in HTML5 are as follows:

- The `<canvas>` element is used for 2D drawing.
- New content-specific elements, such as `<article>`, `<nav>`, `<header>`, `<footer>`, `<section>`, and so on helps to structure the document.
- HTML5 has local storage support.
- The `<audio>` and `<video>` elements is available for media playback. New form controls, such as calendar, date, time, e-mail, URL, search, and so on have been provided by HTML5.
- The Web workers API is added to support background processes without disturbing the main process. The common problems faced by Web applications are slow performance when a large set of data is processed. This is due to the fact that all the processes are executed in a single thread. Web workers help to solve this problem.
- The Web Sockets API provides a continuous connection between a server and a client by using a specific port. Thus, the Web applications become efficient as the data can be easily exchanged between client and server without reloading the page constantly.
- Easier access to location specific data which is made available by devices having Global Positioning System (GPS) capabilities. This improved functionality is achieved with the help of API.
- HTML5 allows Web applications to be executed offline by storing the files and other resources required in the application cache. Web application data is saved locally using Web SQL databases.

**In-Class Question:**

After you finish explaining HTML5 new features, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which of the following tags were used to play an MP3 file before HTML5?

**Answer:**

<embed> tag whose syntax is as follows:

```
<embed height="50" width="50" src="playmusic.mp3">
```

**Slide 21**

Let us understand the Cascading Style Sheets (CSS).

**HTML 5 Cascading Style Sheets (CSS)**

- Works along with HTML to provide visual styles to document elements.
- Is a rule based language that specifies the formatting instructions for content in an HTML document.
- Purpose is to separate content from its formatting.
- Can define the layout and formatting of content in a separate file with a .css extension.
- Allows rules from different .css files to be merged or edited.
  - This task of combining and matching rules from different files is referred to as cascading.

© Aptech Ltd. Introduction to the Web / Session 1 21

Using slide 21, explain the CSS as integral part of HTML5.

A style sheet is a collection of rules that specifies the appearance of data in an HTML document. HTML is a markup language that focuses only on the layout of the content on a Web page. However, applying layouts to more than one occurrence of an HTML element in an HTML page is a tedious job.

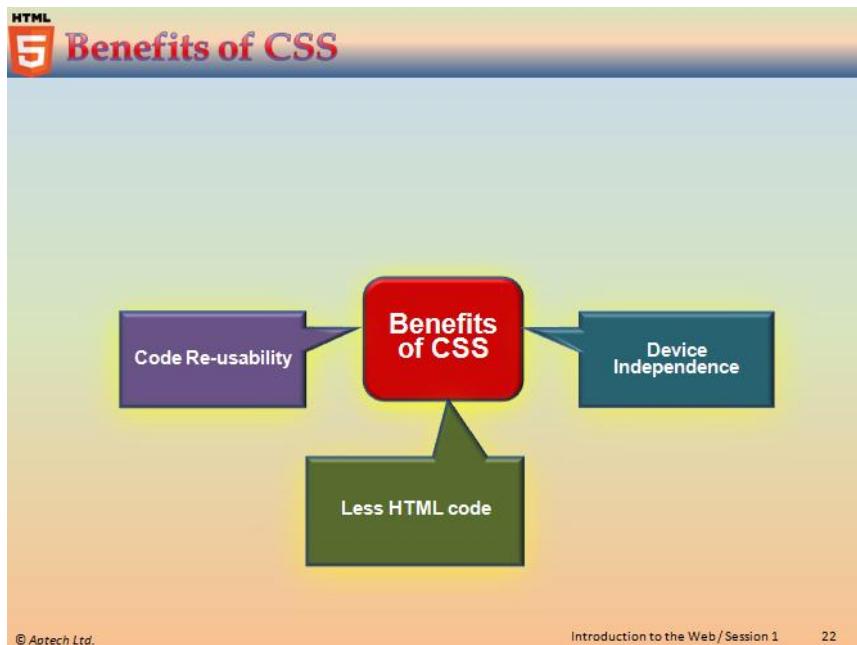
For example, if you want to change the text in the H2 element to bold, this has to be done manually for all the H2 elements. Such a manual task might result into human errors such as missing an occurrence of the H2 element for applying the bold format. This results in format inconsistency among the H2 elements within an HTML page. Further, the specified formatting might not have same appearance across various devices such as computers and mobiles.

Explain CSS is a rule-based language, which specifies the formatting instructions for the content specified in an HTML page. Its purpose is to separate HTML content from its formatting so that Web page designers would not worry about the formatting and layout. This is because they can define the layout and formatting of the content in a separate file saved with an extension of .css. In the .css file, the formatting instructions for an element are referred to as a rule set.

Each rule defines how the content specified within an element should be displayed in a Web browser. While displaying the HTML page, the browser identifies the .css file for the page and applies the rules for the specified elements. You can merge the rules from different .css files or can edit them. This task of combining and matching rules from different files is referred to as cascading.

## Slide 22

Let us understand the benefits of CSS.



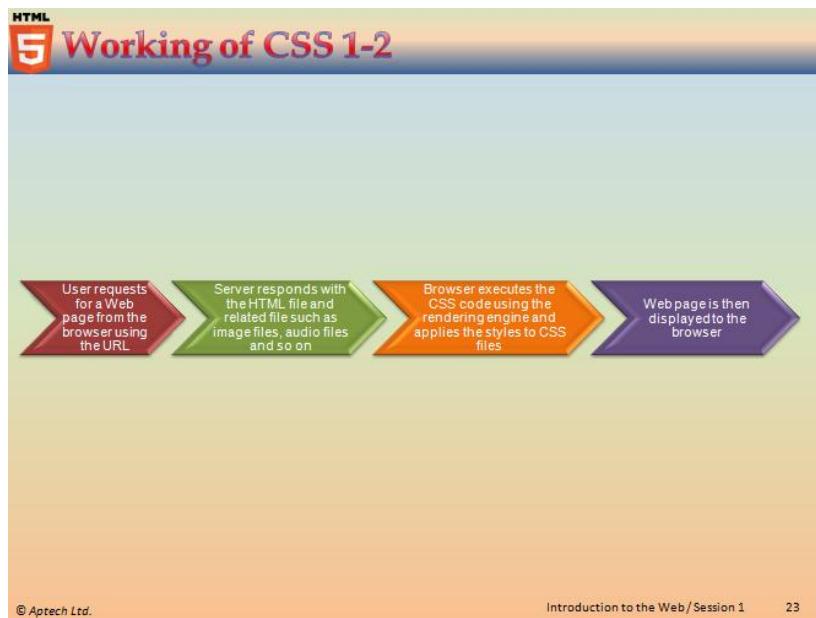
Using slide 22, explain the benefits of the CSS in detail.

Multiple HTML pages can use a CSS document. CSS provides some useful benefits that make it an ideal choice to specify the appearance of the content in an HTML page. These benefits are as follows:

- **Code Reusability:** CSS saves time by specifying the formatting options of an element only once and applying them to multiple HTML pages.
- **Less HTML Code:** CSS helps in reducing the file size of HTML documents by specifying the formatting instructions in another file.
- **Device Independence:** CSS is designed for different devices to provide the same look and feel of the HTML page across them.

## Slides 23 and 24

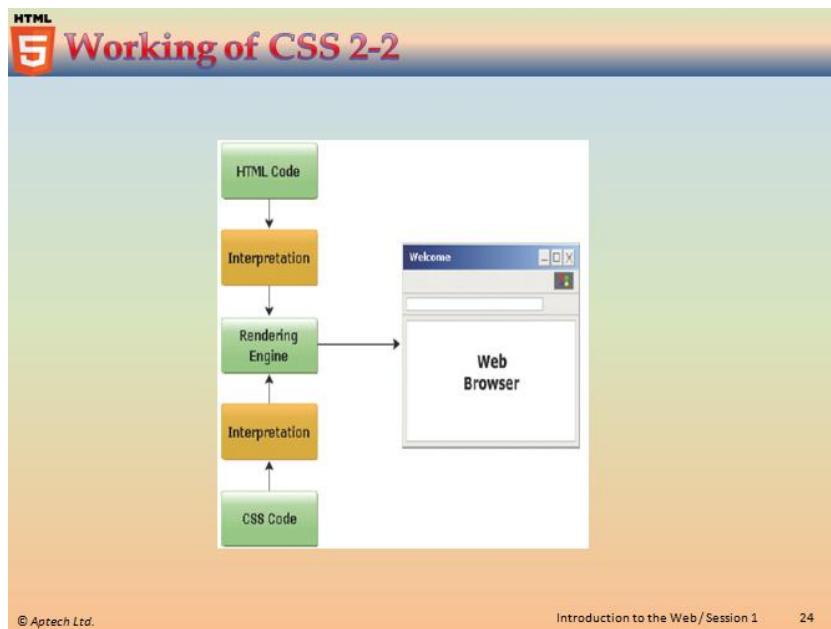
Let us understand the working of CSS.



© Aptech Ltd.

Introduction to the Web / Session 1

23



© Aptech Ltd.

Introduction to the Web / Session 1

24

Using slides 23 and 24, explain about working of CSS.

The CSS code can be embedded within the HTML code or link the HTML file externally to the CSS file. The browser will locate the style sheet irrespective of its location and will apply the style to the HTML page. There are certain steps involved in applying a style sheet to an HTML page. Explain these steps as mentioned on the slide.

Then, explain the working of CSS with help of the displayed on the slide 24.

**In-Class Question:**

After you finish explaining working of CSS, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Who is responsible to apply the formatting instructions to the element on the HTML page?

**Answer:**

Rendering engine.

**Slides 25 and 26**

Let us understand the functionality of JavaScript.

The screenshot shows a presentation slide with a purple header bar containing the text 'HTML 5 JavaScript 1-2'. Below the header, there is a title 'Functionality of JavaScript' in a purple box. A vertical blue line connects this title to a list of seven items, each enclosed in a colored box. The items are:

- Allows a user to create 2D drawable surface in your page without using plug-ins.
- Use Web Workers to turbo charge the JavaScript code to perform advanced computation.
- Accesses any Web service and brings back the data to the application in real time.
- Does not require any special plug-ins to play video.
- Allows to create own playback controls using JavaScript and HTML.
- Uses browser local storage and does not require browser cookies.
- Can perform full video processing in the browser.

At the bottom of the slide, there are copyright and page information: '© Aptech Ltd.', 'Introduction to the Web / Session 1', and '25'.

The screenshot shows a presentation slide with a purple header bar containing the text 'HTML5 JavaScript 2-2'. Below the header is a purple box with the title 'Functionality of JavaScript'. Four blue boxes list the functionality: 1. Helps Web designer to insert code snippets into the HTML page without the need for in-depth programming knowledge. 2. Can be used to execute events on certain user actions. 3. Can manipulate HTML elements using JavaScript. 4. Can collect browser information of a Web site visitor. At the bottom left is the copyright notice '© Aptech Ltd.', and at the bottom right are the slide details 'Introduction to the Web / Session 1' and '26'.

Using slides 25 and 26, explain the functionality of JavaScript.

Explain the need for scripting languages. Mention JavaScript helps to build dynamic Web pages by ensuring maximum user interactivity. JavaScript is a scripting language that supports object-oriented programming style. This means that it provides objects for specifying functionalities. An object has a unique identity, state, and behavior. JavaScript being a light-weight programming language is embedded directly into HTML pages. JavaScript is also free for use by all. It is the most popular scripting language and is supported by the major browsers.

Then explain the functionality of JavaScript. Some of the tasks that can be performed using JavaScript and HTML5 are as follows:

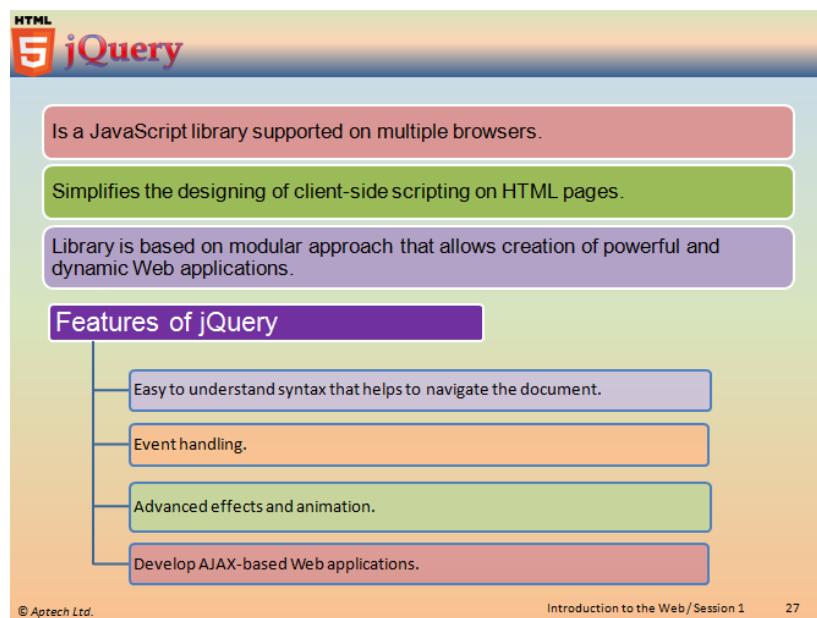
- With HTML5 and JavaScript, you can create a 2D drawable surface in your page without using any plug-ins.
- Use Web Workers to turbo charge the JavaScript code to perform advanced computation or make an application more responsive.
- Access any Web service and bring that data back to your application in real time.
- No need for special plug-ins to play video.
- Create your own video playback controls using HTML and JavaScript.
- There is no need to use browser cookies as the browser local storage can be used.
- Use JavaScript to perform full video processing in the browser. You can also create special effects and even directly manipulate video pixels.

Besides the points mentioned JavaScript can also perform the following functionalities:

- JavaScript helps Web designer to insert code snippets into the HTML pages without the need to have in-depth programming knowledge.
- JavaScript can be used to execute events on certain user actions such as on click of a HTML element, page load, and so on.
- HTML elements can be manipulated by using JavaScript.
- The browser information of a Web site visitor can be collected by using JavaScript.

## Slide 27

Let us understand jQuery API.



Using slide 27, explain the jQuery and its features.

Mention that jQuery is a JavaScript library which is supported on multiple browsers. It simplifies the designing of client-side scripting on HTML pages.

The jQuery library is based on modular approach that allows the creation of powerful and dynamic Web applications. The use of jQuery on HTML pages enable developers to abstract the low-level interaction code with pre-defined library developed on top of the JavaScript. This also helps to keep the client-side script short and concise.

Then, explain the features of jQuery library mentioned on the slide. While explaining AJAX-based development feature, mention AJAX stands for Asynchronous JavaScript and XML.

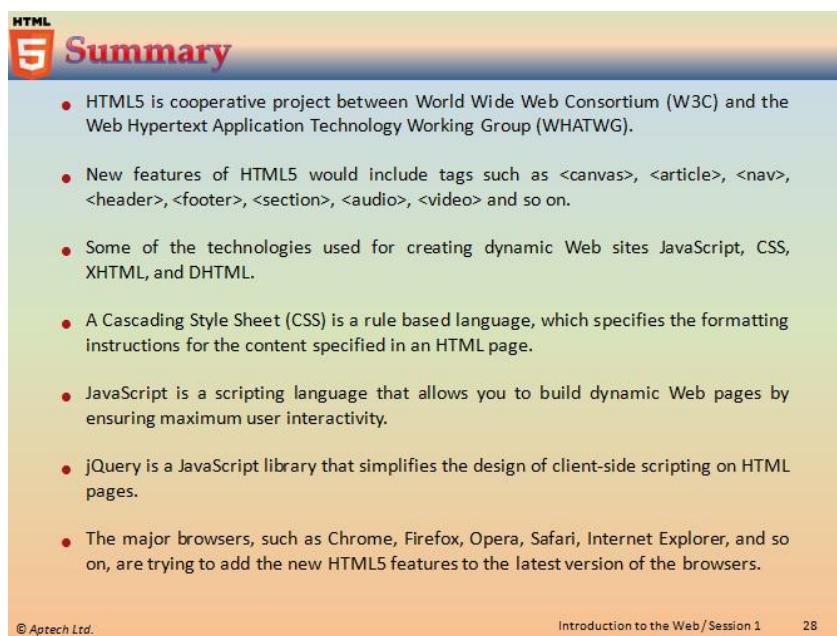
jQuery is a preferred library used by developers, as it is easy to understand than JavaScript. Also, the features of jQuery enable the development of rich Web applications in a shorter period.

**Tips:**

AJAX is the mechanism of exchanging data with a server, and updating parts of a Web page - without reloading the whole page.

**Slide 28**

Let us summarize the session.



**HTML5 Summary**

- HTML5 is cooperative project between World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).
- New features of HTML5 would include tags such as <canvas>, <article>, <nav>, <header>, <footer>, <section>, <audio>, <video> and so on.
- Some of the technologies used for creating dynamic Web sites JavaScript, CSS, XHTML, and DHTML.
- A Cascading Style Sheet (CSS) is a rule based language, which specifies the formatting instructions for the content specified in an HTML page.
- JavaScript is a scripting language that allows you to build dynamic Web pages by ensuring maximum user interactivity.
- jQuery is a JavaScript library that simplifies the design of client-side scripting on HTML pages.
- The major browsers, such as Chrome, Firefox, Opera, Safari, Internet Explorer, and so on, are trying to add the new HTML5 features to the latest version of the browsers.

© Aptech Ltd.      Introduction to the Web / Session 1      28

In slide 28, you will summarize the session. You will end the session, with a brief summary of what has been taught in the session.

### 1.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session. You should also explore the HTML editors, tags, and entities that are offered with the next session.

**Tips:**

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 2 – Introduction to HTML5

---

## 2.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 2.1.1 Objectives

By the end of this session, the learners will be able to:

- Explain the elements constituting an HTML tag
- Describe DOCTYPE declarations
- Explain the basic tags in HTML
- List the different data types, attributes, and entities of HTML5
- Describe container and standalone tags
- Explain the role of HTML5 in mobile devices

### 2.1.2 Teaching Skills

To teach this session, you should be well-versed with concept of HTML tags and DOCTYPE declarations. Also, basic tags in HTML should be known. Along with this, you should prepare yourself with the new data types, attributes, entities, and support of HTML5 in mobile devices.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

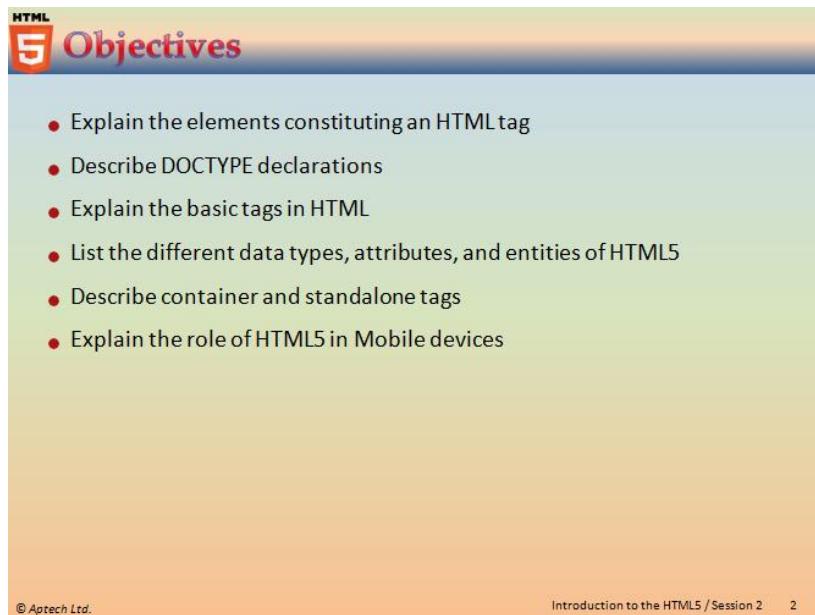
It is recommended that you test the understanding of the students by asking questions in between the class.

#### In-Class Activities:

Follow the order given here during In-Class activities.

**Overview of the Session:**

Then give the students the overview of the current session in the form of session objectives.  
Show the students slide 2 of the presentation.



The slide has a green and orange gradient background. At the top left is a red square icon containing a white '5' with the word 'HTML' above it. To its right is the word 'Objectives' in red. Below this is a bulleted list of six items. At the bottom left is the copyright notice '© Aptech Ltd.' and at the bottom right is the page number 'Introduction to the HTML5 / Session 2 2'.

- Explain the elements constituting an HTML tag
- Describe DOCTYPE declarations
- Explain the basic tags in HTML
- List the different data types, attributes, and entities of HTML5
- Describe container and standalone tags
- Explain the role of HTML5 in Mobile devices

Tell the students that this session introduces them to HTML tags and DOCTYPE declarations. They will learn about HTML5 basics and also different types of data types, attributes, and entities supported in HTML 5. They will also know about the container, standalone tags, and the role of HTML 5 in Mobile devices.

## 2.2 In-Class Explanations

### Slides 3 and 4

Let us understand elements in HTML5.

**HTML 5 Elements 1-2**

- An element organizes the content in a Web page hierarchically, which forms the basic HTML structure.

It consists of tags, attributes, and content. Tags denote the start and end of an HTML element.

A start tag includes an opening angular bracket (<) followed by the element name, zero or more space separated attributes, and a closing angular bracket (>).

Attributes are name/value pairs that describe the element and content format.

An end tag is written exactly as the start tag, but the forward slash (/) precedes the element name.

© Aptech Ltd. Introduction to the HTML5 / Session 2 3

**HTML 5 Elements 2-2**

- Following figure shows an element in HTML tag.

```

graph TD
    Element[Element] --- StartTag[StartTag]
    Element --- Content[Content]
    Element --- EndTag[EndTag]
    StartTag --- Attr[attributename="v value"]
    StartTag --- Content
    EndTag --- Content
    Attr --- Body["<BODY title='Web Content'>This is my first Web page.</BODY>"]
  
```

© Aptech Ltd. Introduction to the HTML5 / Session 2 4

Using slides 3 and 4, explain the elements of a Web page.

Mention that an HTML page organizes the content into the hierarchical tree structure. The tree structure contains HTML elements. The element organizes the content in a Web page hierarchically, which forms the basic HTML structure. Each element consists of tags, attributes, and content.

Tags denote the start and end of an HTML element. A start tag includes an opening angular bracket (<) followed by the element name, zero or more space separated attributes, and a closing angular bracket (>).

Attributes are name/value pairs that describe the element and content format. An end tag is written exactly as the start tag, but the forward slash (/) precedes the element name.

Further, explain syntax of the tags to the students in detail using the figure provided on the slide.

## Slide 5

Let us understand the DOCTYPE declaration.

The slide has a blue header bar with the text "HTML 5 DOCTYPE". The main content area has a light blue gradient background. It contains the following text and bullet points:

- Informs the browser the HTML version number of your document.
- It is the first declaration in the HTML5 document before any other HTML code is written.
- Allows a browser to be more precise in the way it interprets and renders your pages.

The new HTML5 DOCTYPE is as follows:

```
<!DOCTYPE html>
```

- It is the new syntax of HTML5 as well as for all future versions of HTML.
- This DOCTYPE is compatible with the older browsers also.

At the bottom left is the copyright notice "© Aptech Ltd.". At the bottom right is the page number "Introduction to the HTML5 / Session 2 5".

Using slide 5, explain the concept of DOCTYPE declaration.

Mention DOCTYPE element informs the browser the HTML version number of your document. It is the first declaration in the HTML5 document before any other HTML code is written. By using a DOCTYPE, the browser is able to be more precise, in the way it interprets and renders your pages. It is highly recommended to use a DOCTYPE at the beginning of all HTML documents.

The new HTML5 DOCTYPE declaration is as follows:

```
<!DOCTYPE html>
```

Not only this syntax is valid for the DOCTYPE for HTML5, but it is also the DOCTYPE for all future versions of HTML. This DOCTYPE is compatible even with the older browsers.

Mention that it is not an html tag and also not case sensitive. HTML5 is not based on SGML, and therefore does not require a reference to a DTD which is required in HTML 4.01, XHTML 1.0, and XHTML 1.1.

A Document Type Definition (DTD) defines the legal building blocks of an XML document. It defines the document structure with a list of legal elements and attributes.

### In-Class Question:

After you finish explaining DOCTYPE declaration, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What is the use of doctype declaration in a Web page?

### Answer:

The browser is able to be more precise, in the way it interprets and renders Web page if doctype declaration is present.

## Slides 6 to 11

Let us understand the basic tags.

**HTML 5 Basic Tags 1-6**

- An HTML document is made up of different elements, tags, attributes, which specify the content and its format.
- HTML is both a structural and presentational markup language.
- Structural markup specifies the structure of the content, while the presentational markup specifies the format.
- An HTML page is saved with the .html extension.
- The basic structure of an HTML document mainly consists of seven basic elements. These are as follows:

➤ **HTML**

- The **HTML** element is the root element that marks the beginning of an HTML document.
- It contains the start and end tag in the form of `<HTML>` and `</HTML>` respectively.
- It is the largest container element as it contains various other elements.

© Aptech Ltd. Introduction to the HTML5 / Session 2 6

**HTML** **5 Basic Tags 2-6**

➤ HEAD

- The HEAD element provides information about the Web page such as keywords and language used.
- Keywords are important terms existing in a Web page used by the search engines to identify the Web page with respect to the search criterion.

➤ TITLE

- The TITLE element allows you to specify the title of the Web page under the <TITLE> and </TITLE> tags.
- The title is displayed on the Title bar of the Web browser. The TITLE element is included within the HEAD element.

© Aptech Ltd. Introduction to the HTML5 / Session 2 7

**HTML** **5 Basic Tags 3-6**

➤ META

- The meta tag is used for displaying information about the data.
- In HTML5, the content meta tag which was used for specifying the charset or character encoding has been simplified.
- The new <meta> tag is as follows:

```
<meta charset="utf-8" />
```

- UTF-8 is the most commonly used character coding that supports many alphabets.
- There are several other attributes associated with the meta tag that can be used to declare general information about the page.
- This information is not displayed in the browser.
- Meta tags provide search engines, browsers, and Web services with the information that is required to preview or acquire a summary of the relevant data of your document.

© Aptech Ltd. Introduction to the HTML5 / Session 2 8

**HTML** **5 Basic Tags 4-6**

➤ LINK

- The <link> tag is used to define the association between a document and an external resource.
- It is used to link stylesheets. Its type attribute is used to specify the type of link such as 'text/css' which points out to a stylesheet.

```
<link type="text/css" rel="stylesheet" href="first.css">
```

- The type attribute is not included in HTML5.
- The reason is that CSS has been declared as the default and standard style for HTML5. So, the new link is as follows:

```
<link rel="stylesheet" href="first.css">
```

© Aptech Ltd. Introduction to the HTML5 / Session 2 9

**HTML Basic Tags 5-6**

### ➤ SCRIPT

- With HTML5, JavaScript is now the standard and default scripting language.
- The type attribute tag can be removed from the script tags.
- The new script tag is as follows:

```
<script src="first.js"></script>
```

The following example shows the use of the script tag.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>HTML Webinar</title>
<link rel="stylesheet" href="first.css">
<script src="first.js"></script>
</head>
</html>
```

© Aptech Ltd. Introduction to the HTML5 / Session 2 10

**HTML Basic Tags 6-6**

### ➤ BODY

- The BODY element enables you to add content on the Web page specified under the <BODY> and </BODY> tags.
- Content can include text, hyperlinks, and images. You can display the content using various formatting options such as alignment, color, and background.
- Following figure shows the basic HTML elements.

```
<HTML>
<HEAD>
<TITLE>"Welcome Page"</TITLE>
</HEAD>
<BODY>
    This Web page will list the different
    Web technologies used to create Web sites.
</BODY>
</HTML>
```

© Aptech Ltd. Introduction to the HTML5 / Session 2 11

Using slides 6 to 11, explain the basic requirement of the html file. Explain the basic tag in the HTML.

An HTML document is made up of different elements or tags, and attributes which specify the content and its format. Therefore, HTML is both a structural and presentational markup language. Structural markup specifies the structure of the content, while the presentational markup specifies the format.

An HTML page is saved with the .html extension. The basic structure of an HTML document mainly consists of seven basic elements.

Explain **HTML** element is the root element that marks the beginning of an HTML document. It contains the start and end tag in the form of `<HTML>` and `</HTML>` respectively. It is the largest container element as it contains various other elements.

**Tips:**

The DOCTYPE declaration is done before the `<HTML>` tag in the HTML page.

Then, explain `<HEAD>` element and `<Title>` element. Mention that the `<head>` element includes other elements such as title, scripts, styles, and meta information.

Now, explain the `<TITLE>` element. It allows you to specify the title of the Web page under the `<TITLE>` and `</TITLE>` tags and is displayed on the title bar of the Web browser.

Explain `meta` tag is used for displaying information about the data. In HTML5, the content `meta` tag can be used for specifying the charset or character encoding used on the Web page.

Tell them that UTF-8 is also being promoted as the new standard for characters. UTF-8 encodes each Unicode character as a variable number of 1 to 4 octets, where the number of octets depends on the integer value assigned to the Unicode character. It is an efficient encoding of Unicode documents that use mostly US-ASCII characters because it represents each character in the range U+0000 through U+007F as a single octet.

There are several other attributes associated with the `meta` tag that can be used to declare general information about the page. This information is not displayed in the browser. Meta tags provide search engines, browsers, and Web services, the information that is required to preview or acquire a summary of the relevant data of your document.

**Tips:**

In HTML5, it is not very important to self-close tags with a slash at the end. Though self-enclosing is recommended for compatibility reasons.

Then, explain `<link>` tag that is used to link style sheets in the HTML page. The `type` attribute is used to specify the type of link such as '`text/css`' which points out to a style sheet. The `type` attribute is not included in HTML5, because CSS has been declared as the default and standard style for HTML5.

So, the new link can be written as:

```
<link rel="stylesheet" href="first.css">
```

Along with this, mention the `size` attribute is new in HTML5 for `<link>` tag. The `size` attribute specifies the sizes of icons for visual media and is used if `rel="icon"`.

Further, explain the `<script>` element to the students. Mention that JavaScript is now the standard and default scripting language. Hence, you can remove the `type` attribute from the `script` tag.

Thus, the new script tag is as follows:

```
<script src="first.js"></script>
```

**Tips:**

The `async` attribute can be used with the `<link>` tag. The `async` attribute is a boolean attribute and it specifies that the script will be executed asynchronously as soon as it is available.

If the `async` attribute is present, then the script will be executed asynchronously, as soon as it is available. If the `async` attribute is not present, then the script is fetched and executed immediately, before the user agent continues parsing the page.

Finally, explain the `<body>` element to the students. Tell them that this is the most important tag as only contents contained within the `<body>` tag are displayed on the Web page. In other words, it enables you to add content on the Web page. The content can include text, hyperlinks, and images. You can display the content using various formatting options such as alignment, color, and background. Use the figure provided on the slide to explain basic HTML elements.

**In-Class Question:**

After you finish explaining basic elements of HTML 5, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which tag contains the `<meta>` tag in an HTML page?

**Answer:**

`<head>`

## Slides 12 and 13

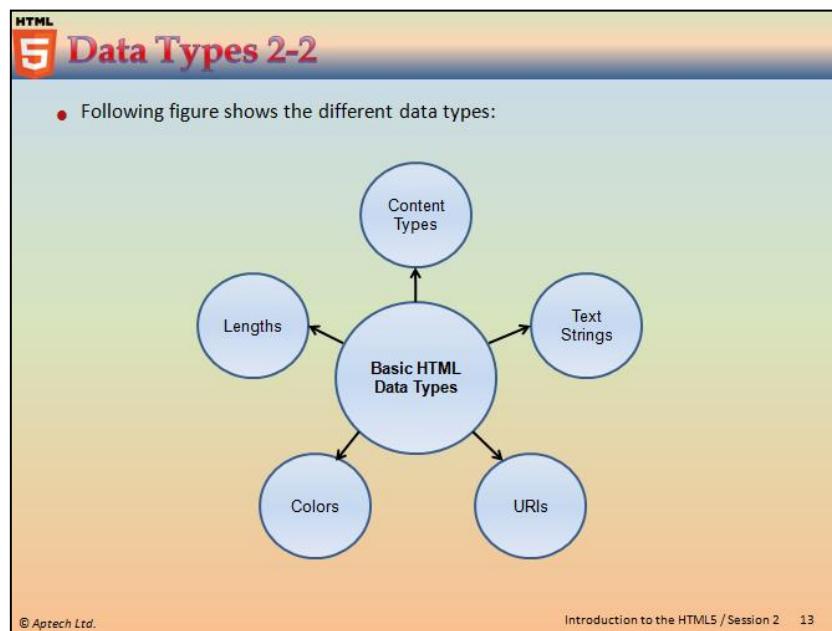
Let us understand the data types in HTML5.

**HTML 5 Data Types 1-2**

- A data type specifies the type of value assigned to the attributes and the type of content that is to be displayed on the Web page.
- Data types help in identifying the type of formatting such as color and length of data.
- Following table describes the different types of content.

Data Type	Description
Text Strings	Specifies textual content, which is readable by the user.
Uniform Resource Identifiers (URIs)	Specifies the location of Web pages or network files.
Colors	Specifies the color to be applied to the content on the Web page.
Lengths	Specifies the spacing among HTML elements. Length values can be in Pixels, Length, or MultiLength. Pixels refer to the smallest dot on the screen.
Content Types	Specifies the type of content to be displayed on a Web page. Content types include 'text/html' for displaying text, 'image/gif' for displaying image of a .gif format, 'video/mpg' for displaying a video file of .mpg format.

© Aptech Ltd. Introduction to the HTML5 / Session 2 12



Using slides 12 and 13, explain the different data types of HTML.

Data types specify the type of value that can be assigned to the attributes. The different types of value or content include texts, images, hyperlinks, video, and audio. Data types also identify the length of the data that can be stored in it.

Explain the different data types using figure displayed on slide 13. The important basic HTML data types are as follows:

- Text Strings:** Specifies textual content which is readable by the user.

- **Uniform Resource Identifiers (URIs)**: Specifies the location of Web pages or network files.
- **Colors**: Specifies the color to be applied to the content on the Web page.
- **Lengths**: Specifies the spacing among HTML elements. Length values can be in pixels, length, or multilength. Pixels refer to the smallest dot on the screen. Length is specified as a percentage value of pixels or available space on the screen. Multilength can be specified as pixel or percentage.
- **Content**: Specifies the type of content to be displayed on a Web page. Examples of content types include ‘text/html’ for displaying text using HTML format, ‘image/gif’ for displaying image of a .gif format, and ‘video/mpg’ for displaying a video file of .mpg format.

Explain the code snippet: <image src="C:/myFolder/image1.gif" alt="image1">

## Slide 14

Let us understand the attributes.

**HTML5 Attributes**

- HTML attributes help to provide some meaning and context to the elements.
- Following table describes some of the global attributes used in HTML5 elements.

Attribute	Description
<b>class</b>	Specifies class names for an element.
<b>contextmenu</b>	Specifies the context menu for an element.
<b>dir</b>	Specifies the direction of the text present for the content.
<b>draggable</b>	Specifies the draggable function of an element.
<b>dropzone</b>	Specifies whether the data when dragged is copied, moved, or linked, when dropped.
<b>style</b>	Specifies the inline CSS style for an element.
<b>title</b>	Specifies additional information about the element.

© Aptech Ltd. Introduction to the HTML5 / Session 2 14

Using slide 14, explain the some of the global attributes of HTML 5. The global attributes can be associated with any element or tag.

Mention HTML attributes that helps to provide some meaning and context to the elements.

Mention the `style` attribute is used for providing the inline style to the element. For example, to change the text color of a paragraph, the `style` attribute can be used, `<p style="color:red;">This is a paragraph.</p>`

While the `class` attribute is used to specify the id for the elements for which styles need to be applied.

## Slide 15

Let us understand the HTML entities.

The slide has a header 'HTML Entities' with a '5' icon. It contains the following text:

- Entities are special characters that are reserved in HTML.
- These entities can be displayed on a HTML5 Web site using the following syntax:

**Syntax:**

```
&entity_name; or &#entity_number;
```

- Following table shows some of the commonly used HTML entities.

Output	Description	Entity Name	Entity Number
	non-breaking space	&nbsp;	&#160;
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	ampersand	&amp;	&#38;
€	euro	&euro;	&#8364;
©	copyright	&copy;	&#169;

© Aptech Ltd. Introduction to the HTML5 / Session 2 15

Using slide 15, explain the HTML entities to the students. Mention entities are special characters that are reserved in HTML. These entities can be displayed on a HTML5 Web site using the following syntax:

`&entity_name;` or `&#entity_number;`

The advantage of using an entity name, instead of a number, is that the name is easier to remember. The disadvantage is that browsers may not support all entity names, instead they may support numbers.

### Tips:

1. Entities name are case sensitive.
2. Entities can also be used to present characters that are not displayed on a keyboard.

**In-Class Question:**

After you finish explaining entities in HTML 5, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What is the entity name and entity number for the copyright symbol?

**Answer:**

&copy; and &#169 are the entity name and the entity number respectively used for the copyright symbol.

**Slide 16**

Let us understand the container and standalone tags.

The slide has a blue header bar with the text 'HTML 5 Container and Standalone Tags'. The main content area is divided into four colored boxes: pink, green, purple, and blue. The pink box contains the text: 'There are two types of HTML elements namely, container and standalone elements.' The green box contains: 'A container element includes the start tag, contents, sub-elements, and end tag.' The purple box contains: 'All the basic HTML elements are container elements.' The blue box contains: 'A standalone element consists of the start tag and attributes followed by the end tag as /> without any content.' At the bottom left is the copyright notice '© Aptech Ltd.' and at the bottom right is the page number 'Introduction to the HTML5 / Session 2 16'.

Using slide 16, explain the container and standalone tags.

Mention there are two types of HTML elements namely, container and standalone elements. A container element includes the start tag, contents, sub-elements, and end tag. All the basic HTML elements are container elements.

A standalone element consists of the start tag and attributes followed by the end tag as without any content.

Provide examples such as `<link>`, `<meta>`, and `<br>` tag are standalone tags. Similarly, tags such as `<body>`, `<head>`, and so on are container tags.

## Slide 17

Let us understand the HTML5 and mobile devices.

**HTML5 and Mobile Devices**

- HTML5 helps to create better and richer mobile applications by using APIs that support advanced Web application features for mobile browsers.
- New age smartphones with Apple iOS and Google Android as operating systems support HTML5 compliant browsers.
- HTML5 tries to integrate all the features to deploy mobile applications that would be compatible in all the platforms.
- HTML5 provides features such as drag-and-drop functionality, video embedding in an application, and even offline capabilities.
- As HTML5 is compatible with most mobile operating systems, upto 30% of the cost for development for different operating systems is saved.
- Also, there is a reduced dependency in third-party components, thus reducing the licensing costs.
- All the required components will be readily available through the browser in HTML5.

© Aptech Ltd. Introduction to the HTML5 /Session 2 17

Using slide 17, explain the use of HTML 5 in mobile devices in detail.

Mention HTML5 has helped to create better and richer mobile applications. For this, APIs are used in HTML5. These APIs support advanced Web application features for mobile browsers.

HTML5 is not supported by older mobile devices. New age smartphones with Apple iOS and Google Android as operating systems support HTML5 compliant browsers. Even Microsoft Windows 7 for Mobile will have a newly developed browser to support HTML5 developed Web sites and applications.

However, due to the various mobile platforms available on mobile devices, development of mobile applications is difficult. HTML5 has tried to integrate all the features to deploy mobile applications that would be compatible in all the platforms. HTML5 provides features such as drag-and-drop functionality, video embedding in an application, and even offline capabilities.

As HTML5 is compatible with most mobile operating systems, up to 30% of the cost for development of different operating systems is saved. Also, there is a reduced dependency in third-party components, thus reducing the licensing costs. All the required components will be readily available through the browser in HTML5.

## Slide 18

Let us understand the benefits of HTML5 for mobile development.

**Benefits of HTML5 for Mobile Development**

The benefits of HTML5 for mobile developments are as follows:

- HTML5 has included APIs, hence additional plug-ins are not required for mobile browsers.
- Mobile development is easier as knowledge of only HTML5, CSS, and JavaScript is majorly required.
- There is a rising growth of HTML5 for mobile applications due to its enhanced compatibility.
- HTML5 is compatible with most operating system platforms.
- The HTML5 based mobile applications can run on browsers of Android, iOS, Blackberry, Windows Phone, and other mobile operating systems.
- The development cost for creating applications in HTML5 is low.
- Applications based on location and maps will have greater support in HTML5.
- Third-party programs are not required in HTML5.

© Aptech Ltd. Introduction to the HTML5 / Session 2 18

Using slide 18, explain the benefits of using HTML 5 for development of mobile device applications.

Mention HTML5 as a standard has a long way to go. The fragmented support for HTML5 in different browsers drives the user experience to an even lower common denominator.

HTML5 apps also have the ability for offline access and usage via the application cache, which means working without a network connection is now possible.

One of the biggest benefits to IT organizations developing mobile applications in HTML5 is the ability to deploy those apps and updates directly to the user community via the browser. No third party or extra step is needed for distribution.

## Slide 19

Let us summarize the session.

The slide has a blue header bar with the text "HTML5 Summary". The main content area has a light green background. It contains a bulleted list of nine points about HTML5 elements:

- An element organizes the content in a Web page hierarchically, which forms the basic HTML structure.
- The DOCTYPE tells the browser the type of your document.
- A data type specifies the type of value assigned to the attributes and the type of content that is to be displayed on the Web page.
- Entities are special characters that are reserved in HTML.
- A container element includes the start tag, contents, sub-elements, and the end tag.
- A standalone element consists of the start tag and attributes followed by the end tag as /> without any content.
- HTML5 provides features such as drag-and-drop functionality, video embedding in an application, and even offline capabilities for mobile devices.

At the bottom left is the copyright notice "© Aptech Ltd." and at the bottom right is the page number "Introduction to the HTML5 / Session 2 19".

In slide 19, you will summarize the session. You will end the session with a brief summary of what has been taught in the session.

### 2.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session. You should also explore the formatting text tags that are offered with the next session.

#### Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 3 – Formatting Text Using Tags

---

## 3.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 3.1.1 Objectives

By the end of this session, the learners will be able to:

- Explain the Heading tag
- Explain the different tags related to formatting
- Explain monospaced font, preformatted text, and block quotation
- Describe the different types of lists
- Explain the procedure to change the background color and image

### 3.1.2 Teaching Skills

To teach this session, you should be well-versed with formatting tags such as Heading tags and other related tags. Also the tags such as monospaced font, preformatted text, and block quotation should be known. Aware yourself with different types of lists that can be displayed on a Web page. Also how to apply background color and image to the Web page.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

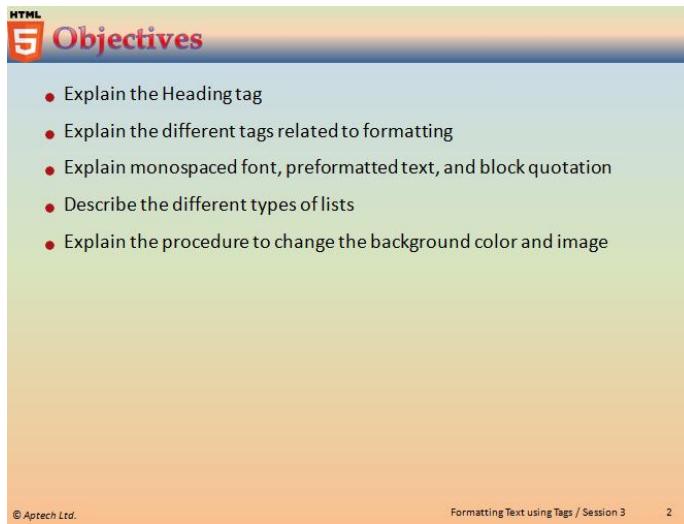
It is recommended that you test the understanding of the students by asking questions in between the class.

#### In-Class Activities:

Follow the order given here during In-Class activities.

**Overview of the Session:**

Then give the students the overview of the current session in the form of session objectives.  
Show the students slide 2 of the presentation.



Tell the students that this session will introduce them to the various formatting tags in HTML. They will learn about the different levels of heading tags and other formatting tags that can be applied to the content displayed on the Web page. They will also know about the different types of lists and procedure to change the background color and image.

## 3.2 In-Class Explanations

### Slide 3

Let us understand the introduction to formatting text using tags.



Using slide 3, explain the different types of formatting that can be applied to the text on a Web page.

Mention text content of the Web page form an important part of a Web site. This text content must not only be informative, but also attractive. It must be easy to read and must have short and crisp sentences for easy understanding of the Web user. To attract the attention of the user, headings must be appropriately provided. Also, text formatting options such as bold, italics, subscript, superscript, and so on must be applied on the text. Bullets can be also used to list the text in a systematic manner. The background color and background image of a Web page can be specified using HTML.

## Slides 4 and 5

Let us understand the different headings in HTML.

Heading elements define headings for contents such as text and images.

Specifies the hierarchical structure of a Web page by grouping the contents.

HTML defines six levels of headings ranging from H1 to H6.

- H1 is the top level heading and is displayed with largest font size
- H6 is the lowest-level heading and is displayed with smallest font size

© Aptech Ltd.      Formatting Text using Tags / Session 3      4

The Code Snippet demonstrates how to specify the six levels of heading in an HTML page.

```
<!DOCTYPE html>
<html>
<head>
    <title>Headings</title>
</head>
<body>
    <h1>H1 Heading</h1>
    <h2>H2 Heading</h2>
    <h3>H3 Heading</h3>
    <h4>H4 Heading</h4>
    <h5>H5 Heading</h5>
    <h6>H6 Heading</h6>
</body>
</html>
```

H1 Heading  
H2 Heading  
H3 Heading  
H4 Heading  
H5 Heading  
H6 Heading

© Aptech Ltd.      Formatting Text using Tags / Session 3      5

Using slides 4 and 5, explain the heading tags.

The heading elements define heading for content such as text and images. They specify a hierarchical structure of a Web page by grouping the contents into different headings.

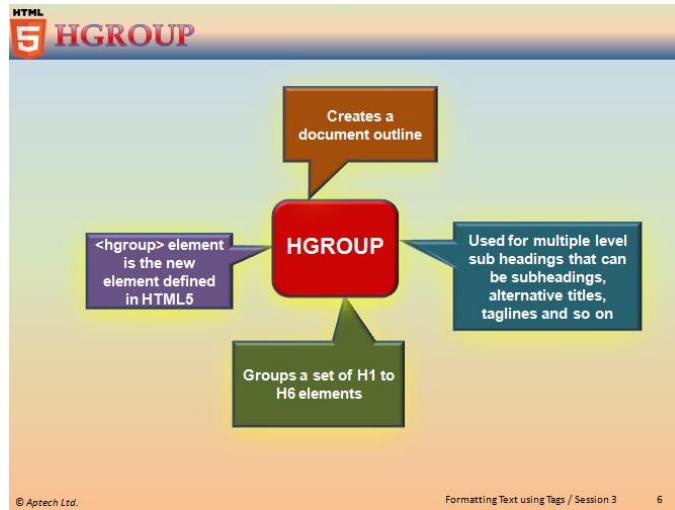
Mention that heading under the H1 tags will be displayed with the largest size. Each subsequent heading will be displayed in a size lower than its previous heading. The heading under the H6 tags will be displayed with the lowest size.

### Tips:

Web browsers automatically add a single margin or space before and after the heading text.

## Slide 6

Let us understand the `hgroup` element.



Using slide 6, explain the `hgroup` tag.

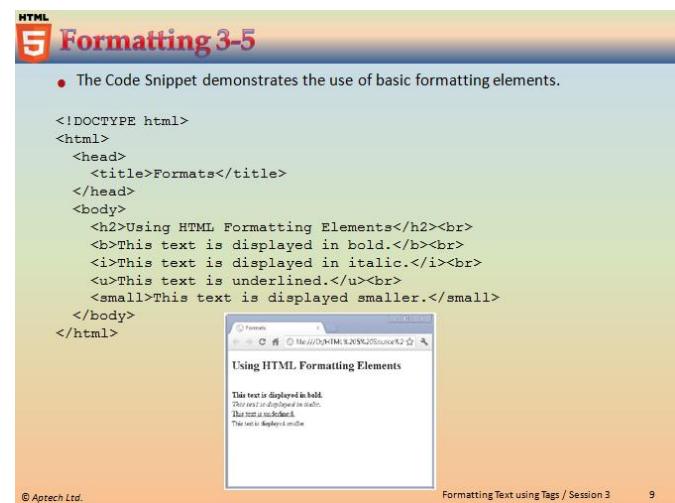
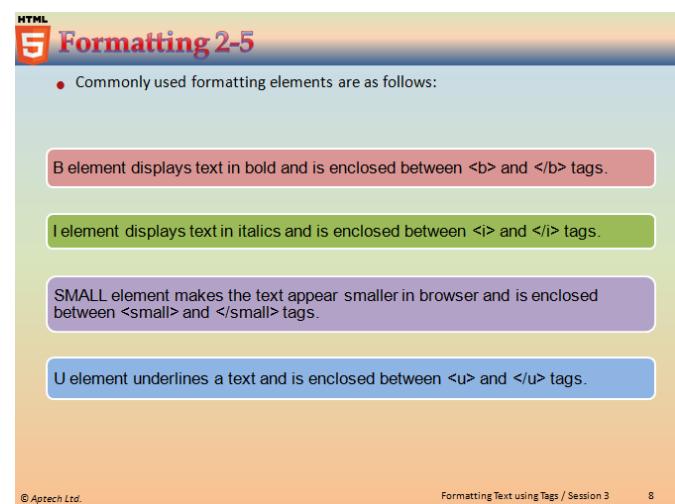
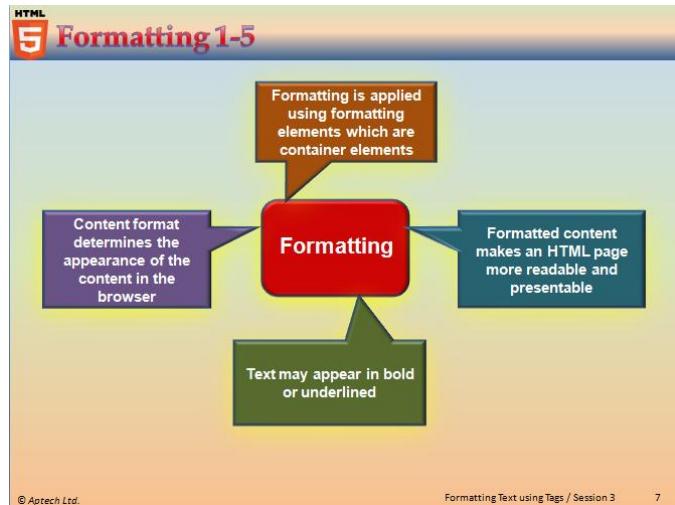
Mention `<hgroup>` element is a new element defined in HTML5. It is used to group titles and their subtitles. The element is used to group a set of H1–H6 elements. These are used for headings that have multiple levels that can include subheadings, alternative titles, taglines, and so on. The main advantage of using the `<hgroup>` tag is to create a document outline.

For example, you might have a level 1 heading, followed by a subheading in a level 2 heading. In this instance, the level 2 heading is different to the other level 2 headings in the document, because it is an extension of the level 1 heading (i.e. it is a subheading of the heading). Therefore, to group the two together, you can use the `<hgroup>` tag. The following example demonstrates this scenario:

```
<article>
  <hgroup>
    <h1>Heading level 1</h1>
    <h2>Heading level 2</h2>
  </hgroup>
  <p>This is example for hgroup element</p>
</article>
```

## Slides 7 to 11

Let us study the formatting in HTML.



**HTML 5 Formatting 4-5**

- Some more formatting elements are as follows:

DEL element encloses deleted text and is placed between `<del>` and `</del>` tags.

INS element encloses inserted text and is placed between `<ins>` and `</ins>` tags.

STRONG element emphasizes the text and is placed between `<strong>` and `</strong>` tags.

SUB element displays a text as subscript and is enclosed between `<sub>` and `</sub>` tags.

SUP element displays a text as superscript and is enclosed between `<sup>` and `</sup>` tags.

© Aptech Ltd.

Formatting Text using Tags / Session 3

10

**HTML 5 Formatting 5-5**

- The Code Snippet demonstrates the use of other formatting elements.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Updating and Shifting Text</title>
  </head>
  <body>
    <h3>Updating, Emphasizing, and Shifting Text</h3>
    This is an example of <del>deleted</del> <ins>inserted</ins> text.<br/>
    The is an example of <strong>Strong</strong> text.<br/>
    The is an example of <sub>subscript</sub>text.<br/>
    The is an example of <sup>superscript</sup> text.<br/>
  </body>
</html>
```

© Aptech Ltd.

Formatting Text using Tags / Session 3

11

Using slides 7 to 11, explain the different formatting tags.

The content format determines how the content will appear in the browser. For example, when you visit a Web site, some text appears in a specific format such as bold or underlined. This means that the formatted content makes an HTML page look readable and presentable. In HTML, formatting is applied to the text by using formatting elements, which are container elements.

The commonly used HTML formatting elements are as follows:

- B: The B element displays the text in bold. The text that needs to be displayed in bold is enclosed between `<b>` and `</b>` tags.
- I: The I element displays the text in italic. The text that needs to be displayed in italic is enclosed between `<i>` and `</i>` tags.
- SMALL: The SMALL element makes the text appear smaller in a browser. The text that needs to be displayed smaller is enclosed between `<small>` and `</small>` tags.
- U: The U element applies an underline to the text. The text that needs to be underlined is enclosed between `<u>` and `</u>` tags.

**Tips:**

Often `<strong>` renders as `<b>`, and `<em>` renders as `<i>`.

Mention HTML provides some more formatting elements that can be applied to the text. These formatting elements are as follows:

- **DEL:** The `DEL` element encloses text, which has been deleted from the document. The text to be deleted is placed in the `<del>` and `</del>` tags.
- **INS:** The `INS` element encloses text, which has been inserted in the document. The text to be inserted is placed in the `<ins>` and `</ins>` tags. The `INS` element can be used with `DEL` element to inform the user about the deleted text, which is replaced by the new text.
- **STRONG:** The `STRONG` element emphasizes the text as compared to its surrounding text. The text to be emphasized is placed in the `<strong>` and `</strong>` tags.
- **SUB:** The `SUB` element displays the text as subscript. The text to be displayed as subscript is enclosed in `<sub>` and `</sub>` tags.
- **SUP:** The `SUP` element displays the text as superscript. The text to be displayed as superscript is enclosed in `<sup>` and `</sup>` tags.

Using slide 11, explain the code snippet using the other formatting tags. Also explain the corresponding output shown in the figure on the slide.

**In-Class Question:**

After you finish explaining formatting, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which tag is used to format a text as subscript?

**Answer:**

`<SUB>` tag is used to format a text as subscript.

## Slides 12 and 13

Let us understand monospaced and preformatted text.

The slide has a blue header bar with the text 'HTML 5 Monospaced and Preformatted Text 1-2'. Below the header, there are four colored callout boxes containing text:

- Red box: Monospaced font allows the same amount of horizontal space between fonts irrespective of font size, shape, and type.
- Green box: Monospaced fonts are used for programming code snippets, instruction texts, and ASCII characters.
- Purple box: <pre> tag is used for preformatted text content.
- Blue box: <pre> tag applies a fixed-font width to the text content.
- Orange box: <pre> tag allows you to copy-paste the content along with the formatting from the source.

At the bottom left is the copyright notice '© Aptech Ltd.' and at the bottom right is the page number 'Formatting Text using Tags / Session 3 12'.

The slide has a blue header bar with the text 'HTML 5 Monospaced and Preformatted Text 2-2'. Below the header, there is a bullet point followed by a table:

- Following table lists some of the predefined tags and their description.

Tag	Description
<em>	Used for emphasized text
<dfn>	Used for definition term
<code>	Used for computer code
<samp>	Used for sample output from a computer program
<cite>	Used for citation

At the bottom left is the copyright notice '© Aptech Ltd.' and at the bottom right is the page number 'Formatting Text using Tags / Session 3 13'.

Using slides 12 and 13, explain the monospaced and preformatted text.

Explain by using monospaced font in HTML5, a Web developer can provide the same amount of horizontal space between the fonts, even if the font size, shape, and type are not the same. Monospaced fonts are used for programming code scripts, instruction texts, and ASCII characters.

The <pre> tag is used to apply preformatted text content to a Web page and has a fixed-font width. It also maintains a standard formatting for spaces and line breaks. The <pre> tag is usually used when you want to copy paste content from a source, but do not want to change the formatting of the same. The content would be copied while maintaining all the line breaks and spaces.

Also explain some of the other predefined tags available for formatting of text in HTML provided on slide 13.

**Tips:**

Use the `<pre>` element when displaying text with unusual formatting or some sort of computer code.

**Slide 14**

Let us understand `<blockquote>` tag.

The screenshot shows a presentation slide with a blue header bar containing the text 'HTML' and a large orange number '5'. Below the header, the title 'Formatting a Block Quotation' is displayed in a large, bold, black font. The main content area contains three bullet points:

- To define a long quotation or block quotation, `<blockquote>` tags are used.
- `<blockquote>` tag indents the quotation in browsers.
- The Code Snippet demonstrates the use of `<blockquote>` tags.

Below the bullet points is a code snippet:

```
<blockquote>
  "When one door closes, another opens; but we often look so long
  and so regretfully upon the closed door that we do not see the
  one which has opened for us." -Alexander Graham Bell
</blockquote>
```

At the bottom of the slide, there is footer text: '© Aptech Ltd.', 'Formatting Text using Tags / Session 3', and '14'.

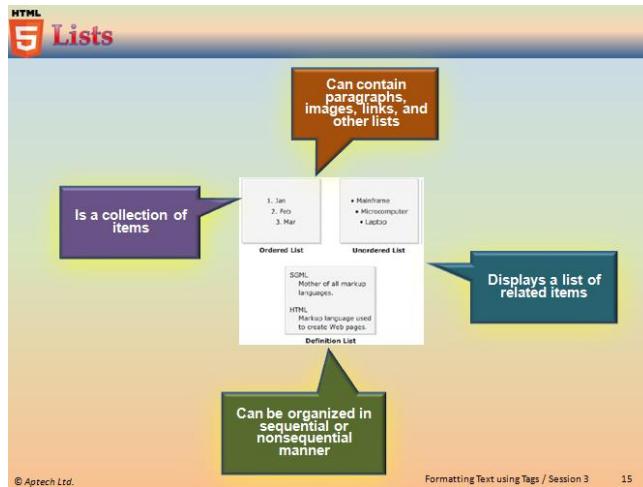
Using slide 14, explain formatting a block quotation.

Mention to define a long quotation or block quotation, the `<blockquote>` tag can be used. When the `<blockquote>` tag is used, the quotation is indented in browsers.

Also, explain the example mentioned on the slide.

## Slide 15

Let us understand the lists in HTML.



Using slide 15, explain the different lists.

A list is a collection of items which might be organized in a sequential or non-sequential manner. You can use a list to display related items that belong to a particular category. For example, to display the types of computers, such as mainframe, microcomputer, and laptops, you would organize these items one below the other under the **Types of Computers** category. Similarly, HTML allows you to display related items in a list on a Web page.

A list in HTML can contain paragraphs, line breaks, images, links, and other lists. The items within a list are displayed on a Web page one below the other using bullets. HTML supports three types of lists. They are as follows:

- Ordered
- Unordered
- Definition

Explain the lists with the figure given on the slide.

### Tips:

Inside a list item, you can put text, line breaks, images, links, other lists, and so on.

## Slides 16 and 17

Let us understand ordered lists.

**HTML Ordered Lists 1-2**

- List is displayed using a numbered or alphabetic bullet
- Two elements used for creating an ordered list are as follows:
  - OL – Creates an ordered list
  - LI – Specifies an item and it is a sub-element of the OL element
- The Code Snippet demonstrates the use of OL and LI tag.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Days in a Week</title>
  </head>
  <body>
    <h2>Days in a Week:</h2>
    <ol>
      <li>Sunday</li>
      <li>Monday</li>
      <li>Tuesday</li>
      <li>Wednesday</li>
      <li>Thursday</li>
      <li>Friday</li>
      <li>Saturday</li>
    </ol>
  </body>
</html>
```

Days in a Week:

1. Sunday
2. Monday
3. Tuesday
4. Wednesday
5. Thursday
6. Friday
7. Saturday

Formatting Text using Tags / Session 3      16

© Aptech Ltd.

**HTML Ordered Lists 2-2**

- Following table lists some of the different numbering styles and their description.

Property's Value	Example
decimal	1, 2, 3...
lower-alpha	a, b, c...
upper-alpha	A, B, C...
lower-roman	i, ii, iii...
upper-roman	I, II, III...

- list-style-type property is used to specify a numbering style for the ordered list.
- It is the property of the style attribute, which is specified with the <ol> tags.

Formatting Text using Tags / Session 3      17

© Aptech Ltd.

Using slides 16 and 17, explain the ordered lists.

Mention an ordered list is a list of items arranged in a particular order. Here, the order of the items is important as it indicates a sequential flow. For example, to display the days in a week or months in a year, you would use numbered bullets.

Similarly, HTML allows you to implement ordered lists where each item in the list is displayed using a numbered or alphabetic bullet.

HTML provides two elements for creating an ordered list. These are as follows:

- OL: Creates an ordered list.
- LI: Specifies an item and it is a sub-element of the OL element.

Also, explain the example and the output in figure for ordered list provided on the slide.

Mention different numbering styles, such as roman numerals or alphabetic bullets can be applied to an ordered list. Explain the different numbering styles that can be specified in an ordered list.

The `list-style-type` property is used to specify a numbering style for the ordered list. It is the property of the `style` attribute, which is specified within the `<ol>` tag.

The `list-style-type` property of the `style` attribute in the code is set to `lower-roman`. The property and its value are separated by a colon. This means that the days of the week will be displayed sequentially by applying the lower-case roman numbers as bullets.

The following example shows how to use the `list-style-type` property to display the list.

```
<ol style='list-style-type: upper-roman'>
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

Some of the property values for `list-style-type` property are as follows:

- disc – displays filled circle
- armenian – displays armenian numbering
- circle – displays circle
- cjk-ideographic – displays plain ideographic number
- decimal – displays number
- decimal-leading-zero – displays leading zeros such as 01, 02, and so on
- lower-alpha – lower-alpha such as a, b, c, and so on
- lower-roman – lower-roman such as i, ii, iii, iv, and so on
- none – no marker
- square – displays square
- upper-alpha – displays a, b, c, and so on
- upper-latin – displays a, b, a, and so on
- upper-roman – displays i, ii, iii, iv, and so on

## Slides 18 to 20

Let us understand unordered lists.

**HTML 5 Unordered Lists 1-3**

- Items are arranged in random order
- Two elements used for creating an unordered list are as follows:
  - UL – Creates an unordered list
  - LI – Specifies an item and it is a sub-element of the OL element
- The Code Snippet demonstrates the use of UL and LI tag.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Features of EasyPad</title>
  </head>
  <body>
    <h2>Features of EasyPad</h2>
    <ul>
      <li>Opens many files at a time</li>
      <li>Unlimited undo and redo</li>
      <li>Reads and writes both
        Windows and Unix files</li>
    </ul>
  </body>
</html>
```

Formatting Text using Tags / Session 3 18

**HTML 5 Unordered Lists 2-3**

- The `list-style-type` property specifies the type of bullet to be applied to an unordered list.
- There are three types of bullets defined for the unordered lists:
  - Disc
  - Square
  - Circle
- The default value is disc, which is applied to the unordered list, even if the `list-style-type` property is not specified.
- The Code Snippet demonstrates how to apply the square bullet to an unordered list.

```
<!DOCTYPE html>
<html>
<head>
<title>Wild Animals</title>
</head>
<body>
<h2>Wild Animals</h2>
<ul style="list-style-type:square">
<li>Lion</li>
<li>Tiger</li>
<li>Leopard</li>
<li>Wolf</li>
</ul>
</body>
</html>
```

Formatting Text using Tags / Session 3 19

**HTML 5 Unordered Lists 3-3**

- The `list-style-type` property of the style attribute is set to square.
- Hence, the unordered list of wild animals will be displayed using the square bullet as shown in the figure.

Formatting Text using Tags / Session 3 20

Using slides 18 to 20, explain the unordered list.

An unordered list is a list where the items are arranged in a random order. This means that you will create an unordered list when the order of related items is not important.

For example, to list the features of a product, you would create an unordered list. Each item in an unordered list is displayed using symbolic bullets such as circles and squares. These bullets are similar to the bullets provided by Microsoft Office Word. HTML provides the `UL` element to create an unordered list.

Explain the code snippet and the corresponding figure provided on the slide.

The `UL` element contains the `<ul>` tag and multiple `<li>` sub-elements. The `<ul>` tag marks the beginning of the unordered list. Each of the sub-elements starts with the `<li>` tag displayed with the default symbolic bullet, which is a small black disc.

The `list-style-type` property specifies the type of bullet to be applied to an unordered list. There are three types of bullets defined for the unordered lists in HTML. These bullet types are namely, disc, square, and circle. The default value is disc which is applied to the unordered list, even if the `list-style-type` property is not specified.

Using slide 19, explain the bullets used in unordered list. Also, explain that the `list-style-type` property of the `style` attribute is set to square in the figure provided on slide 20.

### In-Class Question:

After you finish explaining unordered list, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which property of the `style` tag is used for specifying the bullet type in the unordered list?

### Answer:

The `list-style-type` property

## Slides 21 to 23

Let us understand the definition list.

**HTML 5 Definition List 1-3**

- Refers to a collection of terms with their corresponding descriptions
- Contains the terms along with their descriptions
- Appears with the term indented on the left followed by description on the right or on next line
- Elements required to create a definition list are as follows:

DL – Is a container element that consists of DT and DD sub elements. Specifies that the definition list will be created using these elements.

DT – Specifies the term to be defined or described.

DD – Specifies the definition or description of the term.

© Aptech Ltd.      Formatting Text using Tags / Session 3      21

**HTML 5 Definition List 2-3**

- Steps to create a definition list are as follows:

- Specify the DL element to indicate that you want to create a definition list.
- Use the DT element to specify the term such as Common Noun.
- Use the DD element to specify the description of the term.

© Aptech Ltd.      Formatting Text using Tags / Session 3      22

The screenshot shows a web page titled "Definition List 3-3". It contains an HTML code snippet demonstrating how to create a definition list. The code includes a title, a section header "Types of Nouns", and a definition list with two entries: "Common Noun" and "Proper Noun", each with its respective description.

```
<!DOCTYPE html>
<html>
<head>
    <title>Types of Nouns</title>
</head>
<body>
    <h2>Types of Nouns</h2>
    <dl>
        <dt><b>Common Noun:</b></dt>
        <dd>It is a name of an object in general, such as pencil, pen, paper, and so on.</dd>
        <dt><b>Proper Noun:</b></dt>
        <dd>It is the unique name of a person or a place.</dd>
    </dl>
</body>
</html>
```

© Aptech Ltd.      Formatting Text using Tags / Session 3      23

23 (a)

The screenshot shows a web browser window displaying the rendered output of the HTML code. The title "Types of Nouns" is visible, followed by a definition list with two entries: "Common Noun" and "Proper Noun", each with its corresponding description. A callout arrow points from the text "Definition List" to the start of the definition list in the code.

© Aptech Ltd.      Formatting Text using Tags / Session 3      23

23 (b)

Using slides 21 to 23, explain the definition list.

A definition list refers to a collection of terms with their corresponding descriptions. For example, you can display a glossary on a Web page by creating a definition list, which will contain the terms and their descriptions.

A definition list appears with the term indented on the left followed by the description on the right or on the next line. By default, the description text appears on the next line and is aligned with respect to the term.

You can specify a single line or multiple lines of description for each term. HTML provides three elements to create a definition list. These elements are as follows:

- **DL:** Is a container element that consists of the **DT** and **DD** sub-elements. It specifies that a definition list will be created using these elements.
- **DT:** Specifies the term to be defined or described.

- **DD:** Specifies the definition or description of the term.

Consider a scenario, where you want to create a Web page that will display the types of nouns with their descriptions. To do this, you must create a definition list.

The steps to create a definition list are as follows:

- Specify the **DL** element to indicate that you want to create a definition list.
- Use the **DT** element to specify the term such as Common Noun.
- Use the **DD** element to specify the description of the term.

Using slides 23 (a) and 23 (b) explain the data definition list example and the output for the same.

## Slides 24 and 25

Let us understand background and foreground colors.

**Background and Foreground Colors 1-2**

Background properties specify the background color and image for the Web pages.

Background property is a shorthand property that specifies all the background properties in just one declaration.

**bcolor** attribute specifies the background color of a document.

- Syntax for **bcolor** is:

```
<body bcolor="color_name|hex_number|rgb_number">
```

where,

- color\_name - Specifies the background color with a color name (such as "red")
- hex\_number - Specifies the background color with a hex code (such as "#ff0000")
- rgb\_number - Specifies the background color with an rgb code (such as "rgb(255,0,0)")

© Aptech Ltd.      Formatting Text using Tags / Session 3      24

**Background and Foreground Colors 2-2**

Another way to specify a background color for a Web page is by using the **style="background-color: color"** attribute.

This attribute must be added to the **style** attribute of the **<body>** tag.

The foreground color can be specified by using the **style="color: color"** attribute.

- Example demonstrating the specification of background and foreground color is:

```
<body style="background-color: navy; color: yellow">
```

© Aptech Ltd.      Formatting Text using Tags / Session 3      25

Using slides 24 and 25, explain the process to specify background color and image for Web page.

HTML provides background properties that specify the background color and image for the Web pages. To specify a background for a Web page, use the background property. The background property is a shorthand property that specifies all the background properties in just one declaration. The `bgcolor` attribute specifies the background color of a document.

Explain the syntax for `bgcolor` to apply background color to the page.

Then, explain the other way to specify the background color. Another way to specify a background color for a Web page is by using the `style="background-color: color"` attribute. This attribute must be added to the `<body>` tag.

An example for applying a background color using the style attribute is as follows:

```
<body style="background-color: yellow">
```

The color name 'yellow' can also be replaced by the hex code or the rgb code.

The default text color of the page is specified as the foreground color. The foreground color can be specified by using the `style="color: color"` attribute. An example for applying a background and foreground color using the style attribute is as follows:

```
<body style="background-color: navy; color: yellow">
```

**In-Class Question:**

After you finish explaining background and foreground color, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



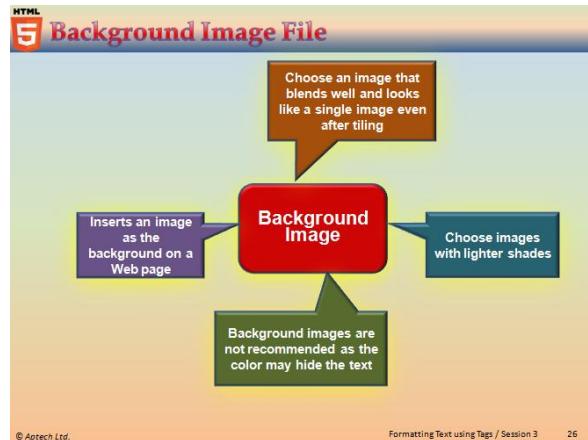
Which property of style tag is used to specify the background color?

**Answer:**

The `background-color` property.

## Slide 26

Let us understand background image file.



Using slide 26 explain the image file used as background.

A Web site developer can also insert an image as the background on a Web page. These background images are not recommended as sometimes, the colors in the image may hide the text content. Hence, it is best to choose images with lighter shades. Also, as the image is tiled, it is best to choose an image that blends well and looks like a single image even after it is tiled.

The code snippet demonstrates use of image in the background.

```
<html>
<body background="bgimage1.jpg">
    <h1>Hello world!</h1>
</body>
</html>
```

### Tips:

The `<body>` background attribute is not supported in HTML5. Use CSS instead.

CSS syntax: `<body style="background-image:url(bgimage.jpg)">`

## Slide 27

Let us summarize the session.

The slide is titled "HTML5 Summary". It contains a bulleted list of HTML tags and their purposes:

- The heading elements define headings for contents such as text and images.
- The `<hgroup>` element is used to group titles and their subtitles.
- Monospaced fonts are used for programming code scripts, instruction texts, and ASCII characters.
- The `<pre>` tag is used to apply preformatted text content to a Web page.
- To define a long quotation or block quotation, the `<blockquote>` tag can be used.
- A list is a collection of items, which might be organized in a sequential or nonsequential manner. HTML supports three types of lists namely, ordered, unordered, and definition.
- HTML provides background properties that specify the background color and image for the Web pages.

© Aptech Ltd.      Formatting Text using Tags / Session 3      27

Use slide 27 to summarize the session. End the session, with a brief summary of what has been taught in the session.

### 3.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session. You should also explore the creating hyperlinks and anchors that are offered with the next session.

#### Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 4 – Creating Hyperlinks and Anchors

---

## 4.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 4.1.1 Objectives

By the end of this session, the learners will be able to:

- Describe hyperlinks
- Explain absolute and relative paths
- Explain how to hyperlink to a Web page and e-mail address
- Explain how to hyperlink to anchors and other content

### 4.1.2 Teaching Skills

To teach this session, you should be well-versed with concept linking Web pages using hyperlinks. Also the concepts of providing the absolute and relative path in the hyperlinks should be known. You should also aware yourself on how to hyperlink to an e-mail address, anchors, and other content.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

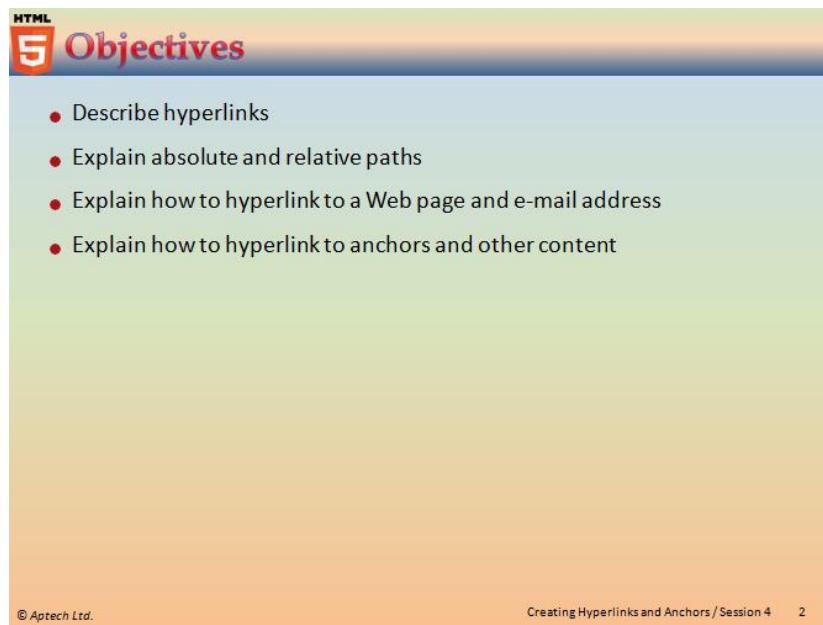
#### Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

**Overview of the Session:**

Give the students the overview of the current session in the form of session objectives.

Show the students slide 2 of the presentation.



The slide has a blue header bar with the text "HTML5 Objectives". Below the header is a list of four bullet points:

- Describe hyperlinks
- Explain absolute and relative paths
- Explain how to hyperlink to a Web page and e-mail address
- Explain how to hyperlink to anchors and other content

At the bottom left is the copyright notice "© Aptech Ltd." and at the bottom right is the page number "Creating Hyperlinks and Anchors / Session 4 2".

Tell the students that this session introduces hyperlinks and anchors. They will learn about hyperlinks, relative path, and absolute path to link the Web pages. They will also understand how to hyperlink an e-mail address and also how to hyperlink to anchor and other content.

## 4.2 In-Class Explanations

### Slides 3 to 5

Explain the concept of hyperlinks.

**HTML 5 Hyperlinks 1-3**

- A hyperlink is referred to as a link, linking to another Web page or to a section in the same Web page.
- The A (anchor) element is used to create a hyperlink.
- One can specify a text or an image as a hyperlink.
- When mouse is moved over such content, the cursor changes into a hand with its index finger pointing towards the content.
- This means that clicking the link will take the user to the respective link.
- To specify the linked page section or linked Web page, attributes of the A element have to be used.
- Following table lists the attributes of the A element.

Attribute	Description
<code>href</code>	Specifies the URL of the Web page to be linked or the value of the name attribute.
<code>hreflang</code>	Indicates the language of the destination URL.
<code>name</code>	Specifies the section of the Web page, which is to be linked.

© Aptech Ltd. Creating Hyperlinks and Anchors / Session 4 3

**HTML 5 Hyperlinks 2-3**

- The `<a>` tag is used to provide a hyperlink.
- This contains the `href` attribute that would contain the link to a URL or path of a Web page.
- An example of a href attribute code is as follows:

```
<a href=" http://www.aptech-worldwide.com/">
```

- The description and reference text that will serve as a hyperlink must be provided before closing the `<a>` tag by using `</a>`.
- An example of a hyperlink along with its output is as follows:

```
<html>
  <head>
  </head>
  <body>
    <a href="http://www.aptech-worldwide.com/">
      Click to view the Aptech Web site</a>
    </body>
  </html>
```

© Aptech Ltd. Creating Hyperlinks and Anchors / Session 4 4



Using slides 3 to 5, explain the hyperlinks. Mention hyperlink is referred to as a link. It refers for linking to another Web page or to a section in the same Web page.

Explain the students on how to use the hyperlink to navigate the Web pages. Anything, a text or an image can be provided as a hyperlink. When you move the mouse over a hyperlink content, the cursor changes into a hand with its index finger pointing towards the content. This means that clicking it, will take you to the respective link.

Explain the basic syntax to provide a hyperlink `<a>` tag that is used to provide a hyperlink. The `href` attribute is used to provide a URL or path of a Web page. An example of a `href` attribute code, `<a href=" http://www.aptech-worldwide.com/">`.

The description and reference text that will serve as a hyperlink must be provided between the opening and closing `<a> Hyperlink Text </a>` tag.

Then, show the example of hyperlink where **click on the link** will be redirected to the `http://www.aptech-woldwide.com`.

#### **Tips:**

In HTML5, the `href` attribute is optional. In case, if the `<a>` tag is not specified, then it acts as a placeholder for the hyperlink.

## Slide 6

Explain the Target attribute of the <a> tag.

**Target Attribute**

- The target attribute of the A element specifies the location where the linked Web page will open when a link is clicked.
- One can assign values to the target attribute.
- Following table lists some of the values of the target attribute.

Value	Description
_blank	Loads the target URL in a new blank window.
_self	Loads the target URL in the same window as that of the current Web page.
_top	Loads the target URL in the complete area of window.

© Aptech Ltd. Creating Hyperlinks and Anchors / Session 4 6

Using slide 6, explain the target attribute in detail. The target attribute of the <a> tag or element specifies the location where the linked Web page will open when a link is clicked.

Then, explain the values that can be assigned to the target attribute as shown in the table listed on the slide.

### Tips:

If you do not specify any target, then the linked page is displayed in the current browser window.

## Slides 7 and 8

Explain the concept of absolute and relative paths.

**Absolute and Relative Paths 1-2**

- Absolute paths are links that contain the complete address to get to a Web page.
- Absolute paths are the best way to link to a Web site.
- The syntax of an absolute path is as follows:

**Syntax:**

```
<a href=" http://www.aptech-worldwide.com/pages/about-
us/aboutus_aboutaptechworldwide.html ">Aptech Web
site</a>
```

- Relative paths are links that are provided when the files of a Web page are in the same folder as the page displaying the link.
- The syntax of a relative path is as follows:

**Syntax:**

```
<a href="aboutus_aboutaptechworldwide.html">
Aptech Web site</a>
```

© Aptech Ltd. Creating Hyperlinks and Anchors / Session 4 7

**Absolute and Relative Paths 2-2**

- To link to the files present in the subfolder, you need to provide the path to the subfolder.
- For example, if the file `aboutus_aboutaptechworldwide.html` is in a subfolder named `about-us` then the syntax is as follows:

**Syntax:**

```
<a href="about-us/aboutus_aboutaptechworldwide.html">
Aptech Web site</a>
```

- Files that are present in folders that are one level up can also be linked using a relative path. The syntax to link to a file one level up is as follows:

**Syntax:**

```
<a href="../aboutus_aboutaptechworldwide.html">Aptech
Web site </a>
```

© Aptech Ltd. Creating Hyperlinks and Anchors / Session 4 8

Using slides 7 and 8, explain the absolute and relative path to the students.

Absolute paths are links that contain the complete address to get to a Web page. Absolute paths are the best way to link to a Web site. Explain the syntax provided on the slide to explain absolute path which includes the domain name of the Web site. For example, `<a href=" http://www.aptech-worldwide.com/pages/about-us/
aboutus_aboutaptechworldwide.html ">Aptech Web site</a>`.

Relative paths are links that are provided when the files of a Web page are in the same folder as the page displaying the link. Explain the syntax of a relative path.

For example, `<a href="aboutus_aboutaptechworldwide.html">Aptech Web site</a>`.

To link to the files present in the subfolder, you need to provide the path to the subfolder.

For example, if the file `aboutus_aboutaptechworldwide.html` is in a subfolder named, **about-us** then the syntax is as follows: `<a href="about-us/aboutus_aboutaptechworldwide.html">Aptech Web site</a>`

Similarly, files that are present in folders that are one level up can also be linked using a relative path. The syntax to link to a file one level up is as follows: `<a href="../aboutus_aboutaptechworldwide.html">Aptech Web site</a>`

### In-Class Question:

After you finish explaining absolute and relative path, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What is the correct HTML for creating a hyperlink?

- `<a url="http://www.aptechworldwide.com">Aptech World Wide</a>`
- `<a>http://www.aptechworldwide.com</a>`
- `<a name="http://www.aptechworldwide.com">Aptech World Wide</a>`
- `<a href="http://www.aptechworldwide.com">Aptech World Wide</a>`

### Answer:

`<a href="http://www.aptechworldwide.com">Aptech World Wide</a>`

## Slide 9

Understand linking an e-mail address.

The slide has a blue header bar with the title 'Hyperlink to an E-mail Address' in red. Below the title is a bulleted list of points. At the bottom of the slide, there are two code snippets. The footer contains the copyright information '© Aptech Ltd.' and the page number '9'.

- Hyperlinks can be even applied to e-mail addresses in the same way as they can be given for Web pages.
- There are various tasks that can be performed when a hyperlink is given to an e-mail, such as starting the default e-mail client, creating a new message, adding the subject line, and so on.
- To add an e-mail to a hyperlink, the href attribute must be used and followed by mailto:email address.
- Following code snippet shows the way to hyperlink an e-mail address.  
`<a href="mailto:customercare@aptech.ac.in">Customer Care</a>`
- To automatically add a subject line in the new e-mail message, the ?subject= attribute must be inserted after the e-mail address.
- Following code snippet shows the way to add a subject line to a hyperlinked e-mail address.  
`<a href="mailto:customercare@aptech.ac.in?subject=E-mail to Customer Care">Customer Care</a>`

Using slide 9, explain how to hyperlink to an e-mail address.

Hyperlinks can be even applied to e-mail addresses in the same way, as they can be given for Web pages. There are various tasks that can be performed when a hyperlink are given to an e-mail. Some of these tasks include starting the default e-mail client, creating a new message, inserting the recipients address, adding the subject line, and so on.

To add an e-mail to a hyperlink, the href= attribute must be used and followed by mailto: address attribute. Clicking mailto link, opens users default email program or software. A new email page is created with '**To**' field containing the address of the name specified on the link by default.

To automatically add a subject line in the new e-mail message, the ?subject= attribute must be inserted after the e-mail address.

### Tips:

1. The mailto link using properties of cc and bcc opens the email program with field **cc** and **bcc** containing email addresses of recipients specified in the value. For example,  
`<a href="mailto:aptechworldwide@aptech.com?cc=admin@aptech.com">Tell us your Concerns</a>.`
2. Web browser that does not regconize the email new type will just treat it as plain text and display a text field for it.

## Slide 10

Explain linking other content on the Web page.

The screenshot shows a slide with a blue header containing the text 'HTML' and a large orange number '5'. The main title 'Hyperlink to Other Content Types' is displayed in red. Below the title, there is a bulleted list of four points:

- Hyperlinks can also be used to link to other files and documents.
- Some commonly linked file types on Web pages using hyperlinks are zipped files (.zip), executable files (.exe), documents (.doc), PDF reader files (.pdf), and so on.
- Hyperlinks can also be used to link to graphical .jpg and .gif files.
- To specify a file instead of the Web page, the name of the file must be provided in the <a> tag as shown in the following code snippet:

`<a href="Compressed.zip">Click to download the  
compressed zip file </a>`

At the bottom left of the slide is the copyright notice '© Aptech Ltd.', and at the bottom right is the page number 'Creating Hyperlinks and Anchors / Session 4 10'.

Using slide 10, explain how to hyperlink other contents.

Hyperlinks can be used to not only refer to another Web page or e-mail address, but also can be used to link to other files and documents. Some of the files that are commonly linked on Web pages using hyperlinks are zipped files (.zip), executable files (.exe), documents (.doc), PDF reader files (.pdf), and so on. Hyperlinks can also be used to link to graphical .jpg and .gif files. To specify a file instead of the Web page, the name of the file must be provided in the <a> tag.

## Slide 11

Let us summarize the session.

**HTML5 Summary**

- A hyperlink is referred to as a link. It refers to linking to another Web page or to a section in the same Web page.
- The A (anchor) element is used to create a hyperlink.
- The target attribute of the A element specifies the location where the linked Web page will open when a link is clicked.
- Absolute paths are links that contain the complete address to get to a Web page.
- Relative paths are links that are provided when the files of a Web page are in the same folder as the page displaying the link.
- To add an e-mail to a hyperlink, the href= attribute must be followed by mailto:email address.
- Hyperlinks can also be used to link to files and documents such as zipped files (.zip), executable files (.exe), documents (.doc), PDF reader files (.pdf), and so on.

© Aptech Ltd. Creating Hyperlinks and Anchors / Session 4 11

Use slide 11 to summarize the session. End the session, with a brief summary of what has been taught in the session.

### 4.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session. You should also explore the topics of the next session, *Introduction to CSS3*.

#### Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 5 – Introduction to CSS3

---

## 5.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 5.1.1 Objectives

By the end of this session, the learners will be able to:

- Identify the new functions of CSS3
- Explain the different types of selectors
- Explain nested tags
- Define Classes and IDs for applying styles
- Explain the process to apply styles to hyperlink

### 5.1.2 Teaching Skills

To teach this session, you should be well-versed with the basic features of Cascading Style Sheets (CSS). You should be aware with the different types of selectors and nested tags used in CSS. Also, aware yourself with the use of Classes and IDs for applying styles.

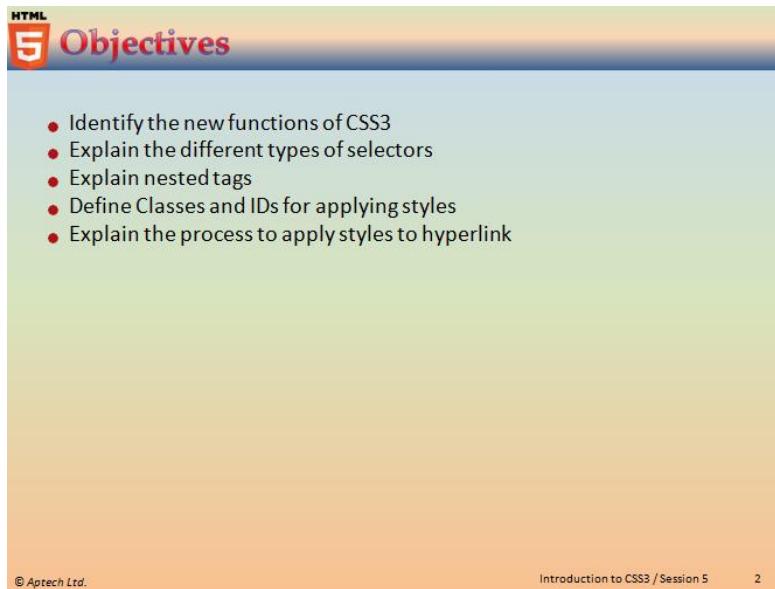
You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

**Overview of the Session:**

Give students the overview of the current session in the form of session objectives. Show the students slide 2 of the presentation.



Tell the students that this session introduces them to the features of CSS used to apply styles to the elements on the HTML Web pages. They will also be learning the new features of CSS3, different types of selectors and nested tags, Classes and IDs for applying styles.

## 5.2 In-Class Explanations

### Slide 3

Explain Cascading Style Sheet (CSS).

The slide has a blue header bar with the text "HTML5 Introduction". Below the header, there is a red box containing the text "Cascading Style Sheet (CSS) is a style sheet language.". There are four colored boxes below it: green (It informs the browser how to present a document.), purple (It uses a markup language for describing the presentation semantics of a document.), and blue (It defines how HTML elements are to be displayed.). At the bottom left is the copyright notice "© Aptech Ltd.", at the bottom center is "Introduction to CSS3 / Session 5", and at the bottom right is the number "3".

Using slide 3, explain the CSS in detail.

CSS is a style sheet language used for informing the browser how to present a document. It uses markup language for describing the presentation semantics of a document. In other words, an HTML document defines the content of the file, whereas the CSS file defines how HTML elements are to be displayed.

The main purpose of CSS is to primarily enable the separation of document content from document presentation which includes elements such as the margins, colors, and fonts.

## Slide 4

Explain the features of CSS3.

The slide has a header 'Cascading Style Sheet 3 (CSS3)' with an 'HTML5' logo. Below the header are six colored boxes containing text:

- Used for adding style such as fonts, colors, and spacing to Web documents.
- Has multiple levels and profiles.
- Updates each level of CSS from the earlier version, by adding new features.
- Denotes version as CSS1, CSS2, CSS3, and CSS4, where the numbers are different for each version or level.
- Is divided into multiple documents called "modules" and each of these modules have new capabilities or extends the features present in CSS2.
- Started drafting of CSS3 when publication of the original CSS2 recommendation was released.

At the bottom left is '© Aptech Ltd.' and at the bottom right are 'Introduction to CSS3 / Session 5' and '4'.

Using slide 4, explain CSS3.

CSS is a mechanism used for adding styles such as fonts, colors, and spacing to Web documents. CSS has multiple levels and profiles. Each level of CSS is updated from the earlier version, by adding new features. CSS version are denoted as CSS1, CSS2, CSS3, and CSS4, where the numbers are different for each version or level.

CSS3 is divided into multiple documents called 'modules'. Each of these modules have new capabilities or extends the features present in CSS2. Drafting of CSS3 started when publication of the original CSS2 recommendation was released. The first CSS3 drafts were released on June 1999. CSS3 extends variety of new ways to create an impact with any designs, with quite a few important changes.

### Tips:

The latest version CSS3 is compatible with the earlier versions of CSS.

## Slides 5 to 8

Explain CSS3 modules.

**HTML 5 Modules 1-4**

- As CSS3 is available as modules and is still evolving, there are many modules having different stability and status.
- Only three modules are released as recommendations and they are as follows:

CSS Color Level 3
CSS Namespaces
Selectors Level 3

- Modules that are stable and in recommendation stage are as follows:

Media Queries
CSS Style Attributes

© Aptech Ltd. Introduction to CSS3 / Session 5 5

**HTML 5 Modules 2-4**

- Modules that are in testing phase and in recommendation stage are as follows:

CSS Backgrounds and Borders Level 3
CSS Image Values and Replaced Content Level 3
CSS Marquee
CSS Multi-column Layout
CSS Speech
CSS Mobile Profile 2.0
CSS TV Profile 1.0

- Modules that are in refining phase and in working draft stage are as follows:

CSS Transforms
CSS Transitions
CSS Values and Units Level 3
CSS Print Profile

© Aptech Ltd. Introduction to CSS3 / Session 5 6

**HTML 5 Modules 3-4**

- Modules that are in revising phase and in working draft and recommendation stage are as follows:
  - CSS Animations
  - CSS Flexible Box Layout
  - CSS Fonts Level 3
  - CSS Paged Media Level 3
  - CSS Text Level 3
  - CSS Basic User Interface Level 3
  - CSS Writing Modes Level 3
- Some of the following modules are in exploring phase and in working draft stage:
  - CSS Cascading and Inheritance Level 3
  - CSS Conditional Rules Level 3
  - CSS Grid Layout
  - CSS Line Grid

© Aptech Ltd. Introduction to CSS3 / Session 5 7

**HTML 5 Modules 4-4**

- Modules that are in rewriting phase and in working draft stage are as follows:
  - CSS Line Layout Level 3
  - CSS Ruby
  - CSS Syntax Level 3
- Modules that are in abandoned phase and in working draft stage are as follows:
  - Behavioral Extensions to CSS
  - CSS Hyperlink Presentation

© Aptech Ltd. Introduction to CSS3 / Session 5 8

Using slides 5 to 8, explain modules available in CSS3.

Explain that since CSS3 is available as modules and is still evolving, there are many modules having different stability and status. Out of the 50 modules published by the CSS working group, only three modules are released as recommendations and they are as follows:

- CSS Color Level 3
- CSS Namespaces
- Selectors Level 3

Rest of the modules are in refining, revising, and testing phase.

**Tips:**

Some of the most popular CSS3 modules that are supported in almost all new versions of the browsers are:

- Selectors
- Box Model
- Backgrounds and Borders
- Image Values and Replaced Content
- Text Effects
- 2D/3D Transformations
- Animations
- Multiple Column Layout
- User Interface

**Slides 9 and 10**

Explain the syntax of CSS.

**Syntax of CSS consists of three parts namely, **selector**, **property**, and **value**.**

**Selector** is an HTML element for which you want to specify the style or the formatting instruction.

**Property** of a selected element is a CSS property that specifies the type of the style to be applied to the selector.

**Value** refers to the value of the CSS property and a CSS property can have multiple values.

Property and the value for a selector are separated with a colon (:).

They are enclosed within the curly brackets ({} ) that is known as the declaration block.

© Aptech Ltd.      Introduction to CSS3 / Session 5      9

© Aptech Ltd. Introduction to CSS3 / Session 5 10

Using slides 9 and 10, explain CSS syntax.

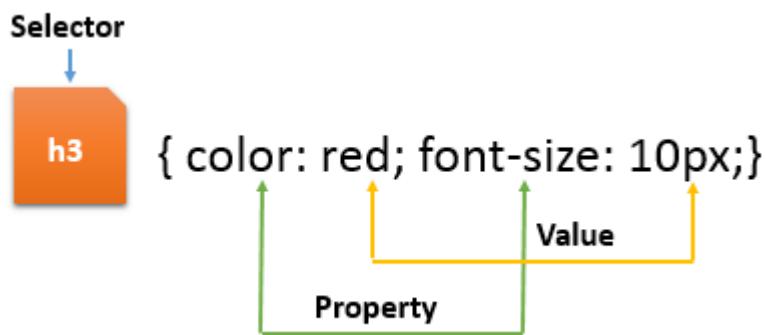
The general syntax of CSS consists of three parts namely, selector, property, and value.

**Selector** - A selector is an HTML element for which you want to specify the style or the formatting instruction.

**Property** - A property of a selected element is a CSS property that specifies the type of the style to be applied to the selector. CSS allows controlling the appearance of the content by providing various properties. These properties include text properties, positioning properties, font properties, color properties, and so on.

**Value** - The property and the value for a selector are separated with a colon (:). They are enclosed within the curly brackets ({} ) that is known as the declaration block.

The following figure explains how to declare a CSS for an HTML element.



Note that you can have various combinations to specify rules for HTML elements.

1. You can specify multiple **property-value** pairs for a selector, which are separated by a semicolon (;) within the declaration block.

2. You can specify multiple selectors for a single property by grouping the selectors. To group the selectors, the selectors are separated by commas followed by a declaration block of properties and values.
  
3. You can specify properties for multiple selectors. Here, the comma-separated selectors are followed with multiple **property-value** pairs.

**Tips:**

CSS selectors are used to find HTML elements based on their ids, classes, types, attributes, and so on.

**In-Class Question:**

After you finish explaining the CSS syntax, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What should be the CSS code to define the red color for the page?

**Answer:**

```
body {color: red;}
```

**Slides 11 to 14**

**HTML 5 Length Measurement Units 1-4**

CSS uses various units of measurements for specifying size of the font, width, and height of margins, and so on.

These units measure the horizontal and vertical length of the content.

CSS supports two types of length measurement units namely, relative and absolute.

© Aptech Ltd. Introduction to CSS3 / Session 5 11

**HTML 5 Length Measurement Units 2-4**

Relative length specifies the length units related to other length property that are calculated in comparison to a current value.

- Following table lists some of the relative length units.

Relative Length	Description
em	Specifies the font size (height) of a particular font. The em unit is relative to the value of the font-size property of the selector.
ex	Specifies the 'x-height' of a particular font. The 'x-height' value is approximately half the font size or the height of the lowercase letter 'x'.
px	Specifies the size in pixels, which is relative to the screen of the device.

© Aptech Ltd. Introduction to CSS3 / Session 5 12

**HTML 5 Length Measurement Units 3-4**

Absolute lengths are specified when the Web page designer is aware of the physical properties of the output device and are specific and fixed values.

- Following table lists some of the absolute length units.

Relative Length	Description
in	Specifies the size in inches, where 1 inch = 2.54 centimeters.
cm	Specifies the size in centimeters'
mm	Specifies the size in millimeters
pt	Specifies the size in points, where 1 point = 1/72th of an inch
pc	Specifies the size in picas, where 1 pica = 12 points

© Aptech Ltd. Introduction to CSS3 / Session 5 13

**HTML 5 Length Measurement Units 3-4**

Percentage allows specifying the length of the content, which is relative to another value.

- Shows the use of percentage in defining the style.

```
H1
{
    font-size: 120%;
    line-height: 200%;
}
```

© Aptech Ltd. Introduction to CSS3 / Session 5 14

Using slides 11 to 14, explain length measurement units. Mention CSS uses various units of measurements for specifying size of the font, width, and height of margins, and so on. These

units measure the horizontal and vertical length of the content. CSS supports two types of length measurement units namely, relative and absolute.

Relative length specifies the length units related to other length property that are calculated in comparison to a current value. Explain the different relative length provided in the table on slide 12.

Then, explain the absolute length. Absolute lengths are specified when the Web page designer is aware of the physical properties of the output device. These are specific and fixed values. Then, explain the absolute length provided in the table on slide 13.

Explain the CSS code on slide 14. Tell the students that percentage allows specifying the length of the content which is relative to another value. Then, explain the CSS code specifies the styles for the H1 element. The font-size property is set to a value of 120%. This means that the size of the header will appear 20% greater than its current size. The line-height property is set to the value 200%. This means that the height of the line will be double the value of the font-size property.

## Slide 15

Explain the types of style sheets.

Three types of style sheets namely, inline, internal or embedded, and external style sheets.

An inline style sheet uses the style attribute within an HTML element to specify the style for HTML elements.

An internal style sheet is also included within the HTML document and is defined using the style element.

© Aptech Ltd. Introduction to CSS3 / Session 5 15

Using slide 15, explain the types of style sheets. Tell the students that the browser reads the style sheet and formats the content on the Web page accordingly.

There are three types of style sheets namely, inline, internal or embedded, and external style sheets. Here, explain the two types of style sheets.

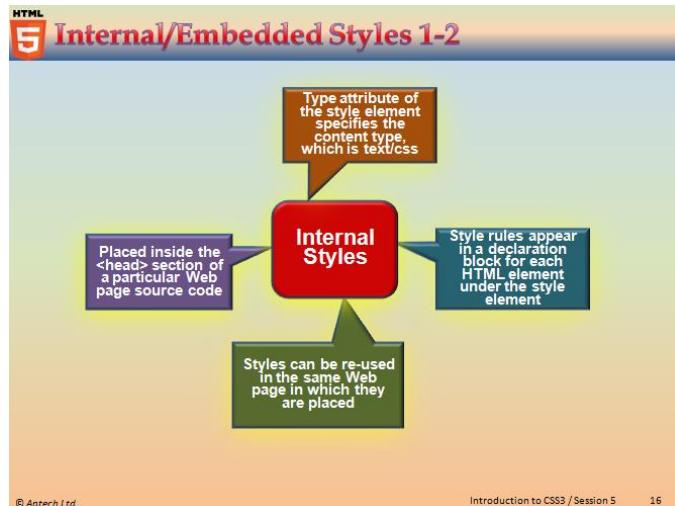
**Inline style sheet** - Uses the style attribute within an HTML element to specify the style for HTML elements.

**Internal style sheet** - Is included within the HTML document. However, it is defined using the style element within the style element. The style rules appear in a declaration block for

each HTML element under the style element. The type attribute of the style element specifies the content type, which is text/css. This means that the content under the style element is CSS code. You can specify any combinations of specifying style rules. The style rules specified for an element will be applied to all the sub-elements. Internal style sheets are useful when styles are to be applied to a specific Web page.

## Slides 16 and 17

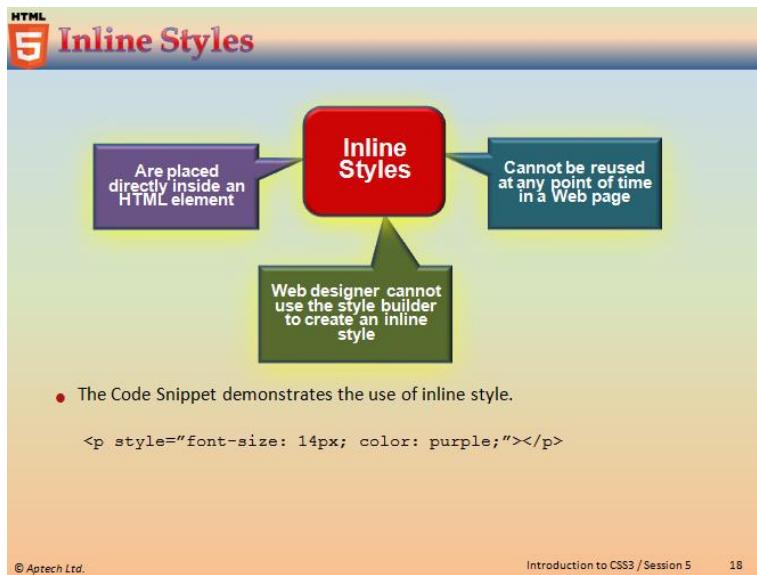
Explain internal style sheet.



Using slides 16 and 17, explain the internal style sheet. Explain the students that the internal style sheets are placed inside the `<head>` section on the HTML Web page.

Explain the CSS properties applied to `<h1>`, `<h2>`, and `<footer>` tags. This can be re-used in the same Web page multiple times.

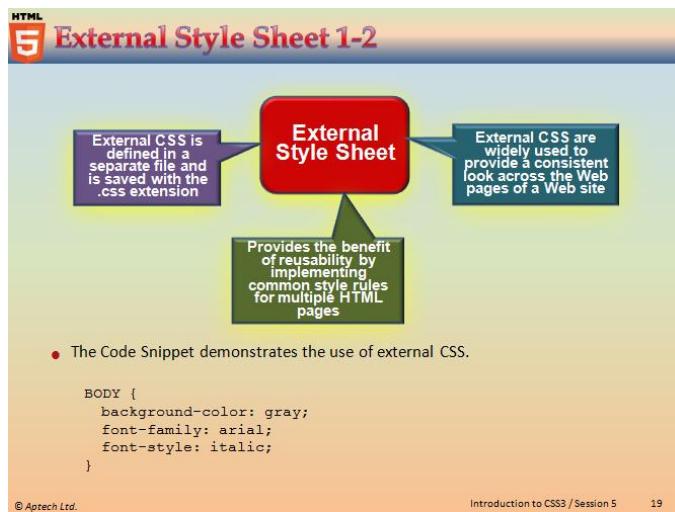
## Slide 18



Using slide 18, explain the inline style sheets. Inline styles are placed directly inside an HTML element. A Web designer cannot use the style builder to create an inline style. Inline style cannot be reused at any point of time in a Web page.

## Slides 19 and 20

Explain external style sheets.



The slide has a blue header bar with the text 'External Style Sheet 2-2'. Below the header, there is a bulleted list and a code snippet.

- Code Snippet shows an example of HTML code using an external CSS style sheet demonstrated earlier.

```
<!DOCTYPE html>
<html>
  <head>
    <LINK rel="stylesheet" type="text/css" href="body.css"/>
    <title>Webex e-Server</title>
  </head>
  <body>
    This is the fastest web server..!!
  </body>
</html>
```

At the bottom of the slide, there is a footer with the text '© Aptech Ltd.', 'Introduction to CSS3 / Session 5', and '20'.

Using slides 19 and 20, explain the external style sheets.

An external CSS is defined in a separate file and is saved with the .css extension. It provides the benefit of reusability by implementing common style rules for multiple HTML pages. Hence, external CSS are widely used to provide a consistent look across the Web pages of a Web site.

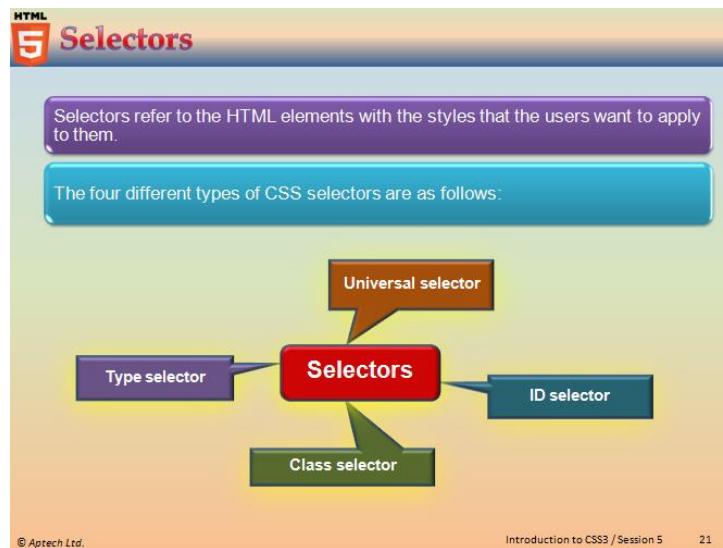
Explain the code snippet given. <Link> tag is used to link the external style sheet with the Web page.

**Tips:**

An external style is used when the styles are to be applied to many pages. Thus, in that case, to change the look of all the pages, you need to make changes only in a single file.

## Slide 21

Explain the CSS selectors.



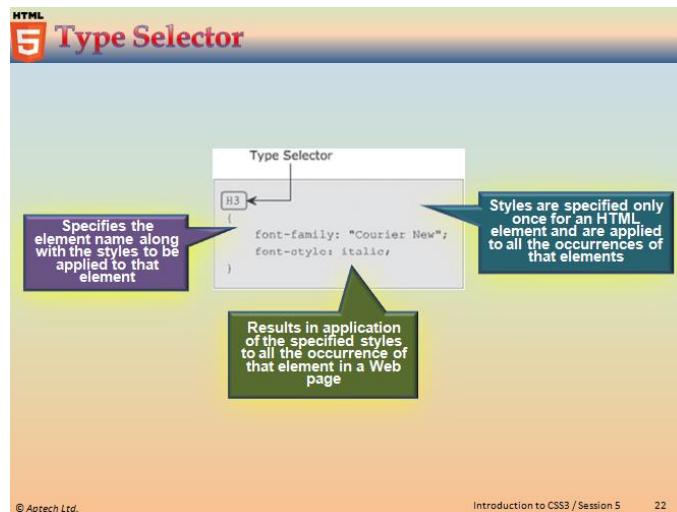
Using slide 21, explain the selectors.

Selectors refer to the HTML elements with the styles, the users want to apply to them. The four different types of CSS selectors are as follows:

- Type selector
- Class selector
- ID selector
- Universal selector

## Slide 22

Explain the type selector.



Using slide 22, explain the type selector.

A type selector simply specifies the element name along with the styles to be applied to that element. This results in applying the specified styles to all the occurrence of that element in a Web page.

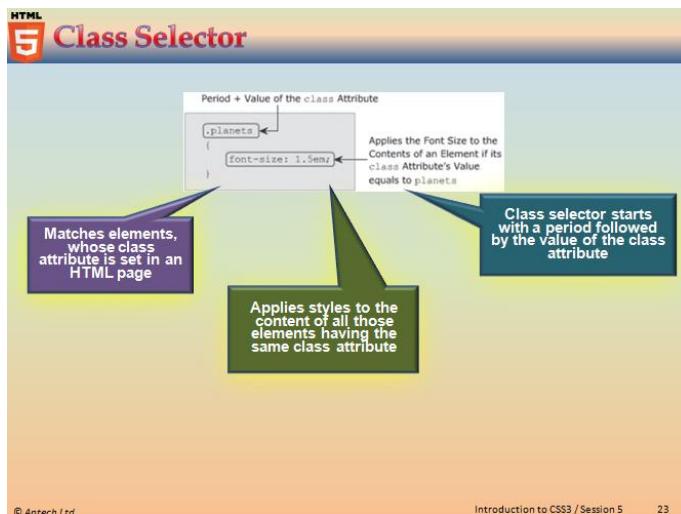
Here, the styles are specified only once for an HTML element and are applied to all the occurrences of that element.

The following example matches all `H1` elements on the page:

```
h1 { font-family: sans-serif }
```

## Slide 23

Explain class selector.



Using slide 23, explain the class selector.

A class selector matches elements whose `class` attribute is set in an HTML page and applies styles to the content of all those elements. For example, if there are `<span>` and `<div>` elements in a Web page with their `class` attributes set, the style specified for the class selector will be applied to both the elements.

For example, the following code shows the matching class selectors for `<div>` and `<span>`.

```
<html>
<head>
<style>
    .foo {
        font-size: 1.8 em;
    }
</style>
</head>
<body>
    <div>A</div>
    <span>B</span>
</body>

```

```

        }
    </style>

</head>

<body>

<span class="foo">Matches Span </span>

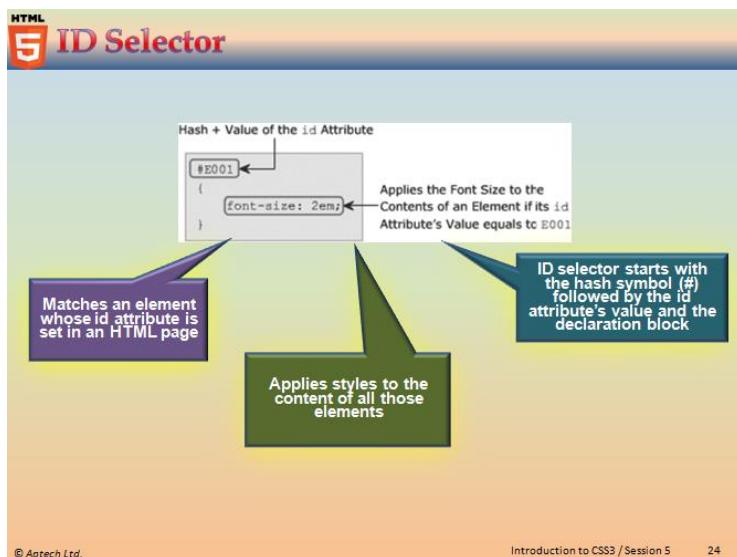
<div class="foo" title="Help">Matches Div </div>

</body>

```

## Slide 24

Explain ID selector.



Using slide 24, explain the ID selector.

An ID selector matches an element whose `id` attribute is set in an HTML page and applies styles to the content of that element. The ID selector specifies styles for an element whose `id` attribute is set to a unique value. An ID selector starts with the hash symbol (#) followed by the `id` attribute's value and the declaration block.

The additional example for ID selector is as follows:

```

<div id="top">

<h1>About Us</h1>

<p class="intro"> We are first in IT Industry to introduce the grid computing.
</p>

<p class="intro"> Go through our services offered in grid computing. </p>

```

```
</div>
```

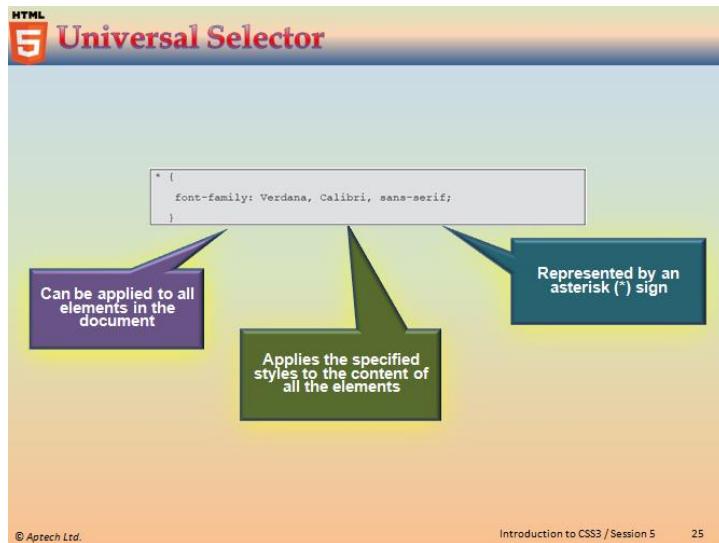
Thus, the css for the mentioned Web page is as follows:

```
#top {
    background-color: #ccc;
    padding: 20px
}

.intro {
    color: red;
    font-weight: bold;
}
```

## Slide 25

Explain universal selector.



Using slide 25, explain universal selector.

The universal selector can be applied to all elements in the document. This means that it applies the specified styles to the content of all the elements. It is represented by an asterisk (\*) sign. For example, universal selector is used to define the font family for all the elements, as shown in code snippet on the slide.

### Tips:

The \* also select all elements inside another element. For example,

```
<div class="intro">
```

```

<p> My name is John <span id="Lastname"> Smith </span> </p>
<p id="my-Address"> I live in Johannesburg </p>
</div>

```

So, to apply CSS to all elements in the `<div>` tag, the selector is as follows:

```

div *
{
    background-color: cyan;
}

```

### Slides 26 (a) and 26 (b)

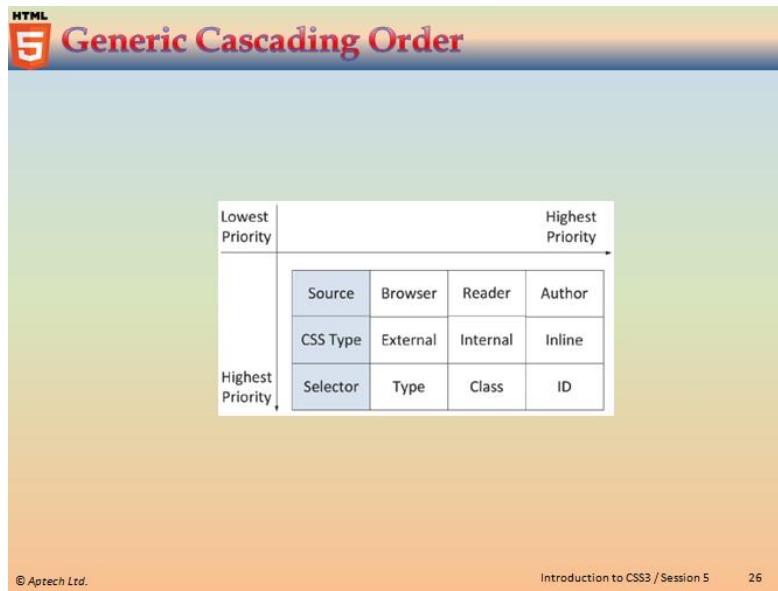
Explain generic cascading order.

**HTML 5 Generic Cascading Order**

- W3C has defined some rules for applying styles to an HTML element. These rules are:
  - Gather all the styles that are to be applied to an element.
  - Sort the declarations by the source and type of style sheet. The source specifies the origin from where the styles are rendered.
  - Highest priority is given to the external style sheet defined by an author. The next priority is of the reader, which can be a software that reads the content, and the last priority is of the browser.
  - Sort the declarations by the priority of a selector, where the ID selector has the highest priority.
  - Sort the declaration according to the specified order.

© Aptech Ltd. Introduction to CSS3 / Session 5 26

**26 (a)**



## 26 (b)

Using slides 26 (a) and 26 (b), explain the generic cascading order.

Consider a scenario where you have multiple style sheets defined for an HTML page. These style sheets might have various selectors and multiple styles defined for an HTML element. Therefore, W3C has defined some rules for applying styles to an HTML element. These rules are as follows:

- Gather all the styles that are to be applied to an element.
- Sort the declarations by the source and type of style sheet. The source specifies the origin from where the styles are rendered.

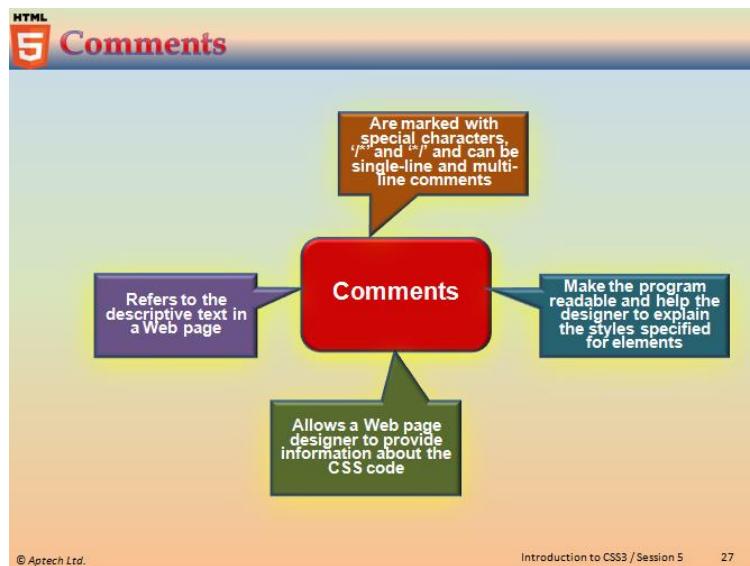
The highest priority is given to the external style sheet defined by an author. The next priority is of the reader, which can be a software that reads the content (screen reader software), and the last priority is of the browser.

- Sort the declarations by the priority of a selector, where the ID selector has the highest priority.
- Sort the declaration according to the specified order.

Explain the generic cascading order provided in the table on slide 26 (b).

## Slide 27

Explain how to apply comments in CSS.



Using slide 27, explain the comments in CSS.

A comment refers to the descriptive text that allows a Web page designer to provide information about the CSS code. Comments make the program more readable and help the designer to explain the styles specified for elements. This is helpful when other Web designers analyze the CSS code.

The browser can identify comments as they are marked with special characters, which are '/\*' and '\*/'. When the browser encounters these symbols, the text within them are ignored and are not displayed in the browser. You can have single-line and multi-line comments in the CSS file.

Example on how to apply comments in a CSS file:

```
/*This is a multiple
lines comment*/
h1
{
    color:red;

/* This is to align text*/
    text-align:center;
}
```

## Slides 28 (a) and 28 (b)

Explain pseudo classes.

**HTML 5 Psuedo Classes 1-3**

- Following table lists the different states of an element:

Sometimes unknowingly the same Web page get open that you have already visited.

You might feel the need for a mechanism that could differentiate the already visited links from the remaining ones.

This is possible by using pseudo classes.

Pseudo classes allow the users to apply different styles to the elements such as buttons, hyperlinks, and so on.

- Syntax for declaring Pseudo classes are as follows:

```
selector_name:state_name {property: value}
```

© Aptech Ltd. Introduction to CSS3 / Session 5 28

### 28 (a)

**HTML 5 Psuedo Classes 1-3**

- Following table lists the different states of an element:

State	Description
active	Defines a different style to an element that is activated by the user.
hover	Defines a different style to an element when the mouse pointer is moved over it.
link	Defines a different style to an unvisited hyperlink.
visited	Defines a different style to the visited hyperlink.

- Syntax for declaring Pseudo classes are as follows:

```
selector_name:state_name {property: value}
```

© Aptech Ltd. Introduction to CSS3 / Session 5 28

### 28 (b)

Using slides 28 (a) and 28 (b), explain the pseudo classes.

Pseudo classes allow the users to apply different styles to the elements such as buttons, hyperlinks, and so on.

Explain using a scenario where a Web site consists of multiple Web pages linked through hyperlinks. Browse through various Web pages by randomly clicking the links within the main page. At times, it might happen that unknowingly the same Web page gets open that

you have already visited. In such a case, you might feel the need for a mechanism that could differentiate the already visited links from the remaining ones. In CSS, this is possible by using pseudo classes.

Explain the syntax to declare a pseudo class in slide 28 (b).

### Tips:

Pseudo-classes names are not case-sensitive.

### Slides 29 and 30

Explain the different pseudo classes.

**HTML 5 Psuedo Classes 2-3**

- Following table lists the selector name and its descriptions:

Selector Name	Description
:link	Is used for selecting all unvisited links
:active	Is used for selecting the active link
:hover	Is used for selecting links on mouse over
:visited	Is used for selecting all visited links
:focus	Is used for selecting the input element which has focus
:first-letter	Is used for selecting the first letter of every <p> element
:first-line	Is used for selecting the first line of every <p> element
:first-child	Is used for selecting every <p> elements that is the first child of its parent
:before	Is used for inserting content before every <p> element
:after	Is used for inserting content after every <p> element

© Aptech Ltd. Introduction to CSS3 / Session 5 29

**HTML 5 Psuedo Classes 3-3**

- Pseudo classes specify the styles to be applied on an element depending on its state.
- In CSS3, a selector can contain multiple pseudo-classes.
- These pseudo-classes should not be mutually exclusive.
- Code snippets demonstrates the use of CSS code specifying the different styles for the visited links, unvisited links, and for the links when the mouse hovers over it.

```
a:link {
    color: white;
    background-color: black;
    border: 2px solid white;
}
a:visited {
    color: white;
    background-color: brown;
    border: 2px solid white;
}
a:hover {
    color: black;
    background-color: white;
    border: 2px solid black;
}
```

Specifies the styles for an unvisited link  
Specifies the styles for a visited link  
Specifies the styles when a mouse hovers over it

© Aptech Ltd. Introduction to CSS3 / Session 5 30

Using slides 29 and 30, explain the list of selectors.

Pseudo classes specify the styles to be applied on an element depending on its state. In CSS3, a selector can contain multiple pseudo-classes. These pseudo-classes should not be mutually exclusive. For example, the selectors `a:visited:hover` and `a:link:hover` are applicable, but `a:link:visited` is not applicable, because `:link` and `:visited` are mutually exclusive selectors.

CSS code specifies the different styles for the visited links, unvisited links, and for the links when the mouse hovers over it.

### Slide 31

Explain the purpose of pseudo elements.

The slide has a header 'HTML 5 Purpose of Pseudo Elements'. Below the header is a bulleted list of six points. At the bottom is a diagram illustrating the syntax for declaring pseudo elements.

- Consider a scenario where you are designing a Web site that explains the important technical terms.
- While defining such terms, you might feel the need to emphasize more on the first letter by applying different styles.
- Pseudo elements provide you with a flexibility to apply styles to a specific part of the content such as a first letter or first line.
- Pseudo element adds some special effects to HTML elements such as `<p>`, `<body>`, and so on.
- Syntax for declaring psuedo elements is:

```

    graph TD
      A[Can be either :first-line or :first-letter.] --> B[selector_name]
      B --> C[pseudo_element]
      C --> D[property: value]
      E[Is an element name.] --> B
      F[Is any CSS property such as color, border, and font.] --> D
  
```

© Aptech Ltd. Introduction to CSS3 / Session 5 31

Using slide 31, explain the purpose of pseudo elements.

Mention pseudo elements provide you with a flexibility to apply styles to a specific part of the content such as a first letter or first line. This allows you to control the appearance of that specific content without affecting the rest of the content.

Explain with help of scenario, where you are designing a Web site that explains the important technical terms. While defining such terms, you might feel the need to emphasize more on the first letter by applying different styles. It becomes difficult if you try to apply styles only on the first letter of a line or paragraph. This can be achieved by using the pseudo elements.

Pseudo element adds some special effects to HTML elements such as `<p>`, `<body>`, and so on.

## Slides 32 and 33

Explain first-line and first-letter pseudo elements.

The slide title is "Psuedo Elements 1-2". A green callout box contains the text: "The :first-line and :first-letter pseudo elements allow you to apply styles to the first line and first letter respectively." A purple callout box below it states: "The :first-line pseudo element allows you to apply styles to the first line." A blue arrow points from the text "The Code Snippet declares the style that will be applied to the first line in the paragraph." to a code snippet. The code is:

```
p:first-line
{
font-family: "Tahoma";
font-weight: bolder;
background-color: #FFFFCC;
}
```

A callout box with a blue arrow points to the word "first-line" in the code, stating: "Specifies the styles to be applied to the first line of the paragraph content".

© Aptech Ltd. Introduction to CSS3 / Session 5 32

The slide title is "Psuedo Elements 2-2". A green callout box contains the text: "The :first-letter pseudo element allows you to apply styles to the first letter." A purple callout box below it states: "The Code Snippet declares the style that will be applied to the first letter in the paragraph." A blue arrow points from the text "The Code Snippet declares the style that will be applied to the first letter in the paragraph." to a code snippet. The code is:

```
p:first-letter
{
font-family: "fantasy";
font-size: xx-large;
font-weight: bold;
}
```

A callout box with a blue arrow points to the word "first-letter" in the code, stating: "Specifies the styles to be applied to the first letter of the paragraph content".

© Aptech Ltd. Introduction to CSS3 / Session 5 33

Using slides 32 and 33, explain the pseudo elements.

- **:first-line:** The :first-line pseudo element allows you to apply styles to the first line. HTML code where the :first-line pseudo element will be used is shown in slide 32.
- **:first-letter:** The :first-letter pseudo element allows you to apply styles to the first letter. HTML code for the :first-letter pseudo element is shown in slide 33.

## Slide 34

Explain how to use pseudo classes with CSS.

**HTML**

### Styles to Hyperlink

- CSS can be used to change the appearance and behavior of hyperlinks.
- There are two other ways to assign hyperlink styles namely, div specific and Link specific.
- A div specific hyperlink styles can be created and assigned to a specific div and will have all the hyperlinks present within the div to follow the specified rules.
- Class specific hyperlink styles generally uses a class than an id. A point to note that an id can only be used once on a page whereas a class can be used multiple times as required.

© Aptech Ltd. Introduction to CSS3 / Session 5 34

Using slide 34, explain the styles to hyperlink.

CSS can be used to change the appearance and behavior of hyperlinks. To do this, use the following selectors/pseudo-classes:

- a
- a:link
- a:visited
- a:hover
- a:active

This selectors/pseudo classes represent the ‘anchor’ element (specified using the HTML a tag) and its various states.

There are two other ways to assign hyperlink styles. They are as follows:

1. Div specific
2. Link specific

A hyperlink styles can be created and assigned to a specific div. This will have all the hyperlinks present within the div to follow the specified rules. It is irrelevant if the div is an (#) id or (.) class.

Specific styling can be assigned to a specific type of hyperlink. This is achieved by creating the style rules in the CSS. For this type of hyperlink styling, a class is used generally than an

id. A point to note that an `id` can only be used once on a page, whereas a `class` can be used multiple times as required.

### Slide 35

Let us summarize the session.

The slide has a header 'HTML 5 Summary' with a logo. The content is a bulleted list of CSS concepts:

- CSS is a mechanism for adding style such as fonts, colors, and spacing to Web documents. CSS has multiple levels and profiles.
- The general syntax of CSS consists of three parts namely, selector, property, and value.
- Selectors refer to the HTML elements with the styles that are applied to them and they can be Type, Class, ID, or Universal selectors.
- A comment refers to the descriptive text that allows a Web page designer to provide information about the CSS code.
- Pseudo classes allow the users to apply different styles to the elements such as buttons, hyperlinks, and so on.
- Pseudo elements allow the developer to apply styles to a specific part of a content such as first letter or first line.
- A hyperlink style can be assigned either through DIV or through link class.

© Aptech Ltd. Introduction to CSS3 / Session 5 35

Use slide 35 to summarize the session. End the session, with a brief summary of what has been taught in the session.

### 5.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session. You should also explore the formatting using style sheets that are offered with the next session.

#### Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 6 – Formatting Using Style Sheets

---

## 6.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 6.1.1 Objectives

By the end of this session, the learners will be able to:

- List and explain text and font styles
- Describe inline spans
- Explain paragraph indentation and application of border
- Explain horizontal paragraph alignment
- Explain vertical spacing within a paragraph

### 6.1.2 Teaching Skills

To teach this session, you should be well-versed with CSS properties to format text using various font properties. You should also aware yourself with the paragraph indentation and applying borders to the various elements on the Web page. Apart from this, you can also aware yourself with horizontal and vertical paragraph alignment and spacing within a paragraph.

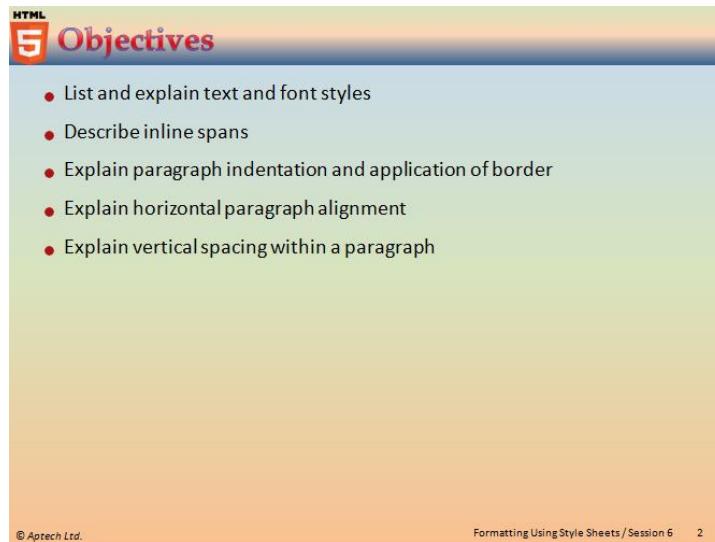
You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

## Overview of the Session:

Then give the students the overview of the current session in the form of session objectives.  
Show the students slide 2 of the presentation.



Tell the students that this session introduces them to the various CSS properties to format the text using font styles, inline spans, paragraph indentation, borders, and so on. They will also know about the horizontal paragraph alignment and vertical spacing within a paragraph.

## 6.2 In-Class Explanations

### Slides 3 and 4

Let us understand the CSS properties to format text.

**HTML**

### 5 Text and Font Style 1-2

- The text properties specify and control the appearance of the text in a Web page.
- A user can change the color of a text, increase or decrease the space between characters, align a text, and so on using the text properties.
- Following table lists different text properties.

Property	Description
color	It is used for specifying the color of the text.
text-align	It is used in specifying the horizontal alignment of text in an element.
text-decoration	It is used for specifying the decoration of the text in an element.
text-indent	It is used for specifying the indentation of first line of text in an element in length or %.
text-transform	It is used in specifying the casing of text in an element.
word-spacing	It is used for increasing or decreasing the space between words.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      3

**HTML**

### 5 Text and Font Style 2-2

- The font properties allow you to specify the font for the text and change the different font attributes of the text such as font, size, and style of the text.
- Following table lists the different font properties.

Property	Description
font-family	It is used for specifying the font and can specify a generic family or a specific family name such as "Serif" or "Times New Roman".
font-size	It is used for specifying the size of the font and can have an absolute or relative value.
font-style	It is used for specifying the style of the font.
font-variant	It is used for specifying whether the text should be displayed in small-caps.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      4

Using slides 3 and 4, explain the text styles.

Mention text properties specify and control the appearance of the text in a Web page. You can change the color of a text, increase or decrease the space between characters, align a text, and so on using the CSS text properties.

Explain the properties of text with its description listed on the slide. Then, explain the code snippet:

```
h1 {text-align:center;}  
p.main  
  
{text-align:justify;  
text-indentation: 50px;  
text-transform: capitalize;}
```

The code demonstrates the use of `text-align` property to align the text of the level-1 heading. Similarly, you can show the use of other properties to format the text in a paragraph.

Using slide 4, explain the font styles.

Mention font properties allow you to specify the font for the text. They allow you to change the different font attributes of the text such as font, size, and style of the text. The browser must support the font specified by the font properties. Otherwise, it will display the default font, which is dependent on the browser.

Explain the properties of the font styles with its description.

#### **Tips:**

Always use lowercase letters for the font URL. Uppercase letters can give unexpected results in IE.

#### **Slides 5 to 12**

Let us understand text styles.

**HTML**  
**5 Text Styles 1-8**

- The different text styles such as `text-align`, `text-indent`, and `text-transform` provide different values that allow specifying the alignment, indentation, and casing of text in an element.
- The `text-align` property allows the text to be centered, or left or right aligned, or justified.
- Following table lists the values of `text-align` property.

Value	Description
<code>left</code>	It is used for aligning the text to the left of the Web page.
<code>right</code>	It is used for aligning the text to the right of the Web page.
<code>center</code>	It is used for aligning the text in the middle of the Web page.
<code>justify</code>	It is used for justifying the text on both sides of the Web page.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      5

**HTML 5 Text Styles 2-8**

- The `text-indent` property is used for specifying the indentation of the text.
- Following table lists the values of `text-indent` property.

Value	Description
<code>length</code>	It is used in specifying fixed indentation and the default value is 0.
<code>%</code>	It is used to specify an indentation as a percentage of the width of the parent element which the selector elements is defined.

- The `text-transform` property is for changing the case of letters in a text.
- Following table lists the values of `text-transform` property.

Value	Description
<code>none</code>	It is used in specifying that the text will be displayed with the same casing as written within the element.
<code>capitalize</code>	It is used in specifying that the first letter of each word will be capitalized.
<code>Uppercase</code>	It is used in specifying only uppercase letters.
<code>Lowercase</code>	It is used in specifying only lowercase letters.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      6

**HTML 5 Text Styles 3-8**

- Following figure shows DIV element HTML code.

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
<LINK rel="stylesheet" type="text/css" href="TextProperties.css"/>
<TITLE>Client</TITLE>
</HEAD>
<BODY>
<H2>Client Contact Information</H2>
<DIV>
<H4>Dynamic Solutions</H4>
<P>Tel Number - 445 558 7744</P>
<P>Fax Number - 703 740 6539</P>
</DIV>
</BODY>
</HTML>
```

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      7

**HTML 5 Text Styles 4-8**

**> CSS Code**

- Following figure displays a CSS code that specifies the text styles for the DIV element.

```
div
{
  text-align:left;
  text-indent:2em;
  text-transform:uppercase;
}
```

- The `text-align` property is set to `left`, which will align the text towards the left.
- The `text-indent` property is set to `2em`, which will indent the text with respect to the font size.
- The `text-transform` property is set to `uppercase`, which will display all the letters in uppercase.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      8

**HTML 5 Text Styles 5-8**

- The text specified in the DIV element is aligned towards the left and all the letters are displayed in uppercase.
- Following figure shows the output.

**Client Contact Information**

DYNAMIC SOLUTIONS

TEL NUMBER - 445 558 7744

FAX NUMBER - 703 740 6539

- The text-decoration and word-spacing properties provides different values that allow the user to specify the decoration and word spacing of text in an element.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      9

**HTML 5 Text Styles 6-8**

- Following table lists the values assigned to the text-decoration property.

Value	Description
none	It is used for displaying normal text without any formatting.
underline	It is used for displaying a line under the text.
overline	It is used for displaying a line over the text.
line-through	It is used for displaying a line through the text.
blink	It is used for flashing the text.

- Following table lists the values assigned to the word-spacing property.

Value	Description
normal	It is used in specifying normal spacing between words and it is the default value.
length	It is used in specifying fixed space between words.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      10

**HTML 5 Text Styles 7-8**

- Following figure shows the header and paragraph HTML code.

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
<TITLE>Solar System</TITLE>
<LINK rel="stylesheet" type="text/css" href="Txtproperties.css"/>
</HEAD>
<BODY>
<H3>Nine Planets</H3>
<P>Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune, Pluto</P>
</BODY>
</HTML>
```

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      11

The slide title is "Text Styles 8-8". It contains the following text and code:

- Following figure displays a CSS code that specifies the text properties for the BODY and H3 elements.

```
body
{
    word-spacing:2px;
}
h3
{
    text-decoration:underline;
}
```

- The word-spacing property is set to 2px for the BODY element.
- The text-decoration property is set to underline for the H3 element.
- Following figure shows the header and paragraph HTML code.

**Nine Planets**

Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune, Pluto

The slide footer includes the copyright notice "© Aptech Ltd." and the page number "Formatting Using Style Sheets / Session 6 12".

Using slides 5 to 12, explain text styles.

The different text styles such as `text-align`, `text-indent`, and `text-transform` provide different values that allow specifying the alignment, indentation, and casing of text in an element. The `text-align` property allows the text to be centered, or left or right aligned, or justified.

Explain with the help of table, the different values for `text-align`.

Using slide 6, explain `text-indent` property and `text-transform` property.

Explain the `text-indent` property is used for specifying the indentation of the text using the table.

Also, explain the `text-transform` property is used for changing the case of letters in a text using the table.

Using slide 7, explain the `Div` element.

Using `<div>` tag, a special division can be formed in a Web page and the Id or Class selectors can be used to apply formatting to specific div on the page.

Using slide 8, explain the CSS code.

Explain the figure that displays a CSS code that specifies the text styles for the `DIV` element. The `text-align` property is set to `left`, which will align the text towards the left. The `text-indent` property is set to `2em`, which will indent the text with respect to the font size. The `text-transform` property is set to `uppercase` which will display all the letters in uppercase.

Using slide 9, explain the output of the CSS code.

The text specified in the DIV element is aligned towards the left and all the letters are displayed in uppercase.

Mention text-decoration and word-spacing properties provides different values that allow the user to specify the decoration and word spacing of text in an element.

**Tips:**

In CSS3, the text-decoration property is a shorthand property for text-decoration-line, text-decoration-color, and text-decoration-style, but this is currently not supported in any of the major browsers.

Using slide 10, explain the lists of values which can be assigned to the text-decoration and word-spacing property.

Using slide 11, explain the <h3> tag and <p> tag used in the code snippet. Using slide 12, explain the CSS code for the body and level-3 heading element.

Explain the figure that displays a CSS code that specifies the text properties for the BODY and H3 elements. The word-spacing property is set to 2px for the BODY element. This will display each word by leaving a distance of two pixels. The text-decoration property is set to underline for the H3 element. This will underline the heading in the Web page.

Show that in the figure, the header is underlined and each word in the header and the paragraph is displayed by leaving a distance of two pixels between them.

**In-Class Question:**

After you finish explaining the text styles, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which property can be applied to make the text flash on the Web page?

**Answer:**

text-decoration property with blink as value is used for making the text blink or flash on the Web page.

## Slides 13 and 14

Let us understand the inline span.

**HTML5 Inline Span 1-2**

- The `<span>` tag groups inline-elements in a document.
- For example, if one word in a sentence needs to be bold or colored without using the `<b>` tag then a `<span>` tag is used which can be present within an existing tag.
- The Code Snippet demonstrates CSS inline style for `<span>` tag.

```
<p>My mother has <span style="color: lightblue">light blue</span> eyes.</p>
Or
<span class="eyesonly">light blue</span>
```

- The Code Snippet demonstrates CSS external style for `<span>` tag.

```
.eyesonly {font-color: lightblue}
```

- The span tag has different attributes; it supports JavaScript event attributes also.

© Aptech Ltd.

Formatting Using Style Sheets / Session 6 13

**HTML5 Inline Span 2-2**

- Following table lists different attributes and values used in `<span>` tag.

Attribute	Value	Description
<code>class</code>	<code>classname</code>	It is used in specifying the text direction for the content in an element.
<code>dir</code>	<code>rtl</code> <code>ltr</code>	It is used in specifying the text direction for the content in an element.
<code>id</code>	<code>id</code>	It is used in specifying a unique id for an element.
<code>lang</code>	<code>language_code</code>	It is used in specifying a language code for the content in an element.
<code>style</code>	<code>style_definition</code>	It is used in specifying an inline style for an element.
<code>title</code>	<code>text</code>	It is used in specifying extra information about an element.
<code>xml:lang</code>	<code>language_code</code>	It is used in specifying a language code for the content in an element, in XHTML documents.

© Aptech Ltd.

Formatting Using Style Sheets / Session 6 14

Using slides 13 and 14, explain modules available in CSS3.

The `<span>` tag groups inline-elements in a document. For example, if one word in a sentence need to be bold or colored without using the `<b>` tag, then a `<span>` tag is used which can be present within an existing tag.

Mention `<span>` tag provides no visual change by itself. The `<span>` tag provides a way to add a hook to a part of a text or a part of a document.

Explain the code snippet given in the slide. Also, explain the attributes and values of `span` tag and its description.

## Tips:

When a text is hooked in a <span> element, you can style it with CSS or manipulate it with JavaScript.

## Slides 15 to 20

Let us understand indenting paragraph.

**HTML 5 Indenting Paragraph 1-6**

- Indenting is the process of setting off the text from its normal position, either to the left or to the right.
- In paragraph style, there are three types of indentation:

**► First line indent**

- The `text-indent` property is used in the CSS for indenting the first line of a paragraph.
- The Code Snippet demonstrates inline style for `<p>` tag and an internal CSS code for first line indent.

```
Inline style
<p style="text-indent: 50px">
Internal CSS
p {text-indent: 50px}
```

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      15

**HTML 5 Indenting Paragraph 2-6**

- The Code Snippet demonstrates the use of the `text-indent` property in the HTML file.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Font Gallery</title>
    <style>
      p {text-indent: 150px}
    </style>
  </head>
  <body>
    <p>
      The font styles properties allow you to specify the font for the text. They allow you to change the different font attributes of the text such as font, size, and style of the text. The browser must support the font specified by the font properties. Otherwise, it will display the default font, which is dependent on the browser.
    </p>
  </body>
</html>
```

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      16

**HTML**  **Indenting Paragraph 3-6**

- Following figure shows the output of text-indent property.

The font styles properties allow you to specify the font for the text. They allow you to change the different font attributes of the text such as font, size, and style of the text. The browser must support the font specified by the font properties. Otherwise, it will display the default font, which is dependent on the browser.

**> Padding**

- The padding property is used to add a specified amount of space between the border of an element and its contents.
- The Code Snippet demonstrates inline style for <p> tag and an internal CSS code for padding property.

```
Inline style
<p style="padding: 20px">
Internal CSS
p {padding: 20px}
```

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      17

**HTML**  **Indenting Paragraph 4-6**

- The Code Snippet demonstrates the use of the text-indent property in the html file.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Font Gallery</title>
    <style>
      p {padding: 20px }
    </style>
  </head>
  <body>
    <p>
      The font styles properties allow you to specify the font for the text. They allow you to change the different font attributes of the text such as font, size, and style of the text. The browser must support the font specified by the font properties. Otherwise, it will display the default font, which is dependent on the browser.
    </p>
  </body>
</html>
```

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      18

**HTML**  **Indenting Paragraph 5-6**

- Following figure shows the padding property.

**Padding** → The font styles properties allow you to specify the font for the text. They allow you to change the different font attributes of the text such as font, size, and style of the text. The browser must support the font specified by the font properties. Otherwise, it will display the default font, which is dependent on the browser.

**> Margin**

- The margin property is used to add a specified amount of white space around an element, on the outside of the element.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      19

**HTML**

## Indenting Paragraph 6-6

- Following code snippet demonstrates inline style for `<p>` tag and an internal CSS code for margin property.

```
Inline style
<p style="margin: 20px">
Internal css
p {margin: 20px}
```

- Following figure shows the output of margin property.

**margin**

The font styles properties allow you to specify the font for the text. They allow you to change the different font attributes of the text such as font, size, and style of the text. The browser must support the font specified by the font properties. Otherwise, it will display the default font, which is dependent on the browser.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      20

Using slides 15 to 20, explain indenting paragraph. Indenting is the process of setting off the text from its normal position, either to the left or to the right. In paragraph style, there are three types of indentation:

**First line indent** - The `text-indent` property is used in the CSS for indenting the first line of a paragraph.

Code Snippet demonstrates inline style for `<p>` tag and an internal CSS code for first line indent.

**Tips:**

Negative values are allowed. The first line will be indented to the left if the value specified is negative.

Using the slide 17, explain the output and concept of padding.

Mention padding property is used to add a specified amount of space between the border of an element and its contents. The padding clears an area around the content (inside the border) of an element. The padding is affected by the background color of the element. The top, right, bottom, and left padding can be changed independently using separate properties.

Using slide 18, explain the code for padding property applied to `<p>` tag.

Using slides 19 and 20, explain the output and concept of margin.

The `margin` property is used to add a specified amount of white space around an element. The `margin` clears an area around an element (outside the border). The margin does not have a background color and is completely transparent.

The top, right, bottom, and left margin can be changed independently using separate properties.

Also, explain code snippet and output of the inline style for `<p>` tag and an internal CSS code for `margin` property.

**In-Class Question:**

After you finish explaining the indenting paragraph, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which property is used to specify amount between its border and content of an element?

**Answer:**

`padding` property is used to specify amount between its border and content of a element.

**Slides 21 to 27**

Let us understand the border style.

**HTML 5 Border Style 1-7**

- Borders are rectangular outlines that surround an element.
- Borders present around text and an image emphasize the content inside the text box.
- CSS border properties specify the style, color, and width of the border.
- Following table lists the border-style properties.

Property	Description
<code>border-left-style</code>	It sets an element's left border.
<code>border-right-style</code>	It sets an element's right border.
<code>border-top-style</code>	It sets an element's top border.
<code>border-bottom-style</code>	It sets an element's bottom border.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      21

**HTML 5 Border Style 2-7**

- Following table lists the values of the `border-style` properties.

Value	Description
<code>Dashed</code>	It is used for specifying a dashed border.
<code>Dotted</code>	It is used for specifying a dotted border.
<code>Double</code>	It is used for specifying two borders.
<code>groove</code>	It is used for specifying a 3D grooved border.
<code>Inset</code>	It is used for specifying a 3D inset border.
<code>outset</code>	It is used for specifying a 3D outset border.
<code>ridge</code>	It is used for specifying a ridged border.
<code>solid</code>	It is used for specifying a solid border.

© Aptech Ltd.

Formatting Using Style Sheets / Session 6 22

**HTML 5 Border Style 3-7**

- Following figure shows an HTML code.

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
<TITLE>MagnaSoftwares</TITLE>
<LINK rel="stylesheet" type="text/css" href="styles.css" />
</HEAD>
<BODY>
<DIV id="heading">
<H2>Welcome to MagnaSoftwares</H2>
</DIV>
</BODY>
</HTML>
```

- Following figure shows CSS code of border style.

```
#heading
{
background:#FFFFCC;
text-align:center;
border-left-style:ridge;
border-right-style:groove;
border-top-style:dashed;
border-bottom-style:double;
}
```

© Aptech Ltd.

Formatting Using Style Sheets / Session 6 23

**HTML 5 Border Style 4-7**

- Explanation for code.

```
border-left-style: ridge;
    • Applies a ridged border to the left.
border-right-style: groove;
    • Applies a 3D grooved border to the right.
border-top-style: dashed;
    • Applies a dashed border at the top.
border-bottom-style: double;
    • Applies two borders at the bottom.
```

- Following figure shows the output of `border-style` properties.

© Aptech Ltd.

Formatting Using Style Sheets / Session 6 24

**HTML 5 Border Style 5-7**

### ► Shorthand property

- To make the code concise CSS allows certain shorthand properties to reduce the length of the code.
- The shorthand property for setting the border is **border-style**.
- Following figure shows Sample HTML Code.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Corpse - Worlds Largest Flower</title>
    <link rel="stylesheet" type="text/css" href="flower.css"></link>
</head>
<body>
<figure></figure>
<h2>World's Largest Flower</h2>
<p>Corpse flower is the world's largest flower.<br/>
Its diameter is about a metre.<br/>
It grows in openings in rainforests on limestone hills of Sumatra, Indonesia.</p></td>
</body>
</html>
```

© Aptech Ltd.

Formatting Using Style Sheets / Session 6 25

**HTML 5 Border Style 6-7**

- Following figure shows CSS code of shorthand **border-style** properties.

```
.largest_flower
{
    border-style:groove inset outset dashed;
}
```

- Explanation for code.**

border-style: groove inset outset dashed;

- Applies a 3D grooved border at the top, 3D inset border at the right, 3D outset border at the bottom, and dashed border at the left.

© Aptech Ltd.

Formatting Using Style Sheets / Session 6 26

**HTML 5 Border Style 7-7**

- Following figure shows output of shorthand **border-style** properties.



**World's Largest Flower**

Corpse flower is the world's largest flower.  
Its diameter is about a metre.  
It grows in openings in rainforests on limestone hills of Sumatra, Indonesia.

© Aptech Ltd.

Formatting Using Style Sheets / Session 6 27

Using slides 21 to 27, explain the border styles.

Mention borders are rectangular outlines that surround an element. Borders present around text and image, emphasize the content inside the text box. CSS border properties specify the style, color, and width of the border.

Explain the properties of border given in the table.

Using slide 22, explain the `border-style` properties. Explain the value that can be specified to `border-style` properties along with its description.

Using slides 23 and 24, explain the HTML code and the output for the same.

### **Tips:**

The `border-style` property can have from one to four values. The following table lists the style values and their description:

Style	Description
<code>border-style:dotted solid double dashed;</code>	<ul style="list-style-type: none"> <li>• top border is dotted</li> <li>• right border is solid</li> <li>• bottom border is double</li> <li>• left border is dashed</li> </ul>
<code>border-style:dotted solid double;</code>	<ul style="list-style-type: none"> <li>• top border is dotted</li> <li>• right and left borders are solid</li> <li>• bottom border is double</li> </ul>
<code>border-style:dotted solid;</code>	<ul style="list-style-type: none"> <li>• top and bottom borders are dotted</li> <li>• right and left borders are solid</li> </ul>
<code>border-style:dotted;</code>	<ul style="list-style-type: none"> <li>• all four borders are dotted</li> </ul>

Using slide 25, explain the shorthand property.

Mention to make the code concise as CSS 3.0 allows certain shorthand properties. With the help of these shorthand properties, the length of the code is reduced. The shorthand property for setting the border is `border-style`.

Explain the HTML code snippet. Using slides 26 and 27, explain the CSS code and output for the HTML code.

### **Explanation for the code:**

`border-style: groove inset outset dashed;` applies a 3D grooved border at the top, 3D inset border at the right, 3D outset border at the bottom, and dashed border at the left. Explain the figure that shows output of shorthand border-style properties.

**Tips:**

None of the border properties will have ANY effect unless the `border-style` property is set.

**Slides 28 to 33**

Let us understand `border-color` property.

**Border Color 1-6**

- The `border-color` property in CSS applies colors to all the four borders.
- One can also apply four different colors to four borders.
- There are other border color properties that allow a user to individually specify colors of the left, right, top, or bottom border.
- Following table lists the different border color properties.

Property	Description
<code>border-bottom-color</code>	It is used to specify the color for the bottom border.
<code>border-left-color</code>	It is used to specify the color for the left border.
<code>border-right-color</code>	It is used to specify the color for the right border.
<code>border-top-color</code>	It is used to specify the color for the top border.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      28

**Border Color 2-6**

- The `border-color` property accepts different color values that determine the different shades of color to be applied to the borders.
- Following table lists the values of the different border-color properties.

Value	Description
<code>color</code>	It is used in specifying the color to be applied to the border by using either the RGB or hexadecimal value, or the color name itself.
<code>transparent</code>	It is used for specifying that the border is transparent.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      29

**HTML 5 Border Color 3-6**

- Following figure shows an HTML code with properties.

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
<TITLE>HealthCare</TITLE>
<LINK rel="stylesheet" type="text/css" href="Tips.css" />
</HEAD>
<BODY>
<DIV class="tips">
<H2>Five Essential Health Tips</H2>
<OL>
<LI>Quit Smoking</LI>
<LI>Reduce Stress</LI>
<LI>Protect yourself from Pollution</LI>
<LI>Avoid Excessive Drinking</LI>
<LI>Regular Exercise</LI>
</OL>
</DIV>
</BODY>
</HTML>
```

© Aptech Ltd.

Formatting Using Style Sheets / Session 6 30

**HTML 5 Border Color 4-6**

- Following figure shows the CSS code of border-color properties.

```
.tips
{
background:#FFDDDD;
border-bottom-color:#FF0000;
border-top-color:#FF0000;
border-right-color:#0000FF;
border-left-color:#0000FF;
}
```

- Explanation for code.**
- border-bottom-color: #FF0000;  
Displays the bottom border in red color.
- border-top-color: #FF0000;  
Displays the top border in red color.
- border-right-color: #0000FF;  
Displays the right border in blue color.
- border-left-color: #0000FF;  
Displays the left border in blue color.

© Aptech Ltd.

Formatting Using Style Sheets / Session 6 31

**HTML 5 Border Color 5-6**

### ➤ Shorthand property

- The shorthand property for setting the color of the border is border-color.
- Following figure shows an HTML code of a table with border-color properties.

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
<TITLE>Car_Gallery</TITLE>
<LINK rel="stylesheet" type="text/css" href="Gallery.css" />
</HEAD>
<BODY>
<H2>Car_Gallery</H2>
<TABLE border="1">
<TR>
<TD><B>Ferrari</B><BR/><IMG alt="Ferrari" class="carmodel" src="Ferrari.jpg"/></TD>
<TD><B>Chevrolet</B><BR/><IMG alt="Chevrolet" class="carmodel" src="Chevrolet.jpg"/></TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

© Aptech Ltd.

Formatting Using Style Sheets / Session 6 32

The slide has a green-to-yellow gradient background. At the top left is a small 'HTML5' logo. The title 'Border Color 6-6' is in large red font. Below the title is a bulleted list: 'Following figure shows CSS code off shorthand border-color.' A code block follows:

```
body
{
    text-align:center;
}
.carmodel
{
    border-style:solid;
    border-color: Red Blue Green Yellow;
}
```

Below the code is another bulleted list: 'Explanation for code.' It includes the line 'border-color: Red Blue Green Yellow;' followed by a description: 'Displays the top border in red, right border in blue, bottom border in green and left border in yellow color.'

At the bottom left is the copyright notice '© Aptech Ltd.' and at the bottom right is the page number 'Formatting Using Style Sheets / Session 6 33'.

Using slides 28 to 33, explain border-color property.

The border-color property in CSS applies colors to all the four borders. You can also apply four different colors to four borders. There are other border color properties that allow you to individually specify colors of the left, right, top, or bottom border.

Using slide 29, explain the border-color property value and its description.

The border-color property accepts different color values that determine the different shades of color to be applied to the borders. The table lists the values of the different border-color properties.

Using slides 30 and 31, explain the HTML and CSS code snippet with link tag and using the external CSS file.

Using slide 33, explain the shorthand property and its example.

The shorthand property for setting the color of the border is border-color. Explain the figure that shows an HTML code of a table with border-color properties.

#### Explanation for the code:

`border-color: Red Blue Green Yellow;` displays the top border in red, right border in blue, bottom border in green, and left border in yellow color.

#### Tips:

The border-color property does not work if it is used alone. Use the border-style property to set the borders first before applying colors.

## Slides 34 to 40

Let us understand the border-width property.

**HTML 5 Border Width 1-7**

- The border-width property is a shorthand property used to specify the width for all the four borders.
- There are other border-width properties that allow a user to individually specify the left, right, top, or bottom borders.
- Following table lists the different border-width properties.

Property	Description
border-bottom-width	It is used to specify the width of the bottom border.
border-left-width	It is used to specify the width of the left border.
border-right-width	It is used to specify the width of the right border.
border-top-width	It is used to specify the width of the top border.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6    34

**HTML 5 Border Width 2-7**

- The values of the border width properties specify the way the border will appear.
- Following table lists the values of the different border-width properties.

Value	Description
medium	It is used in specifying a medium border.
length	It is used in accepting an explicit value that specifies the thickness of border.
thick	It is used for displaying a thick border.
thin	It is used in specifying a thin border.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6    35

**HTML 5 Border Width 3-7**

- Following figure shows an HTML code for border-width properties.

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
<TITLE>EasyBank</TITLE>
<LINK rel="stylesheet" type="text/css" href="banner.css" />
</HEAD>
<BODY>
<DIV class="banner">
<H2>EasyBank - Whole world one Bank</H2>
</DIV>
</BODY>
</HTML>
```

- Following figure shows the CSS code of border-width properties.

```
.banner
{
    text-align:center;
    background-color:#C0C0C0;
    border-style:solid;
    border-right-style:none;
    border-left-style:none;
    border-top-width: thick;
    border-bottom-width: thick;
    font-family:fantasy;
}
```

© Aptech Ltd.      Formatting Using Style Sheets / Session 6    36

**HTML 5 Border Width 4-7**

- **Explanation for code.**

```
border-top-width: thick;
border-bottom-width: thick;
```

  - Displays a thick top border.
  - Displays a thick bottom border.
- Following figure shows the output of border-width properties.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      37

**HTML 5 Border Width 5-7**

### ➤ Shorthand property

- The shorthand property for setting the border is border-width.
- Following figure shows an HTML code using the shorthand border-width properties.

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
<TITLE>UNICEF</TITLE>
<LINK rel="stylesheet" type="text/css" href="UNICEF.css" />
</HEAD>
<BODY>
<H2>About UNICEF</H2>
<P Class="aboutus">
UNICEF is an organization that supports and works for children's
rights, development and protection.
</P>
</BODY>
</HTML>
```

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      38

**HTML 5 Border Width 6-7**

- Following figure shows the CSS code using the shorthand property, border-width.

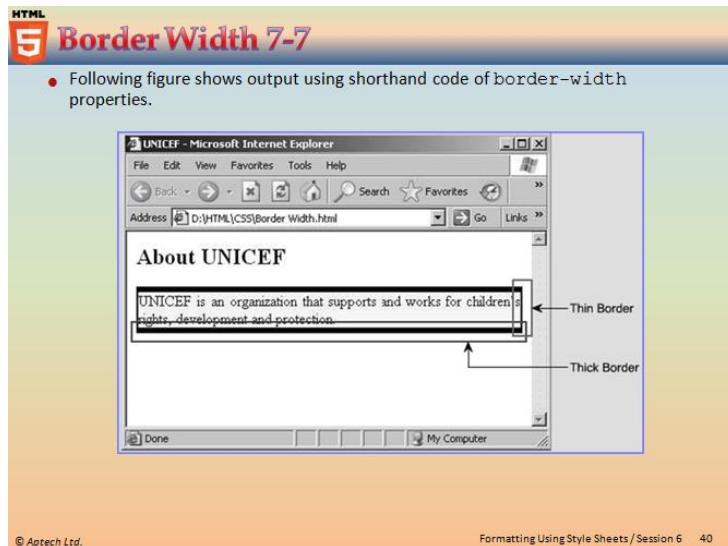
```
.aboutus
{
text-align:justify;
background-color:#FFFFCC;
border-style:solid;
border-width: thick thin thick thin;
}
```

- **Explanation for code.**

```
border-width: thick thin thick thin;
```

  - Specifies a top and bottom border as thick and right and left border as thin.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      39



Using slides 34 to 40, explain the `border-width` property. The `border-width` property is a shorthand property that specifies the width for all the four borders. There are other `border-width` properties that allow you to individually specify the left, right, top, or bottom borders. Explain the table that lists the different `border-width` properties.

Using slide 35, explain the `border-width` properties to specify the way the border will appear.

The width of the border can be specified or altered by using the predefined values of the border width properties.

Mention `border-width` property is used to set the width of the border. The width is set in pixels or by using one of the three pre-defined values: `thin`, `medium`, or `thick`.

Using slide 36, explain the HTML and CSS code for `border-width` property.

#### **Explanation for the code:**

`border-top-width: thick;` displays a thick top border.

`border-bottom-width: thick;` displays a thick bottom border.

Using slide 38, explain the shorthand property and its example.

The shorthand property for setting the border is `border-width`. Explain the figure that shows an HTML code using the shorthand `border-width` properties.

#### **Explanation for the code:**

`border-width: thick thin thick thin;` specifies a top and bottom border as thick and right and left border as thin.

Using slides 39 and 40, explain the CSS code and output for the same.

## Slides 41 to 46

Let us understand shorthand border.

**HTML 5 Shorthand Border 1-6**

- The border shorthand property in CSS specifies all the properties such as style, width, and color for all the four borders.
- It allows the user to specify the different properties in just one declaration.
- One can also set these properties individually by using the different shorthand border properties.
- Following table lists the different shorthand border properties.

Property	Description
border-bottom	It is used to specify the width, style, and color of the bottom border.
border-left	It is used to specify the width, style, and color of the left border.
border-right	It is used to specify the width, style, and color of the right border.
border-top	It is used to specify the width, style, and color of the top border.

© Aptech Ltd.

Formatting Using Style Sheets / Session 6 41

**HTML 5 Shorthand Border 2-6**

- Following figure shows an HTML code for shorthand border properties.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Important Note</title>
  <link rel="stylesheet" type="text/css" href="impnote.css"/>
</head>
<body>
<h3>Notice:</h3>
<div class="impnote">
<ul style="list-style:square">
<li>Mobiles are not allowed during class hours.</li>
<li>Each student should carry his/her identity card regularly.</li>
</ul>
</div>
</body>
</html>
```

© Aptech Ltd.

Formatting Using Style Sheets / Session 6 42

**HTML** **5 Shorthand Border 3-6**

- Following figure shows a CSS code using different shorthand border properties.

```
.impnote
{
background-color:#FFFFCC;
border-top:dashed thin #FF0000;
border-bottom:ridge thick #0000FF;
border-right:dotted thin #FF8040;
border-left:inset medium #FF00FF;
}
```

- Explanation for code.**
  - `border-top: dashed thin #FF0000;`
    - Displays a thin top border with a dashed line in red color.
  - `border-bottom: ridge thick #0000FF;`
    - Displays a thick ridged bottom border in blue color.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6    43

**HTML** **5 Shorthand Border 4-6**

```
border-right: dotted thin #FF8040;
```

- Displays a thin right border with a dotted line in orange color.

```
border-left: inset medium #FF00FF;
```

- Displays a medium 3D inset left border in purple color.

- Following figure shows the output of border properties.

**Notice:**

- Mobiles are not allowed during class hours.
- Each student should carry his/her identity card regularly.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6    44

**HTML** **5 Shorthand Border 5-6**

- Following figure shows an HTML and CSS code of image border property.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Flower Gallery</title>
<style>
.flower
{
    border:solid thin #FF0000;
}
</style>
</head>
<body>
<h2>Flower</h2>
<table>
<tr>
<td></td>
<td align="top"><h3>Lilac is a species of flowering plants in the olives family. They are shrubs that ranges from 2 to 10m in height.</h3></td>
</tr>
<tr>
<td></td>
<td align="top"><h3>Sunflower is a flowering plants whose stem can grow as high as 3m tall.</h3></td>
</tr>
</table>
</body>
</html>
```

© Aptech Ltd.      Formatting Using Style Sheets / Session 6    45

**HTML 5 Shorthand Border 6-6**

- **Explanation for code.**
  - border: solid thin #FF0000;
  - Specifies that all the four borders must be solid in style, thin by width, and red in color.
- Following figure shows the output of Image Border property.

Flower

Lilac is a species of flowering plants in the olives family. They are shrubs that ranges from 2 to 10m in height.

Sunflower is a flowering plants whose stem can grow as high as 3m tall.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      46

Using slides 41 to 46, explain the shorthand border. The border shorthand property in CSS specifies all the properties such as style, width, and color for all the four borders. It allows the user to specify the different properties in just one declaration. These properties can be set individually by using the different shorthand border properties.

Explain the table that lists the different shorthand border properties. The values of the different border properties determine the type of effect to be applied to the borders.

Mention to shorten the code, it is possible to specify all the individual border properties in one property. This is called a shorthand property.

Using slides 42 to 44, explain the HTML and CSS code snippet and its output.

#### **Explanation for the code:**

border-top: dashed thin #FF0000; displays a thin top border with a dashed line in red color.

border-bottom: ridge thick #0000FF; displays a thick ridged bottom border in blue color.

border-right: dotted thin #FF8040; displays a thin right border with a dotted line in orange color.

border-left: inset medium #FF00FF; displays a medium 3D inset left border in purple color. Explain the figure that shows the output of border properties.

Similarly, using slides 45 and 46, explain the HTML and CSS code snippet with the output.

### Explanation for the code:

`border: solid thin #FF0000;` specifies that all the four borders must be solid in style, thin by width, and red in color. Explain the figure that shows the output of image border property.

### Slides 47 and 48

Let us understand horizontal alignment.

**HTML**

## 5 Horizontal Alignment 1-2

- In CSS, `text-align` property is used for horizontal alignment of text in an element.
- This property aligns the inline content of a block.
- Following table lists all values of `text-align` property.

Value	Description
<code>left</code>	Aligns the text to the left.
<code>right</code>	Aligns the text to the right.
<code>center</code>	Centers the text.
<code>justify</code>	Aligns text to both left and right margins by adding space between words (like in newspapers and magazines).
<code>inherit</code>	Specifies that the value of the <code>text-align</code> property should be inherited from the parent element.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      47

## 5 Horizontal Alignment 2-2

The `text-align` property applies only to block-level elements, such as paragraphs.

Hence, `text-align` cannot change the alignment of a single word without changing the alignment of the entire line.

For Western languages, which are read from left to right, the default value of `text-align` is `left`.

The text aligns on the left margin and has a ragged right margin.

Languages such as Hebrew and Arabic has default align to right since they are read from right to left.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      48

Using slides 47 and 48, explain the horizontal alignment.

In CSS, `text-align` property is used for horizontal alignment of text in an element. This property aligns the inline content of a block. Explain the table that lists all values of `text-align` property.

The `text-align` property applies only to block-level elements, such as paragraphs. Hence, `text-align` cannot change the alignment of a single word without changing the alignment of the entire line.

For western languages, which are read from left to right, the default value of `text-align` is `left`. The text aligns on the left margin and has a ragged right margin. Languages such as Hebrew and Arabic have default align to right since they are read from right to left.

### Tips:

A block element is an element that takes up the full width available and has a line break before and after it.

Examples of block elements:

- `<h1>`
- `<p>`
- `<div>`

### Slide 49

Let us understand vertical alignment.

**HTML5 Vertical Alignment**

- In CSS `line-height` property is used for vertical alignment of text in an element.
- This property is also a component of the 'font' shorthand property.
- It can be applied on block-level elements, table cells, table caption, and so on.

Value	Description
<code>normal</code>	A normal line height. This is default.
<code>number</code>	A number that will be multiplied with the current font size to set the line height.
<code>length</code>	A fixed line height in px, pt, cm, and so on.
<code>%</code>	A line height in percent of the current font size.
<code>inherit</code>	Specifies that the value of the <code>line-height</code> property should be inherited from the parent element.

© Aptech Ltd.      Formatting Using Style Sheets / Session 6      49

Using slide 49, explain the vertical alignment.

In CSS, `line-height` property is used for vertical alignment of text in an element. This property is also a component of the 'font' shorthand property. It can be applied on block-level elements, table cells, table captions, and so on.

Explain the table that lists the values of `line-height` property.

Explain the following code which uses `line-height` property:

```
p.small {line-height:90%}
p.big {line-height:200%}
```

**In-Class Question:**

After you finish explaining vertical alignment, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which property is used to align the content vertically?

**Answer:**

line-height property.

**Slide 50**

Let us summarize the session.

**HTML 5 Summary**

- The text styles specify and control the appearance of the text in a Web page.
- Indenting is the process of offsetting text from its normal position, either to the left or to the right.
- CSS border property specifies the style, color, and width of the border.
- The border-color property accepts different color values that determine the different shades of color to be applied to the borders.
- The values of the different border properties determine the type of effect to be applied to the borders.
- In CSS, text-align property is used for horizontal alignment of text in an element.
- In CSS, line-height property is used for vertical alignment of text in an element.

© Aptech Ltd.                      Formatting Using Style Sheets / Session 6      50

Use slide 50 to summarize the session. You will end the session, with a brief summary of what has been taught in the session.

**6.3 Post Class Activities for Faculty**

You should familiarize yourself with the topics of the next session. You should also explore the Displaying Graphics and CSS3 Animation that are offered with the next session.

**Tips:**

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 7 – Displaying Graphics and CSS3 Animation

---

## 7.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 7.1.1 Objectives

By the end of this session, the learners will be able to:

- Explain graphic formatting in Web pages
- Explain graphic insertion, sizing, and padding
- Explain CSS3 Animation
- Describe the use of CSS3 on mobile devices

### 7.1.2 Teaching Skills

To teach this session, you should be well-versed with graphic formatting in Web pages. Also the graphic insertion, sizing, and padding should be known. Along with this, you should prepare yourself with how to apply CSS3 animation. You should also explore few Web sites to get the idea on how to CSS3 on mobile devices.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### **Tips:**

It is recommended that you test the understanding of the students by asking questions in between the class.

#### **In-Class Activities:**

Follow the order given here during In-Class activities.

**Overview of the Session:**

Then, give the students the overview of the current session in the form of session objectives. Show the students slide 2 of the presentation.

**Objectives**

- Explain graphic formatting in Web pages
- Explain graphic insertion, sizing, and padding
- Explain CSS3 Animation
- Describe the use of CSS3 on mobile devices

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 2

Tell the students that this session will introduce them how to apply graphic formatting to the HTML Web pages. The session explores CSS3 features to learn how to perform graphic insertion, sizing, and padding. They will also learn about the CSS3 animation and the use of CSS3 on mobile devices.

## 7.2 In-Class Explanations

### Slides 3 to 6

Let us understand about graphic format.

**HTML 5 Graphic Format 1-4**

- There are many graphic formats available; the most commonly used are Joint Photographic Experts Group (JPEG), Graphics Interchange Format (GIF), and Portable Network Graphics (PNG).
- The difference between each graphic format depends on the following characteristics:

**> Color Depth**

- It is defined by the number of distinct colors that are represented by a hardware or software.
- Color depth is defined by the number of bits per pixel (bpp) and it is also called as bit depth.
- Higher color depth indicates higher range of colors used.

**> Compression/file size**

- Compression stores the original images in a reduced number of bytes using an algorithm.
- This image can be expanded back to the original size using a decompression algorithm.

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 3

**HTML 5 Graphic Format 2-4**

- The two types of image file compression algorithms used are as follows:

**Lossless compression**

- File size is reduced but preserves a copy of the original uncompressed image.
- Avoids accumulating stages of re-compression when editing images.

**Lossy compression**

- A representation of the original uncompressed image is preserved.
- The image appears to be a copy of the original image but in real it is not a copy.
- Lossy compression achieves smaller file sizes when compared with lossless compression.
- Lossy compression algorithms allow variable compression that comprises on image quality for file size.

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 4

**HTML 5 Graphic Format 3-4**

**> Animation**

- Some graphic format consists of a series of frames that are played one after the other giving an impression of animation.
- Following figure shows an animated graphic.

**> Transparency**

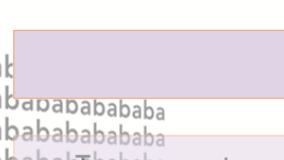
- It is very common on the Web to display an image on a Web page that appears directly against the background color of the page.
- The background color of the Web page shows through the transparent portion of the image.

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 5

**HTML**

## Graphic Format 4-4

- In a transparent image, one and only one color can be hidden.
- If the color chosen to make transparent is same as the background of the inserted image, then an irregularly shaped image appears to float on the page.
- Following figure shows a transparent Image.



ababab  
ababab  
ababababababababa  
ababababababababa  
ababababababababa  
ababababababababa  
ababababababababa

Transparent

Using slides 3 to 6, explain the graphic format.

There are many graphic formats available; the most commonly used are Joint Photographic Experts Group (JPEG), Graphics Interchange Format (GIF), and Portable Network Graphics (PNG). The difference between each graphic format depends on the following characteristics:

**Color Depth** – It is defined by the number of distinct colors that are represented by a hardware or software. Color depth is defined by the number of Bits Per Pixel (BPP) and it is also called as bit depth. Higher the color depth indicates higher range of colors used. For example, a color depth of 8-bit for GIF image would offer maximum of 256 variations. Similarly, a color depth of 24-bit will give 16,000,000 variations.

**Compression/file size** – Graphic files are large, so images are compressed using various techniques. Compression stores the original images in a reduced number of bytes using an algorithm. This image can be expanded back to the original size using a decompression algorithm. In some compression formats, images with less complexity results in smaller compressed file sizes.

Using slide 4, explain the two types of image file compression algorithms namely, lossless and lossy.

Using slides 5 and 6, explain the **animation** and **transparency** graphic format.

### In-Class Question:

After you finish explaining graphic format, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What is the effect of a compression done on an image?

### Answer:

Compression stores the original images in a reduced number of bytes using an algorithm.

### Slides 7 and 8

Let us understand graphic format for the Web.

**HTML 5 Graphic Format for the Web 1-2**

- For Web pages, use of JPEG and PNG graphics are recommended as it provides maximum compatibility with all the devices accessing the Web page.
- For photos, use of JPEG graphic format and for screen-shots and drawings use of PNG graphic format is recommended.

**> JPEG**

- Uses a lossy compression which means that the image quality is lost in the process of compressing the image.
- For continuous tone pictures such as photos, JPEG should be used.
- Most JPEG editors allow the user to specify the amount of detail that the user is prepared to lose.
- If the quality is reduced, then the loss is visible; JPEG is about half the size of PNG.

**> PNG**

- Uses lossless compression, which means there is no loss of any image detail.
- Designed for transferring images on the Internet and not for professional-quality print graphics.
- Therefore, it does not support non-RGB color spaces such as CMYK.
- Supports high color and partial transparency using alpha channels.

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 7

**HTML 5 Graphic Format for the Web 2-2**

**> GIF**

- Uses a lossless compression which means that there is no loss in quality when the image is compressed.
- The uncompressed image stores its information in a linear fashion.
- Each line of pixels is read from left to right.
- An interlaced GIF file stores the lines of the image in a different order.
- Animated graphics are stored in GIF format.

Compatibility and appearance are the keywords on the Web.

The inserted images must be visible and undistorted when appearing on any recipient's device.

The Web designer can make assumptions that the Web site will open in a computer which will have minimum resolution of 800x600 pixel display capability.

If a mobile based Web page needs to be created then the specifications will change.

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 8

Using slides 7 and 8, explain the various graphic format available for the Web.

For Web pages, use of JPEG and PNG graphics are recommended, as it provides maximum compatibility with all the devices that might be accessing the Web page. For photos, use of JPEG graphic format and for screen-shots and drawings use of PNG graphic format is recommended. Both these formats compress the picture information to reduce the download time and increase the downloading speed.

Then, explain the features of each graphic format to the students.

### Slides 9 to 14

Let us understand how to add graphics on the Web page.

**HTML 5 Graphic Insertion 1-6**

- The IMG element is an empty element, which allows the user to insert an image in a Web page.
- It allows insertion of images and diagrams.
- The commonly used graphic formats that are supported are namely, GIF, JPEG, BITMAP (BMP), and PNG.
- The <img> tag reserves a space for the image and does not insert the image in the HTML page.
- It creates a link between the image and the HTML page.

© Aptech Ltd.

Displaying Graphics and CSS3 Animation / Session 7 9

**HTML 5 Graphic Insertion 2-6**

- Following table lists the commonly used attributes of the IMG element.

Data Type	Description
<code>src</code>	Specifies the path of an image that is to be displayed.
<code>height</code>	Specifies the height of an image.
<code>width</code>	Specifies the width of an image.

- Following Code Snippet demonstrates displaying an image using the IMG element.
 

```
<body>

</body>
```
- The code uses the `src` attribute of the IMG element to insert a JPEG image.
- The attribute specifies the name of the image and also indicates that the image is present in the same folder where the HTML file is saved.
- The width and height of the image is set to 225 and 151 pixels respectively by using the `width` and `height` attribute.
- A pixel refers to the smallest dot on the monitor screen.

© Aptech Ltd.

Displaying Graphics and CSS3 Animation / Session 7 10

**HTML 5 Graphic Insertion 3-6**

- An image can also be stored in a subfolder of the folder containing the HTML file.
- In such cases, a reference to the image is made by using the sub folder name as shown in the following Code Snippet.

```
<body>

</body>
```

- To align the image, the `float` style attribute can be used to specify the inline style for the element.
- This will force the image to be aligned to the left or right side of the screen and wrap the surrounding text around the image.
- Following Code Snippet demonstrates the use of the `float` style.

```
<body>

</body>
```

© Aptech Ltd.

Displaying Graphics and CSS3 Animation / Session 7 11

**HTML 5 Graphic Insertion 4-6**

- Following table lists the values of `float` property in the `<img>` tag.

Value	Description
left	The element floats to the left.
right	The element floats to the right.
none	The element does not float and is the default value.
inherit	The element specifies that the value of the <code>float</code> property should be inherited from the parent element.

HTML5 introduced a new `<figure>` tag that acts as a container containing the `<img>` tag.

It is not a replacement for `<img>` tag, but acts as a container into which the `<img>` tag is placed.

The `<figure>` tag specifies self-contained content, such as illustrations, diagrams, photos, code listings, and so on.

The content of the `<figure>` element is related to the main flow, its position is independent of the main flow.

© Aptech Ltd.

Displaying Graphics and CSS3 Animation / Session 7 12

**HTML 5 Graphic Insertion 5-6**

- Following Code Snippet demonstrates the use of `<figure>` tag.

```
<figure>

</figure>
```

- The main advantage of using `<figure>` tag is that it allows the user to use the `<figcaption>` tag along with it.
- The `<figcaption>` tag allows the user to add a caption to the image.
- The caption always appears along with the image even if the image floats in Web site layout.
- Following Code Snippet demonstrates the use of `<figcaption>` tag.

```
<figure>

<figcaption>This diagram shows the logo of a
product.</figcaption>
</figure>
```

© Aptech Ltd.

Displaying Graphics and CSS3 Animation / Session 7 13

**HTML 5 Graphic Insertion 6-6**

- The `<figure>` tag can also assign styles and other attributes to the `<figure>` element using an external or internal style sheet.
- A single caption to a group of images can be added using the `<figure>` tag.
- Following Code Snippet demonstrates how to assign a single caption to a group of images.

```
<figure>



<figcaption>The different types of flowers</figcaption>
</figure>
```

- Following figure shows output of a single caption to a group of images.

The different types of flowers

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 14

Using slides 9 to 14, explain the graphic insertion. Tell the students that the graphics can be added on the Web pages using `IMG` and `Figure` element.

The `IMG` element is an empty element, which allows the user to insert an image on a Web page. It allows insertion of images and diagrams. The commonly used graphic formats that are supported are namely, GIF, JPEG, BMP, and PNG.

The `<img>` tag reserves a space for the image and does not insert the image in the HTML page. It creates a link between the image and the HTML page.

Then, explain the attributes of `<img>` tag namely `src`, `height`, and `width`.

Use Code Snippet to explain how to display an image in a Web page using the `IMG` element.

The code uses the `src` attribute of the `IMG` element to insert a JPEG image. The attribute specifies the name of the image and also indicates that the image is present in the same folder where the HTML file is saved. The width and height of the image is set to 225 and 151 pixels respectively by using the `width` and `height` attribute. A pixel refers to the smallest dot on the monitor screen.

Using slide 11, explain the `src` attribute if stored in subfolder of the folder.

An image can also be stored in a subfolder of the folder containing the HTML file. In such cases, a reference to the image is made by using the sub folder name as shown in Code Snippet.

Mention the `alt` attribute which is the other important attribute of the `<img>` tag. The required `alt` attribute specifies an alternate text for an image, if the image cannot be displayed.

The `alt` attribute provides alternative information for an image if a user for some reason cannot view it (because of slow connection, an error in the `src` attribute, or if the user uses a screen reader).

Using slide 12, explain the `float` property and `<figure>` tag.

HTML5 introduced a new `<figure>` tag. The `<figure>` tag acts as a container containing the `<img>` tag. In other words, it is not a replacement for `<img>` tag, but acts as a container into which the `<img>` tag is placed. The `<figure>` tag specifies self-contained content, such as illustrations, diagrams, photos, code listings, and so on.

Using slide 13, explain the `<figure>` and `<figcaption>` tag.

To align the image the `float` style attribute can be used to specify the inline style for the element. This will force the image to be aligned to the left or right side of the screen and wrap the surrounding text around the image. Code Snippet demonstrates the use of the `float` style.

While the content of the `<figure>` element is related to the main flow, its position is independent of the main flow, and if removed it does not affect the flow of the document.

The main advantage of using `<figure>` tag is that it allows the user to use the `<figcaption>` tag along with it. The `<figcaption>` tag allows the user to add a caption to the image. The caption always appears along with the image even if the image floats in Web site layout. Last code snippet in the slide demonstrates the use of `<figcaption>` tag.

Using slide 14, explain the `<figure>` tag attributes.

The `<figure>` tag can also assign styles and other attributes to the `<figure>` element using an external or internal style sheet. A single caption to a group of images can be added using the `<figure>` tag. Code Snippet demonstrates how to assign a single caption to a group of images.

#### Tips:

To create a tooltip for an image, use the `title` attribute. The `title` attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element. In HTML5, the `title` attribute can be used on any HTML element.

## Slides 15 and 16

Let us understand CSS image sizing and padding.

**HTML 5 CSS Image Sizing and Padding 1-2**

- Size of an image is specified in pixels.
- The `height` and `width` property sets the height and width of the image.
- One can specify the width and the height will be resized or vice versa.
- Following Code Snippet demonstrates CSS code for setting the image height and width property.

```
p.ex
{
height:100px;
width:100px;
}
```

- Following table lists different CSS properties and values of images.

Property	Description	Values
<code>height</code>	Sets the height of an element	<ul style="list-style-type: none"> <li>Auto</li> <li>Length</li> <li>%</li> <li>Inherit</li> </ul>

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 15

**HTML 5 CSS Image Sizing and Padding 2-2**

Property	Description	Values
<code>max-height</code>	Sets the maximum height of an element	<ul style="list-style-type: none"> <li>none</li> <li>length</li> <li>%</li> <li>Inherit</li> </ul>
<code>max-width</code>	Sets the maximum width of an element	<ul style="list-style-type: none"> <li>none</li> <li>length</li> <li>%</li> <li>Inherit</li> </ul>
<code>min-height</code>	Sets the minimum height of an element	<ul style="list-style-type: none"> <li>length</li> <li>%</li> <li>Inherit</li> </ul>
<code>min-width</code>	Sets the minimum width of an element	<ul style="list-style-type: none"> <li>length</li> <li>%</li> <li>Inherit</li> </ul>
<code>width</code>	Sets the width of an element	<ul style="list-style-type: none"> <li>auto</li> <li>length</li> <li>%</li> <li>Inherit</li> </ul>

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 16

Using slides 15 and 16, explain the CSS image sizing and padding.

Size of an image is specified in pixels. The `height` and `width` property sets the height and width of the image. Explain the Code Snippet that demonstrates how to use CSS for setting the image height and width property.

Using slide 16, explain the different properties of `image` tag.

## Slides 17 to 19

Let us understand padding and shorthand padding.

**HTML 5 Padding 1-3**

- The CSS padding property is used to specify the space between the element border and the element content.
- The background color of the element affects the padding property.
- Using separate properties such as top, right, bottom, and left, different padding values can be specified and the padding can be changed separately.
- Following table lists the various values used in padding property.

Value	Description
length	This property specifies a fixed value for padding in pixels, pt, em, and so on.
%	This property specifies a value for padding in % of the containing element.

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 17

**HTML 5 Padding 2-3**

- Following Code Snippet demonstrates the CSS code used for specifying different padding values for different sides.

```
padding-top:10px;
padding-bottom:10px;
padding-right:15px;
padding-left:15px;
```

- In the code, the value for padding was set for all the sides.
- Instead of using different padding for different sides, users can use a shorthand property.
- A shorthand property is one where all the padding properties for the different sides are specified in one property.
- The shorthand property for all the padding properties is padding.
- Following Code Snippet demonstrates the use of the shorthand property for padding.

```
padding:25px 50px 75px 100px;
where,
top padding is 25px, right padding is 50px, bottom padding is 75px,
and left padding is 100px.
```

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 18

**HTML 5 Padding 3-3**

- Following table lists all CSS padding properties.

Property	Description
padding	The browser calculates the height and is the default value
padding-bottom	Defines the length in pixels (px)
padding-left	Defines the height of the containing block in percent format
padding-right	Specifies that the value of the property should be inherited from the parent element
padding-top	Sets the top padding of an element

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 19

Using slides 17 to 19, explain padding and shorthand padding.

The CSS padding property is used to specify the space between the element border and the element content. It is used to separate them from the surrounding element. The

background color of the element affects the padding property. Using separate properties such as top, right, bottom, and left, different padding values can be specified and the padding can be changed separately.

Explain the table provided on the slide that lists the various values used in padding property. Also, explain the Code Snippet that demonstrates the CSS code used for specifying different padding values for different sides.

In the code, the value for padding was set for all the sides. Instead of using different padding for different sides, users can use a shorthand property. A shorthand property is one where all the padding properties for different sides are specified in one property. This will result in a shortened code. The shorthand property for all the padding properties is padding. The property can be used to specify one to four values for each of the side.

Using slide 19, explain the list of CSS properties with description given.

#### **In-Class Question:**

After you finish explaining padding, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What is shorthand property for applying padding to an element?

#### **Answer:**

A shorthand property is one where all the padding properties for the different sides are specified in one property.

#### **Slides 20 to 23**

Let us understand thumbnail graphics.

**HTML 5 Thumbnail Graphics 1-4**

- A thumbnail is a small image, or a part of a larger image.
- Clicking the thumbnail image will link to the larger original image, which can be viewed and downloaded.
- Even a hover effect can be given through CSS and JavaScript.
- Following Code Snippet demonstrates an HTML code for inclusion of a thumbnail image.

```

h1, h2{
    font-size:180%;
    font-weight: normal;
    color:#555;
}
p{
    margin:1em 0;
}
p.text{
    width:500px;
}
a{
    color:#f20;
    text-decoration:none;
}

```

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 20

**20 (a)**

**HTML**

## Thumbnail Graphics 1-4

```

h1, h2{
    font-size:180%;
    font-weight:normal;
    color:#555;
}
p{
    margin:1em 0;
}
p.text{
    width:500px;
}
a{
    color:#f20;
    text-decoration:none;
}

```

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 20

**20 (b)**

**HTML**

## Thumbnail Graphics 2-4

```

a:hover{
    color:#999;
}
img{
    border:none;
}
/* // general */
/* thumbnail list */
ul#thumbs, ul#thumbs li{
    margin:0;
    padding:0;
    list-style:none;
}
ul#thumbs li{
    float:left;
    margin-right:0px;
    border:1px solid #999;
    padding:2px;
}

```

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 21

**21 (a)**

**HTML**

## Thumbnail Graphics 2-4

```

ul#thumbs a{
    display:block;
    float:left;
    width:125px;
    height:135px;
    line-height:50px;
    overflow:hidden;
    position:relative;
    z-index:1;
}
ul#thumbs a img{
    float:left;
    position:absolute;
    top:0px;
    left:0px;
}

```

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 21

**21 (b)**

**HTML5** **Thumbnail Graphics 3-4**

```

/* mouse over */
ul#thumbs a:hover{
    overflow:visible;
    z-index:1000;
    border:none;
}
ul#thumbs a:hover img{
    border:1px solid #999;
    background:#fff;
    padding:2px;
}
/* // mouse over */
/* clearing floats */
ul#thumbs:after, li#thumbs:after{
    content:".";
    display:block;
    height:0;
    clear:both;
    visibility:hidden;
}

```

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 22

**22 (a)**

**HTML5** **Thumbnail Graphics 3-4**

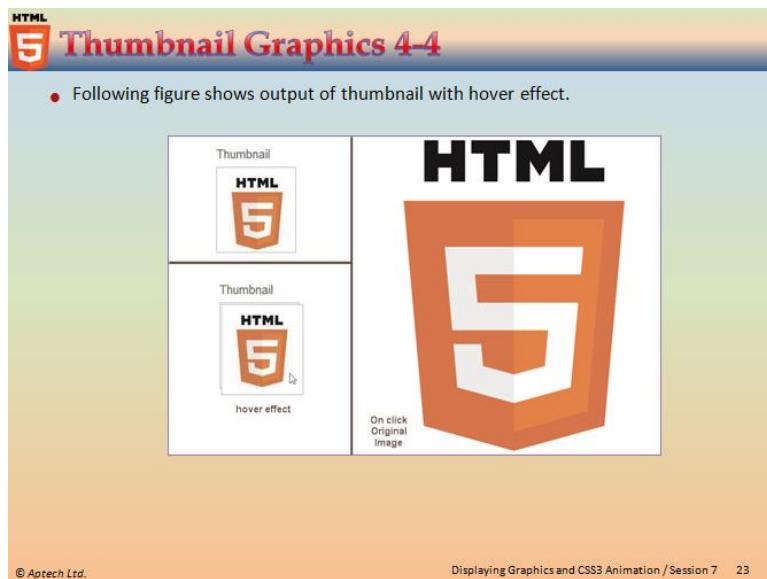
```

ul#thumbs, li#thumbs{
    display:block;
}
ul#thumbs, li#thumbs{
    min-height:1%;
}
* html ul#thumbs, * html li#thumbs{
    height:1%;
}
/* // clearing floats */
/* // thumbnail list */
</style>
</head>
<body>
    <h2>Thumbnail</h2>
    <ul id="thumbs">
        <li><a href="HTML5.png" target="_blank">
            </a></li>
    </ul>
</body> </html>

```

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 22

**22 (b)**



Using slides 20 to 23 explain the students on the use of thumbnail graphics.

The speed of loading a page of a Web site is reduced if high-resolution graphics are used. High-resolution graphics are required to improve the effectiveness of the site and cannot be avoided. Hence, to avoid this issue, thumbnails are used. A thumbnail is a small image, or a part of a larger image. Clicking thumbnail image will link to the larger original image which can be viewed and downloaded. Even a hover effect can be given through CSS and JavaScript.

Explain the CSS code snippet provided on slides 20 (a), 20 (b), 21 (a), 21 (b) slide, 22 (a), and 22 (b).

Explain the style applied to the unordered list and anchor tag. Also, the style applied to `<img>` and `<anchor>` tag further for positioning the element. Explain the heading tag of level two, unordered list and `<img>` tag

Using slide 23 to explain the output with respect to the code discussed in previous slides.

## Slides 24 to 28

Let us understand the working with CSS3 transitions.

**HTML5 Working with CSS3 Transitions 1-5**

- In 2007, Apple introduced the CSS transition, which later became a proprietary feature of Safari called CSS Animation.
- Representatives from Apple and Mozilla began adding the CSS transitions module to the CSS Level 3 specification, closely modeled on what Apple had already added to Webkit and moz.

Browsers that support CSS3 Transitions are as follows:

- Apple Safari 3.1 and later which requires the prefix –webkit–
- Google Chrome which requires the prefix –webkit–
- Mozilla Firefox 3.7 alpha and later which requires the prefix –moz–
- Opera 10.5x and later which requires the prefix –o–

- Currently, Internet Explorer 9 does not support CSS3 Transitions.

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 24

**HTML5 Working with CSS3 Transitions 2-5**

For performing CSS transitions the two required specifications are as follows:

- The CSS property that needs the effect
- The duration of the effect

- Following Code Snippet demonstrates the use of transition effect on the width property for 3 seconds.

```
div
{
    transition: width 3s;
    -moz-transition: width 3s; /* Firefox 4 */
    -webkit-transition: width 3s; /* Safari and Chrome */
    -o-transition: width 3s; /* Opera */
}
```

- The effect will start when the specified CSS property changes value.
- The CSS property changes its value typically when a user moves a mouse over an element.
- Thus, the user can set the hover for <div> elements.

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 25

**HTML5 Working with CSS3 Transitions 3-5**

- Following Code Snippet demonstrates setting the hover for <div> elements.

```
div:hover
{
    width:200px;
}
```

- Following table lists all the transition properties.

Property	Description
transition	Is a shorthand property and is used for setting the four transition properties into a single property.
transition-property	Is used for specifying the name of the CSS property for which the transition value is set.
transition-duration	Is used for defining the duration of the transition. Default value is 0.
transition-timing-function	Is used for describing how the speed during a transition will be calculated. Default value is 'ease'.
transition-delay	Is used for defining the start of the transition. Default value is 0.

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 26

**HTML Working with CSS3 Transitions 4-5**

- Following Code Snippet demonstrates an HTML and CSS code using all transition properties.

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
div
{
width:100px;
height:100px;
background:#000000;
transition-property:width;
transition-duration:2s;
transition-timing-function:linear;
transition-delay:1s;
/* Firefox 4 */
-moz-transition-property:width;
-moz-transition-duration:2s;
-moz-transition-timing-function:linear;
-moz-transition-delay:1s;
```

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 27

27 (a)

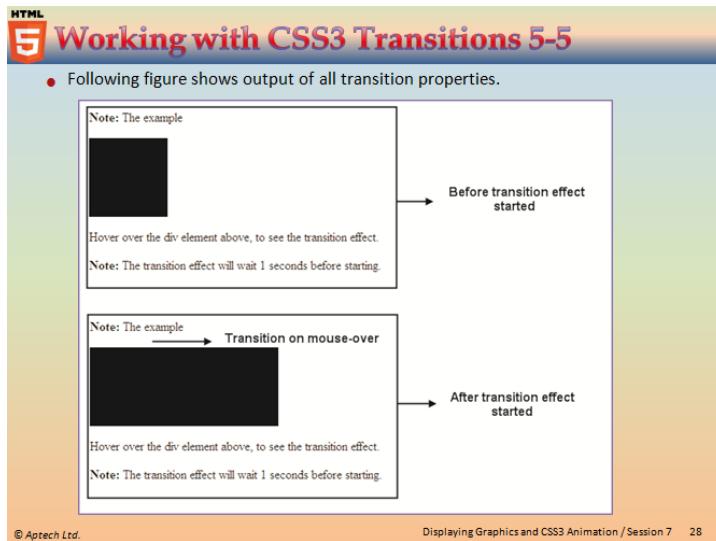
**HTML Working with CSS3 Transitions 4-5**

- Following Code Snippet demonstrates an HTML and CSS code using all transition properties.

```
/* Safari and Chrome */
-webkit-transition-property:width;
-webkit-transition-duration:2s;
-webkit-transition-timing-function:linear;
-webkit-transition-delay:1s;
}
div:hover
{
width:500px;
}
</style> </head>
<body>
<p><b>Note:</b> The example</p>
<div></div>
<p>Hover over the div element above, to see the transition
effect.</p>
<p><b>Note:</b> The transition effect will wait 1 seconds
before starting.</p>
</body></html>
```

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 27

27 (b)



Using slides 24 to 28 explain working with CSS3 transitions.

Tell the students that interactivity is one of the important aspects of animation. Earlier, a combination of HTML, CSS, and JavaScript were used to animate objects on the Web. In 2007, Apple introduced the CSS transition, which later became a proprietary feature of Safari called CSS Animation. Representatives from Apple and Mozilla began adding the CSS transitions module to the CSS Level 3 specification, closely modelled on what Apple had already added to Webkit and moz.

All browsers do not support CSS3 transitions. Browsers that support CSS3 Transitions are as follows:

- Apple Safari 3.1 and later which requires the prefix – webkit–
- Google Chrome which requires the prefix – webkit–
- Mozilla Firefox 3.7 alpha and later which requires the prefix – moz–
- Opera 10.5x and later which requires the prefix – o–

Also, Windows Internet Explorer 9 and Internet Explorer 10 add even more support for many CSS3 modules.

Using slide 25, explain the requirement specification for CSS transitions.

For performing CSS transitions the two required specifications are as follows:

- The CSS property that needs the effect
- The duration of the effect

Explain the code snippet that demonstrates the use of transition effect on the width property for three seconds.

Explain that the effect will start when the specified CSS property changes value. The CSS property changes its value typically when a user moves a mouse over an element. Thus, the user can set the hover for `<div>` elements.

Using slide 26, explain the code snippet for the `div` element and the transition properties.

Explain the list of all the transition properties.

**Tips:**

The `transition-duration` property specifies how many seconds (s) or milliseconds (ms) a transition effect takes to complete. Always specify the `transition-duration` property, otherwise the duration is 0, and the transition will have no effect.

The `transition-timing-function` property specifies the speed curve of the transition effect. This property allows a transition effect to change speed over its duration.

Using slides 27 (a) and 27 (b), explain the HTML and CSS code using all transition properties.

Explain the style tag used for the CSS code and the `<p>`, `<div>` tag used. Using slide 28 explain the output of the transition properties.

## Slide 29

Let us understand CSS3 animation.

The slide has a blue header bar with the text "HTML 5 CSS3 Animation". Below the header, there is a bulleted list: "CSS3 animations can animate transitions of one CSS style configuration to another." A callout box contains the text "The two components of animation are as follows:" with two items: "An animation style describing the animation." and "A keyframes set that specifies the start and end states of the animation's CSS style and possible intermediate waypoints along the way." Another callout box contains the text "The advantages of CSS3 animations over script-based animation techniques are as follows:" with three items: "Easy to use and anybody can create them without the knowledge of JavaScript.", "Executes well even under reasonable system load.", and "Allows the browser to control the animation sequence, optimize performance and efficiency". At the bottom of the slide, there is a footer bar with the text "© Aptech Ltd." and "Displaying Graphics and CSS3 Animation / Session 7 29".

Using slide 29, explain the CSS3 animation.

Tell the students that CSS3 animations can animate transitions of one CSS style configuration to another. The two components of animation are as follows:

- An animation style describing the animation.
- A keyframe set that specifies the start and end states of the animation's CSS style and possible intermediate waypoints along the way.

Some of the advantages of CSS3 animations over script-based animation techniques are as follows:

- Easy to use and anybody can create them without the knowledge of JavaScript.

- Executes well even under reasonable system load. As simple animations perform poorly in JavaScript, the rendering engine uses the frame-skipping techniques to allow smooth flow of animation.
- Allows the browser to control the animation sequence, optimize performance and efficiency by reducing the update frequency of animations executing in tabs that aren't currently visible.

### In-Class Question:

After you finish explaining CSS3 animation, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What is the keyframes set used for?

### Answer:

keyframes set specifies the start and end states of the animation's CSS style and possible intermediate waypoints along the way.

### Slides 30 to 35

Let us understand configuring the animation.

**HTML5 Configuring the Animation 1-6**

- A CSS animation sequence can be created by styling the element with the `animation` property.
- This property can be used to configure the timing, duration, and sequence of the animation.
- `@keyframes` rule defines the appearance of the animation.
- The keyframe is used to describe the rendering of the element in the animation sequence.
- Following table lists the `@keyframes` rule and all the animation properties.

Property	Description
<code>@keyframes</code>	Is used for specifying the animation.
<code>animation</code>	Is a shorthand property representing all the animation properties, except the <code>animation-play-state</code> property.
<code>animation-name</code>	Is used for specifying the name of the <code>@keyframes</code> animation.
<code>animation-duration</code>	Is used for specifying the duration of an animation cycle in seconds or milliseconds. Default value is 0.
<code>animation-timing-function</code>	Is used for describing the progress of animation over one cycle of its duration. Default value is 'ease'.

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 30

**HTML** **Configuring the Animation 2-6**

Property	Description
animation-delay	Is used for specifying the start value of animation. Default value is 0.
animation-iteration-count	Is used for specifying the number of times an animation is played. Default value is 1.
animation-direction	Is used for specifying whether or not the animation should play in reverse on alternate cycles. Default value is 'normal'.
animation-play-state	Is used for specifying the state of the animation, that is whether it is running or paused. Default value is 'running'.

- The syntax for @keyframes is as follows:

**Syntax:**

```
@keyframes myfirst
{
  from {background: red;}
  to {background: yellow;}
}
```

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 31

**HTML** **Configuring the Animation 3-6**

```

}
@-moz-keyframes myfirst /* Firefox */
{
from {background: red;}
to {background: yellow;}
}
@-webkit-keyframes myfirst /* Safari and Chrome */
{
from {background: red;}
to {background: yellow;}
}

● Following Code Snippet demonstrates HTML and CSS code of @keyframes rule and all the animation properties.

<!DOCTYPE html>
<html>
<head>
<style type="text/css">
```

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 32

**HTML** **Configuring the Animation 4-6**

```

div
{
width:200px;
height:200px;
background:red;
position:relative;
border-radius:100px;
animation-name:myfirst;
animation-duration:4s;
animation-timing-function:linear;
animation-delay:1s;
animation-iteration-count:infinite;
animation-direction:alternate;
animation-play-state:running;
/* Firefox: */
-moz-border-radius:100px;
-moz-animation-name:myfirst;
-moz-animation-duration:4s;
-moz-animation-timing-function:linear;
-moz-animation-delay:1s;
-moz-animation-iteration-count:infinite;
-moz-animation-direction:alternate;
-moz-animation-play-state:running;
```

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 33

33 (a)

**HTML5 Configuring the Animation 4-6**

```

/* Safari and Chrome: */
-webkit-border-radius:100px;
-webkit-animation-name:myfirst;
-webkit-animation-duration:4s;
-webkit-animation-timing-function:linear;
-webkit-animation-delay:1s;
-webkit-animation-iteration-count:infinite;
-webkit-animation-direction:alternate;
-webkit-animation-play-state:running;
}
@keyframes myfirst
{
0% {background:red; left:0px; top:0px;}
25% {background:yellow; left:300px; top:0px;}
50% {background:blue; left:300px; top:300px;}
75% {background:green; left:0px; top:300px;}
100% {background:red; left:0px; top:0px;}
@-moz-keyframes myfirst /* Firefox */
{
0% {background:red; left:0px; top:0px;}
25% {background:yellow; left:300px; top:0px;}
}

```

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 33

**33 (b)**

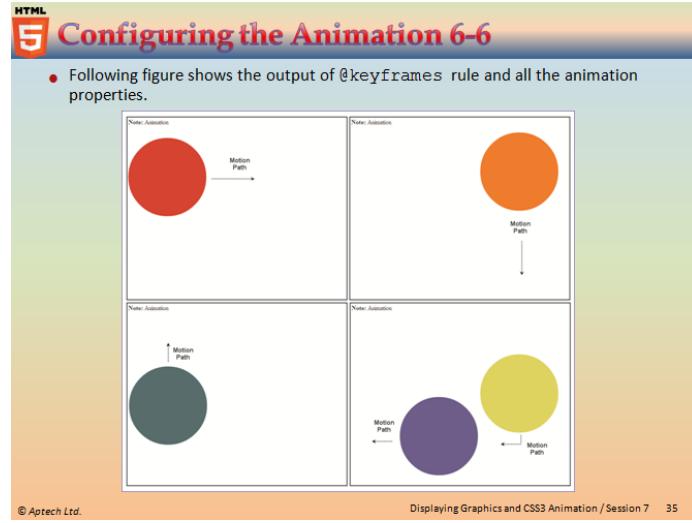
**HTML5 Configuring the Animation 5-6**

```

50% {background:blue; left:300px; top:300px;}
75% {background:green; left:0px; top:300px;}
100% {background:red; left:0px; top:0px;}
}
@-webkit-keyframes myfirst /* Safari and Chrome */
{
0% {background:red; left:0px; top:0px;}
25% {background:yellow; left:200px; top:0px;}
50% {background:blue; left:200px; top:200px;}
75% {background:green; left:0px; top:200px;}
100% {background:red; left:0px; top:0px;}
}
</style>
</head>
<body>
<p><b>Note:</b> Animation</p>
<div></div>
</body>
</html>

```

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 34



Using slides 30 to 35 explain configuring the animation.

A CSS animation sequence can be created by styling the element with the `animation` property. This property can be used to configure the timing, duration, and sequence of the animation. `@keyframes` rule define the appearance of the animation. The keyframe is used to describe the rendering of the element in the animation sequence.

Explain the table that lists the `@keyframes` rule and all the animation properties.

Mention the `@keyframes` rule is where the animation is created. Specify a CSS style inside the `@keyframes` rule and the animation will gradually change from the current style to the new style.

When an animation is created in the `@keyframe` rule, you must bind it to a selector, otherwise the animation will have no effect. Bind the animation to a selector (element) by specifying at least these two properties:

- The name of the animation
- The duration of the animation

The animation created using `@keyframes` must be bound with the selector for effective execution. For this, specify the name of the animation and the duration of the animation to the selector.

Using slide 33 (a), 33 (b), and 34 explain the HTML and CSS code that demonstrates HTML and CSS code of `@keyframes` rule and all the animation properties.

Using slide 35 explain the output of `@keyframes` rules and animation properties used.

**Tips:**

If the duration part is not specified, the animation will have no effect, because the default value is 0.

## Slides 36 and 37

Let us understand usage of CSS3 on mobile device.

**HTML 5 Using CSS3 on Mobile Devices 1-2**

- There are different ways to provide Web pages for mobile devices.
- The user can make use of style sheet for the handheld devices (all mobile browsers do not recognize it).
- iPhone's Safari and Opera's Mini browsers support a new feature of CSS3 called media queries.
- These queries allow the user to specify a conditional expression for media type.
  - Following Code Snippet shows the use of a conditional expression for displaying a link element where the maximum screen width for mobile devices is 480 pixels.
 

```
<link rel="stylesheet" href="styles/mobile.css" media="only screen and (max-device-width: 480px)"/>
```
  - The user can also specify another link element for screen media with a minimum screen width of 481 pixels.
  - In other words, the style sheet for this element can be used for standard computer screens.

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 36

**HTML 5 Using CSS3 on Mobile Devices 2-2**

- Most mobile Web sites are created to precede the domain name of the main site with m for example [m.aptech-education.com](http://m.aptech-education.com).
- To detect a mobile device, a Web site can use JavaScript on the client, a scripting language on the server, or Wireless Universal Resource File (WURFL) on the server.

The five ways to provide Web pages for mobile devices are as follows:

- Define a style sheet for mobile devices
- Include a link to a mobile version of the Web site
- Use JavaScript to detect mobile devices and redirect
- Use a server-side scripting language to detect and redirect
- Use the WURFL to detect mobile devices

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 37

Using slides 36 and 37, explain using CSS3 on mobile devices.

Explain the students that there are different ways to provide Web pages for mobile devices. The user can make use of style sheet for the handheld devices (all mobile browsers do not recognize it). iPhone's Safari and Opera's Mini browsers support a new feature of CSS3 called media queries. These queries allow the user to specify a conditional expression for media type. Code Snippet shows the use of a conditional expression for displaying a link element where the maximum screen width for mobile devices is 480 pixels.

Mention that the user can also specify another link element for screen media with a minimum screen width of 481 pixels. In other words, the style sheet for this element can be used for standard computer screens.

Mention HTML5 in combination with CSS3 is very useful, because it allows creation of user interfaces similar to native apps, without the restrictions of each platform and without the need to multiply the versions of the app for each one of them in smartphones. This way, apps are easier to develop and they can be more easily cross-platform spread.

Separate Websites must be developed for mobile devices. The home page of the main site should provide a link that connects to the mobile Website. This technique identifies the mobile device of the user and renders the mobile Website automatically in the best view possible. Most mobile Websites are created to precede the domain name of the main site with m for example m.aptech-education.com To detect a mobile device, a Web site can use JavaScript on the client, a scripting language on the server, or Wireless Universal Resource File (WURFL) on the server.

Then, explain the ways to provide Web pages for mobile devices.

### **Slides 38 to 40**

Let us understand the optimum browser compatibility.

**Optimum Browser Compatibility 1-3**

- Web browser compatibility measures are undertaken to provide predictability and consistency across the preferable Web browsers of the targeted end users.
- Cross browser compatibility means a Website that is attuned and reliable in looks, layout, color, functionality, interactivity, and proportion.
- Cross browser compatibility is across all existing Web browsers, regardless of the browsers' insignificance or popularity differences from version to version.
- Multi-browser compatibility is constant and it is functionally rendered across the most commonly used browsers in a client's target market.
- HTML5 uses different standards and is supported by various browsers. These browsers provide different version of support.
- Rendering engines are a set of tools that are used in most browsers that supports different HTML features.

© Aptech Ltd.      Displaying Graphics and CSS3 Animation / Session 7      38

**Optimum Browser Compatibility 2-3**

- Some of the rendering engines of different browsers are as follows:

<b>Gecko</b>	<ul style="list-style-type: none"> <li>The Gecko engine is the main engine of Mozilla Firefox, and a number of related browsers.</li> <li>It has support for various HTML5 features.</li> </ul>
<b>Trident</b>	<ul style="list-style-type: none"> <li>The Trident engine is used by different versions of Internet Explorer (IE).</li> <li>Currently, HTML5 is not majorly supported by the Trident engine.</li> </ul>
<b>WebKit</b>	<ul style="list-style-type: none"> <li>The WebKit engines is supported mainly for the Safari browser used in Apple Macs, iPhones, iPads, and other Apple products.</li> <li>This engine is based on the open source KHTML project.</li> </ul>
<b>Presto</b>	<ul style="list-style-type: none"> <li>Presto is the engine used in the Opera browsers.</li> <li>Opera browsers are considered to be a technically superior browser, but market share of Opera browsers is still low.</li> </ul>

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 39

**Optimum Browser Compatibility 3-3**

- The best practices for optimum browser compatibility are as follows:

<b>Test the Web site in different browsers</b>	<ul style="list-style-type: none"> <li>Review the Web site's appearance and functionality on multiple browsers to ensure that all the users are getting the same experience according to the design.</li> <li>Preferably test on different versions of the same browser also as they can show the Web site differently.</li> </ul>
<b>Write a good clean HTML code</b>	<ul style="list-style-type: none"> <li>To ensure that the page looks same in all browsers is to write Web pages using valid HTML and CSS codes, and then test it in many browsers.</li> <li>Using External CSS can help pages render and load faster.</li> </ul>

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 40

Using slides 38 to 40, explain the optimum browser compatibility.

Mention Web browser compatibility measures are undertaken by Web developers who are committed to producing Web products that provide predictability and consistency across the preferable Web browsers of the targeted end users. Cross browser compatibility means a Web site that is attuned and reliable in looks, layout, color, functionality, interactivity, and proportion across all existing Web browsers, regardless of the browsers' insignificance or popularity differences from version to version. Multi-browser compatibility is constant and it is functionally rendered across the most commonly used browsers in a client's target market. HTML5 uses different standards and is supported by various browsers. These browsers provide different version of support.

Then, explain the different rendering engines used for different browsers. Finally, conclude the explanation by describing the best practices for optimum browser compatibility.

**Tips:**

Mention some more points to be considered are as follows:

- A lot of cross-browser issues amount to this: you didn't specify something and different browsers make different assumptions.
- No site will look perfect in every browser that exists. If a user doesn't have Flash, or JavaScript, or advanced CSS, and so on. You want your site to be usable anyway.
- Try loading your site with bare HTML - no styles, no scripts. Are menu options available? Does primary content precede secondary content? Is the site usable, even if ugly?
- Avoid browser-specific markup or only use it when its failure in other browsers won't be significant to the site experience.

**Slide 41**

Let us summarize the session.

The slide has a header 'HTML 5 Summary' with a blue gradient background. Below the header is a bulleted list of nine items, each preceded by a red circular bullet point. The list describes various CSS properties and their functions. At the bottom left is the copyright notice '© Aptech Ltd.' and at the bottom right is the page number 'Displaying Graphics and CSS3 Animation / Session 7 41'.

- The text styles specify and control the appearance of the text in a Web page.
- Indenting is the process of offsetting text from its normal position, either to the left or to the right.
- CSS border property specifies the style, color, and width of the border.
- The border-color property accepts different color values that determine the different shades of color to be applied to the borders.
- The values of the different border properties determine the type of effect to be applied to the borders.
- In CSS, the text-align property is used for horizontal alignment of text in an element.
- In CSS, the line-height property is used for vertical alignment of text in an element.

© Aptech Ltd. Displaying Graphics and CSS3 Animation / Session 7 41

In slide 41, you will summarize the session. You will end the session, with a brief summary of what has been taught in the session.

### 7.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session. You should also explore the Creating Navigational Aids and Division-Based Layout that are offered with the next session.

**Tips:**

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 8 – Creating Navigational Aids and Division-Based Layout

---

## 8.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 8.1.1 Objectives

By the end of this session, the learners will be able to:

- Explain HTML5 semantic tags
- Explain HTML5 semantic tag layouts
- Explain the usage of navigation bar
- Describe a text-based and graphical navigation bar
- Explain image mapping
- Explain divisions in HTML5

### 8.1.2 Teaching Skills

To teach this session, you should be well-versed with HTML5 semantic tags. Also, the HTML5 semantic tag layouts and the usage of navigation bar should be known. Along with this, you should prepare yourself with text-based and graphical navigation bar. The session also covers the image mapping and divisions in HTML5.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

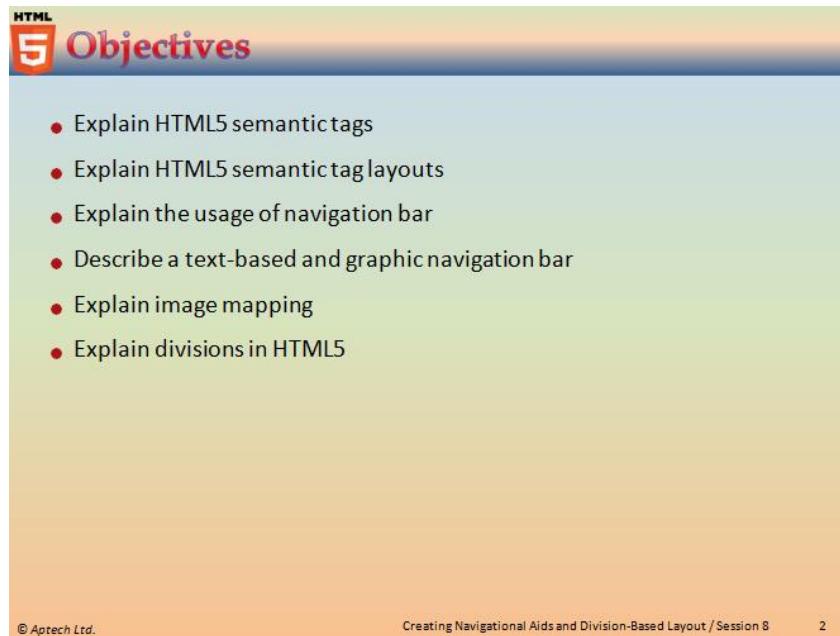
It is recommended that you test the understanding of the students by asking questions in between the class.

#### In-Class Activities:

Follow the order given here during In-Class activities.

**Overview of the Session:**

Then, give the students the overview of the current session in the form of session objectives. Show the students slide 2 of the presentation.



The slide has a blue header bar with the text "HTML5 Objectives". Below the header is a list of six bullet points:

- Explain HTML5 semantic tags
- Explain HTML5 semantic tag layouts
- Explain the usage of navigation bar
- Describe a text-based and graphic navigation bar
- Explain image mapping
- Explain divisions in HTML5

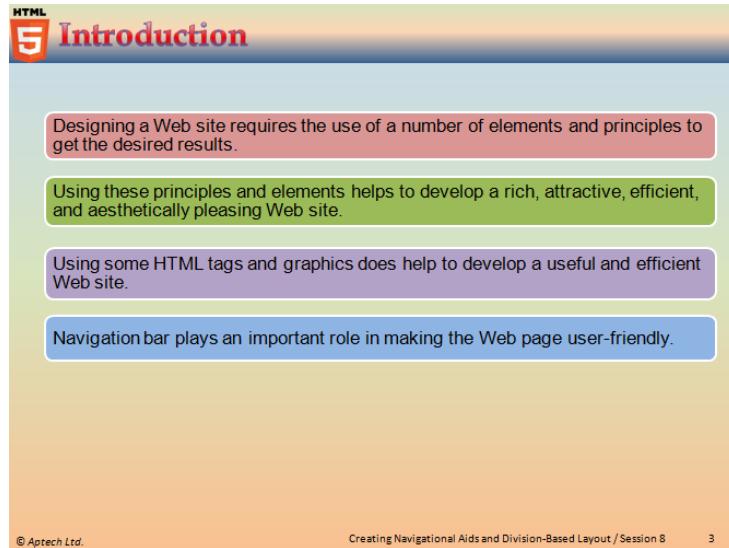
At the bottom left is the copyright notice "© Aptech Ltd.". At the bottom center is the page number "2".

Tell the students that this session introduces HTML5 semantic tags. They will learn about HTML5 semantic layout and the usage of navigation bar. They will also know about the text-based, graphical navigation bar, and divisions in HTML5.

## 8.2 In-Class Explanations

### Slide 3

Let us understand creating navigational aids and division-based layout.

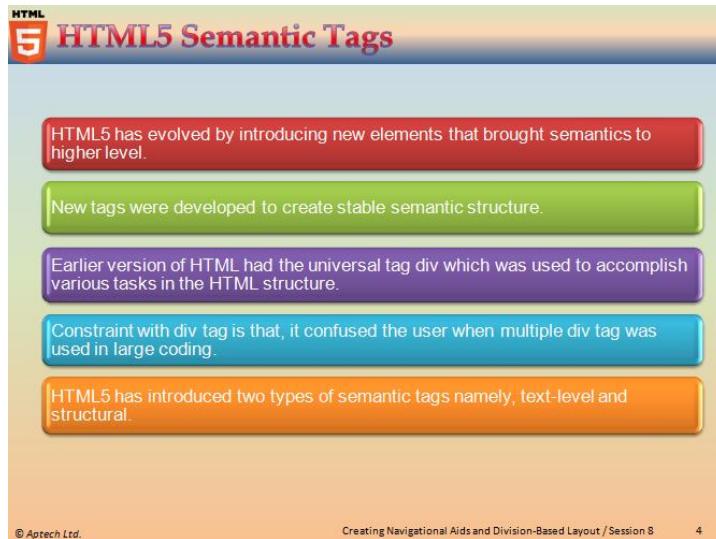


Using slide 3, explain the introduction of creating navigational aids and division-based layout.

For designing a Website, a number of elements and principles are used to get the desired results. Using these principles and elements helps to develop a rich, attractive, efficient, and aesthetically pleasing Website. In other words, using some HTML tags and graphics does help to develop a useful and efficient Website. Navigation bar plays an important role in making the Web page user-friendly.

## Slide 4

Let us understand the HTML5 semantic tags.



Using slide 4, explain the HTML5 semantic tag.

Mention HTML5 has evolved by introducing new elements that brought semantics to higher level. New tags were developed to create stable semantic structure. The earlier version of HTML had the universal tag div which was used to accomplish various tasks in the HTML structure.

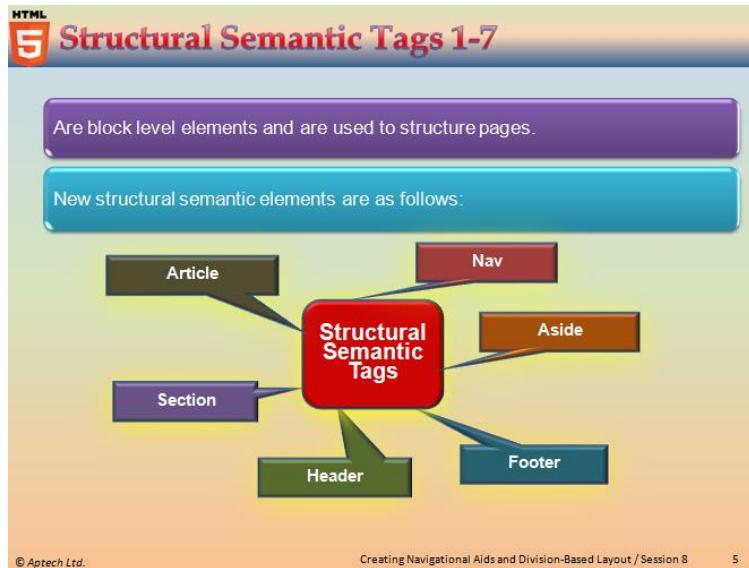
HTML5 has introduced two types of semantic tags. They are namely, text-level and structural.

### Tips:

The semantic element clearly describes its meaning to both the browser and the developer. Examples of **non-semantic** elements are `<div>` and `<span>`. These tells nothing about its content. Examples of **semantic** elements are `<form>`, `<table>`, and `<img>`. These clearly defines its content.

## Slide 5

Let us understand the structural semantic tags.



Using slide 5 explain the structural semantic tags.

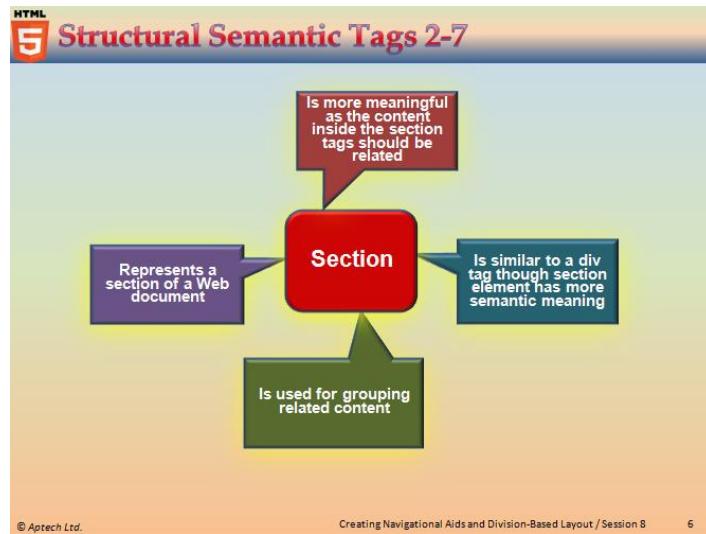
They are the block level elements and are used to structure pages. The new structural semantic elements are as follows:

- Section
- Header
- Footer
- Aside
- Nav
- Article

All these tags are newly introduced in HTML5.

## Slide 6

Let us understand the section semantic tag.

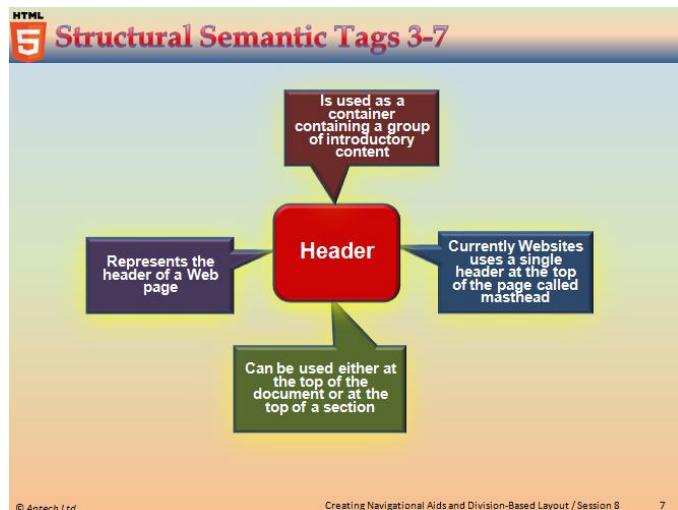


Use slide 6 to explain the section element. Mention section element represents a section of a Web document. It is used for grouping related content and is different from other content groups present on the Web page. It is similar to a div tag though section element has more semantic meaning. In other words, section element is more meaningful as the content inside the section tags should be related.

Mention <section> tag defines sections in a document such as chapters, headers, footers, or any other sections of the document.

## Slide 7

Let us understand the header semantic tag.



Use slide 7 to explain the header element.

Mention `header` element represents the header of a Web page. It can be used either at the top of the document or at the top of a section. Though most of the Websites currently uses a single header at the top of the page called masthead, but a Web developer can have multiple headers in a single HTML5 document. This element is used as a container containing a group of introductory content or a set of navigational links.

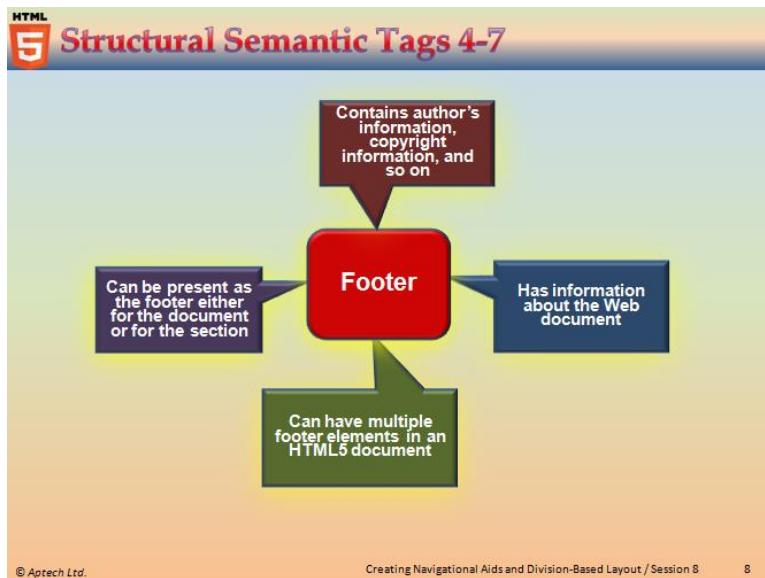
Mention `<header>` tag specifies a header for a document or section. The `<header>` element should be used as a container for introductory content or set of navigational links. There can be several `<header>` elements in one document.

### Tips:

A `<header>` tag cannot be placed within a `<footer>`, `<address>`, or another `<header>` element.

## Slide 8

Let us understand the `footer` semantic tag.



Using slide 8, explain the `footer` element. Mention `footer` is similar to `header` and can be present as a footer either for the document or for the section. There can be multiple footer elements in an HTML5 document. A footer element has information about the Web document.

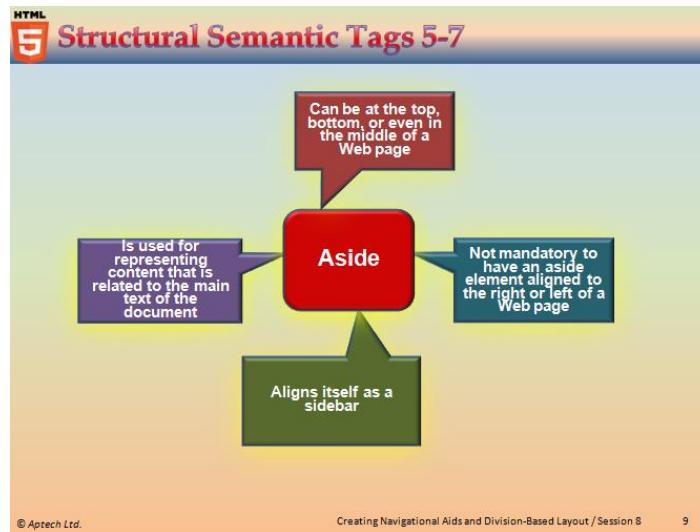
The typical contents which are placed in footer are as follows:

- Authors information Copyright information
- Text-based navigation bar

Any metadata for the section can also be included in a footer tag.

## Slide 9

Let us understand the `aside` semantic tag.



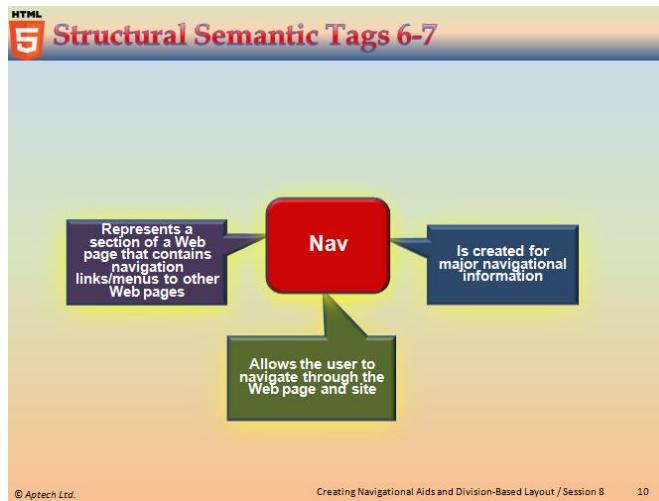
Using slide 9 explain the `aside` element. The `aside` element is used for representing the content that is related to the main text of the document. It aligns itself as a sidebar. As compared with other structural tags its importance is not related with its position within a document, but rather its relationship with the content. It is not mandatory to have an `aside` element aligned to the right or left of a Web page. It can be at the top, bottom, or even in the middle of a Web page.

### Tips:

The `<aside>` content could be placed as a sidebar in an article.

## Slide 10

Let us understand the `nav` semantic tag.



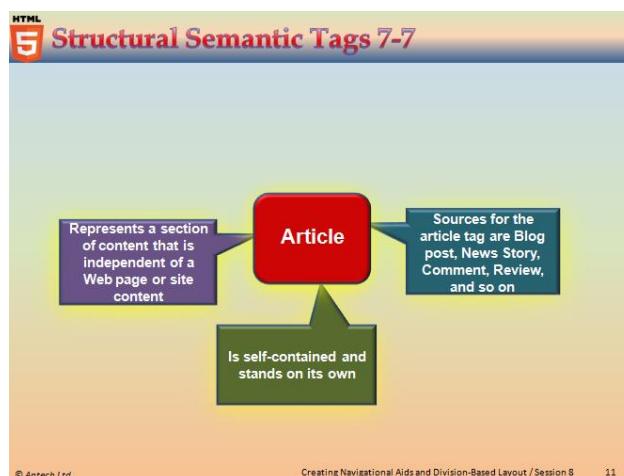
Using slide 10, explain the `nav` element. The `nav` element represents a section of a Web page that contains navigation links/menus to other Web pages or to other parts within the Web page. In other words, it allows the user to navigate through the Web page and site. This section is created for major navigational information such as a navigation bar for the entire site or for a subsection menu.

Mention that NOT all links of a document should be inside a `<nav>` element. The `<nav>` element is intended only for major block of navigation links.

Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

## Slide 11

Let us understand the `article` semantic tag.



Using slide 11, explain the `article` element.

The `article` element represents a section of content that is independent of a Web page or site content. It is self-contained and stands on its own. The possible sources for the `article` tag are as follows:

- Blog post
- News story
- Comment
- Review
- Forum post

### In-Class Question:

After you finish explaining structural semantic tags, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What is `header` element used for?

### Answer:

`Header` element is used as a container containing a group of introductory content or a set of navigational links and can be used at the top of the document or top of the section.

### Slides 12 to 18

Let us understand text-level semantic tags.



**HTML 5 Text-level Semantic Tags 2-7**

<mark> tag is used for defining marked or highlighted text.

<mark> tag can be used for highlighting words in a Web page that a visitor searched for.

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 13

**HTML 5 Text-level Semantic Tags 3-7**

<time> tag is used for defining either the time, or a date in the Gregorian calendar.

Can be used to encode dates and times in a machine-readable format and is used optionally with a time and a time-zone offset.

- Following table lists attribute and value of <time> tag.

Attribute	Value	Description
datetime	datetime	Provides the date/time given by the element's content.
pubdate	pubdate	It is used for specifying publication date and time of the document.

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 14

**HTML 5 Text-level Semantic Tags 4-7**

- The Code Snippet demonstrates how to display the date and time.

```
<!DOCTYPE html>
<html>
  <body>
    <time datetime="13:00">1pm</time>
    <time datetime="2011-07-15">July 15th, 2011</time>
    <time datetime="2011-07-15T13:00">1pm on July 14th</time>
  </body>
</html>
```

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 15

**HTML 5 Text-level Semantic Tags 5-7**

- <meter> tag displays markup or scalar measurement within a defined range.
- Absolute scalar values, such as height or weight, are not represented automatically by the meter tag.
- It is also used for displaying fractional value.

- The Code Snippet demonstrates how to display the <meter> tag.

```
<meter value="2" min="0" max="10">2 out of 10</meter>
```

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 16

**HTML 5 Text-level Semantic Tags 6-7**

- Following table lists attribute and value of <meter> tag.

Attribute	Value	Description
form	form_id	Is used for specifying one or more forms that <meter> element belongs to.
high	number	Is used for specifying the high range value.
low	number	Is used for specifying a range of value that is to be considered as low and should be greater than min attribute value.
max	number	Is used for specifying the maximum value of the range.
min	number	Is used for specifying the minimum value of the range.
optimum	number	Is used for specifying the optimal value for the <meter> tag.
value	number	Is used for specifying the current value of the <meter> tag.

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 17

**HTML 5 Text-level Semantic Tags 7-7**

- <progress> tag can be used with JavaScript to display the progress of a task.
- Following table lists attribute and value of <progress> tag.

Attribute	Value	Description
max	number	Is used for specifying the work as a floating point number that the task requires in total.
value	number	Is used for specifying how much task has been completed.

- The Code Snippet demonstrates how to display the <progress> tag.

```
<progress value="24" max="120"></progress>
```

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 18

Use slides 12 to 18 to explain the text-level semantic tags.

The text level semantic tags are currently inline elements and they are as follows:

- Mark
- Time
- Meter
- Progress

Mention all these tags are introduced in HTML5.

Using slide 13 explain the `<mark>` tag. Mention, `<mark>` tag is used for defining marked or highlighted text because of its relevance to the context. For example, a mark tag can be used for highlighting words in a Web page that a visitor searched for.

Using slides 14 and 15 explain the `<time>` tag. The `<time>` tag is used for defining either the time or a date in the Gregorian calendar. It is used optionally with a time and a time-zone offset. This element can be used to encode dates and times in a machine-readable format. For example, a Web user can add birthday reminders or scheduled events to the user's calendar and enable the search engines to produce better search results.

Explain the lists of attribute and value of `<time>` tag in table. Also, explain the code snippet which displays the date and time.

Using slide 16, explain the `<meter>` tag. The `<meter>` tag displays markup or scalar measurement within a defined range. Absolute scalar values, such as height or weight, are not represented automatically by the `meter` tag. For this, the user must specify the height and weight within the known range of values. It is also used for displaying fractional value.

Also, explain the code snippet for the `<meter>` tag.

Using slide 17, to explain the list of attribute of `meter` tag with value and description.

Using slide 18, explain the `<progress>` tag. The `<progress>` tag can be used with JavaScript to display the progress of a task. Explain the table which lists the attributes and value of `<progress>` tag.

#### In-Class Question:

After you finish explaining text-level semantic tags, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



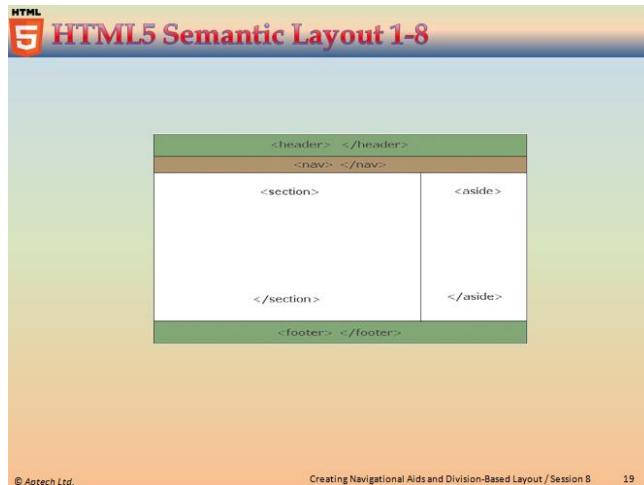
Which attribute of `<meter>` tag is used to specify the current value for it?

#### Answer:

Value attribute of `<meter>` tag is used to specify the current value for it.

## Slides 19 to 26

Let us understand the HTML5 semantic layout.



© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 19

This slide focuses on the `<header>` element. It defines the element as providing introductory information and states that it is used only for the body of the Web page or for sections inside the body. A code snippet demonstrates the use of the `<header>` tag:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>My First Page</title>
</head>
<body>
<header>
<h1>Sample Blog </h1>
</header>
</body>
</html>
```

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 20

This slide focuses on the `<nav>` element. It defines it as a section containing navigation links and notes that navigational elements help identify large blocks of navigational data. A code snippet demonstrates the use of the `<nav>` tag:

```
<body>
<header>
<h1>Sample Blog</h1>
</header>
<nav>
<ul>
<li> home </li>
<li> help </li>
<li> contact </li>
</ul>
</nav>
</body>
```

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 21

**HTML5 Semantic Layout 4-8**

<section> is the main information bar that contains the most important information of the document.

It can be created in different formats. For example, it can be divided into several blocks or columns.

- The Code Snippet demonstrates the use of <section> tag.

```

<body>
  <header>
    <h1>Sample Blog </h1>
  </header>
  <nav>
    <ul>
      <li> home </li>
      <li> help </li>
      <li> contact </li>
    </ul>
  </nav>
  <section>
    <h1>Links</h1>
    <ul>
      <li><a href="#">Link 1</a></li>
      <li><a href="#">Link 2</a></li>
      <li><a href="#">Link 3</a></li>
    </ul>
  </section>
</body>

```



© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 22

**HTML5 Semantic Layout 5-8**

<aside> element is a column or a section that generally contains data linked to the main information.

This element is used for typographical effects, such as for sidebars, for groups of nav elements, for advertising purposes, and so on.

- The Code Snippet demonstrates the use of <aside> tag.

```

<!DOCTYPE html>           <section>
<html lang="en">          <h1>Links</h1>
  <body>                <ul>
    <header>              <li><a href="#">Link 1</a></li>
    <h1>Sample Blog </h1>  <li><a href="#">Link 2</a></li>
    </header>            <li><a href="#">Link 3</a></li>
    <nav>                </ul>
      <ul>
        <li> home </li>
        <li> help </li>
        <li> contact </li>
      </ul>
    </nav>                </section>
    <aside>
      <blockquote>Archive Number One</blockquote>
      <br>
      <blockquote>Archive Number Two</blockquote>
    </aside>
  </body>
</html>

```



© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 23

**HTML5 Semantic Layout 6-8**

<footer> element give an end to the document's body.

A footer typically contains information about the sections.

Can include the author or company details, links to related documents, copyright data, and so on.

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 24

**HTML5 Semantic Layout 7-8**

- The Code Snippet demonstrates the use of `<footer>` tag.

```

<body>
  <header>
    <h1>Sample Blog</h1>
  </header>
  <nav>
    <ul>
      <li> home </li>
      <li> help </li>
      <li> contact </li>
    </ul>
  </nav>
  <section>
    <h1>Links</h1>
    <ul>
      <li><a href="#">Link 1</a></li>
      <li><a href="#">Link 2</a></li>
      <li><a href="#">Link 3</a></li>
    </ul>
  </section>
  <aside>
    <blockquote>Archive Number One</blockquote>
    <br>
    <blockquote>Archive Number Two</blockquote>
  </aside>
  <footer>
    Copyright &copy; 2012-2013
  </footer>
</body>
</html>

```

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 25

**HTML5 Semantic Layout 8-8**

`<article>` element helps to insert a self-contained composition in an application, page, document, or site.

- The Code Snippet demonstrates the use of `<article>` tag.

```

<body>
  <header>
    <h1>Sample Blog</h1>
  </header>
  <nav>
    <ul>
      <li> home </li>
      <li> help </li>
      <li> contact </li>
    </ul>
  </nav>
  <section>
    <article>
      First Blog entry
    </article>
    <article>
      Second Blog entry
    </article>
  </section>
  <aside>
    <blockquote>Archive Number One</blockquote>
    <br>
    <blockquote>Archive Number Two</blockquote>
  </aside>
  <footer>
    Copyright &copy; 2012-2013
  </footer>
</body>
</html>

```

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 26

Using slides 19 to 26 explain the various elements that are used for creating semantic layout on the Web page.

The `<header>` element provides introductory information. This information can include titles, subtitles, logos, and so on. It can also include the navigational aids.

The `<head>` tag provides information about the entire document, whereas the `<header>` tag is used only for the body or the sections inside the body of the Web page.

Then, explain the code snippet that demonstrates the use of `<header>` tag.

In the code, the `<header>` element shows the commencement of the body. This is the visible part of the document. Inside the `<header>`, the `<h1>` element is used to indicate the importance of the heading.

Explain attributes of `<nav>` tag with code snippet. The `nav` element is a section which contains the links to other pages or links to different sections within the page. In other words, it is a section containing the navigation links.

Navigational elements are helpful in large blocks of navigational data and are generally not preferred for small navigational displays. Explain the code snippet that demonstrates the use of `<nav>` tag.

In the code, the `<nav>` element is present between the `<body>` tags, but after the closure of `<header>` tag.

Using slide 22 explain the `<section>` element and its usage. It is the main information bar that contains the most important information of the document and it can be created in different formats. For example, it can be divided into several blocks or columns. For example, a Website's home page could be divided into sections for an introduction, news updates, and contact information.

Explain the code snippet that demonstrates the use of `<section>` tag.

Similar to navigation bar, the main information bar is a separate section. Therefore, the main information bar appears after the `</nav>` closing tag.

Using slide 23 explain the `<aside>` tag. Explain the code snippet that demonstrates the use of `<aside>` tag.

The `<aside>` element can be placed in any part of the site layout. It can also be used in any way as long as the content is not considered as the main content of the document.

Using slides 24 and 25, explain the `<footer>` tag. HTML5 provides the `<footer>` element to give an end to the document's body. Footer typically contains information about the sections. This can include the author or company details, links to related documents, copyright data, and so on.

Explain the code snippet that demonstrates the use of `<footer>` tag.

Using slide 26 explain the `<article>` tag. The `<article>` element could be an interactive widget, an entry in a blog, an article in a newspaper or magazine, a post in a forum, a comment submitted by a user, or any other independent content.

Explain the code snippet that demonstrates the code for `<article>` tag. In the code, the `<article>` tag are placed individually one after another because each one is an independent part of the `<section>`.

## Slide 27

Let us understand the navigation bar.

The slide has a blue header bar with the title 'Navigation Bar'. Below the header, there are six colored boxes containing text:

- Is one of the most important elements in Web design.
- Web-layouts do not have any specific physical representation except for a consistent navigation menu.
- Navigation is one segment of a Web site's information architecture.
- In Web designing, navigation menu are always on navigation bars, which can be horizontal or vertical.
- Navigation bar is a section of a Website or online page intended to support visitors in browsing through the online document.
- Web pages will have a primary and a secondary navigation bar on all pages which will include links to the most important sections of the site.

Using slide 27, explain the navigation bar.

Navigation is one of the most important elements in Web design. Web-layouts do not have any specific physical representation that a user can depend on except for a consistent navigation menu. It is one of the most important design elements which provide the users with a sense of orientation and guide them through the Website. Thus, it can be said that navigation is one segment of a Website's information architecture, but it plays an important role as it is the most visible segment to the end user.

Mention in Web designing, a navigation menu is always on navigation bars, which can be horizontal or vertical. A navigation bar is a section of a Website or online page intended to support visitors in browsing through the online document. Typically, Web pages will have a primary and a secondary navigation bar on all pages which will include links to the most important sections of the Website.

## Slides 28 and 29

Let us understand the text-based navigation bar.

The slide has a title 'Text-based Navigation Bar 1-2' with an 'HTML 5' icon. It contains six colored callout boxes:

- Red box: Some users browse Web site with graphics turned off, or use browsers with minimum graphics capability.
- Green box: For such situations, it is essential to provide text-based navigation bars which are created as stand-alone navigation bars.
- Purple box: Text-based navigation bars are not associated with icons but are easy to create, and can be displayed in any Web browsers.
- Cyan box: Advantage of using a text-based navigation bar is that it reduces the loading time of a page.
- Orange box: Although a text-based navigation bar is easy to create but is not interesting, since there is very less interaction or visual appeal to the visitor.
- Red box: It can be displayed either horizontally or vertically.

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 28

The slide has a title 'Text-based Navigation Bar 2-2' with an 'HTML 5' icon. It contains a bullet point and a code snippet:

- The Code Snippet demonstrates the HTML code for a text-based navigation bar.

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <nav>
      <a href="/home/"><font size="6">Home</font></a> | 
      <a href="/news/"><font size="6">News</font></a> | 
      <a href="/contact/"><font size="6">Contact</font></a> | 
      <a href="/about/"><font size="6">About</font></a>
    </nav>
    <h1>This is a Text-based Navigation Bar</h1>
  </body>
</html>
```

The preview shows a horizontal navigation bar with four links: Home, News, Contact, and About. Below the bar, the text "This is a Text-based Navigation Bar" is displayed.

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 29

Using slides 28 and 29, explain the text-based navigation bar.

Navigation menu is the most used element than any other element on any Web page. Therefore, it is important to make sure that the Website visitors should be able to easily navigate through the site structure.

Some users browse Website with graphics turned off, or use browsers with minimum graphics capability. For such situations, it is essential to provide text-based navigation bars which are created as stand-alone navigation bars. The developer in addition can also provide a graphical bar.

Text-based navigation bars are not associated with icons, but are easy to create and can be displayed in any Web browsers. The advantage of using a text-based navigation bar is that it reduces the loading time of a page. Although a text-based navigation bar is easy to create, it is not interesting because there is very less interaction or visual appeal to the visitor. Text

links are hard to distinguish from the regular text that appears on a Web page. It can be displayed either horizontally or vertically. The font (best to use Web safe fonts), color, and link colors can be determined by the user through the font pane.

Explain the code snippet and output for the navigation bar.

`<A>` tags are used in the navigation bar, but due to use of `<nav>` element they are grouped.

The `href` attribute has the URL address of the Web page which should be visited on clicking the hyperlink.

### In-Class Question:

After you finish explaining text-based navigation, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which element is used for creating the navigation bar in HTML5?

### Answer:

`Nav` element.

### Slide 30

Let us understand the graphical navigation bar.

**HTML 5 Graphical Navigation Bar**

- | Is more captivating than text-based navigation bar as it uses icons.
- | Increases the usability of the page with a good choice of icon for the navigation bar.
- | Also, makes the Web site more noticeable for the user visiting the Web site.
- | Disadvantage is that, since it uses images, it takes longer time for a page to load.
- | Also, the Web page will be useless for users using a non-graphic browser.

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 30

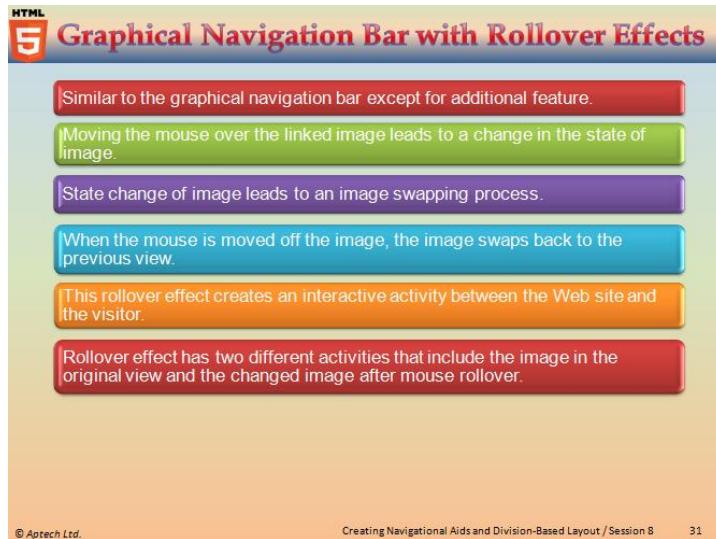
Using slide 30 explain graphical navigation bar.

Graphical navigation bar is more captivating than text-based navigation bar as it uses icons. The usability of the page increases with a good choice of icon for the navigation bar. It can

also make the Website more noticeable for the user visiting the Website. In other words, graphical navigation bar is better than text-based navigation as it gives a visual appeal to the visiting users. The only disadvantage is that, since it uses images, it takes longer time for a page to load. Also, the Web page will be useless for users using a non-graphic browser.

### Slide 31

Let us understand the graphical navigation bar with rollover effects.



Using slide 31, explain the graphical navigation bar with rollover effects.

Mention that, they are similar to the graphical navigation bar except for the additional feature. Moving the mouse over the linked image leads to a change in the state of image. In other words, the state change of image leads to an image swapping process. When the mouse is moved off the image, the image swaps back to the previous view. This rollover effect creates an interactive activity between the Website and the visitor. This rollover effect has two different activities that include the image in the original view and the changed image after mouse rollover.

## Slides 32 to 34

Let us understand the image map.

**HTML 5 Image Map 1-3**

- Are images with clickable areas.
- Areas in image-maps when clicked will link to another page.
- Have to be used intelligently to make it effective.
- Uses the <map> tag to define an image-map.
- <map> element contains a number of <area> elements for defining the clickable areas in the image map.
- The id attribute of the <map> tag when specified, must have the same value as the name attribute.

© Aptech Ltd.

Creating Navigational Aids and Division-Based Layout / Session 8

32

**HTML 5 Image Map 2-3**

- Following table lists the <map> tag attribute and its value.

Attribute	Value	Description
name	mapname	It is used for specifying the name of an image-map.

**Guidelines to create an image map:**

- Use the <img> tag to insert and link an image. In the <img> tag, use the usemap attribute to define the image map name.
- Use the <map> tag to create a map with the same name. Inside this <map> tag, define the clickable areas with the <area> tag.

© Aptech Ltd.

Creating Navigational Aids and Division-Based Layout / Session 8

33

The slide is titled "Image Map 3-3". It contains the following text:

- The Code Snippet demonstrates the use of image map in an HTML code.

```
<!DOCTYPE html>
<html>
  <body>
    
    <map name="cakemap">
      <area shape="circle" coords="0,0,200,600" href="4.html"
      alt="cake" />
    </map>
  </body>
</html>
```

Below the code is a screenshot of a web browser. The main window shows a blue background with a white circle labeled "Clickable Area". A smaller window titled "About" is open, displaying a graphical navigation bar with links for Home, About, Contact, and More.

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 34

Using slides 32 to 34 explain the image map concept.

Image maps are images with clickable areas. These areas in image-maps when clicked will link to another page. The image maps have to be used intelligently to make it effective. If they are not used appropriately they can confuse the users. The `<map>` tag is used to define an image-map. The `<map>` element contains a number of `<area>` elements for defining the clickable areas in the image map. In HTML5, if the id attribute of the `<map>` tag is specified, then it must have the same value as the name attribute.

Using slide 33 explain the attribute of the `<map>` tag.

#### **Tips:**

The `usemap` attribute in the `<img>` tag is associated with the `<map>` element's name attribute and creates a relationship between the image and the map.

Then, explain the guidelines for creating an image map. Explain the code used for demonstrating the image map.

Mention that in the code, `shape` attribute of `area` tag can have `rect` and `poly` as value based on which the `corords` attribute value will vary.

As these shapes are links, the attributes such as `target`, `media`, and `download` can also be associated with it.

Mention, that images with multiple click areas is known as hotspot.

#### **In-Class Question:**

After you finish explaining image map, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which attribute is used to define the image map name?

### Answer:

`usemap` attribute is used to define the image map name.

### Slides 35 and 36

Let us understand the divisions.

**HTML Divisions 1-2**

- <div> tag defines a division in an HTML Web page.
- Is used to group block-elements and format them with CSS.
- New structural semantic tags reasonably reduce a lot of <div> tag's usage.
- <div> tag can be used when there is no other semantically appropriate element left that suits the purpose in a Web page development.
- It can be commonly used for stylistic purposes such as wrapping some semantically marked-up content in a CSS-styled container.

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 35

**HTML Divisions 2-2**

- The Code Snippet demonstrates the use of <div> tag.

```
<body>
  <div id="wrapper">
    <header>
      <h1>Hello</h1>
      <nav>
        <! -- ... -->
      </nav>
    </header>
  </div>
</body>
```

- Tips for using <div> tag in Web site development are as follows:
  - <div> tag is a block-level element
  - <div> tag can contain any other tag
  - <div> tag can be found inside any element that can contain flow elements, such as other <div>, <address>, <section>, and <table>

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 36

Using slide 35, explain the division in HTML page or <div> tag.

The `<div>` tag defines a division in an HTML Web page. It is used to group block-elements and format them with CSS. The new structural semantic tags reasonably reduce a lot of `<div>` tag's usage, but `<div>` tag is still important in the HTML5 world.

The `<div>` tag can be used when there is no other semantically appropriate element left that suits the purpose in a Web page development. It can be commonly used for stylistic purposes such as wrapping some semantically marked-up content in a CSS-styled container.

Using slide 36, explain the HTML code for `<div>` tag and tips for using it.

Some of the features of `<div>` tag in Website development are as follows:

- The `<div>` tag is a block-level element.
- The `<div>` tag can contain any other tag and can also be nested.
- In HTML5, the `<div>` tag can be found inside any element that can contain flow elements, such as other `<div>`, `<address>`, `<section>`, and `<table>`.

### Slides 37 and 38

Let us understand division positioning and formatting.

**HTML5 Division Positioning and Formatting 1-2**

- Elements can be positioned using the top, bottom, left, and right properties.
- These properties will not work unless the position of the property is set.
- There are five position properties in DIV elements namely, static, relative, absolute, fixed, and inherit.
- Only three properties are used namely, absolute, relative, and fixed.
- Positioning can be applied to any block element.
- Default position for a block element (DIV) is static.

© Aptech Ltd. Creating Navigational Aids and Division-Based Layout / Session 8 37

The slide has a header 'HTML 5 Division Positioning and Formatting 2-2'. Below the header is a bullet point: 'Following table shows the values and its description used in DIV element.' A table follows:

Relative Length	Description
static	Positions the element in order, as they appear in the document flow. It is the default value.
absolute	Positions the element relative to its first position.
fixed	Positions the element relative to the browser window.
relative	Positions the element relative to its normal position.
inherit	Positions the element with respect the value that is inherited from the parent element.

At the bottom of the slide are the copyright notice '© Aptech Ltd.', the page title 'Creating Navigational Aids and Division-Based Layout / Session 8', and the page number '38'.

Using slides 37 and 38, explain the division positioning and formatting.

Elements can be positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position of the property is set. They also work differently depending on the positioning method. There are five position properties in DIV elements namely, static, relative, absolute, fixed, and inherit. For easy usage, only three properties are used namely, absolute, relative, and fixed. Positioning can be applied to any block element. The default position for a block element (DIV) is static.

Explain the table provided on slide 38 that shows the various values that can be used in DIV element to position elements.

#### Tips:

The <div> element is very often used together with CSS.

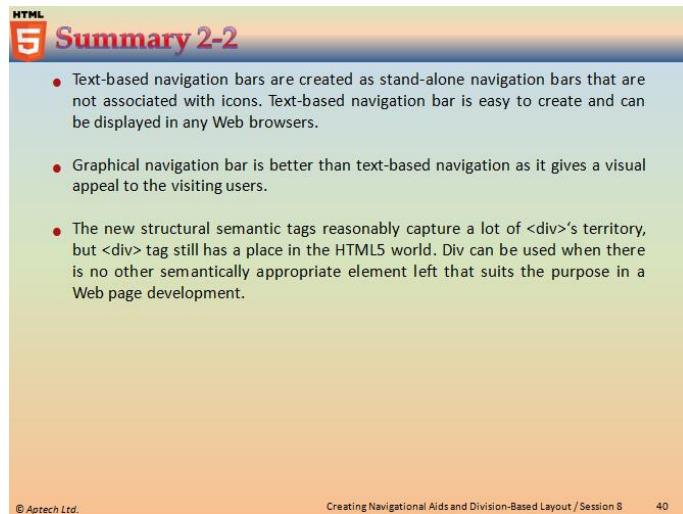
#### Slides 39 and 40

Let us summarize the session.

The slide has a header 'HTML 5 Summary 1-2'. Below the header is a bulleted list:

- HTML 5 has introduced two types of semantic tags. They are namely, text-level and structural. Structural semantic tags are as follows:
  - Section
  - Header
  - Footer
  - Aside
  - Nav
  - Article
- Text level semantic tags are as follows:
  - Mark
  - Time
  - Meter
  - Progress
- Navigation is the most significant element in Web design. Since Web-layouts does not have any physical representation, a user can depend on consistent navigation menu.

At the bottom of the slide are the copyright notice '© Aptech Ltd.', the page title 'Creating Navigational Aids and Division-Based Layout / Session 8', and the page number '39'.



The slide has a blue header bar with the text "HTML5" and "Summary 2-2". The main content area has a yellow gradient background. It contains a bulleted list of three items:

- Text-based navigation bars are created as stand-alone navigation bars that are not associated with icons. Text-based navigation bar is easy to create and can be displayed in any Web browsers.
- Graphical navigation bar is better than text-based navigation as it gives a visual appeal to the visiting users.
- The new structural semantic tags reasonably capture a lot of <div>'s territory, but <div> tag still has a place in the HTML5 world. Div can be used when there is no other semantically appropriate element left that suits the purpose in a Web page development.

At the bottom left is the copyright notice "© Aptech Ltd." and at the bottom right are the page numbers "Creating Navigational Aids and Division-Based Layout / Session 8" and "40".

In slides 39 and 40, you will summarize the session. You will end the session, with a brief summary of what has been taught in the session.

### 8.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session. You should also explore the Creating Tables that are offered with the next session.

#### Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 9 – Creating Tables

---

## 9.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 9.1.1 Objectives

By the end of this session, the learners will be able to:

- Describe how to create and format tables
- Explain the table size and the width of a column
- Explain the process of merging table cells
- Explain the page layout for tables

### 9.1.2 Teaching Skills

To teach this session, you should be well-versed with the process of creating and formatting tables. Also, you should aware yourself with the attributes of the table element, such as size and width of the columns, celling padding, applying borders, and so on. Along with this, prepare yourself with the concept of merging table cells. Also, visit some Web sites to understand how to use tables as page layout for the Web pages.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

#### In-Class Activities:

Follow the order given here during In-Class activities.

## Overview of the Session:

Then give the students the overview of the current session in the form of session objectives. Show the students slide 2 of the presentation.

**Objectives**

- Describe how to create and format tables
- Explain the table size and the width of a column
- Explain the process of merging table cells
- Explain the page layout for tables

© Aptech Ltd. Creating Tables / Session 9 2

Tell the students that this session introduces process of creating and formatting tables. They will learn about how to resize the table and adjust column width. They will also know about the process of merging table cells and choosing tables as page layouts for the Web pages.

## 9.2 In-Class Explanations

### Slides 3 to 5

Let us understand how to create and format tables.

**Creating and Formatting Tables 1-3**

- A table is made up of rows and columns. The intersection of each row and column is called as a cell.
- A row is made up of a set of cells that are placed horizontally.
- A column is made up of set of cells that are placed vertically.
- The user can represent the data in a tabular format by using the `<table>` element in HTML.
- The `<tr>` element divides the table into rows and the `<td>` element specifies columns for each row.
- By default, a table does not have a border.
- The `border` attribute of the `<table>` element specifies a border for making the table visible in a Web page.

© Aptech Ltd. Creating Tables / Session 9 3

**HTML 5 Creating and Formatting Tables 2-3**

- The Code Snippet demonstrates how to create a table.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Languages</title>
  </head>
  <body>
    <h2>Main Languages</h2>
    <table border="1">
      <tr>
        <td>English</td>
        <td>German</td>
      </tr>
      <tr>
        <td>French</td>
        <td>Italian</td>
      </tr>
    </table>
  </body>
</html>
```

© Aptech Ltd. CreatingTables / Session 9 4

**HTML 5 Creating and Formatting Tables 3-3**

- The code uses the `<table>` element to create a table.
- The `border` attribute of `<table>` element gives a border to the table, which is 1 pixel wide.
- The `<tr>` element within the `<table>` element creates rows.
- The `<td>` element creates two cells with the values English and German in the first row and French and Italian in the second row.
- Following figure displays the table created.

© Aptech Ltd. CreatingTables / Session 9 5

Using slides 3 to 5, explain the process of creating and formatting tables.

Mention that table is made up of rows and columns. The intersection of each row and column is called as a cell. A row is made up of a set of cells that are placed horizontally. A column is made up of set of cells that are placed vertically.

The user can represent the data in a tabular format by using the `<table>` element in HTML. The `<tr>` element divides the table into rows and the `<td>` element specifies columns for each row. By default, a table does not have a border. The `border` attribute of the `<table>` element specifies a border for making the table visible in a Web page.

Using slides 4 and 5, explain the HTML code for table and its output.

Explain the code that uses the `<table>` element to create a table. The `border` attribute of `<table>` element gives a border to the table, which is 1 pixel wide. The `<tr>` element

within the `<table>` element creates rows. The `<td>` element creates two cells with the values **English** and **German** in the first row and **French** and **Italian** in the second row.

Explain the figure provided in the slide for displaying the table.

### In-Class Question:

After you finish explaining table creation process, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which tag is used for defining a row in table?

### Answer:

`<td>` tag is used for defining a row in the table.

## Slides 6 to 8

Let us understand table heading tag.

**HTML 5 Table Headings 1-3**

- The user can specify the heading for each column in HTML.
- To specify the heading for columns in a table, use the `<th>` element.
- The text included within the `<th>` element appears in bold.
- The Code Snippet demonstrates how to create a table with a heading.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>List of Students </title>
  </head>
  <body>
    <h2>List of Students</h2>
    <table border="1">
      <tr>
        <th>Name</th>
        <th>Age</th>
        <th>Place</th>
      </tr>
```

© Aptech Ltd. Creating Tables / Session 9 6

**HTML 5 Table Headings 2-3**

```
<tr>
<td>Mark</td>
<td>17</td>
<td>Madrid</td>
</tr>
<tr>
<td>John</td>
<td>19</td>
<td>London</td>
</tr>
</table>
</body>
</html>
```

- In this code, the `<table>` element creates a table with a border of 1 pixel.
- The `<th>` element provides three column headings namely, **Name**, **Age**, and **Place**.

© Aptech Ltd. CreatingTables / Session 9 7

**HTML 5 Table Headings 3-3**

- The second and the third row lists the details of the students in the three columns.
- Following figure displays the output of the table with headings.

Name	Age	Place
Mark	17	Madrid
John	19	London

© Aptech Ltd. CreatingTables / Session 9 8

Using slides 6 to 8, explain the table headings and HTML code.

Explain that one can specify the heading for each column in HTML. To specify the heading for columns in a table, use the `<th>` element.

The text included within the `<th>` element appears in bold. Explain the code snippet that demonstrates how to create a table with a heading.

In this code, the `<table>` element creates a table with a border of 1 pixel. The `<th>` element provides three column headings namely, **Name**, **Age**, and **Place**.

Using slide 8, explain the output of the `<th>` tag.

Explain the second and third rows that list the details of the students in the three columns.

The first row is heading while the other two rows shows the data.

**Tips:**

1. The text in `<th>` element is bold and centered by default.
2. The `<thead>` tag is used to group header content in an HTML table. The `<thead>` element is used in conjunction with the `<tbody>` and `<tfoot>` elements to specify each part of a table (header, body, footer).

Browsers can use these elements to enable scrolling of the table body independently of the header and footer. Also, when printing a large table that spans multiple pages, these elements can enable the table header and footer to be printed at the top and bottom of each page.

The `<thead>` tag must be used in the following context: As a child of a `<table>` element, after any `<caption>` and `<colgroup>` elements, and before any `<tbody>`, `<tfoot>`, and `<tr>` elements.

**Slides 9 to 11**

Let us understand `colspan` attribute.

**HTML 5 Colspan Attribute 1-3**

- Spanning refers to a process of extending a cell across multiple rows or columns.
- To span two or more columns, use the `colspan` attribute of the `<td>` and `<th>` elements.
- The `colspan` attribute allows the user to span a cell along a horizontal row.
- The value of the `colspan` attribute specifies the number of cells across which a specific cell shall be expanded.
- The Code Snippet demonstrates how to create a table and span header cells across two cells vertically.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Employee Details</title>
  </head>
  <body>
    <h2>Employee Details</h2>
    <table border="1">
```

© Aptech Ltd. Creating Tables / Session 9 9

**HTML 5 Colspan Attribute 2-3**

```
<tr>
  <th colspan="2">IT</th>
  <th colspan="2">Accounts</th>
</tr>
<tr>
  <th>Name</th>
  <th>Location</th>
  <th>Name</th>
  <th>Location</th>
</tr>
<tr>
  <td>David</td>
  <td>New York</td>
  <td>John</td>
  <td>London</td>
</tr>
```

© Aptech Ltd. Creating Tables / Session 9 10

**HTML Colspan Attribute 3-3**

```

<tr>
  <td>Kathy</td>
  <td>New Jersey</td>
  <td>Peter</td>
  <td>Los Angeles</td>
</tr>
</table>
</body>
</html>

```

- The code creates a table with a border of 1 pixel.
- The `<th>` element specifies two column headings namely, **IT** and **Accounts**.
- Each of these header cells horizontally span across the two cells by setting the `colspan` attribute of the `<th>` element to 2.
- Each of these headings has two sub-headings namely, **Name** and **Location**, which specify the name and location of employees.
- The first and second rows display the details of the employees.

© Aptech Ltd. Creating Tables / Session 9 11

Using slides 9 to 11, explain the `colspan` attribute of `<th>` or `<td>` tag.

The user might feel the need to span two or more cells while working with tables. Spanning refers to a process of extending a cell across multiple rows or columns. To span two or more columns, use the `colspan` attribute of the `<td>` and `<th>` elements. The `colspan` attribute allows the user to span a cell along a horizontal row. The value of the `colspan` attribute specifies the number of cells across which a specific cell shall be expanded.

Then explain the code snippet which demonstrates how to create a table and span header cells across two cells vertically.

Explain the code that creates a table with a border of 1 pixel. The `<th>` element specifies two column headings namely, **IT** and **Accounts**. Each of these header cells horizontally span across the two cells by setting the `colspan` attribute of the `<th>` element to 2. Each of these headings has two sub-headings namely, **Name** and **Location**, which specify the name and location of employees. The first and second rows display the details of the employees.

## Slides 12 to 14

Let understand rowspan attribute.

**HTML 5 Rowspan Attribute 1-3**

- The rowspan attribute spans a data cell across two or more rows.
- It allows the user to span a data cell along a vertical column.
- Like the colspan attribute, the rowspan attribute can be used within the <td> and <th> elements.
- The Code Snippet demonstrates how to span a cell across multiple rows.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Automobile Gallery</title>
  </head>
  <body>
    <table border="1">
      <tr>
        <th>Manufacturer</th>
        <th>Model</th>
        <th>Price</th>
      </tr>
      <tr>
        <th rowspan="3">Audi</th>
        <td>A4</td>
        <td>34.5</td>
      </tr>
```

© Aptech Ltd. Creating Tables / Session 9 12

**HTML 5 Rowspan Attribute 2-3**

```
<tr>
  <td>A5</td>
  <td>42.6</td>
</tr>
<tr>
  <td>A6</td>
  <td>30.75</td>
</tr>
<tr>
  <th rowspan="2">BMW</th>
  <td>328i</td>
  <td>28.25</td>
</tr>
<tr>
  <td>530d</td>
  <td>47.5</td>
</tr>
</table>
</body>
</html>
```

- The code creates a table with a border width of 1 pixel.

© Aptech Ltd. Creating Tables / Session 9 13

**HTML 5 Rowspan Attribute 3-3**

- The three <th> elements within the <tr> element specify column headings namely, **Manufacturer**, **Model**, and **Price**.
- The rowspan attribute of the <th> element combines the three rows of the **Manufacturer** column into a common brand namely, **Audi**.
- The three different models and the respective prices of the **Audi** brand are displayed in three different rows.
- Similarly, the rowspan attribute of the <th> element combines the next two rows of the **Manufacturer** column into a common brand called **BMW**.
- Following figure displays the rowspan attribute effect.

Manufacturer	Model	Price
Audi	A4	34.5
Audi	A5	42.6
Audi	A6	30.75
BMW	328i	28.25
BMW	530d	47.5

Effect of rowspan

© Aptech Ltd. Creating Tables / Session 9 14

Using slides 12 to 14, explain the attribute `rowspan` and HTML code for it. The `rowspan` attribute spans a data cell across two or more rows. It allows spanning a data cell along a vertical column. Like the `colspan` attribute, the `rowspan` attribute can be used within the `<td>` and `<th>` elements.

Explain the code snippet for spanning a cell across multiple rows.

Using slide 14, explain the output for `rowspan` attribute.

Explain the code that creates a table with a border width of 1 pixel. The three `<th>` elements within the `<tr>` element specify column headings namely, **Manufacturer**, **Model**, and **Price**. The `rowspan` attribute of the `<th>` element combines the three rows of the **Manufacturer** column into a common brand namely, **Audi**. The three different models and the respective prices of the **Audi** brand are displayed in three different rows. Similarly, the `rowspan` attribute of the `<th>` element combines the next two rows of the **Manufacturer** column into a common brand called **BMW**.

Explain the figure that displays the `rowspan` effect.

## Slides 15 to 18

Let us understand horizontal alignment.

**HTML 5 Horizontal Alignment 1-4**

- Alignment determines the representation of text along the left, right, or center positions.
- In HTML, by default, the data within the table is aligned on the left side of the cell.
- HTML5 has deprecated the `align` attribute.
- The four possible values for setting the horizontal alignment are as follows:

<b>left:</b>	• Aligns the data within a cell on the left side. This is the default value for table content.
<b>center:</b>	• Aligns the data within the cell on the center. This is the default value for table headings.
<b>right:</b>	• Aligns the data within the cell on the right side.
<b>justify:</b>	• Aligns the data within the cell by adjusting the text at the edges.

© Aptech Ltd. Creating Tables / Session 9 15

**HTML 5 Horizontal Alignment 2-4**

- To set the alignment with style you can use the text-align attribute to specify the horizontal alignment.
- The Code Snippet demonstrates how to center align the table data.

```
<!DOCTYPE HTML>
<html>
<head>
<title>Automobile Gallery</title>
</head>
<body>
<table border="1">
<tr>
<th>Sr.No.</th>
<th>Medicine Name</th>
<th>Price</th>
</tr>
<tr style="text-align: center;">
<td>1</td>
<td>Captopril</td>
<td>12.45</td>
</tr>

```

© Aptech Ltd. Creating Tables / Session 9 16

**HTML 5 Horizontal Alignment 3-4**

```
<tr style="text-align: center;">
<td>2</td>
<td>Ceftriaxone</td>
<td>6.94</td>
</tr>
<tr style="text-align: center;">
<td>3</td>
<td>Ciprofloxacin</td>
<td>56.21</td>
</tr>
</table>
</body>
</html>
```

- The code aligns the data within the row using a style within the `<tr>` element.
- The table content is center aligned by setting the value of the `text-align` attribute to center.

© Aptech Ltd. Creating Tables / Session 9 17

**HTML 5 Horizontal Alignment 4-4**

- Following figure displays the horizontal alignment.

Sr.No.	Medicine Name	Price
1	Captopril	12.45
2	Ceftriaxone	6.94
3	Ciprofloxacin	56.21

Center-align data

© Aptech Ltd. Creating Tables / Session 9 18

Using slides 15 to 18, explain the horizontal alignment or the align attribute.

Alignment determines the representation of text along the left, right, or centre positions. In HTML, by default, the data within the table is aligned on the left side of the cell. Sometimes,

the user might need to align the data to some other position for improving the readability or focusing on some data. HTML5 has deprecated the align attribute.

The four possible values for setting the horizontal alignment are as follows:

- **left:** Aligns the data within the cell on the left side. This is the default value for table content.
- **center:** Aligns the data within the cell on the center. This is the default value for table headings.
- **right:** Aligns the data within the cell on the right side.
- **justify:** Aligns the data within the cell by adjusting the text at the edges.

To set the alignment with style, you can use the `text-align` attribute to specify the horizontal alignment.

Using slides 16 to 18, explain the horizontal alignment HTML code and its output.

Explain the code that aligns the data within the row using a style in the `<tr>` element. The table content is `center` aligned by setting the value of the `text-align` attribute to `center`.

Explain the figure that displays the horizontal alignment.

## Slides 19 to 22

Let us understand the vertical alignment.

**HTML 5 Vertical Alignment 1-4**

- Users can vertically align the position of data earlier by using the `valign` attribute.
- HTML5 has deprecated the `valign` attribute.
- The possible values of vertical alignment are as follows:

<b>top:</b>	• Vertically aligns the data within the cell at the top.
<b>middle:</b>	• Vertically aligns the data within the cell at the center.
<b>bottom:</b>	• Vertically aligns the data within the cell at the bottom.

- To set the alignment with the style, you can use the `text-align` attribute to specify the vertical alignment use the following syntax:

**Syntax:**

```
<td style= "text-align: center; vertical-align: middle">
Aptech Web site </a>
```

© Aptech Ltd. CreatingTables / Session 9 19

**HTML 5 Vertical Alignment 2-4**

- The style can also be applied to individual rows, cells, or to the entire table.
- The Code Snippet demonstrates how to align the data vertically within the table using the style attribute.

```
<!DOCTYPE HTML>
<html>
<head>
<title>CelinaBatteries</title>
</head>
<body>
<table border="1">
<tr>
<th>Sr.No.</th>
<th>Product Id</th>
<th>Product Description</th>
</tr>
<tr>
<td style="text-align: center; vertical-align: middle">1
</td>
<td style="text-align: center; vertical-align: middle">P101
</td>
<td>1.5 Volts AA Ultra Alkaline</td>
</tr>
```

© Aptech Ltd. Creating Tables/Session 9 20

**HTML 5 Vertical Alignment 3-4**

```
<tr>
<td style="text-align: center; vertical-align: middle">
</td>
<td style="text-align: center; vertical-align: middle">
M105
</td>
<td>9 Volts pp3 Super Alkaline</td>
</tr>
</table>
</body>
</html>
```

- The text-align attribute is set to the value center, which specifies that the data within the rows are centrally aligned.
- The vertical-align is used to specify the vertical alignment in the table.

© Aptech Ltd. Creating Tables/Session 9 21

**HTML 5 Vertical Alignment 4-4**

- Following figure displays the vertical alignment.

Effect of vertical alignment

© Aptech Ltd. Creating Tables/Session 9 22

Using slides 19 to 22, explain the vertical alignment.

Mention users can vertically align the position of data earlier by using the valign attribute. HTML5 has deprecated the valign attribute.

The possible values of vertical alignment are as follows:

- **top**: Vertically aligns the data within the cell at the top.
- **middle**: Vertically aligns the data within the cell at the center.
- **bottom**: Vertically aligns the data within the cell at the bottom.

To set the alignment with the style, you can use the `text-align` attribute to specify the vertical alignment use the following syntax:

```
<td style= "text-align: center; vertical-align: middle">
```

The style can also be applied to individual rows, cells, or to the entire table.

Using slides 20 to 22, explain the vertical alignment HTML code and its output.

Mention code snippet for how to align the data vertically within the table using the `style` attribute.

The `text-align` attribute is set to the value `center`, which specifies that the data within the rows are centrally aligned. The `vertical-align` is used to specify the vertical alignment in the table.

Explain the figure that displays the vertical alignment.

### Slide 23

Let us understand margin attributes.

**Margin Attributes**

- The data in a table might appear cluttered, which may affect the readability.
- This might make it difficult to comprehend data as the data.
- To overcome this issue, use the cell margin attributes.
- Cell padding allows the user to control the look of the content on a page.

**> Padding**

- Padding is the amount of space between the content and its outer edge.
- For tables, padding is referred as a space between the text and the cell border.
- Suppose, if the user wants to set the padding attribute for the individual cells then padding attribute can be used in a style as follows:

```
<td style="padding: 4px">
```

© Aptech Ltd. Creating Tables / Session 9 23

Using slide 23, explain the margin attribute.

The data in a table might appear cluttered, which may affect the readability. This might make it difficult to comprehend data as the data. To overcome this issue, use the cell margin attributes. Cell padding allows the user to control the look of the content on a page.

## Padding:

Padding is the amount of space between the content and its outer edge. For tables, padding is specified as a space between the text and the cell border.

Suppose, if the user wants to set the padding attribute for the individual cells, then he/she can use the padding attribute in a style as follows:

```
<td style="padding: 4px">
```

## Slides 24 to 26

Let us understand the `caption` element.

**HTML 5 Caption Element 1-3**

- To specify the main heading for the table, use the `<caption>` element.
- The `<caption>` element defines a caption for the table. It is a sub-element of the `<table>` element.
- It must be present immediately after the `<table>` tag.
- The `<caption>` element allows the user to specify a title for your entire table.
- There can be only one caption for a table.
- The Code Snippet demonstrates how to specify a heading for a table.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Travel Expense Report</title>
  </head>
  <body>
    <table border="1">
      <caption>Travel Expense Report</caption>
      <tr>
        <th>&nbsp;</th>
        <th>Meals</th>
        <th>Hotels</th>
        <th>Transport</th>
      </tr>
    </table>
  </body>
</html>
```

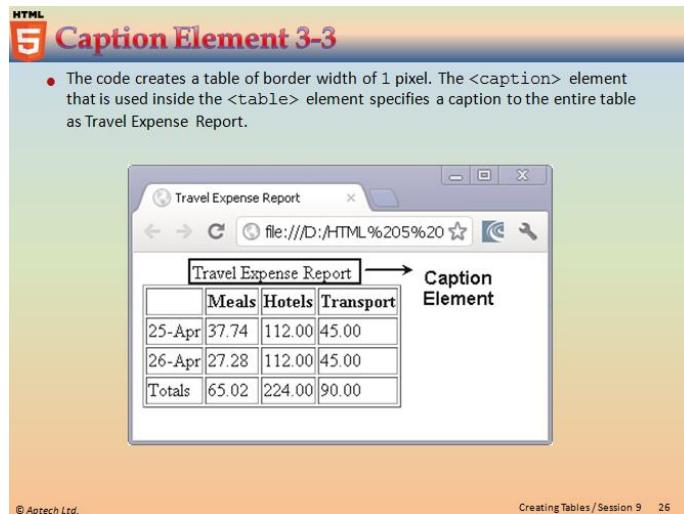
© Aptech Ltd. Creating Tables / Session 9 24

**HTML 5 Caption Element 2-3**

25-Apr	37.74	112.00	45.00
26-Apr	27.28	112.00	45.00
Totals	65.02	224.00	90.00

```
<tr>
  <td>25-Apr</td>
  <td>37.74</td>
  <td>112.00</td>
  <td>45.00</td>
</tr>
<tr>
  <td>26-Apr</td>
  <td>27.28</td>
  <td>112.00</td>
  <td>45.00</td>
</tr>
<tr>
  <td>Totals</td>
  <td>65.02</td>
  <td>224.00</td>
  <td>90.00</td>
</tr>
</table>
</body>
</html>
```

© Aptech Ltd. Creating Tables / Session 9 25



Using slides 24 to 26, explain the `caption` element and HTML code for it.

Mention a user can add a heading to a table in HTML. To specify the main heading for the table, use the `<caption>` element. The `<caption>` element defines a caption for the table. It is a sub-element of the `<table>` element. It must be present immediately after the `<table>` tag.

Unlike the `<th>` element that is used to specify a heading to an individual row or column, the `<caption>` element allows the user to specify a title for the entire table. There can be only one caption for a table.

Explain the code snippet for specifying a heading for a table using `caption` element.

Using slide 26, explain the output for the `<caption>` tag code.

The code creates a table of border width of 1 pixel. The `<caption>` element that is used inside the `<table>` element specifies a caption to the entire table as **Travel Expense Report**. Explain the figure that displays the table captions.

## Slides 27 and 28

Let us understand table size and width of columns.

**HTML 5 Table Size and Width of a Column 1-2**

- The table size can be expanded when the user wants to add rows and columns in the table.
- The user can use the `<style>` section to set the default width for the table to 100% of the browser window.
- To set the width of a column in pixels, one can use style attribute in the `<td>` tag.
- The Code Snippet demonstrates how to create a table with specific width for a column.

```
<!DOCTYPE HTML>
<html>
<head>
<title>Tables</title>
</head>
<body>
<h2>Table</h2>
<table border="1">
<tr>
<td style ="width: 200px">Flowers</td>
<td style ="width: 80px">Fruits</td>
</tr>
</table>
</body>
</html>
```

© Aptech Ltd. Creating Tables / Session 9 27

**HTML 5 Table Size and Width of a Column 2-2**

```
<tr>
<td style ="width: 200px">Vegetables</td>
<td style ="width: 80px">Trees</td>
</tr>
</table>
</body>
</html>
```

- The code creates a table of border width of 1 pixel.
- The `<style>` element is used to set table width to 100%.
- The width of the columns is set by using the style attribute.
- Following figure displays the table size and column width.

© Aptech Ltd. Creating Tables / Session 9 28

Using slides 27 and 28, explain the table size and column width.

Mention that user can decide the size of the table based on his/her requirements while creating a Web site. The table size can be expanded when the user wants to add rows and columns in the table. The user can use the `<style>` section to set the default width for the table to 100% of the browser window. For setting the width of a column in pixels, you can use style attribute in the `<td>` tag.

Explain the code snippet for creating a table with specific width for a column.

Explain the figure that displays the table size and column width.

## Slides 29 to 31

Let us understand merging table cells.

**Merging Table Cells 1-3**

- To change the cells of a table to different height and width, colspan and rowspan attributes can be used.
- Consider a scenario, where the user wants to merge a cell into adjacent cells to the right-hand side.
  - The colspan attribute can be used to specify the number of columns to span.
  - The rowspan attribute can be used to specify the number of rows.
- The Code Snippet demonstrates creating a table having five columns and five rows, but many of the cells span multiple columns or rows.

```
<!DOCTYPE HTML>
<html>
<head>
<title>Favorite Destination</title>
</head>
<body>
<h2>Report</h2>
<table border="1" width="100%" height="100%">
<tr>
<td colspan="2" rowspan="2">Results</td>
<td colspan="3">Range</td>
</tr>
```

© Aptech Ltd. Creating Tables / Session 9 29

**Merging Table Cells 2-3**

```
<tr>
<td>18 to 20</td>
<td>25 to 50</td>
<td>over 50</td>
</tr>
<tr>
<td rowspan="3">Your favorite vacation destination</td>
<td>Dubai</td>
<td>25%</td>
<td>50%</td>
<td>25%</td>
</tr>
<tr>
<td>Bangkok</td>
<td>40%</td>
<td>30%</td>
<td>30%</td>
</tr>
```

© Aptech Ltd. Creating Tables / Session 9 30

**Merging Table Cells 3-3**

```
<tr>
<td>Switzerland</td>
<td>30%</td>
<td>20%</td>
<td>50%</td>
</tr>
</table>
</body>
</html>
```

- The code creates a table having a border of 1 pixel, table with five columns and five rows, and uses the colspan and rowspan attributes respectively.
- Following figure displays the merging table cells.

The screenshot shows a table titled "Report". The first row has three columns: "Results" (colspan=2) and "Range" (colspan=3). The second row has four columns: "18 to 20" (colspan=2), "25 to 50", "Over 50", and an empty cell. The third row has four columns: "Your favorite vacation destination" (rowspan=3), "Dubai", "Bangkok", and "Switzerland". The fourth row has three columns: "25%", "50%", and "25%". The fifth row has three columns: "40%", "30%", and "30%". The sixth row has three columns: "30%", "20%", and "50%".

© Aptech Ltd. Creating Tables / Session 9 31

Using slides 29 to 31, explain the concept of merging table cells. Explain the need for merging cell.

Suppose if the user wants to change the cells of a table to different height and width, then `colspan` and `rowspan` attributes can be used. Consider a scenario, where the user wants to merge a cell into adjacent cells to the right side. The `colspan` attribute can be used to specify the number of columns to span. Similarly, the user can use the `rowspan` attribute to specify the number of rows.

Explain the code snippet that demonstrates creating a table having five columns and five rows.

Using slide 31, explain the output for merging cell.

The code creates a table having a border of 1 pixel. It also creates a table with five columns and five rows and uses the `colspan` and `rowspan` attributes respectively. Explain the figure that displays the merging table cells.

**In-Class Question:**

After you finish explaining the merging table cell concept, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which attributes are used for merging table columns and rows?

**Answer:**

`Colspan` and `rowspan` attributes are used for merging columns and rows, respectively.

## Slides 32 and 33

Let us understand how to apply border to table using styles.

**HTML 5** **Apply Borders by Using Styles 1-2**

- CSS can be used for applying borders as it is the best reliable and flexible method.
- One can format the table by using style based border for `<table>` and `<td>` tags.
- To evaluate the attributes used are as follows:

<b>border-width:</b>	• Used to control the thickness of the border and the values are specified in pixels.
<b>border-color:</b>	• Used to control the color of the border and specifies the color by either name, or RGB value, or hexadecimal number.
<b>border-style:</b>	• Used to control the line style. Users can choose between solid, dashed, groove, dotted, outset, ridge, inset, or none.

© Aptech Ltd. Creating Tables / Session 9 32

**HTML 5** **Apply Borders by Using Styles 2-2**

- To set all these attributes at one time, the user can use the `border` attribute and place the settings in the order of width, color, and style respectively.
- To format the sides of the border individually, replace the `border` attribute with `border-bottom`, `border-top`, `border-right`, or `border-left` attribute.
- The user can apply these attributes to the entire table or individual cells and also create rules in the `<style>` area.

© Aptech Ltd. Creating Tables / Session 9 33

Using slides 32 and 33, explain the process of applying border using styles.

Mention that users can use CSS for applying borders as it is the best reliable and flexible method. The user must select the CSS method for Web sites that will be active for many years as the old formatting methods will not be used in future. You can format the table by using style based border for `<table>` and `<td>` tags. To evaluate this, the attributes used are as follows:

- The `border-width` attribute is used to control the thickness of the border and the values are specified in pixels.
- The `border-color` attribute is used to control the color of the border and specifies the color by name, or RGB value, or hexadecimal number.
- The `border-style` attribute is used to control the line style. Users can choose value from solid, dashed, groove, dotted, outset, ridge, inset, or none.

Suppose, if the user wants to set all these attributes at one time, then the user can use the border attribute and place the settings in the following order namely, width, color, and style respectively. The user can also format the sides of the border individually by replacing the border attribute with border-bottom, border-top, border-right, or border-left attribute. The user can apply these attributes to the entire table or individual cells and also create rules in the <style> area.

Explain the CSS code for applying border:

```
table, th, td
{
    border: 1px solid black;
}
```

The code snippet demonstrates use of border attribute which will apply border to the table in all with the width 1px, color black, and style solid.

### Slides 34 to 38

Let us understand tables for page layout.

**HTML**

## 5 Tables for Page Layout 1-5

- Tables are used for structuring the content and to organize the data in an appropriate manner.
- Tables allow the user to arrange the data horizontally or vertically according to the requirement.
- Each and every Web site has a unique way of presenting data to their customers or users.
- Many Web sites use pop-ups for providing information to their customers.
- The Code Snippet demonstrates a simple example of using table for structuring the content of a Web page.

```
<!DOCTYPE HTML>
<html>
<head>
    <title>Page Layout </title>
</head>
<style>
```

© Aptech Ltd. Creating Tables / Session 9 34

## 5 Tables for Page Layout 2-5

```
#navlayout {
    width: 100%;
    float: left;
    margin: 0 3em 0;
    padding: 0;
    list-style: none;
    background-color: #f2f2f2;
    border-bottom: 1px solid #ccc;
    border-top: 1px solid #ccc; }

#navlayout li {
    float: left; }

#navlayout li a {
    display: block;
    padding: 8px 15px;
    text-decoration: none;
    font-weight: bold;
    color: #069;
    border-right: 1px solid #ccc; }

#navlayout li a:hover {
    color: #c00;
    background-color: #fff; }

</style>
```

© Aptech Ltd. Creating Tables / Session 9 35

**HTML 5 Tables for Page Layout 3-5**

```
<body>
  
  <h1>Blossoms Gallery</h1>
  <h5><i>The Best sellers for flowers since 1979</i></h5>
  <navlayout>
    <hr>
    <ul id="nav">
      <li><a href="#">Home</a></li>
      <li><a href="#">Contact Us</a></li>
      <li><a href="#">About Us</a></li>
      <li><a href="#"> FAQs</a></li>
    </ul>
  </nav>
```

© Aptech Ltd.

CreatingTables/Session 9 36

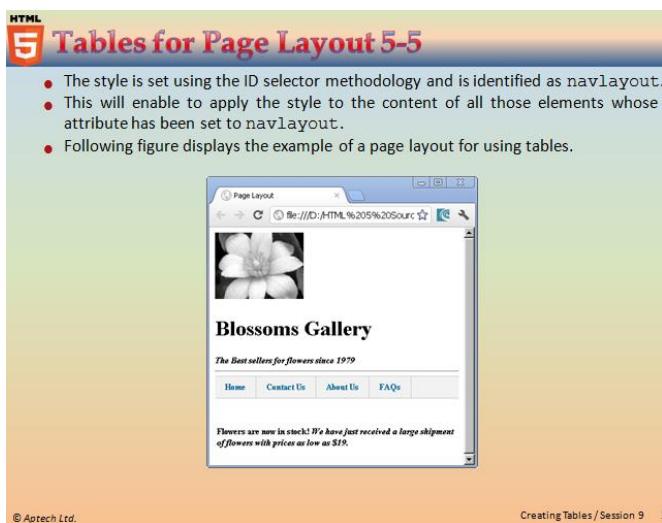
**HTML 5 Tables for Page Layout 4-5**

```
<table>
  <tr>
    <td>
      <b>Flowers are now in stock! </b>
      <i> We have just received a large shipment of flowers
           with prices as low as $19.
      </i>
    </td>
  </tr>
</table>
```

- The code creates a page layout for a Web site. The data is arranged in a tabular format and an embedded style is used for defining the style.
- The style is defined using the `style` element placed immediately after the `<head>` section.
- Defining a style in this manner helps to reuse the style in the same Web page.

© Aptech Ltd.

CreatingTables/Session 9 37



© Aptech Ltd.

CreatingTables/Session 9 38

Using slides 34 to 38, explain the use of table for page layout.

Nowadays, there are many new techniques used for developing attractive Web pages. Tables are used for structuring the content. In other words, tables are used by the user to organize the data in an appropriate manner.

With the help of tables, the user can arrange the data horizontally or vertically according to his/her requirements. Community Web sites such as Facebook has different page layouts, the user uses the navigation tabs to move from one page to another. Similarly, the look and feel of each page is different.

While accessing Web sites such as Yahoo, Rediff, and so on, users can view that the home page is very informative with a number of links, images, and so on. Each and every Web site has its unique way of presenting data to their customers or users. Many Web sites use pop-ups for providing information to their customers.

Using slides 35 to 37, explain the table used for page layout in the HTML code.

Explain the code snippet which demonstrates a simple example of using table for structuring the content of a Web page.

The code creates a page layout for a Web site. The data is arranged in a tabular format and an embedded style is used for defining the style. The style is defined using the style element placed immediately after the `<head>` section.

Using slide 38, explain the output of using table for page layout.

Defining a style in this manner helps to reuse the style in the same Web page.

The style is set using the ID selector methodology and is identified as `navlayout`. This will enable to apply the style to the content of all those elements whose `id` attribute has been set to `navlayout`.

Explain the figure displaying the page with table page layout.

### In-Class Question:

After you finish explaining the concept of table for page layout, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which section of HTML code is used for defining the styles for a Web page?

### Answer:

`<head>` element contains the `<style>` tag inside which the styles can be defined.

## Slide 39

Let us summarize the session.

The slide has a header 'HTML 5 Summary' with a red '5' icon. The content is a bulleted list of nine points:

- Tables allow the user to view your data in a structured and classified format.
- Padding is the amount of space between the content and its outer edge.
- The caption element defines a caption for a table. It is a sub-element of the <table> element.
- Spanning refers to a process of extending a cell across multiple rows or columns.
- The rowspan attribute spans a data cell across two or more rows.
- The colspan attribute allows the user to specify the number of columns a cell should span.
- The border attribute of the table element allows the user to specify a border for making the table visible in a Web page.
- Tables allow the user to organize the data. It enables the developer to design a Web page having an attractive page layout.

© Aptech Ltd. Creating Tables / Session 9 39

Use slide 39 to summarize the session. You will end the session, with a brief summary of what has been taught in the session.

### 9.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session. You should also explore the HTML Forms that are offered with the next session.

#### Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 10 – HTML Forms

---

## 10.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 10.1.1 Objectives

By the end of this session, the learners will be able to:

- Describe HTML5 forms
- Explain the working of new input types in HTML5
- Explain the new Form attributes
- Explain the new Form elements

### 10.1.2 Teaching Skills

To teach this session, you should be well-versed with the elements used in creating HTML forms. You should also know about the new input types in HTML5. The new input types in HTML5 includes new form attributes and elements.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

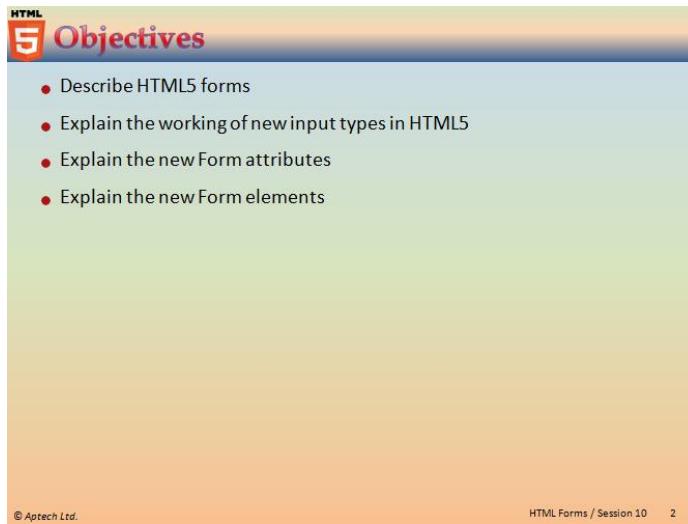
#### In-Class Activities:

Follow the order given here during In-Class activities.

## Overview of the Session:

Give the students the overview of the current session in the form of session objectives.

Show the students slide 2 of the presentation.

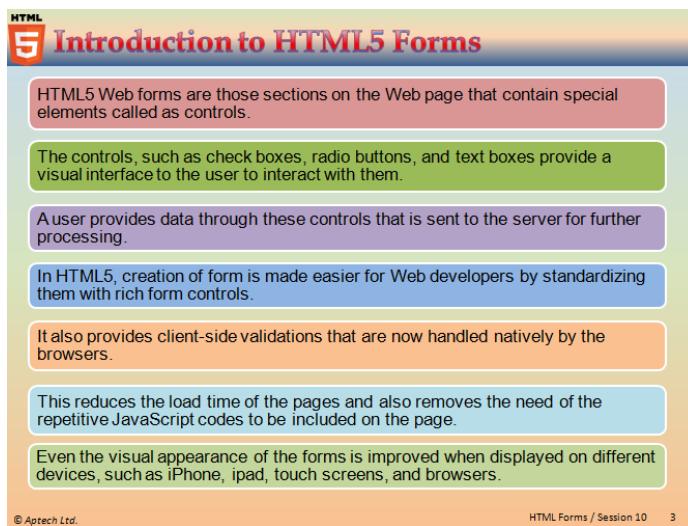


Tell the students that this session introduces them to various HTML 5 forms. They will learn also learn new input types, new form attributes, and new form element introduced in HTML5.

## 10.2 In-Class Explanations

### Slide 3

Let us understand forms in HTML5.



Using slide 3, explain the features of HTML forms and new enhancements in HTML5 forms.

Web forms created on HTML are those sections on the Web page that contain special elements called as controls.

The controls, such as check boxes, radio buttons, text boxes, and so on provides a visual interface to the user to interact with them. A user provides data through these controls that is sent to the server for further processing.

Thus, it is very important to create forms with lot of consideration for usability and accessibility by the users.

Explain the students that important elements such as `<form>` and `<input>` used to create form and add controls on it.

The syntax to create form on the Web page is as follows:

```
<form>
    . . .
    <!-- Elements or controls accepting input -->
    . . .
</form>
```

The syntax to create form element is as follows:

```
<form>
    <input type="" value="" />
</form>
```

Here, the `type` attribute can be a single line text box, check box, password, radio button, submit button, and so on.

The code to create a login form on the Web page is as follows:

```
<form>
    name: <input type="text" name="firstname"><br>
    Password: <input type="password" name="pwd"> <br>
    <input type="submit" value="Submit">
</form>
```

The submit button sends data from the form controls to a server for processing the input. The way to send the data to the server depends on the value of the `method` attribute specified with the `<form>` element. Also, page where the data needs to be sent is specified in the `action` attribute.

Example:

```
<form method="GET" action="form_action.jsp">
```

```

name: <input type="text" name="firstname"><br>
Password: <input type="password" name="pwd"> <br>
<input type="submit" value="Submit">
</form>

```

- **GET** - Appends the form-data into the address of the page as name/value pair.  
Example, <http://www.myWebsite.com/. . ./form.html?firstname=john&pwd=john@1234>
- **POST** - Appends the form-data inside the body of the HTTP request protocol. The data is not visible on the address bar and hence treated as a secure way to pass data to the server.

Then, you can even show some more codes on HTML forms containing check boxes, radio buttons, and so on.

Then, explain that among the other features of HTML5, there has been a great enhancement to Web forms. In HTML5, creation of form is made easier for Web developers by standardizing them with rich form controls. It also provides client-side validations that are now handled natively by the browsers. This not only reduces the load time of the pages, but also removes the need of the repetitive JavaScript codes to be included on the page.

Even the visual appearance of the forms is improved when displayed on different devices, such as iPhone, ipad, touch screens, and browsers. This enhances the user experience while interacting with them.

#### Slide 4

Let us understand new features in HTML5 forms.

**New Features in HTML5 Forms**

- HTML5 Web forms bring great improvements related to form creation for the Web developers and also for users interacting with them.

The following are the changes introduced in HTML5 forms:

- New form elements
- New input types
- New attributes
- Browser-based validation
- CSS3 styling techniques
- Forms API

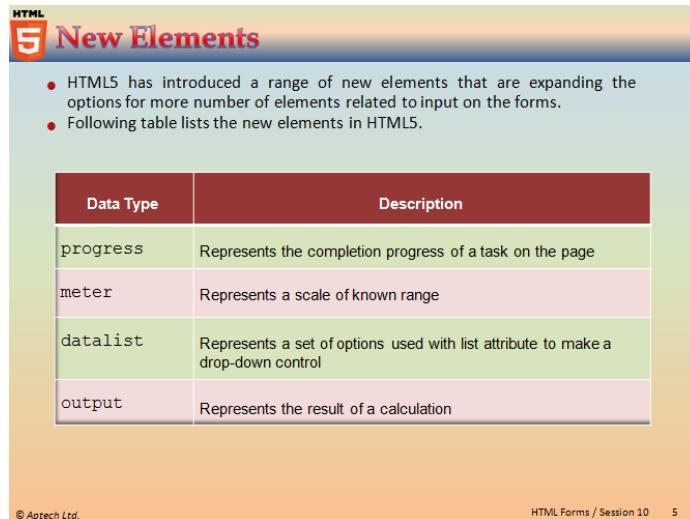
© Aptech Ltd.      HTML Forms / Session 10      4

Using slide 4, explain the new features of HTML 5 form.

Mention HTML5 Web forms bring great improvements related to form creation for the Web developers and also for users interacting with them. Describe the changes introduced in HTML5 forms as listed on the slide.

## Slide 5

Let us understand new elements in HTML5.



The slide has a blue header bar with the text "HTML 5 New Elements". Below the header is a bulleted list:

- HTML5 has introduced a range of new elements that are expanding the options for more number of elements related to input on the forms.
- Following table lists the new elements in HTML5.

Data Type	Description
<code>progress</code>	Represents the completion progress of a task on the page
<code>meter</code>	Represents a scale of known range
<code>datalist</code>	Represents a set of options used with list attribute to make a drop-down control
<code>output</code>	Represents the result of a calculation

At the bottom of the slide, there is a footer bar with the text "© Aptech Ltd.", "HTML Forms / Session 10", and the number "5".

Using slide 5, explain the new elements of HTML5 form.

HTML5 has introduced a range of new elements that are expanding the options for more number of elements related to input on the forms.

Describe the list of the new elements in HTML5.

### Tips:

`keygen` is also a new element of HTML 5 form. The `<keygen>` tag specifies a key-pair generator field used for forms. When the form is submitted, the private key is stored locally and the public key is sent to the server.

## Slides 6 and 7

Let us understand new input types introduced in HTML5.

**HTML 5 New Input Types 1-2**

- The input element is a data field that allows the user to edit the data on the form.
- It has an attribute named `type` which controls the data type and characteristics of the input element.
- Following table lists the new input types supported by HTML5.

Type	Description
<code>email</code>	Represents the completion progress of a task on the page
<code>search</code>	Represents a scale of known range
<code>url</code>	Represents a set of options used with list attribute to make a drop-down control
<code>tel</code>	Represents the result of a calculation
<code>number</code>	Represents a numeric value in the input field

© Aptech Ltd.      HTML Forms / Session 10      6

**HTML 5 New Input Types 2-2**

Type	Description
<code>range</code>	Represents a numeric value to be selected from a range of numbers
<code>date</code>	Represents a calendar which is shown whenever the field is clicked
<code>Week</code>	Represents date in year-week format
<code>month</code>	Represents a value with year-month format
<code>time</code>	Represents a value in hours and minutes format
<code>datetime</code>	Represents a full date and time input field with a time zone
<code>datetime-local</code>	Represents a full date and time with no time zone
<code>color</code>	Represents a predefined interface for selecting color

© Aptech Ltd.      HTML Forms / Session 10      7

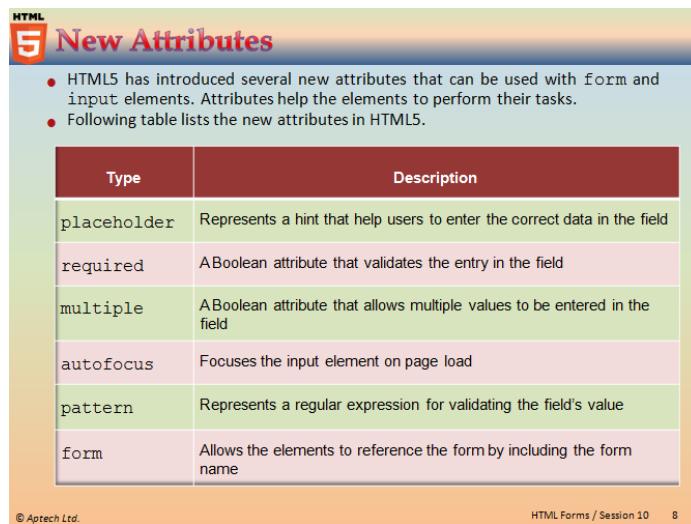
Using slides 6 and 7, explain the new input types. Explain the list of the new input types supported by HTML5 that specify the kind of input expected from the users on the Web page.

### Tips:

All browsers do not provide support all the new form elements. If any of the element is not supported, then it is displayed as a text field.

## Slide 8

Let us understand new attributes in HTML5.



The slide has a header 'HTML 5' and a title 'New Attributes'. It contains two bullet points: 'HTML5 has introduced several new attributes that can be used with form and input elements. Attributes help the elements to perform their tasks.' and 'Following table lists the new attributes in HTML5.' Below this is a table with six rows, each containing an attribute name and its description. The table has a red header row. At the bottom, there is a footer with '© Aptech Ltd.' and 'HTML Forms / Session 10 8'.

Type	Description
placeholder	Represents a hint that helps users to enter the correct data in the field
required	A Boolean attribute that validates the entry in the field
multiple	A Boolean attribute that allows multiple values to be entered in the field
autofocus	Focuses the input element on page load
pattern	Represents a regular expression for validating the field's value
form	Allows the elements to reference the form by including the form name

Using slide 8, explain the new attributes introduced in HTML 5. Explain the list of new attributes in HTML5.

HTML5 has introduced several new attributes that can be used with form and input elements. Attributes help the elements to perform their tasks.

Mention, there are many other attributes such as `autocomplete`, `step`, `formnovalidate`, `formaction`, and so on.

### In-Class Question:

After you finish explaining new attributes in HTML5, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which attribute is used for focusing the element on page load?

### Answer:

`Autofocus` attribute is used for focusing the element on page load.



Which attribute if specified, then the user must provide the value for the input field before the form is submitted?

### Answer:

`required` attribute

## Slide 9

Let us understand browser-based validation.

The slide has a header 'HTML5 Browser-based Validation'. Below it are five colored boxes containing text:

- HTML4 supported the use of custom JavaScript or libraries to perform validation on the client-side browsers.
- These validations ensure that the input fields are checked before the form is submitted to the server for further processing.
- The new attributes in HTML5, such as required and pattern can be used with the input elements to perform validation.
- This relieves the Web developers from writing the custom JavaScript code for performing client-side validation on the Web pages.
- HTML5 also provides advanced validation techniques that can be used with JavaScript to set custom validation rules and messages for the input elements.

At the bottom left is the copyright notice '© Aptech Ltd.' and at the bottom right is 'HTML Forms / Session 10 9'.

Using slide 9, explain the concept of browser based validation.

HTML4 supported the use of custom JavaScript or libraries to perform validation on the client-side browsers. These validations ensure that the input fields are checked before the form is submitted to the server for further processing.

The new attributes in HTML5, such as `required` and `pattern` can be used with the input elements to perform validation. This relieves the Web developers from writing the custom JavaScript code for performing client-side validation on the Web pages.

HTML5 also provides advanced validation techniques that can be used with JavaScript to set custom validation rules and messages for the input elements.

Mention, that browser based validation or client-side validation may not only reduce the round trip, but may also prevent from vulnerabilities such as injection flaws and malicious file execution.

## Slides 10 and 11

Let us understand CSS styling techniques.

**CSS Styling Techniques 1-2**

- A Web developer can enhance the form elements with the pseudo-class selectors, such as :required, :valid, and :invalid.
- The input fields which cannot be left blank while submitting the form can be displayed with an outline by styling the input field using CSS.
- The Code Snippet shows the CSS code for formatting non-empty and incorrect data input in the input element fields on the form.

```
<style>
input:required
{
    outline: 1px red solid;
    color: green ;
}
input:required:valid
{
    background-size:10px 10px;
    background-position: right top;
    background-repeat: no-repeat;
}
```

© Aptech Ltd.      HTML Forms / Session 10    10

**CSS Styling Techniques 2-2**

```
input:required:invalid
{
    background-size:10px 10px;
    background-position: right top;
    background-repeat: no-repeat;
}
</style>
</head>
<body>
<form method="get" action="try.php">
    Name: <input type="text" name="name" required="true" /><br/>
    Email: <input type="email" name="emailid" required="true" />
    <input type="submit" value="submit" />
</form>
....
```

© Aptech Ltd.      HTML Forms / Session 10    11

Using slides 10 and 11, explain the CSS styling technique for forms.

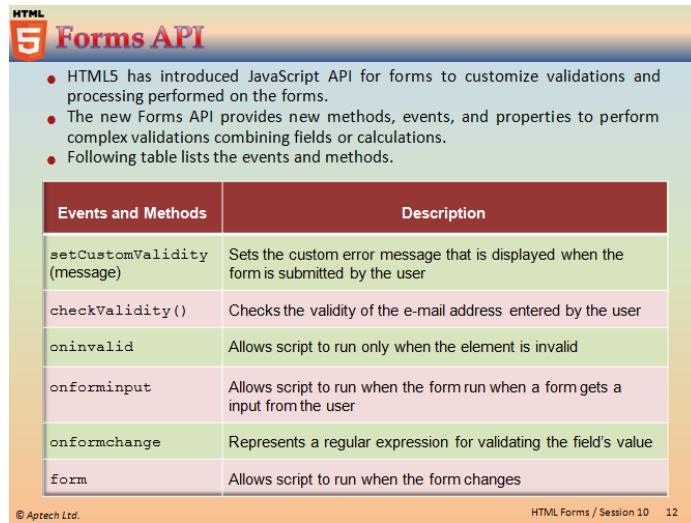
A Web developer can enhance the form elements with the pseudo-class selectors, such as :required, :valid, and :invalid.

For example, the input fields which cannot be left blank while submitting the form can be displayed with an outline. To achieve this, input field with required attribute can be styled using CSS. Applying CSS styles make it easier for user to navigate and complete the form.

Explain the code snippet that shows the CSS code for formatting non-empty and incorrect data input in the input element fields on the form.

## Slide 12

Let us understand forms API.



**HTML5 Forms API**

- HTML5 has introduced JavaScript API for forms to customize validations and processing performed on the forms.
- The new Forms API provides new methods, events, and properties to perform complex validations combining fields or calculations.
- Following table lists the events and methods.

Events and Methods	Description
<code>setCustomValidity(message)</code>	Sets the custom error message that is displayed when the form is submitted by the user
<code>checkValidity()</code>	Checks the validity of the e-mail address entered by the user
<code>oninvalid</code>	Allows script to run only when the element is invalid
<code>onforminput</code>	Allows script to run when the form run when a form gets a input from the user
<code>onformchange</code>	Represents a regular expression for validating the field's value
<code>form</code>	Allows script to run when the form changes

© Aptech Ltd.      HTML Forms / Session 10      12

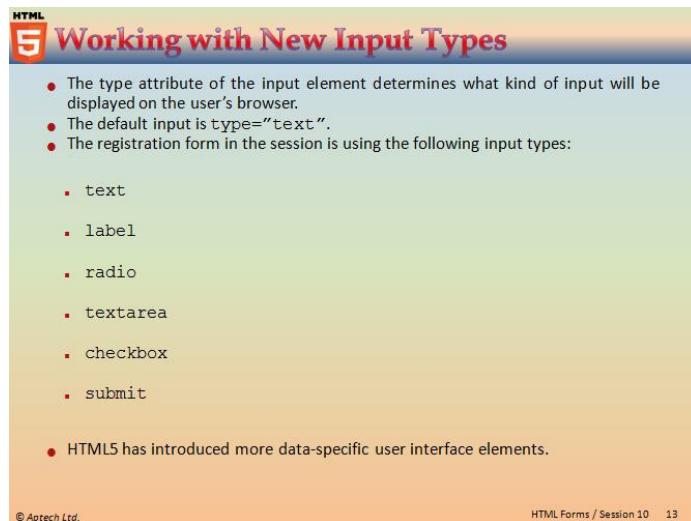
Using slide 12, explain the form API.

HTML5 has also introduced JavaScript API for forms. This API is used to customize validations and processing performed on the forms.

The new form's API provides new methods, events, and properties to perform complex validations combining fields or calculations. Explain the list of events and methods.

## Slide 13

Let us understand the value for new input types in HTML5.



**HTML5 Working with New Input Types**

- The type attribute of the input element determines what kind of input will be displayed on the user's browser.
- The default input is `type="text"`.
- The registration form in the session is using the following input types:
  - `text`
  - `label`
  - `radio`
  - `textarea`
  - `checkbox`
  - `submit`
- HTML5 has introduced more data-specific user interface elements.

© Aptech Ltd.      HTML Forms / Session 10      13

Using slides 13, explain the working of new input types.

The `type` attribute of the input element determines what kind of input will be displayed on the user's browser. The default input is `type="text"`. Apart, from "text", you can also

mention the other values for the `type` attribute that are specific to the data. Some of these are:

- text
  - label
  - radio
  - textarea
  - checkbox
  - submit

Slides 14 and 15

Let us understand e-mail address input type.

**HTML**

## E-mail Address 1-2

- The type="email" is used for specifying one or more e-mail addresses. To allow multiple addresses in the e-mail field, separate each address with comma-separator.
- In the registration form, the input type is changed from text to email as shown in the following code snippet:

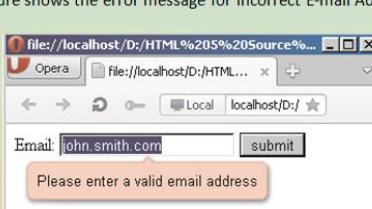
```
<form method="get" action="test.html">
    <label for="emailid">Email:</label>
    <input type="email" value="" id="emailid"
           name="emailaddress" maxlength="255" />
    <input type="submit" value="submit"/>
</form>
```

- In the code snippet, <label> tag defines a label for the element on the form.
- The for attribute of <label> tag binds it with the related element, that is email element, on the form.
- The value of for attribute must match with the value of id attribute assigned to the element.
- The email contains two attributes, namely id and name.
- The id attribute specifies a unique id for the element.
- The value of the id attribute should be unique within the document.

HTML

## E-mail Address 2-2

- The name attribute specifies a name for the `input` element.
- The look of the input is still like a plain text field, but changes are applied behind the scenes.
- Browsers, such as Firefox, Chrome, and Opera will display a default error message if user submits the form with some unrecognizable contents.
- Following figure shows the error message for incorrect E-mail Address in Chrome.



The screenshot shows an Opera browser window with the URL `file:///localhost/D:/HTML%205%20Source%`. The page contains a single input field labeled "Email:" with the value "john.smith.com". Below the input field is a "submit" button. A red callout box with a white border and black text is positioned over the input field, displaying the message "Please enter a valid email address". The browser interface includes standard navigation buttons (back, forward, search), a toolbar, and a status bar at the bottom.

Using slides 14 and 15, explain the e-mail input type. Mention, `type="email"` is used for specifying one or more e-mail addresses. To allow multiple addresses in the e-mail field, separate each address with comma-separator.

Explain the code snippet on the slide.

Tell them that:

1. In the registration form, the input type is changed from `text` to `e-mail` as shown in code snippet.
2. The `for` attribute of `<label>` tag binds it with the related element, that is `email` element, on the form. The value of `for` attribute must match with the value of `id` attribute assigned to the element.
3. The `email` type contains two attributes, namely `id` and `name`. The `id` attribute specifies a unique id for the element. The value of the `id` attribute should be unique within the document. It can be used as a reference for styles in style sheet or to access elements using DOM API in JavaScript.
4. The `name` attribute specifies a name for the input element. It can be used for referencing the elements in JavaScript or form data after a form is submitted to the server for processing. The look of the input is still like a plain text field, but changes are applied behind the scenes. Browsers, such as Firefox, Chrome, and Opera will display a default error message if user submits the form with some unrecognizable contents.

## Slide 16

Let us understand input type URL.

The slide has a yellow-to-green gradient background. At the top left is a logo with 'HTML' above a blue square containing a white '5' and the word 'URL' in blue. Below the logo is a bulleted list of three items. Underneath the list is a code snippet of HTML code. Below the code is another bulleted list of three items. At the bottom is a screenshot of an Opera browser window showing a validation error message.

- The `type="url"` input element is used for specifying a Web address.
- The look of the `url` field is a normal text field.
- The Code Snippet shows the code of `url` input type.

```
<label for="url">Enter your Web page address:</label>
<input type="url" value="" id="urlname" name="urltext"
      maxlength="255" />
<input type="submit" value="submit"/>
```

- Browsers, such as Opera, Firefox, and Chrome support validation for the `url` input type.
- While validating the URL, browsers only checks for entry with forward slash (/).
- Following figure shows the error message for incorrect URL in Chrome.

**Opera** file:///localhost/D:/HTML%205%20Source%20Codes/... Local localhost/D/ Search  
Enter Web address: site.com submit  
Please enter a valid web address

© Aptech Ltd. HTML Forms / Session 10 16

Using slide 16, explain the type `url`.

Mention, `type="url"` input element is used for specifying a Web address. The look of the `url` field is a normal text field.

Explain the code snippet that shows the code of `url` input type.

Browsers, such as Opera, Firefox, and Chrome supports validation for the `url` input type. While validating the URL, browser only checks for entry with forward slash (/). For example,

a URL such as `x://mysite.com` will be considered as valid, even though `x://` is not a real protocol.

## Slide 17

Let us understand telephone number input type.

**HTML 5 Telephone Number**

- The `type="tel"` input element is used for accepting telephone numbers.
- The `tel` type does not impose a particular pattern.
- It supports characters, numbers, and special characters except new lines or carriage returns.
- A user can enforce a pattern for `tel` input type by using placeholder, pattern attribute, or a JavaScript code for performing client-side validation.
- The Code Snippet shows the code for including input type on the registration form.

```
<label for="telno">Telephone Number:</label>
<input type="tel" value="" id="telno" name="telephone_no"
maxlength="10" />
```

© Aptech Ltd.      HTML Forms / Session 10      17

Using slide 17, explain the type `tel` used for telephone number.

The `type="tel"` input element is used for accepting telephone numbers. As compared to `email` and `url` types, the `tel` type does not impose a particular pattern. It supports characters, numbers, and special characters except new lines or carriage returns. The reason for not imposing any pattern for `tel` type is that different countries support various lengths and punctuations in phone numbers. Thus, there cannot be a standard format for them.

A user can enforce a pattern for `tel` input type by using `placeholder` or `pattern` attribute. A JavaScript code can also be provided for performing client-side validation on the `tel` input type.

Explain the code snippet which shows the code for including `tel` input type on the registration form.

## Slide 18

Let us understand number input type.

**HTML 5 Number**

- The input type="number" is used for accepting only number values.
- The input element displayed for number type is a spinner box.
- The user can either type a number or click the up or down arrow to select a number in the spinner box.
- The Code Snippet shows the code for including number input type on the form.

```
<label for="stud_age">Age:</label>
<input type="number" value="15" id="stud_age"
       name="studentage" min="15" max="45" step="1" />
<input type="submit" value="submit"/>
```

- In the code snippet, the number input type has attributes, such as min and max to specify the minimum and maximum value for the input.
- Following figure shows the input type in Opera.

Opera browser window showing a form with an input field for age set to 15. The input field has a value of "15" and a "submit" button below it.

© Aptech Ltd.      HTML Forms / Session 10      18

Using slide 18, explain the type number.

The input type="number" is used for accepting only number values. The input element displayed for number type is a spinner box. The user can either type a number or click the up or down arrow to select a number in the spinner box.

Explain the code snippet which shows the code for including number input type on the form. In the code, the number input type has attributes, such as min and max to specify the minimum and maximum value for the input.

## Slides 19 and 20

Let us understand range input type.

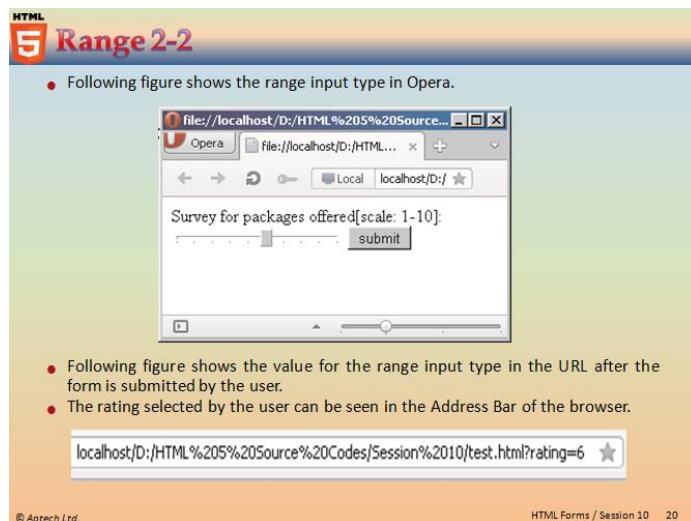
**HTML 5 Range 1-2**

- The input type="range" is similar to number type and displays a slider control on the page.
- The range type is used when the exact value is not required in the input.
- For example, an online survey form asking the clients to rate the products may not receive exact values in the ratings.
- The Code Snippet shows the code for including range input type with attributes min and max.

```
<label>Survey for packages offered[scale: 1-10]:</label>
<input type="range" name="rating" min="1" max="10" />
<input type="submit" value="submit"/>
```

- In code snippet, the range input type contains attributes, such as min, max, step, and value.
- The min and max attributes are used to specify the minimum and maximum value allowed for a range and are set to 1 and 10 respectively.
- The step attribute specifies the intervals for incrementing the value.
- The value attribute specifies the default value for the range.
- The default value is the midpoint of the range specified.

© Aptech Ltd.      HTML Forms / Session 10      19



Using slides 19 and 20, explain the type range and code snippet for it.

The `input type="range"` is similar to `number` type and displays a slider control on the page. The `range` type is used when the exact value is not required in the input. In other words, the value from this type is not accurate. For example, an online survey form asking the clients to rate the products may not receive exact values in the ratings.

Explain the code snippet that shows the code for including `range` input type with attributes `min` and `max`.

The `min` and `max` attributes are used to specify the minimum and maximum values allowed for a range and are set to 1 and 10 respectively. The `step` attribute specifies the intervals for incrementing the value. The value of `step` attribute is 1 by default. The `value` attribute specifies the default value for the range. The default value is the midpoint of the range specified.

## Slides 21 to 27

Let us understand the date and time input type.

**Date**

- HTML5 has introduced several new input types for date and time.
- The format for all these date and time types is according to the ISO standards.
- At present only Opera provides the support for date element by displaying a calendar control.

```
<label for="startdate">Date:</label>
<input type="date" id="startdate" name="startdate"
       min="2000-01-01"/>
<input type="submit" value="Submit" id="btnSubmit"></input>
```

- This input type contains only date in year, month, and day format. The time part is not supported by date type.
- The code snippet shows the code of the date input type.
- In the code snippet, all date input types have min and max attributes to set the range for the dates.
- The default value for date input type is based on the browser.
- Thus, it is advisable to set the minimum and maximum value for the date type.

© Aptech Ltd.

HTML Forms / Session 10 21

Following figure shows the input type.

Following figure shows the value sent for the date input type after the form is submitted by the user.

file:///localhost/D:/HTML%205%20Source%20Codes/Session%2010/test.html?startdate=2012-07-10

© Aptech Ltd.

HTML Forms / Session 10 22

**Month**

- The type="month" includes only the year and month in the input.
- The code snippet shows the syntax of month input type.

```
<label for="stmonth">Month:</label>
<input type="month" id="stmonth" name="startmonth" />
<input type="submit" value="Submit"/>
```

- Browser such as Opera will display the date picker for selecting month.
- On selecting any day from the calendar, the whole month is selected.
- Following figure shows the date picker for the month input type.

© Aptech Ltd.

HTML Forms / Session 10 23

**HTML 5 Date and Time 4-7**

### > Week

- The input type="week" provides a similar interface as displayed for date type and selects the entire week.
- The Code Snippet shows the syntax of the week input type.

```
<label>Week:</label>
<input type="week" name="week" />
<input type="submit" value="submit"/>
```

- Following figure shows the week input type in Opera.

Opera browser window showing a week calendar for March 2012. The calendar highlights the week starting from Monday, March 12, 2012. A 'Submit' button is visible at the bottom right of the calendar.

© Aptech Ltd.      HTML Forms / Session 10      24

**HTML 5 Date and Time 5-7**

### > Time

- The input type="time" displays a time of day in hours and minutes format (24-hour clock).
- The Code Snippet shows the syntax of the time input type.

```
<label>Time:</label>
<input type="time" name="time" />
<input type="submit" value="submit"/>
```

- Following figure shows the time input type in Opera.

Opera browser window showing a form with a 'Time' input field containing '00:09'. A 'Submit' button is next to it. Below the input field is a slider control.

© Aptech Ltd.      HTML Forms / Session 10      25

**HTML 5 Date and Time 6-7**

### > Datetime

- The input type="datetime" includes full date and time in the input.
- The input includes a date part and a time part which is represented as Coordinated Universal Time (UTC).
- Thus, UTC time will be displayed with 'T' followed by a 'Z'.
- The Code Snippet shows the syntax of datetime input type.

```
<label for="mydatetime">Date-Time:</label>
<input type="datetime" name="mydatetime" />
<input type="submit" value="submit"/>
```

© Aptech Ltd.      HTML Forms / Session 10      26

> Datetime-local

- This input type is similar to datetime input type, except that the time zone is omitted for input type="datetime-local".

© Aptech Ltd. HTML Forms / Session 10 27

Using slides 19 to 27, explain the input type date. HTML5 has introduced several new input types for date and time. The format for all these date and time types is according to the ISO standards. At present only Opera provides the support for date element by displaying a calendar control. The date input type contains only date in year, month, and day format. The time part is not support by date type.

Explain the code snippet that shows the code of the date input type.

In the code, all date input types have min and max attributes to set the range for the dates. The default value for date input type is based on the browsers. Different browsers take different dates as default. Thus, it is advisable to set the minimum and maximum value for the date type.

Using slide 22, show the output of input type date.

Figure shows the date input type. Next figure shows the value sent for the date input type after the form is submitted by the user.

Using slide 23, explain the input type month. The type="month" includes only the year and month in the input. Code Snippet shows the syntax of month input type.

Browser such as Opera will display the date picker for selecting month. On selecting any day from the calendar, the whole month is selected. Figure shows the date picker for the month input type.

Using slide 24 explain the input type week. The input type="week" provides a similar interface as displayed for date type and selects the entire week. Code Snippet shows the syntax of the week input type. Figure shows the week input type in Opera.

In the figure, 12th March is selected and the combo box also displays the 2012-W12 as the 12th week is selected.

Similar to date input types, there are new `time` input types introduced in HTML5.

Using slide 26, explain the input type `time`.

Mention, `input type="time"` displays a time of day in hours and minutes format (24-hour clock). Code Snippet shows the syntax of time input type. Figure shows the time input type in Opera.

Using slides 26 and 27, explain the input type `datetime` and `color`.

The `input type="datetime"` includes full date and time in the input. The input includes a date part and a time part which is represented as Coordinated Universal Time (UTC). Thus, UTC time will be displayed with 'T' followed by a 'Z'. For example, an input, such as 2012-07-03T12%3A05Z is interpreted as 2012-07-03T12:01.

Code Snippet shows the syntax of `datetime` input type. Figure on slide 27 shows the `datetime` input type in Opera.

#### **Datetime-local:**

This input type is similar to `datetime` input type, except that the time zone is omitted for `input type="datetime-local"`.

#### **In-Class Question:**

After you finish explaining the date and time input types, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which input type selects the week based on the date as input?

#### **Answer:**

`Input type week` selects the week based on the date as input.

## Slide 28

Let us understand the input type `color`.

**HTML5 Color**

- HTML5 provides a predefined interface for selecting the colors using input type="color".
- The input value from the color input field is a hexadecimal number.
- For example, #00FF00 represents a hexadecimal RGB color value.
- The Code Snippet shows the usage of color input type to display a color picker on the Web page.

```
<label>Color:</label>
<input type="color" name="mycolor" />
<input type="submit" value="submit"/>
```

- Following figure shows the color input type in Opera.

© Aptech Ltd.      HTML Forms / Session 10      28

Using slide 28, explain the input type `color`. Mention, HTML5 provides a pre-defined interface for selecting the colors using input type="color". The input value from the `color` input field is a hexadecimal number. For example, #00FF00 represents a hexadecimal RGB color value.

At present, the `color` input type is supported on Opera browser and some new smart phones. Code snippet on slide 28 shows the usage of `color` input type to display a color picker on the Web page.

Figure shows the `color` input type in Opera.

## Slide 29

Let us understand new form attributes.

**HTML5 New Form Attributes**

- HTML5 has provided several new attributes that perform the validations without writing JavaScript snippets for them.
- These attributes perform the following tasks:
  - Check data provided by users with the regular expression pattern assigned to the fields
  - Inform users with appropriate errors
  - Check that the required fields are not left empty by the users
  - Enable multiple values for the fields, if provided
- These attributes can be used to support scripting drawbacks, without actually hard coding them in the Web pages.
- Browsers that do not understand these new attributes will ignore them.

© Aptech Ltd.      HTML Forms / Session 10      29

Using slide 29, explain the new form attributes introduced in HTML5.

Earlier, Web developers were required to write JavaScript snippets for performing the validations on the data entered by the users in form fields. HTML5 has provided several new attributes that perform the validations without writing JavaScript snippets for them.

Explain the characteristics of new attributes.

The new attributes can be used to support scripting drawbacks, without actually hard coding them in the Web pages. Browsers that do not understand these new attributes will ignore them.

Slides 30 and 31

Let us understand the required attribute.

**HTML**

**5 Required 1-2**

- This is a boolean attribute that informs the browser to submit the form only when the required fields are not left empty by the users.
- The input type elements, such as button, range, and color cannot be set for required attribute as they have a default value.
- Different Web browsers such as Opera, Chrome, and Firefox provide different error messages, such as 'This is a required field', or 'Please fill out this field' for required attribute.
- The Code Snippet shows assignment of required attribute to the name field on the registration form.

```
<label>Name:  </em>
</label> <br>
<input type="text" value="" name="first" size="8" tabindex="1"
    required="true"/>
<input type="text" value="" name="last" size="14" tabindex="2"
    required="true"/>
<input type="submit" value="submit"/>
```

**HTML**

## Required 2-2

- Following figure shows the message of required attribute in Opera.

The screenshot shows an Opera browser window with the URL `file:///localhost/D:/HTML%205%20Source%20Codes/S...` . The page contains a simple form with a label "Name: \*". A red validation message box with the text "This is a required field" is overlaid on the "submit" button area, indicating that the name field is mandatory.

Using slides 30 and 31, explain the required attribute in detail.

Mention, `required` is a Boolean attribute that informs the browser to submit the form only when the `required` fields are not left empty by the users. The `input type` elements, such as `button`, `range`, and `color` cannot be set for required attribute as they have a default value.

Different Web browsers such as Opera, Chrome, and Firefox provide different error messages, such as 'This is a required field', or 'Please fill out this field' for required attribute.

Explain the code snippet showing assignment of `required` attribute to the name field on the registration form.

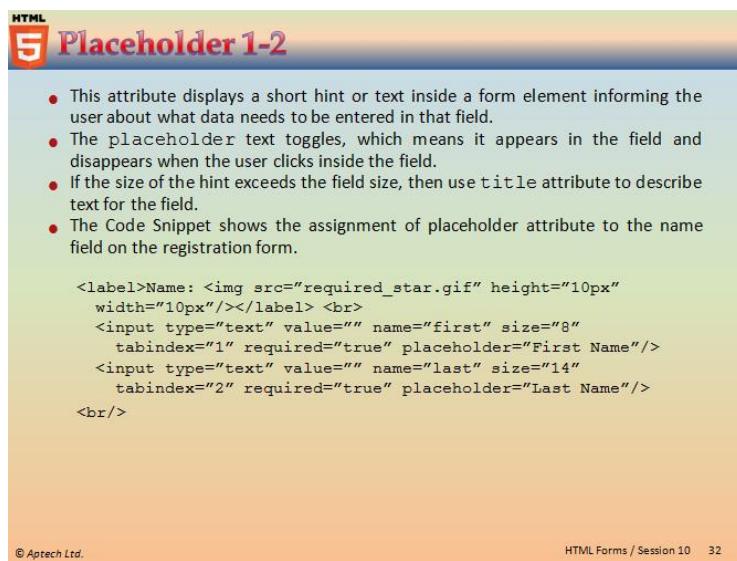
Also, explain the output for the `required` attribute. Figure shows the message of required attribute in Opera.

#### **Tips:**

The required attribute works with the following input types: `text`, `search`, `url`, `tel`, `email`, `password`, date pickers, `number`, `checkbox`, `radio`, and `file`.

#### **Slides 32 and 33**

Let us understand the `placeholder` attribute.



The screenshot shows a slide titled "Placeholder 1-2" with the HTML5 logo. The slide content includes a bulleted list of tips and a code snippet. The footer indicates it's from "Aptech Ltd." and "HTML Forms / Session 10 32".

- This attribute displays a short hint or text inside a form element informing the user about what data needs to be entered in that field.
- The `placeholder` text toggles, which means it appears in the field and disappears when the user clicks inside the field.
- If the size of the hint exceeds the field size, then use `title` attribute to describe text for the field.
- The Code Snippet shows the assignment of `placeholder` attribute to the name field on the registration form.

```
<label>Name: </label> <br>
<input type="text" value="" name="first" size="8" tabindex="1" required="true" placeholder="First Name"/>
<input type="text" value="" name="last" size="14" tabindex="2" required="true" placeholder="Last Name"/>
<br/>
```

© Aptech Ltd.      HTML Forms / Session 10 32



Using slides 32 and 33, explain the `placeholder` attribute.

Mention, `placeholder` attribute displays a short hint or text inside a form element. This informs the user about what data needs to be entered in that field. The placeholder text toggles, which means it appears in the field and disappears when the user clicks inside the field. Earlier, Web developers provided this functionality through JavaScript snippets which is now done in the browsers with the help of `placeholder` attribute. At present, all the browsers such as Chrome, Safari, Opera, and Firefox support the `placeholder` attribute. If the size of the hint exceeds the field size, then use `title` attribute to describe text for the field.

Code Snippet shows the assignment of `placeholder` attribute to the name field on the registration form.

Figure shows the message of `placeholder` attribute in Opera.

#### In-Class Question:

After you finish explaining the `placeholder` attribute, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which attribute is used to give hint to the user about the data to be entered?

#### Answer:

`Placeholder` attribute is used to give hint to the user about the data to be entered.

## Slides 34 and 35

Let us understand pattern attribute.

**HTML 5 Pattern 1-2**

- The pattern attribute uses regular expressions for validating the fields.
- The data entered by the user must match with the pattern specified in the regular expression.
- This helps to limit the input accepted from the user.
- To inform the users about the expected pattern for the data, use the title attribute, which is displayed as a tool tip when pointer is moved over the field.
- The Code Snippet shows the assignment of pattern attribute to the phone number field on the registration form.

```
<label>Phone number:</label>
<input type="tel" value="" size="4" maxlength="5" tabindex="11" required="true" placeholder="Code" pattern="[+0-9]{1,4}" title="Format: (+)99(99)"/>
<label>-</label>
<input type="tel" value="" size="10" maxlength="12" tabindex="13" required="true" placeholder="Number" pattern="[0-9]{8,}" title="Minimum 8 numbers"/>
```

© Aptech Ltd.      HTML Forms / Session 10    34

**HTML 5 Pattern 2-2**

- In the code snippet, `[+0-9]` pattern indicates that only special character '+' as well as numbers are allowed.
- Also, `{1, 4}` refers to the length of the numbers, that is between 1 and 4.
- Similarly, `{8,}` means minimum eight numbers are allowed in the tel input type field.
- Following figure shows the message of pattern attribute in Opera.

© Aptech Ltd.      HTML Forms / Session 10    35

Using slides 34 and 35, explain the pattern attribute.

Mention, pattern attribute uses regular expressions for validating the fields. The data entered by the user must match with the pattern specified in the regular expression. This helps to limit the input accepted from the user.

While including regular expressions through pattern attribute, it informs the users about the expected pattern for the data. This can be achieved in the current browsers using the title attribute, which is displayed as a tool tip when the users move the pointer over the field.

Example, a zip code should be validated with five numbers on the form. There is no pre-defined input type to restrict the input to numbers of specified length. Thus, pattern attribute can be used to create user-defined check values for the field. Also, a title attribute can be used to customize the error message displayed for the field.

Explain the code snippet which shows the assignment of pattern attribute to the phone number field on the registration form.

In the code, `[+0-9]` pattern indicates that only special character '+' as well as numbers are allowed. Also, `{1, 4}` refers to the length of the numbers, that is between 1 and 4. Similarly, `{8, }` means minimum eight numbers are allowed in the `tel` input type field.

Figure shows the message of pattern attribute in Opera.

## Tips:

Use the global title attribute to describe the pattern to help the user.

Slides 36 and 37

Let us understand multiple attribute.

HTML  
5

## Multiple 1-2

- This is a boolean attribute that allows multiple values for some input types.
- This was available only for select input type in the earlier version of HTML.
- HTML5 allows multiple attribute with input types, such as email and file.
- If assigned, it allows selection of multiple files, or include several e-mail addresses in the email field separated by comma separator.
- The Code Snippet shows the assignment of multiple attribute to the e-mail address field on the registration form.

```
<label>Email Address:</label>
<input type="email" value="" name="emaillid" maxlength="255"
tabindex="5" required="true" placeholder="Email Address"
multiple/>
```

- In the code snippet, multiple attribute will allow insertion of multiple e-mail addresses in the field.
- Every e-mail address will be validated individually by the browser.

© Interact I Ltd.

HTML: Forms / Session 10

36

The screenshot shows an Opera browser window with the title "Multiple 2-2". The address bar displays "file:///localhost/D:/HTML%205%20Source%20...". The main content area contains a form with an "Email Address:" label and a text input field containing "john1 smith". A red error message box is overlaid on the form, stating "Please enter a valid email address".

Using slides 36 and 37, explain the multiple attribute.

Mention, `multiple` is a Boolean attribute that allows multiple values for some input types. This was available only for `select` input type in the earlier version of HTML.

HTML5 allows `multiple` attribute with input types, such as `email` and `file`. If assigned, it allows selection of multiple files, or include several e-mail addresses in the `email` field separated by comma separator. At present, browsers such as Chrome, Opera, and Firefox support `multiple` attribute for `email` and `file` element.

Explain code snippet which shows the assignment of `multiple` attribute to the e-mail address field on the registration form. In the code, `multiple` attribute will allow insertion of multiple e-mail addresses in the field. Every e-mail address will be validated individually by the browser.

Figure shows the validation of `multiple` e-mail address.

### Slides 38 and 39

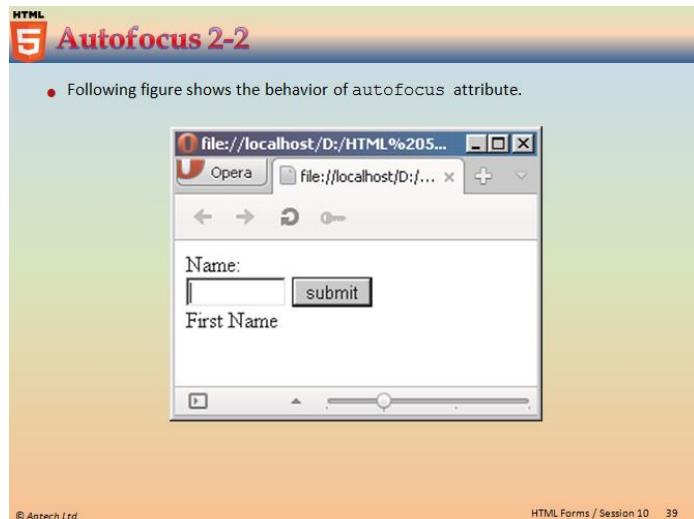
Let us understand `autofocus` attribute.

**Autofocus 1-2**

- The `autofocus` attribute will focus on the input field on page load.
- However, depending upon the situation, it will not move the focus away if the user has selected some other field.
- Only one element can be focused with `autofocus` attribute on a particular page while loading.
- The Code Snippet shows the assignment of `autofocus` attribute to the first name field on the registration form.

```
<label>Name:</label>
<br>
<input type="text" value="" name="first" size="8"
       tabindex="1" placeholder ="First Name" autofocus/>
<input type="submit" value="submit"/>
<br>
<label>First Name</label>
```

© Aptech Ltd.      HTML Forms / Session 10      38



Using slides 38 and 39, explain the `autofocus` attribute.

Earlier, Web developers were using JavaScript code to set the focus to the input field on page load. The purpose was to force the focus over the input field, even if the user selected some other element while page is still loading. As a result of the JavaScript code, control moves to the input field upon completion of page load. This way, regardless of what the user selected, the focus would always be on the input field.

To provide an easier solution for this behavior, HTML5 has introduced `autofocus` attribute for the form elements. The `autofocus` attribute will focus on the input field on page load. However, depending upon the situation, it will not move the focus away if the user has selected some other field. Only one element can be focused with `autofocus` attribute on a particular page while loading.

Explain code snippet which shows the assignment of `autofocus` attribute to the first name field on the registration form.

Figure shows the behavior of `autofocus` attribute.

## Slide 40

Let us understand form in HTML5.

The slide has a blue header bar with the text 'HTML 5 Form'. The main content area has a green-to-orange gradient background. It contains three bullet points and a code snippet:

- Earlier, all the form controls need to be provided between the opening and closing <form> tag.
- In HTML5, elements can be inserted at any place in the document and they can reference the form using the form attribute.
- The Code Snippet shows the association of an element with the form on the Web page.

```
<body>
  <input type="text" name="mytext" id="mytext" form="myform"/>
  . . .
  <form id="myform">
  . . .
  </form>
</body>
```

Below the code snippet, there are two more bullet points:

- In the code snippet, the form is declared with an id attribute.
- The value of the id attribute is assigned to the input element using form attribute.

At the bottom left is a small copyright notice: © Aptech Ltd. At the bottom right is the text: HTML Forms / Session 10 40.

Using slide 40, explain the `form` element.

Earlier, all the `form` controls need to be provided between the opening and closing `<form>` tag. In HTML5, elements can be inserted at any place in the document and they can reference the form using the `form` attribute.

Explain code snippet which shows the association of an element with the form on the Web page. In the code, the form is declared with an `id` attribute. The value of the `id` attribute is assigned to the `input` element using `form` attribute.

New attributes for `form` element are:

- `autocomplete`
- `novalidate`

## Slides 41 and 42

Let us understand `autocomplete` attribute.

**HTML5 Autocomplete Attribute 1-2**

- HTML5 offers an `autocomplete` attribute which provides control on prefilled values displayed in the fields.
- It must be specified on the `form` element which applies for all input fields or on particular input fields.
- The input element that can support autocomplete are `text`, `url`, `tel`, `password`, `datepickers`, `range`, and `color`.
- The `autocomplete` feature comprises two states namely, `on` and `off`. The `on` state indicates that the data that is not sensitive can be remembered by the browser.
- The `off` state indicates that the data will not be remembered. Such data may be sensitive and not safe for storing with the browsers.
- By default, many browsers have the autocomplete feature enabled in them.
- The browsers that do not support autocomplete, can be turned on or off for this behavior by specifying `autocomplete` attribute.

© Aptech Ltd.      HTML Forms / Session 10    41

**HTML5 Autocomplete Attribute 2-2**

- Following figure shows the behavior of `autocomplete` attribute in Chrome.



- The Code Snippet demonstrates disabling the default behavior of `autocomplete` attribute.

```
E-mail: <input type="email" name="email" autocomplete="off" />
<input type="submit" value="submit"/>
```

© Aptech Ltd.      HTML Forms / Session 10    42

Using slide 42, explain the `autocomplete` attribute.

Mention, many browsers help user in filling forms by providing data in the input fields, such as e-mail and tel, based on their earlier input. In many situations, the `autocomplete` behavior may not be secure, especially with certain fields accepting password or credit card number data.

HTML5 offers an `autocomplete` attribute which provides control on prefilled values displayed in the fields. It must be specified on the `form` element which applies for all input fields or on particular input fields. The input element that can support autocomplete are `text`, `url`, `tel`, `password`, `datepickers`, `range`, and `color`.

The `autocomplete` feature comprises two states namely, On and Off. The On state indicates that the data that is not sensitive can be remembered by the browser. Similarly,

the Off state indicates that the data will not be remembered. Such data may be sensitive and not safe for storing with the browsers. Hence, user needs to explicitly provide the data each time while filling the form.

By default, many browsers have the autocomplete feature enabled in them. The browsers that do not support autocomplete, can be turned On or Off for this behavior by specifying autocomplete attribute either on the form or specific input elements.

Figure shows the behavior of autocomplete attribute in Chrome.

Explain code snippet which demonstrates to disable the default behavior of autocomplete attribute.

#### Tips:

In some browsers you may need to activate the autocomplete function for this to work.

### Slide 43

Let understand new form elements.

**HTML5 New Form Elements**

- HTML5 has introduced some new elements that can be incorporated in the Web pages.
- These new elements are specifically designed to be used with the JavaScript.
- When combined with JavaScript, these new elements can be more functional.
- At present, all the browsers do not provide the support for these new elements.
- If the control is not supported by the browser, then it displays element as a text field.
- Opera provides the support for all the new form elements.
  - Datalist
  - Progress
  - Meter
  - Output

© Aptech Ltd.      HTML Forms / Session 10      43

Using slide 43, explain the new form elements.

HTML5 has introduced some brand new elements that can be incorporated in the Web pages. These new elements are specifically designed to be used with JavaScript, as and when combined with JavaScript, these new elements can be more functional.

At present, Opera provides the support for all the new form elements. The new form elements introduced in HTML5 are as follows:

- Datalist
- Progress
- Meter

- Output

## Slides 44 to 46

Let us understand the `datalist` element.

**Datalist 1-3**

Datalist is a form-specific element. It provides a text field with a set of predefined list of options that are displayed in a drop-down list.

When the text field receives focus, a list of options is displayed to the user.

The `<datalist>` element is very similar to standard `<select>` element available in earlier HTML.

The only difference in datalist is that it allows the user to enter data of their choice or select from the suggested list of options.

The lists of options for the `<datalist>` element are placed using the `option` element.

Then, the datalist is associated with an input element using the `list` attribute.

The value of the `list` attribute is the value of `id` attribute provided with the `<datalist>` element.

© Aptech Ltd.

HTML Forms / Session 10 44

**Datalist 2-3**

- At present, only Opera browser provides the support for the datalist.
- The Code Snippet shows the syntax of providing the `<datalist>` element on the form.

```
<label> Select the mode of payment: </label>
<input type="text" name="payment" list="paymentlist" />
<datalist id="paymentlist">
    <option value="Cash-on-Delivery">
    <option value="Net Banking">
    <option value="Credit Card">
    <option value="Debit Card">
    <option value="e-Gift Voucher">
</datalist>
<input type="submit" value="submit"/>
```

- As shown in the code snippet, a datalist requires `value` attribute to be added with the `<option>` tag.
- Values nested between the opening and closing `<option>` tag will not be displayed in the datalist menu.

© Aptech Ltd.

HTML Forms / Session 10 45

**Datalist 3-3**

- Following figure shows the `<datalist>` element in Opera.

The screenshot shows an Opera browser window with the title bar "file:///localhost/D:/HTML%205%20Source%20...". The main content area displays a form with a label "Select the mode of payment:" followed by a text input field and a "submit" button. Below the input field is a dropdown menu containing five options: "Cash-on-Delivery", "Net Banking", "Credit Card", "Debit Card", and "e-Gift Voucher".

© Aptech Ltd.

HTML Forms / Session 10 46

Using slides 44 to 46, explain the new form element `datalist`.

Datalist is a form-specific element. It provides a text field with a set of pre-defined list of options that are displayed in a drop-down list. When the text field receives focus, a list of options is displayed to the user.

The `datalist` element is very similar to standard `select` element available in earlier HTML. The only difference in `datalist` is that it allows the user to enter data of their choice or select from the suggested list of options.

The list of options for the `<datalist>` element are placed using the `option` element. Then, the `datalist` is associated with an `input` element using the `list` attribute. The value of the `list` attribute is the value of `id` attribute provided with the `<datalist>` element. The same `datalist` can be associated with several input fields. At present, only Opera browser provides the support for the `datalist`.

Explain the code snippet for `datalist` and output for it. In the code, a `datalist` requires `value` attribute to be added with the `<option>` tag. Values nested between the opening and closing `<option>` tag will not be displayed in the `datalist` menu.

As shown in figure, the `datalist` is displayed with simple text field that shows the suggested list of options in a drop-down list on focus.

#### Tips:

The `<datalist>` tag is used to provide an "autocomplete" feature on `<input>` elements. Users will see a drop-down list of pre-defined options as they input data.

#### In-Class Question:

After you finish explaining the `<datalist>` tag, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What is difference between `datalist` and `select` tag?

#### Answer:

The `datalist` tag allows the user to enter data of their choice or select from the suggested list of options, while `select` tag only allows user to select the option.

## Slides 47 and 48

Let us understand progress element.

**HTML 5 Progress 1-2**

- The `progress` element represents the current status of a task, which gradually changes as the task heads for completion.
- This is not a form-specific element.
- For example, when the user downloads any file from a particular Web page, the download task is represented as a progress bar.
- The Code Snippet shows the syntax for providing `progress` element on the form.

```
<label> Downloading status: </label>
<progress value="35" max="100" ></progress>
<input type="submit" value="submit"/>
```

- As shown in the code snippet, the `progress` element contains two attributes namely, `max` and `value`.
- The `max` attribute declares the maximum value for the task to be processed for its completion.
- The `value` attribute indicates how much task has been processed so far.

© Aptech Ltd. HTML Forms / Session 10 47

**HTML 5 Progress 2-2**

- Following figure shows the `progress` element in Opera.

The screenshot shows the Opera browser window with the URL `file:///localhost/D:/HTML%205%20Source%...`. Inside the browser, there is a form with a label "Downloading status:" followed by a `progress` bar and a "submit" button. The progress bar is partially filled, indicating a value of about 35%. Below the browser window, there is a toolbar with various icons.

© Aptech Ltd. HTML Forms / Session 10 48

Using slides 47 and 48, explain the `progress` element.

The `progress` element represents the current status of a task, which gradually changes as the task heads for completion.

This is not a form-specific element. For example, when the user downloads any file from a particular Web page, the download task is represented as a progress bar.

Explain code snippet which shows the syntax for providing `progress` element on the form. In the code, the `progress` element contains two attributes namely, `max`, and `value`. The `max` attribute declares the maximum value for the task to be processed for its completion.

The `value` attribute indicates how much task has been processed so far. Figure shows the `progress` element in Opera.

## Slides 49 and 50

Let us understand `meter` element.

**Meter 1-2**

- The `meter` element represents a measurement scale for a known range.
- The known range has a definite minimum and maximum values to measure the data on the scale.
- For example, a `meter` element can be used to represent measurements, such as disk usage space, fraction value, or significance of a query result.
- All these have a known maximum value defined for them.
- The `meter` element cannot indicate age, height, or weight, as maximum values for them cannot be specified.
- The Code Snippet shows the code of the `meter` element.

```
<label> Total score of marks: </label>
0 &nbsp; <meter min="0" max="400" value="180"
title="numbers scored" low="120" high="300">
</meter> &nbsp; 400<br/>
<input type="submit" value="submit"/>
```

- In the code snippet, the `meter` element contains six attributes that are used to determine the measurements in the known range.
- The `min` and `max` attribute specifies the minimum and maximum value that sets bounds for the range.

© Aptech Ltd.

HTML Forms / Session 10 49

**Meter 2-2**

- The default value for the `max` attribute is 1.
- The `value` attribute specifies the current measured value.
- The `high` and `low` attributes specifies the range of values that can be considered as high or low for the given range.
- For example, in the given range of scores, the range of values below 120 will be considered low, but anything above 300 will be considered as high.
- There is another attribute named `optimum` which refers to the ideal value for the measurement.
- Following figure shows the `meter` element in Opera.

© Aptech Ltd.

HTML Forms / Session 10 50

Using slides 49 and 50, explain the `meter` element.

Mention that `meter` element represents a measurement scale for a known range. The known range has definite minimum and maximum values to measure the data on the scale.

For example, a meter can be used to represent measurements, such as disk usage space, fraction value, or significance of a query result. All these have a known maximum value defined for them.

The `meter` element cannot indicate age, height, or weight, as maximum values for them cannot be specified.

Explain that code snippet shows the code of the `meter` element. In the code, the `meter` element contains six attributes that are used to determine the measurements in the known range. The `min` and `max` attribute specifies the minimum and maximum value that sets

bounds for the range. The default value for the max attribute is 1. The value attribute specifies the current measured value. The high and low attributes specifies the range of values that can be considered as high or low for the given range.

For example, in the given range of scores, the range of values below 120 will be considered low, but anything above 300 will be considered as high.

Mention another attribute named optimum for meter element which refers to the ideal value for the measurement.

Figure shows the meter element in Opera.

## Slides 51 and 52

Let us understand output element.

**HTML 5 Output 1-1**

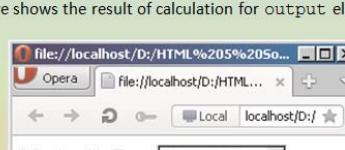
- The output element displays the results of a calculation on a form.
- The result values displayed in the output element are processed from the other form elements.
- For example, the output element might be used to display the total cost on the purchase items after calculating discount amount in a registration form or purchase order form.
- The Code Snippet shows the calculation of data from other form elements to be displayed in the output element.

```
<form oninput="x.value = parseInt(type.value)*  
    parseInt(duration.value)">  
    <label>Membership Type:</label>  
    <select name="type">  
        <option value="400">Gold - $400</option>  
        <option value="500">Silver - $500</option>  
        <option value="600">Platinum - $600</option>  
    </select>  
    <label>Duration [years]:</label>  
    <input type="number" value="0" name="duration"  
        min="1" max="5" step="1" />  
    <label> Annual Payment Fees: $.</label>  
    <output name="x" for="type duration"></output>
```

**HTML**

## Output 2-2

- In the code snippet, `for` attribute relates the output element with the elements whose values are taken for calculation.
- The form `oninput` event handles the `input` event which gets fired whenever the value of the elements change on receiving input from the user.
- A JavaScript code can also be written to update the values for the output element.
- Following figure shows the result of calculation for output element.



The screenshot shows an Opera browser window with two tabs open, both displaying the URL `file:///localhost/D:/HTML%205%205o...`. The active tab contains a form with the following fields:

- Membership Type: `Silver - $500` (selected in a dropdown menu)
- Duration [years]: `3` (entered in a text input field)
- Annual Payment Fees: `$ 1500` (calculated and displayed below the form)

Using slides 51 and 52, explain the output element.

The `output` element displays the results of a calculation on a form. The result values displayed in the `output` element are processed from other form elements. For example, the `output` element might be used to display the total cost on the purchase items after calculating discount amount in a registration form or purchase order form.

Explain the code snippet that shows the calculation of data from other form elements to be displayed in the `output` element. In the code, `for` attribute relates the `output` element with the elements whose values are taken for calculation. The form `oninput` event handles the input event which is fired whenever the value of the elements change on receiving input from the user.

A JavaScript code can also be written to update the values for the `output` element.

Figure shows the result of calculation for `output` element.

### Slide 53

Let us summarize the session.

**HTML5 Summary**

- HTML5 provides a great enhancement to Web forms.
- Creation of form is made easier for Web developers by standardizing them with rich form controls.
- HTML5 introduces new form elements such as new input types, new attributes, browser-based validation, CSS3 styling techniques, and forms API.
- HTML5 provides new input types that are data-specific user interface elements such as email, url, number, range, date, tel, and color.
- The new form elements introduced in HTML5 are datalist, progress, meter, and output.
- HTML5 has provided several new attributes that performs the validations without writing JavaScript snippets for them.
- In HTML5, one can use the submit input type for form submission.

© Aptech Ltd.      HTML Forms / Session 10      53

In slide 53, you will summarize the session. You will end the session, with a brief summary of what has been taught in the session.

### 10.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session. You should also explore the HTML5 audio and video elements that are offered with the next session.

#### Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 11 – HTML5 Audio and Video

---

## 11.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 11.1.1 Objectives

By the end of this session, the learners will be able to:

- Describe the need for multimedia in HTML5
- List the supported media types in HTML5
- Explain the audio elements in HTML5
- Explain the video elements in HTML5
- Explain the accessibility of audio and video elements
- Describe how to deal with non-supporting browsers

### 11.1.2 Teaching Skills

To teach this session, you should be well-versed with the concepts of multimedia files. Then, you should also aware yourself with the supported media types in HTML5 such as audio and video elements. Along with this prepare yourself with accessibility of audio and video elements. You should read on some articles on how to deal with non-supporting browsers on Websites.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

#### In-Class Activities:

Follow the order given here during In-Class activities.

## Overview of the Session:

Give the students the overview of the current session in the form of session objectives.

Show the students slide 2 of the presentation.

**Objectives**

- Describe the need for multimedia in HTML5
- List the supported media types in HTML5
- Explain the audio elements in HTML5
- Explain the video elements in HTML5
- Explain the accessibility of audio and video elements
- Describe how to deal with non-supporting browsers

© Aptech Ltd.      HTML5 Audio and Video / Session 11      2

Tell the students that this session introduces them to the need for multimedia in HTML5. They will learn about supported media types in HTML5 along with the audio and video elements. They will also learn about accessibility of audio and video elements and how to deal with the non-supporting browsers.

## 11.2 In-Class Explanations

### Slide 3

Let us understand audio and video in HTML5.

**Introduction**

Traditionally, Web browsers were capable of handling only graphics and text.

User had to install a distinct program, plug-in, or an ActiveX control to play some video.

Earlier, Web designers and Web developers used to set up Web pages to play audio and video on the Web using Adobe Flash player.

© Aptech Ltd.      HTML5 Audio and Video / Session 11      3

Using slide 3, explain elements used before HTML 5 for multimedia.

Traditionally, Web browsers were capable of handling only graphics and text. Suppose, if a user had to play some video, then, a distinct program, plug-in, or an ActiveX control had to be installed.

Earlier, Web designers and Web developers used to set up Web pages using Adobe Flash player to play audio and video on the Web.

#### Slide 4

Let us understand multimedia in HTML5.

The slide has a header 'HTML 5 Multimedia in HTML5'. Below it are four bullet points in colored boxes:

- Multimedia is a combination of various elements such as video, graphics, sound, and text.
- Common way of inserting a multimedia content on Web pages is by embedding a video or audio file in the Web page.
- HTML5 has made lives easier by introducing <audio> and <video> elements.
- HTML5 has provided the developers with the features to embed media on the Web pages in a standard manner.

Using slide 4, explain multimedia in HTML5.

Multimedia is a combination of various elements such as video, graphics, sound, and text. A common way of inserting multimedia content on Web pages is by embedding a video or audio file in a Web page.

Consider the earlier situations where a Website developer did not have the facility of including videos or audios directly on their Website until and unless the browser had the required plug-in installed. These days, Website developers want their visitors to not only download, but also view movies online on their Website. Today, HTML5 provides this facility.

Explain the students that HTML5 has made lives easier by introducing <audio> and <video> elements. Thus, the user need not depend on Flash to access the audio and video files.

## Slide 5

Let us understand media types in audio and video.

The screenshot shows a slide with a title bar 'HTML 5 Supported Media Types in Audio and Video'. Below the title are six colored boxes containing text:

- Red box: There are various video and audio codecs which are used for handling of video and audio files.
- Green box: Codec is a device or a program used for encoding and decoding digital data stream.
- Purple box: Different codecs have different level of compression quality.
- Cyan box: For storing and transmitting coded video and audio together, a container format is used.
- Orange box: There are a number of container formats which includes Ogg (.ogg), the Audio Video Interleave (.avi), Flash Video (.flv), and many others.
- Dark Red box: Different browsers support different container format. WebM is a new open source video container format supported by Google.

At the bottom of the slide, there is footer text: '© Aptech Ltd.', 'HTML5 Audio and Video / Session 11', and '5'.

5 (a)

The screenshot shows a slide with a title bar 'HTML 5 Supported Media Types in Audio and Video'. Below the title is a bullet point: 'Following table lists the common audio and video formats:' followed by a table:

Container	Video Codec	Audio Codec
Mp4	H.264	AAC
Ogg	Theora	Vorbis
WebM	VP8	Vorbis

At the bottom of the slide, there is footer text: '© Aptech Ltd.', 'HTML5 Audio and Video / Session 11', and '5'.

5 (b)

Using slides 5 (a) and 5 (b), explain the supported media types in audio and video.

Mention that there are various video and audio codecs which are used for handling of video and audio files.

The codec is a device or a program used for encoding and decoding digital data stream.

These different codecs have different level of compression quality.

For storing and transmitting coded video and audio together, a container format is used.

There are a number of container formats which includes Ogg (.ogg), the Audio Video Interleave (.avi), Flash Video (.flv), and many others. WebM is a new open source video

container format supported by Google. Different browsers support different container format.

Explain the list of the common audio and video formats listed in the slide.

**Tips:**

MP4 is the new and upcoming format for Internet video. It is supported in all HTML5 supported browsers.

## Slide 6

Let us understand audio formats.



The slide has a blue header bar with the text "HTML5" and "Audio Formats". Below the header is a bulleted list:

- There are three supported file formats for the <audio> element in HTML5.
- Following table lists the audio file formats supported by the Web browsers:

A table follows, showing browser support for MP3, WAV, and Ogg formats. The table rows are: Opera 10.6 (No, Yes, Yes), Apple Safari 5 (Yes, Yes, No), Google Chrome 6 (Yes, Yes, Yes), FireFox 4.0 (No, Yes, Yes), and Internet Explorer 9 (Yes, No, No). At the bottom of the slide are copyright information ("© Aptech Ltd."), the session title ("HTML5 Audio and Video / Session 11"), and the page number ("6").

Browser Support	MP3	WAV	Ogg
Opera 10.6	No	Yes	Yes
Apple Safari 5	Yes	Yes	No
Google Chrome 6	Yes	Yes	Yes
FireFox 4.0	No	Yes	Yes
Internet Explorer 9	Yes	No	No

Using slide 6, explain the audio formats.

There are the three supported file formats for the <audio> element in HTML5. Table lists the audio file formats supported by the Web browsers.

## Slide 7

Let us understand video formats.

**HTML5 Video Formats**

- There are the three supported file formats for the <video> element in HTML5.
- Following table lists the video file formats supported by the Web browsers:

Browser Support	MP3	WAV	Ogg
Opera 10.6	No	Yes	Yes
Apple Safari 5	Yes	No	No
Google Chrome 6	Yes	Yes	Yes
FireFox 4.0	No	Yes	Yes
Internet Explorer 9	Yes	No	No

© Aptech Ltd.      HTML5 Audio and Video / Session 11      7

Using slide 7, explain the video formats.

There are the three supported file formats for the <video> element in HTML5. Table lists which of the browsers support these three formats namely MP4, WebM, and Ogg.

## Slide 8

Let us understand audio element in HTML5.

**HTML5 Audio Elements in HTML5**

- <audio> element will help the developer to embed music on the Web site.
- <audio> tag specifies the audio file to be used in the HTML document.
- src attribute is used to link the audio file.
- The Code Snippet displays the embedding of an audio file in the Web page using the <audio> tag.

```
<!doctype html>
<html>
  <head>
    <title>audio element</title>
  </head>
  <body>
    <audio src="d:\sourcecodes\audio.controls autoplay loop>
html5 audio not supported
</audio>
  </body>
</html>
```

A screenshot of a web browser showing an audio player interface with playback controls and a progress bar at 00:06.

Audio formats frequently used are wav, ogg, and mp3.

© Aptech Ltd.      HTML5 Audio and Video / Session 11      8

Using slide 8, explain the audio element in HTML5.

The audio element will help the developer to embed music on the Website and allow the user to listen to music. The <audio> tag specifies the audio file to be used in the HTML document. It contains the src attribute that is used to link the audio file.

Explain the code snippet which displays the embedding of an audio file in a Web page using the `<audio>` tag. The music is played in the background when the page is loaded on the browser.

The `src` attribute is mandatory, the `<audio>` tag includes several other options. Explain the code snippet to add an audio file on the Web page.

Then, explain the existing HTML5 specification does not specify the formats supported by the browser in the `<audio>` tag. The audio formats frequently used are wav, ogg, and mp3.

Then, mention that if you want to specify an alternative audio file to be selected based on its supported format in the browser. Then, the `<source>` tag is used as a child element of `<audio>` tag.

The `<source>` tag has three attributes namely, `src`, `media`, and `type`.

**Example:**

```
<audio>
  <source src="maddi.ogg" type="audio/ogg">
  <source src="maddi.mp3" type="audio/mpeg">
</audio>
```

**In-Class Question:**

After you finish explaining the audio elements in HTML 5, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What are the attributes of the `<source>` tag used within the audio element?

**Answer:**

`src`, `media`, and `type`.

## Slide 9

Let us understand `audio` tag attributes.

The slide has a header 'HTML5' and a title 'Audio Tag Attributes'. It includes the following text:

- Attributes provide additional information to the browser about the tag.
- HTML5 has a number of attributes for controlling the look and feel of various functionalities.
- HTML5 has the following attributes for the `<audio>` element.

A bulleted list states: 'Following table lists some of the `<audio>` tag attributes.'

Audio Attributes	Description
<code>autoplay</code>	This attribute identifies whether to start or not the audio once the object is loaded
<code>autobuffer</code>	This attribute starts the buffering automatically
<code>controls</code>	This attribute identifies the audio playback controls that should be displayed such as resume, pause, play, and volume buttons
<code>loop</code>	This attribute identifies whether to replay the audio once it has stopped
<code>preload</code>	This attribute identifies whether the audio has to be loaded when the page loads and is ready to execute

At the bottom, there is a footer: '© Aptech Ltd.' and 'HTML5 Audio and Video / Session 11'.

Using slide 9, explain the `audio` tag attributes in detail.

HTML tags normally consist of more than one attribute. Attributes provide additional information to the browser about the tag. HTML5 has a number of attributes for controlling the look and feel of various functionalities. According to HTML5 specifications, the `<audio>` element has the following attributes: `muted`, `autoplay`, and `controls`.

The `muted` attribute is a Boolean attribute. When present, it specifies that the audio output should be muted.

The `controls` attribute is a Boolean attribute. When present, it specifies that audio controls that should be displayed. Audio controls include:

- Play
- Pause
- Seeking
- Volume

The `autoplay` attribute is a Boolean attribute. When present, the audio will automatically start playing as soon as it can do so without stopping.

## Slide 10

Let us understand how to create audio files.

To play the audio in older browsers then the `<embed>` tag will be used.

`<embed>` tag has two attributes, `src` and `autoplay`.

`src` attribute is used to specify the source of the audio.

`autoplay` attribute controls the audio and determines whether the audio should play as soon as the page loads.

- The Code Snippet demonstrates the use of `<embed>` tag in the `<audio>` element.

```
<!DOCTYPE HTML>
<html>
  <body>
    <audio autoplay loop>
      <source src="sampaudio.mp3">
      <source src="sampaudio.ogg">
      <embed src="sampaudio.mp3">
    </audio>
  </body>
</html>
```

HTML5 Audio and Video / Session 11      10

Using slide 10, explain the process of creating audio files.

Suppose, if the user plays the audio in older browsers then the `<embed>` tag will be used. The `<embed>` tag has two attributes, `src` and `autoplay`. The `src` attribute is used to specify the source of the audio and the `autoplay` attribute controls the audio and determines whether the audio should play as soon as the page loads.

Explain code snippet which demonstrates the use of `<embed>` tag in the `<audio>` element.

Mention that `<audio>` element in HTML5 supports multiple formats. The content included within the `<embed>` tag is automatically played by default. Suppose, if the user does not want to play the audio file automatically then he/she can set the value of the `autoplay` attribute to "false".

Mention, `<embed>` tag also supports another attribute named `loop`. The `loop` attribute determines whether the audio clip will be replayed continuously or not. If the value of the `loop` attribute is set to true or infinite then, the music will be played continuously. If the `loop` attribute is not specified, then it is same as setting the value to false.

### Tips:

Any text inside the between `<audio>` and `</audio>` will be displayed in browsers as an alternative text when browser do not support the `<audio>` tag.

## Slide 11

Let us understand `video` elements in HTML5.

The screenshot shows a presentation slide with the title "Video Elements in HTML5". Below the title is a bulleted list of points about the `<video>` element. To the right of the list is a code snippet in a code editor window. Below the code editor is a screenshot of a web browser displaying a video of a bear. The browser interface includes a play button, a progress bar, and a timestamp of "00:12".

- `<video>` element is a new feature added in HTML5.
- `<video>` element is for embedding the video content on the Web page.
- `<video>` element if not supported by the browser then the content between the start tag and end tag is displayed.
- `src` attribute is used to link to the video file.
- The Code Snippet demonstrates the use of the `<video>` element.

```
<!DOCTYPE HTML>
<html>
  <head>
  </head>
  <body>
    <video src="D:\Source codes\movie.mp4">
      Your browser does not support the video.
    </video>
  </body>
</html>
```

© Aptech Ltd.

HTML5 Audio and Video / Session 11 11

Using slides 13, explain the `video` element in HTML5.

The `video` element is a new feature added in HTML5. The user can use the `<video>` element for embedding the video content on the Web page. The easiest way to specify the video is by using the `src` attribute which gives the URL of the video file to be used. Suppose, if the browser does not support the `<video>` element then the content between the start tag and end tag is displayed on the browser.

Mention, `<video>` element allows multiple `<source>` elements. `<source>` elements can link to different video files. The browser will use the first recognized format.

Explain code snippet which demonstrates the use of the `<video>` element. In the code, the `src` attribute is used for specifying the location of the mp4 video file format used by the `<video>` tag. While adding the `<video>` element in the code, the user can specify messages between the `<video>` and `</video>` tag to check if the browser is supporting the `<video>` tag or not.

## Slide 12

Let us understand `video` tag attributes.

**HTML5 Video Tag Attributes**

HTML5 specification provides a list of attributes that can be used with the `<video>` element.

HTML5 has the following attributes for the `<video>` element.

- Following table lists some of the `<video>` tag attributes.

Video Attributes	Description
<code>autoplay</code>	Specifies that the browser will start playing the video as soon as it is ready
<code>muted</code>	Allows to mute the video initially, if this attribute is existing
<code>controls</code>	Allows displaying the controls of the video, if the attribute exists
<code>loop</code>	Specifies that the browser should repeat playing the existing video once more if the loop attribute exists and accepts a boolean value
<code>preload</code>	Specifies whether the video should be loaded or not when the page is loaded

© Aptech Ltd.      HTML5 Audio and Video / Session 11      12

Using slide 12, explain the attributes of `video` tag.

The HTML5 specification provides a list of attributes that can be used with the `video` element. Explain the table listed on the slide for `<video>` tag attributes.

Some of the additional attributes of video elements are `height`, `width`, `poster`, and `src`.

## Slide 13

Let us understand how to preload a video.

**HTML5 Preloading the Video**

- `<video>` element comprises a `preload` attribute that allows the browser to download or buffering the video while the Web page containing the video is being downloaded.
- `preload` attribute has the following values:

**None** - allows the browser to load only the page. The video will not be downloaded while the page is being loaded.

**Metadata** - allows the browser to load the metadata when the page is being loaded.

**Auto** - is the default behavior as it allows the browser to download the video when the page is loaded. The browser can avoid the request.

© Aptech Ltd.      HTML5 Audio and Video / Session 11      13

Using slide 13, explain the preloading the video using `preload` attribute.

The `video` element comprises a `preload` attribute that allows the browser to download or buffer the video while the Web page containing the video is being downloaded. If the video

is preloaded, then it decreases the initial delay once the user has started the playback. The preload attribute has the following values:

- **None:** This attribute allows the browser to load only the page. The video will not be downloaded while the page is being loaded.
- **Metadata:** This attribute allows the browser to load the metadata when the page is being loaded.
- **Auto:** This is the default behavior as it allows the browser to download the video when the page is loaded. The browser can avoid the request.

## Slide 14

Let us understand the use of `preload` attribute in `video` element.

The Code Snippet demonstrates the use of `none` and `metadata` values for the `preload` attribute.

```
<!DOCTYPE HTML>
<html>
  <head>
  </head>
  <body>
    <video width="160" height="140" src="D:\Source Codes\movie.mp4" controls preload="none" muted>
      Your browser does not support the video.
    </video>
    <video width="160" height="140" src="D:\ Source Codes\movie.mp4" controls preload="metadata" muted>
      Your browser does not support the video.
    </video>
  </body>
</html>
```

© Aptech Ltd.      HTML5 Audio and Video / Session 11      14

Using slide 14, explain the HTML code with `preload` attribute. In the code, there are two video elements. Each video element contains the `preload` attribute with value `none` and `metadata` respectively. Figure displays the effect of both the values `none` and `metadata` on the Web page.

## Slide 15

Let us understand the sizing of the video.

**Setting the Video Size**

- User can specify the size of the video with the height and width attribute of the <video> element.
- If these attributes are not provided then the browser sets the video with the key dimensions of the video.
- The Code Snippet demonstrates how to apply the height and width attributes to the <video> element.

```
<!DOCTYPE HTML>
<html>
<head>
</head>
<title> Video Size</title>
<style>
video{
background-color: black;
border: medium double black;
}
</style>
<body>
<video src="D:\Source Codes\movie.mp4" controls preload="auto" width="360"
height="340">
Your browser does not support the video.
</video>
</body>
</html>
```



HTML5 Audio and Video / Session 11 15

Using slide 15, explain the process of setting the video size.

The user can specify the size of the video with the `height` and `width` attribute of the `video` element. Suppose, if these attributes are not provided then the browser sets the video with the key dimensions of the video. This will result in changing the page layout as the Web page is adjusted to accommodate the video.

Explain code snippet which demonstrates how to apply the `height` and `width` attributes to the `<video>` element. In the code, the `style` attribute is used to specify the `background-color` and `border` style of the video. The code also specifies the `preload`, `height`, and `width` attributes for the `video` element.

## Slide 16

Let us understand how to convert the video files.

**HTML5** Converting the Video Files

- There are many problems with browser vendors for supporting the various video formats on the Web sites.
- Following are some of the video formats supported by the significant browsers:

**Ogg/Theora** - is an open source, royalty-free, and patent-free format available. This format is supported by browsers such as Opera, Chrome, and Firefox.

**WebM** - is a royalty-free and patent-free format supported by Google. This format is supported by browsers such as Opera, Chrome, and Firefox.

**H.264/MP4** - are supported on iPhone and Google Android devices.

**Micro Video Controller** - converter creates all files that the user requires for HTML5 <video> element that works on the cross browser.

© Aptech Ltd.      HTML5 Audio and Video / Session 11      16

Using slide 16, explain the process of converting the video files.

Mention video element used in HTML5 is a great feature, but how will the user get the video files in a correct format. There are many problems with browser vendors for supporting various video formats on the Websites. Thus, there are converters available which can convert the format of video files appropriately to be supported in HTML5 browsers.

The **VLC converter** can be used on Windows to covert video files into Ogg format. For MP4 videos, you can use **HandBrake** which is an open-source application available for Windows, Mac OS X, and Linux. Similarly, **WebM Encoder 1.2** is a simple utility to covert video files to the WebM format.

## Slides 17 and 18

Let us understand accessibility of audio and video elements.

**HTML5** Accessibility of Audio and Video Elements 1-2

- Enterprises across the world are employing people with varied skills and abilities.
- It may include people with limited abilities or disabilities such as people with visual, cognitive, or mobility impairments.
- Accessibility is the level of ease with which computers can be used and be available to a wide range of users.
- While developing an application a lot of assumptions are to be considered and some of them are as follows:

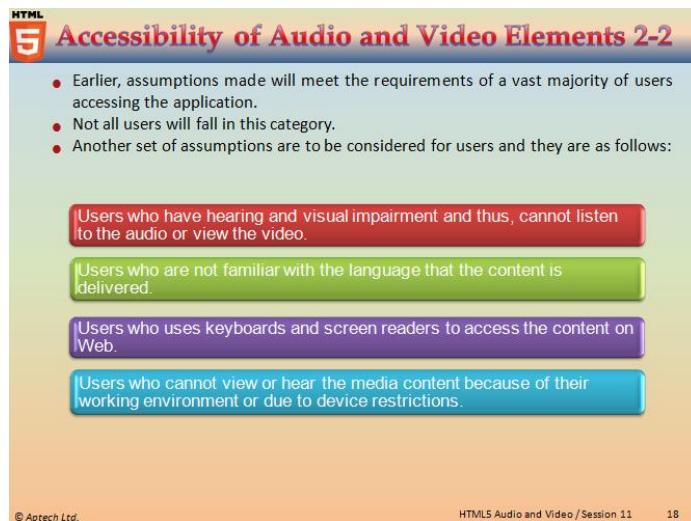
Users can check the content on laptop, mobile, tablet, or desktop.

Users can listen to the audio by using headphones or speakers.

Users can understand the language in which the media was delivered.

Users can successfully play and download the media.

© Aptech Ltd.      HTML5 Audio and Video / Session 11      17



The slide has a header 'HTML5 Accessibility of Audio and Video Elements 2-2' with a red and yellow gradient background. Below the header is a bulleted list of assumptions:

- Earlier, assumptions made will meet the requirements of a vast majority of users accessing the application.
- Not all users will fall in this category.
- Another set of assumptions are to be considered for users and they are as follows:

Below this, there are four colored callout boxes listing specific user groups:

- Red box: Users who have hearing and visual impairment and thus, cannot listen to the audio or view the video.
- Green box: Users who are not familiar with the language that the content is delivered.
- Purple box: Users who use keyboards and screen readers to access the content on Web.
- Blue box: Users who cannot view or hear the media content because of their working environment or due to device restrictions.

At the bottom left is the copyright notice '© Aptech Ltd.' and at the bottom right are the page numbers 'HTML5 Audio and Video / Session 11' and '18'.

Using slides 17 and 18, explain the accessibility of audio and video element.

Mention, enterprises across the world are employing people with varied skills and abilities. They may even include people with limited abilities or disabilities such as people with visual, cognitive, or mobility impairments.

Accessibility is the level of ease with which computers can be used and be available to a wide range of users, including people with disabilities. Applications can be accessed through various sources. If the application considers the requirements of the target audience, then it will be appreciated and used by number of users.

Explain the students about the assumptions for multimedia content.

Then, explain the HTML5 powerful feature named Web Video text Tracks (WebVTT) to make applications accessible to such users.

WebVTT is a file format used to mark up the external text tracks. This format allows the user to give a textual description of the content in the video. This description is then used by different accessibility devices to define the content to those users who cannot see it.

## Slides 19 to 21

Let us understand track elements.

**HTML 5 Track Element 1-3**

- Track element provides an easy, standard way to add captions, subtitles, chapters, and screen reader descriptions to the <audio> and <video> elements.
- Track elements are also used for other types of timed metadata.
- Source data for this track element is in a form of a text file that is made up of a list of timed cues.
- Cue is a pointer at an accurate time point in the length of a video.
- Cues contain data in formats such as Comma-Separated Values (CSV) or JavaScript Object Notation.
- Track element is not supported in many major browsers and is now available in IE 10 and Chrome 18+.

© Aptech Ltd.

HTML5 Audio and Video / Session 11

19

**HTML 5 Track Element 2-3**

- Following table lists the track element attributes.

Container	Description
src	Contains the URL of the text track data
srclang	Contains the language of the text track data
kind	Contains the type of content for which the track definition is used
default	Indicates that this will be the default track if the user does not specifies the value
label	Specifies the title to be displayed for the user

© Aptech Ltd.

HTML5 Audio and Video / Session 11

20

**HTML5 Track Element 3-3**

- The Code Snippet demonstrates how a track element is used in combination with <video> element for providing subtitles.

```
<video controls>
  <source src="myvideo.mp4" type="video/mp4" />
  <source src="myvideo.webm" type="video/webm" />
  <track src="eng.vtt" label="English subtitles" kind="subtitles"
    srclang="en" >
</video>
```

- The Code Snippet demonstrates how a track element is used in combination with <video> element providing subtitles in another language.

```
<video controls>
  <source src="myvideo.mp4" type="video/mp4" />
  <source src="myvideo.webm" type="video/webm" />
  <track src="de.vtt" label="German subtitles" kind="subtitles"
    srclang="de" >
</video>
```

© Aptech Ltd.      HTML5 Audio and Video / Session 11      21

Using slides 19 to 21, explain the `track` element in HTML5.

The `track` element provides an easy, standard way to add captions, subtitles, chapters, and screen reader descriptions to the `<audio>` and `<video>` elements.

Then, explain the table provided on slide 20 that lists the attributes of the `track` element.

Using slide 21, explain the HTML codes for track element. First code specifies the `src`, `label`, and `srclang` attributes in the `track` element. Here, the `srclang` is set to `en` that is English language. Second code demonstrates a track element used in combination with `<video>` element providing subtitles in the French language.

### In-Class Question:

After you finish explaining the track element, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which attribute is used to specify the type of element to be tracked in track tag?

### Answer:

`kind` attribute.

## Slide 22

Let us understand the accessibility support for audio and video elements.

**HTML5 Accessibility for Audio and Video Element**

- Accessibility supports for <audio> and <video> elements are as follows:
  - Audio Support

Firefox - Expose controls with accessibility APIs, however individual controls do not interact with keyboard. Access to keyboard is provided by the Firefox specific shortcuts.

Opera - Has only keyboard support.

IE 9 - Expose controls with accessibility APIs, however individual controls do not interact with keyboard.

- Video Support

Firefox - Cannot interact with individual controls.

Opera - Has only keyboard support.

IE 9 - Does not allow individual controls to interact with keyboard.

© Aptech Ltd.      HTML5 Audio and Video / Session 11      22

Using slide 22, explain the accessibility support for audio and video element in the various browsers.

Mention that the lower versions of these browsers do not provide the support for multimedia elements.

## Slide 23

Let us summarize the session.

**HTML5 Summary**

- Multimedia is a combination of various elements such as video, graphics, sound, and text.
- There are various media types used for audio and video files on different Web sites.
- The <audio> element will help the developer to embed music on the Web site and allow the user to listen to music.
- Users can play the audio in older browsers using the <embed> tag.
- The <video> element is used for embedding the video content on the Web page.
- Preload attribute identifies whether the audio has to be loaded when the page loads and is ready to execute.
- WebM is a new open source video container format supported by Google.

© Aptech Ltd.      HTML5 Audio and Video / Session 11      23

In slide 23, you will summarize the session. You will end the session, with a brief summary of what has been taught in the session.

### **11.3 Post Class Activities for Faculty**

You should familiarize yourself with the topics of the next session. You should also explore the features of JavaScript and its objects that are offered with the next session.

**Tips:**

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 12 – Introduction to JavaScript

---

## 12.1 Pre-Class Activities

Before you commence the session, you should familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 12.1.1 Objectives

By the end of this session, the learners will be able to:

- Explain scripting
- Explain the JavaScript language
- Explain the client-side and server-side JavaScript
- List the variables and data types in JavaScript
- Describe the JavaScript methods to display information
- Explain escape sequences and built-in functions in JavaScript
- Explain events and event handling
- Explain jQuery
- Describe how to use the jQuery Mobile

### 12.1.2 Teaching Skills

To teach this session, you should be well-versed with scripting concepts. Also the JavaScript variable, data types, built-in functions, and methods should be known. Along with this, you should prepare yourself with how to perform event handling in JavaScript. Also, make yourself aware with the jQuery API.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

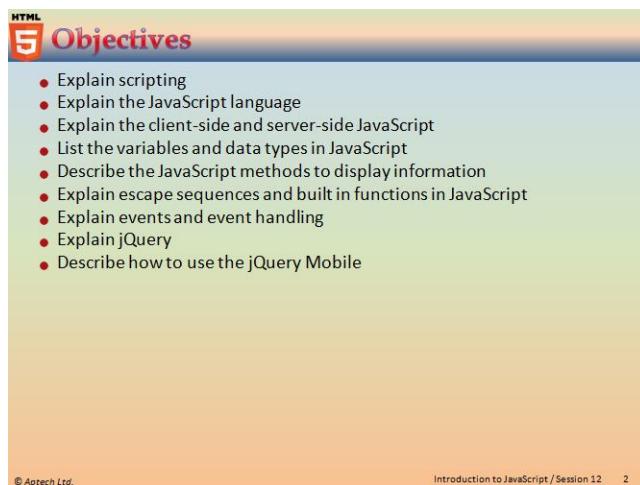
**In-Class Activities:**

Follow the order given here during In-Class activities.

**Overview of the Session:**

Give the students the overview of the current session in the form of session objectives.

Show the students Slide 2 of the presentation.



Tell the students that this session introduces them to basics of scripting. It introduces them to JavaScript language which is used to provide dynamic functionality and validation on the HTML pages. They will learn about JavaScript variable, data types, built-in functions, and methods. They will also know about JavaScript event handling. Finally, the session will introduce them to a ready-to-use jQuery library which provides functions to add interactivity and dynamic behavior to a Web page.

## 12.2 In-Class Explanations

### Slides 3 to 5

Let us understand the scripting.

**HTML 5 Scripting 1-3**

- Scripting refers to a series of commands that are interpreted and executed sequentially and immediately on occurrence of an event.
- This event is an action generated by a user while interacting with a Web page.
- Examples of events include button clicks, selecting a product from a menu, and so on.
- A scripting language refers to a set of instructions that provides some functionality when the user interacts with a Web page.
- Scripting languages are often embedded in the HTML pages to change the behavior of the Web pages according to the user's requirements.

© Aptech Ltd. Introduction to JavaScript / Session 12 3

**HTML 5 Scripting 2-3**

- Following figure displays the need for scripting.

**Dynamic Interaction with Scripting**

© Aptech Ltd. Introduction to JavaScript / Session 12 4

**HTML 5 Scripting 3-3**

- There are two types of scripting languages. They are as follows:

- Client-side Scripting:**
  - Refers to a script being executed on the client's machine by the browser.
- Server-side Scripting:**
  - Refers to a script being executed on a Web server to generate dynamic HTML pages.

- Following figure displays the types of scripting.

© Aptech Ltd. Introduction to JavaScript / Session 12 5

Using slides 3 to 5, explain the scripting.

Consider an organization that provides a Web site that allows its customers to view their products. The company has received frequent customer feedbacks to provide the shopping facility online. Therefore, the company has decided to add the shopping facility in their Web site by creating dynamic Web pages. These Web pages will allow the user to shop for the products online. Here, the main task of the developer is to validate the customer's inputs while they shop online. For example, details such as credit card number, email, and phone number entered by the customer must be in a proper format. Further, the developer also needs to retrieve the chosen products and their quantity to calculate the total cost.

The developer can handle all these critical tasks by using a scripting language. A scripting language refers to a set of instructions that provides some functionality when the user interacts with a Web page.

Scripting languages are often embedded in the HTML pages to change the behavior of the Web pages according to the user's requirements.

Explain them that some of the scripts can be written to be executed on the occurrence of an event on the Web page. Examples of events include button clicks, selecting a product from a menu, and so on.

Explain the need for scripting and types of scripting languages as explained on the slides.

## Slides 6 and 7

Let us understand JavaScript.

**JavaScript 1-2**

JavaScript is a scripting language that allows building dynamic Web pages by ensuring maximum user interactivity.

JavaScript language is an object-based language, which means that it provides objects for specifying functionalities.

In real life, an object is a visible entity such as a car or a table having some characteristics and capable of performing certain actions.

Similarly, in a scripting language, an object has a unique identity, state, and behavior.

The identity of the object distinguishes it from the other objects of the same type.

The state of the object refers to its characteristics, whereas the behavior of the object consists of its possible actions.

The object stores its identity and state in fields (also called variables) and exposes its behavior through functions (actions).

© Aptech Ltd.

Introduction to JavaScript / Session 12

6

**JavaScript 2-2**

- Following figure displays some real world objects.

	
Identity: AXA 43 S State: Color-Red, Wheels-Four Behavior: Running	Identity: T002 State: Color-Brown Behavior: Stable
<b>Objects</b>	

© Aptech Ltd.

Introduction to JavaScript / Session 12

7

Using slides 6 and 7, explain the JavaScript language.

Explain the object-oriented features of JavaScript language to the students. Tell them that an object-based language allows you to work with real-world objects for specifying functionalities. As In real life, an object is a visible entity such as a car or a table. Similarly, software every object has some characteristics and is capable of performing certain actions. Similarly, in a scripting language, an object has a unique identity, state, and behavior. Explain the figure provided in the slide.

The object stores its identity and state in fields (also called variables) and exposes its behavior through functions (actions).

## Slide 8

Let us understand versions of JavaScript.

**Versions of JavaScript**

- The first version of JavaScript was developed by Brendan Eich at Netscape in 1995 and was named JavaScript 1.0.
- Following table lists the various versions of JavaScript language.

Version	Description
1.1	Is supported from 3.0 version of the Netscape Navigator and Internet Explorer.
1.2	Is supported by the Internet Explorer from version 4.0.
1.3	Is supported by the Internet Explorer from version 5.0, Netscape Navigator from version 4.0, and Opera from version 5.0.
1.4	Is supported by servers of Netscape and Opera 6.
1.5	Is supported by the Internet Explorer from version 6.0, Netscape Navigator from version 6.0, and Mozilla Firefox from version 1.0.
1.6	Is supported in the latest versions of the Internet Explorer and Netscape Navigator browsers. It is also supported by Mozilla Firefox from version 1.5.
1.7	Is supported in the latest versions of the Internet Explorer and Netscape Navigator browsers. It is also supported by Mozilla Firefox from version 2.0.

© Aptech Ltd. Introduction to JavaScript / Session 12 8

Using slide 8, explain the versions of JavaScript.

The first version of JavaScript was developed by Brendan Eich at Netscape in 1995 and was named JavaScript 1.0. Netscape Navigator 2.0 and Internet Explorer 3.0 supported JavaScript 1.0. Over the period, it gradually evolved with newer versions where each version provided better features and functionalities as compared to their previous versions.

Explain the table that lists the various versions of JavaScript language.

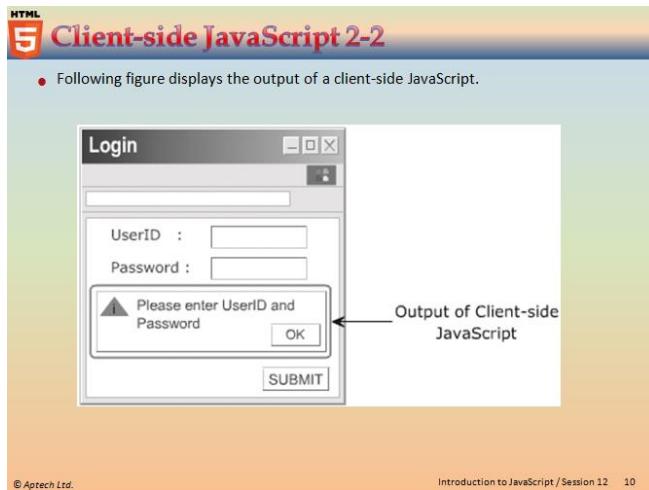
## Slides 9 and 10

Let us understand client-side JavaScript.

**Client-side JavaScript 1-2**

- A Client-side JavaScript (CSJS) is executed by the browser on the user's workstation.
- A client-side script might contain instructions for the browser to handle user interactivity.
- These instructions might be to change the look or content of the Web page based on the user inputs.
- Examples include displaying a welcome page with the user name, displaying date and time, validating that the required user details are filled, and so on.
- A JavaScript is either embedded in an HTML page or is separately defined in a file, which is saved with .js extension.
- In client-side scripting, when an HTML is requested, the Web server sends all the required files to the user's computer.
- The Web browser executes the script and displays the HTML page to the user along with any tangible output of the script.

© Aptech Ltd. Introduction to JavaScript / Session 12 9



Using slides 9 and 10, explain the client-side JavaScript.

JavaScript is a scripting language, which can be executed on the client-side and on the server-side. A client-side JavaScript (CSJS) is executed by the browser on the user's workstation. A client-side script might contain instructions for the browser to handle user interactivity. These instructions might be to change the look or content of the Web page based on the user inputs. Examples include displaying a welcome page with the username, displaying date and time, validating that the required user details are filled, and so on.

Mention, JavaScript is either embedded in an HTML page or is separately defined in a file, which is saved with .js extension. In client-side scripting, when an HTML is requested, the Web server sends all the required files to the user's computer. The Web browser executes the script and displays the HTML page to the user along with any tangible output of the script.

Figure displays the output of a client-side JavaScript.

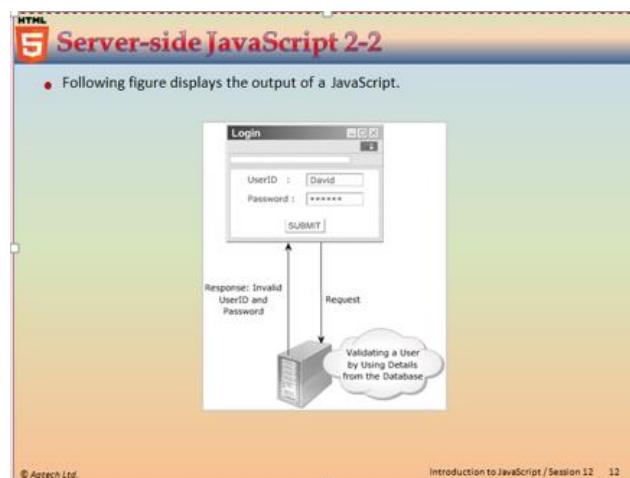
## Slides 11 and 12

Let us understand server-side JavaScript.

**Server-side JavaScript 1-2**

- A Server-side JavaScript (SSJS) is executed by the Web server when an HTML page is requested by a user and the output is displayed by the browser.
- A server-side JavaScript can interact with the database, fetch the required information specific to the user, and display it to the user.
- Server-side scripting fulfills the goal of providing dynamic content in Web pages.
- Unlike client-side JavaScript, HTML pages using server-side JavaScript are compiled into bytecode files on the server.
- A JavaScript is either embedded in an HTML page or is separately defined in a file, which is saved with .js extension.
- Compilation is a process of converting the code into machine-independent code.
- This machine-independent code is known as the bytecode, which is an executable file that the Web server runs to generate the desired output.

© Aptech Ltd. Introduction to JavaScript / Session 12 11



Using slides 11 and 12, explain the server-side JavaScript.

A server-side JavaScript (SSJS) is executed by the Web server when an HTML page is requested by a user. The output of a server-side JavaScript is sent to the user and is displayed by the browser. In this case, a user might not be aware that a script was executed on the server to produce the desirable output.

A server-side JavaScript can interact with the database, fetch the required information specific to the user, and display it to the user. This means that server-side scripting fulfills the goal of providing dynamic content in Web pages.

Explain the figure that shows the output of the server-side JavaScript.

## Slides 13 and 14

Let us understand the `<script>` tag.

The `<script>` tag defines a script for an HTML page to make them interactive.

The browser that supports scripts interprets and executes the script specified under the `<script>` tag when the page loads in the browser.

You can directly insert a JavaScript code under the `<script>` tag.

You can define multiple `<script>` tags either in the `<head>` or in the `<body>` elements of an HTML page.

In HTML5, the `type` attribute specifying the scripting language is no longer required as it is optional.

© Aptech Ltd. Introduction to JavaScript / Session 12 13

- The Code Snippet demonstrates the use of the tag.

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      document.write("Welcome to the Digital World");
    </script>
  </head>
  <body>
    ....
  </body>
</html>
```

There are two main purposes of the `<script>` tag, which are as follows:

- Identifies a given segment of script in the HTML page.
- Loads an external script file.

© Aptech Ltd. Introduction to JavaScript / Session 12 14

Using slides 13 and 14, explain the `<script>` tag.

Mention, `<script>` tag defines a script for an HTML page to make them interactive. The browser that supports scripts interprets and executes the script specified under the `<script>` tag when the page loads in the browser.

Mention, JavaScript code under the `<script>` tag. You can define multiple `<script>` tags either in the `<head>` or in the `<body>` elements of an HTML page. In HTML5, the `type` attribute specifying the scripting language is no longer required as it is optional.

Then, explain the HTML code using `<script>` tag and uses of `<script>` tag.

There are two main purposes of the `<script>` tag, which are as follows:

- Identifies a given segment of script in the HTML page
- Loads an external script file

**In-Class Question:**

After you finish explaining `<script>` tag, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which tag is used to insert JavaScript code in a HTML file?

**Answer:**

`<script>` tag is used to insert JavaScript code in a HTML file.

**Slide 15**

Let us understand variables in JavaScript.

The slide has a header 'HTML 5 Variables in JavaScript'. It contains five bullet points in colored boxes:

- A variable refers to a symbolic name that holds a value, which keeps changing.
- For example, age of a student and salary of an employee can be treated as variables.
- In JavaScript, a variable is a unique location in computer's memory that stores a value and has a unique name.
- The name of the variable is used to access and read the value stored in it.
- A variable can store different types of data such as a character, a number, or a string.

At the bottom left is '© Aptech Ltd.' and at the bottom right is 'Introduction to JavaScript / Session 12 15'.

Using slide 15, explain the variables in JavaScript.

A variable refers to a symbolic name that holds a value, which keeps changing. For example, age of a student and salary of an employee can be treated as variables. A real life example for variables includes the variables used in algebraic expressions that store values.

In JavaScript, a variable is a unique location in the computer's memory that stores a value and has a unique name. The name of the variable is used to access and read the value stored in it.

A variable can store different types of data such as a character, a number, or a string. Therefore, a variable acts as a container for saving and changing values during the execution of the script.

## Slides 16 to 19

Let us understand how to declare variables.

**HTML 5 Declaring Variables 1-4**

- Declaring a variable refers to creating a variable by specifying the variable name.
- For example, one can create a variable named to store the name of a student.

In JavaScript,

- the `var` keyword is used to create a variable by allocating memory to it.
- a keyword is a reserved word that holds a special meaning.
- the variable can be initialized at the time of creating the variable or later.
- Initialization refers to the task of assigning a value to a variable.
- once the variable is initialized, you can change the value of a variable as required.
- variables allow keeping track of data during the execution of the script.
- while referring to a variable, you are referring to the value of that variable.
- one can declare and initialize multiple variables in a single statement.

© Aptech Ltd. Introduction to JavaScript / Session 12 16

**HTML 5 Declaring Variables 2-4**

- Following figure displays how to declare variables.

- Following syntax demonstrates how to declare variables in JavaScript.

**Syntax:**  
`var <variableName>;`

where,

- `var`: Is the keyword in JavaScript.
- `<variableName>`: Is a valid variable name.

© Aptech Ltd. Introduction to JavaScript / Session 12 17

**HTML 5 Declaring Variables 3-4**

- Following syntax demonstrates how to initialize variables in JavaScript.

**Syntax:**  
`<variableName> = <value>;`

where,

- `=`: Is the assignment operator used to assign values.
- `<value>`: Is the data that is to be stored in the variable.

- The syntax demonstrates how to declare and initialize multiple variables in a single statement, which are separated by commas.

**Syntax:**  
`var <variableName1> = <value1>, <variableName2> = <value2>;`

© Aptech Ltd. Introduction to JavaScript / Session 12 18

**HTML** Declaring Variables 4-4

- The Code Snippet declares two variables namely, studID and studName and assign values to them.

```
var studID;
var studName;
studID = 50;
studName = "David Fernando";
```

- The snippet assigns values to studID and studName variables by using the assignment operator (=).
- The value named David Fernando is specified within double quotes.
- The Code Snippet demonstrates how to declare and initialize multiple variables in a single statement in JavaScript.

```
var studName = David, studAge = 15;
```

© Aptech Ltd. Introduction to JavaScript / Session 12 19

Using slides 16 to 19, explain the syntax to declare a variable in JavaScript.

Declaring a variable refers to creating a variable by specifying the variable name. For example, you can create a variable named `studName` to store the name of a student. Here, the variable name `studName` is referred to as an identifier.

In JavaScript, the `var` keyword is used to create a variable by allocating memory to it. A keyword is a reserved word that holds a special meaning in JavaScript.

Mention, variable can be initialized at the time of creating the variable or later. Initialization refers to the task of assigning a value to a variable. Once the variable is initialized, you can change the value of a variable as required. Variables allow keeping track of data during the execution of the script. While referring to a variable, you are referring to the value of that variable. In JavaScript, you can declare and initialize multiple variables in a single statement.

Then, explain the syntax to declare variables.

```
var <variableName>;
```

Then, demonstrate how to initialize variables in JavaScript.

```
<variableName> = <value>;
```

The syntax demonstrates how to declare and initialize multiple variables in a single statement, which are separated by commas.

```
var <variableName1> = <value1>, <variableName2> = <value2>;
```

Using slide 19, explain the JavaScript code snippet.

Mention, code assigns values to `studID` and `studName` variables by using the assignment operator (`=`). The value named “**David Fernando**” is specified within double quotes.

Next, explain that the code snippet shows how to declare and initialize multiple variables in a single statement in JavaScript.

## Slide 20

Let us understand variable naming rules.

The slide has a blue header bar with the text "HTML 5 Variable Naming Rules". Below the header, there is a bulleted list of rules:

- JavaScript is a case-sensitive language.
- The variables X and x are treated as two different variables.
- JavaScript consists of certain rules for naming a variable as follows:

Below this, there is a purple box containing the heading "In JavaScript, a variable name" followed by six sub-rules, each in its own colored box:

- can consist of digits, underscore, and alphabets.
- must begin with a letter or the underscore character.
- cannot begin with a number and cannot contain any punctuation marks.
- cannot contain any kind of special characters such as +, \*, %, and so on.
- cannot contain spaces.
- cannot be a JavaScript keyword.

At the bottom left is the copyright notice "© Aptech Ltd.". At the bottom right is the page number "Introduction to JavaScript / Session 12 20".

Using slide 20, explain the variable naming rules.

Mention, JavaScript is a case-sensitive language. This means that if you specify X and x as variables, both of them are treated as two different variables. Similarly, in JavaScript, there are certain rules, which must be followed while specifying variables names.

Then, explain the rules for a variable name listed on the slide.

## Slides 21 to 23

Let us understand data types in JavaScript.

**HTML 5 Data Types in JavaScript 1-3**

- To identify the type of data that can be stored in a variable, JavaScript provides different data types.
- Data types in JavaScript are classified into two broad categories namely, primitive and composite data types.
- Primitive data types contain only a single value, whereas the composite data types contain a group of values.

**> PRIMITIVE DATA TYPES**

- A primitive data type contains a single literal value such as a number or a string.
- A literal is a static value that you can assign to variables.

© Aptech Ltd. Introduction to JavaScript / Session 12 21

**HTML 5 Data Types in JavaScript 2-3**

- Following table lists the primitive data types.

Primitive Data Type	Description
<code>boolean</code>	Contains only two values namely, true or false
<code>null</code>	Contains only one value namely, null. A variable of this value specifies that the variable has no value. This null value is a keyword and it is not the same as the value, zero
<code>number</code>	Contains positive and negative numbers and numbers with decimal point. Some of the valid examples include 6, 7.5, -8, 7.5e-3, and so on
<code>string</code>	Contains alphanumeric characters in single or double quotation marks. The single quotes is used to represent a string, which itself consists of quotation marks. A set of quotes without any characters within it is known as the null string

© Aptech Ltd. Introduction to JavaScript / Session 12 22

**HTML 5 Data Types in JavaScript 3-3**

**> COMPOSITE DATA TYPES**

- A composite data type stores a collection of multiple related values, unlike primitive data types.
- In JavaScript, all composite data types are treated as objects.
- A composite data type can be either predefined or user-defined in JavaScript.
- Following table lists the composite data types.

Composite Data Type	Description
Objects	Refers to a collection of properties and functions. Properties specify the characteristics and functions determine the behavior of a JavaScript object
Functions	Refers to a collection of statements, which are instructions to achieve a specific task
Arrays	Refers to a collection of values stored in adjacent memory locations

© Aptech Ltd. Introduction to JavaScript / Session 12 23

Using slides 21 to 23, explain the data types in JavaScript.

A Web page designer can store different types of values such as numbers, characters, or strings in variables. However, the Web page designer must know what kind of data a particular variable is expected to store. To identify the type of data that can be stored in a variable, JavaScript provides different data types. A Web page designer need not specify the data type while declaring variables. Thus, JavaScript is referred to as the loosely typed language. This means that a variable holding a number can also hold a string value later. The values of variables are automatically mapped to their data types when the script is executed in the browser.

Mention, data types in JavaScript are classified into two broad categories namely, primitive and composite data types. Primitive data types contain only a single value, whereas the composite data types contain a group of values.

Explain about primitive data types.

A primitive data type contains a single literal value such as a number or a string. A literal is a static value that you can assign to variables.

Using slide 22, explain the different types of primitive data types.

Explain, table which lists the primitive data types.

Using slide 23, explain the composite data type and its types.

Mention, composite data type stores a collection of multiple related values, unlike primitive data types. In JavaScript, all composite data types are treated as objects. A composite data type can be either predefined or user-defined in JavaScript.

Explain, table which lists the composite data types.

## Slides 24 to 26

Let us understand methods in JavaScript.

**HTML 5 Methods 1-3**

- JavaScript allows you to display information using the methods of the document object.
- The document object is a predefined object in JavaScript, which represents the HTML page and allows managing the page dynamically.
- Each object in JavaScript consists of methods, that fulfill a specific task.
- There are two methods of the document object, that display any type of data in the browser. These methods are as follows:
  - `write()`: Displays any type of data.
  - `writeln()`: Displays any type of data and appends a new line character.
- The syntax demonstrates the use of `document.write()` method, which allows you to display information in the displayed HTML page.

**Syntax:**  
`document.write("<data>" + variables);`

where,

- data:** Specifies strings enclosed in double quotes.
- variables:** Specify variable names whose value should be displayed on the HTML page.

© Aptech Ltd. Introduction to JavaScript / Session 12 24

**HTML 5 Methods 2-3**

- The syntax demonstrates the use of `document.writeln()` method, which appends a new line character.

**Syntax:**  
`document.writeln("<data>" + variables);`

- The Code Snippet demonstrates the use of `write()` method.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> JavaScript language </title>
    <script>
      document.write("<p> JavaScript:");
      document.writeln("is a scripting");
      document.write("and a case-sensitive language.");
    </script>
  </head>
  <p>
    JavaScript: is a scripting and a case-sensitive language.
  </p>
</html>
```

© Aptech Ltd. Introduction to JavaScript / Session 12 25

**HTML 5 Methods 3-3**

- The code uses the `writeln()` method to display the text after the colon.
- It appends a new line character after the text. Then, the text within the `write()` method is displayed on the same line after leaving a space.
- The same paragraph is displayed in the body of the HTML page.
- Note that the text in the `<p>` element appears on different lines.
- In HTML, by default the text will not be displayed in the new line in the browser even though the ENTER key is pressed while writing the code.
- Rather, it will be displayed on the same line with a space. The `writeln()` method also follows this same format.
- Following figure displays the use of `write()` and `writeln()` methods.

© Aptech Ltd. Introduction to JavaScript / Session 12 26

Using slides 24 to 26, explain the methods in JavaScript.

JavaScript allows you to display information using the methods of the document object. The document object is a predefined object in JavaScript, which represents the HTML page and allow managing the page dynamically.

Tell the students that the HTML page elements are referred to as object in JavaScript. To work with these objects JavaScript accesses them as objects. The object model on HTML page is also referred to as HTML Document Object Model (DOM).

The browser creates a DOM of the Web page when it is loaded.

Figure shows the HTML DOM Tree of the elements specified on the page.

There are two methods of the document object, which display any type of data in the browser. These methods are as follows:

- `write()` : Displays any type of data.
- `writeln()` : Displays any type of data and appends a new line character.

Explain students the syntax for demonstrating the use of `document.write()` method.

```
document.write("<data>" + variables);
```

Similarly, the syntax that demonstrates the use of `document.writeln()` method that appends a new line character is: `document.writeln("<data>" + variables);`

Also explain the code snippet and output for the same. The code uses the `writeln()` method to display the text after the colon without leaving a space. It finally appends a new line character after the text. Then, the text within the `write()` method is displayed on the same line after leaving a space.

The same paragraph is displayed in the body of the HTML page. Note that the text in the `<p>` element appears on different lines. In HTML, the text on the second line, and a case sensitive language will not be displayed in the new line in the browser even though the ENTER key is pressed while writing the code. Rather, it will be displayed on the same line with a space. The `writeln()` method also follows this same format.

Explain the figure that displays the use of `write()` and `writeln()` methods.

## Slide 27

Let us understand how to use comments.

**HTML5 Using Comments**

- Comments provide information about a piece of code in the script.
- When the script is executed, the browser identifies comments as they are marked with special characters and does not display them.
- JavaScript supports two types of comments. These are as follows:

**> SINGLE LINE COMMENTS**

- Single-line comments begin with two forward slashes (//). You can insert single-line comments as follows:

```
// This statement declares a variable named num.  
var num;
```

**> MULTI-LINE COMMENTS**

- Multi-line comments begin with a forward slash followed by an asterisk /\*) and end with an asterisk followed by a forward slash (\*/).
- You can insert multiple lines of comments as follows:

```
/* This line of code  
declares a variable */  
var num;
```

© Aptech Ltd. Introduction to JavaScript / Session 12 27

Using slide 27, explain the comments in JavaScript.

A Web page designer might code complex script to fulfil a specific task. In JavaScript, a Web page designer specifies comments to provide information about a piece of code in the script. Comments describe the code in simple words so that somebody who reads the code can understand the code. Comments are small piece of text that makes the program more readable. While the script is executed, the browser can identify comments as they are marked with special characters and do not display them.

JavaScript supports two types of comments. These are as follows:

- **Single-line Comments**
- **Multi-line Comments**

Explain these comments to the students.

## Slides 28 and 29

Let us understand escape sequence characters.

**HTML 5 Escape Sequence Characters 1-2**

- An escape sequence character is a special character that is preceded by a backslash (\).
- Escape sequence characters are used to display special non-printing characters such as a tab space, a single space, or a backspace.
- In JavaScript, the escape sequence characters must be enclosed in double quotes.
- Following table lists the escape sequence characters.

Escape Sequence	Non-Printing Character
\b	Back space
\f	Formfeed
\n	Newline
\r	Carriage return
\t	Horizontal tab
\'	Single quote
\"	Double quote

© Aptech Ltd. Introduction to JavaScript / Session 12 28

**HTML 5 Escape Sequence Characters 2-2**

Escape Sequence	Non-Printing Character
\\\	Backslash
\\\aaa	Matches a Latin-1 encoding character using octal representation, where aaa are three octal numbers. For example, \251 represents the copyright symbol
\\\xaa	Matches a Latin-1 encoding character using hexadecimal representation, where aa are two hexadecimal numbers. For example, \x61 represents the character 'a'
\\\uaaaa	Represent the Unicode encoding character, where aaaaa are four hexadecimal numbers. For example, the character \u0020 represents a space

- The Code Snippet demonstrates the use of escape sequence characters in JavaScript.

```
<script>
    document.write("You need to have a \u0022credit card\u0022, if
                    you want to shop on the \'Internet\'.");
</script>
```

- The code uses a Unicode encoding character namely, \u0022, which represents double quotes that will contain the term credit card.
- The word Internet is placed in single quotes that are specified using the backslash character.

© Aptech Ltd. Introduction to JavaScript / Session 12 29

Using slides 28 and 29, explain the escape sequence characters used in JavaScript.

An escape sequence character is a special character that is preceded by a backslash (\). Escape sequence characters are used to display special non-printing characters such as a tab space, a single space, or a backspace. These non-printing characters help in displaying formatted output to the user to maximize readability.

The backslash character specifies that the following character denotes a non-printing character. For example, \t is an escape sequence character that inserts a tab space similar to the Tab key of the keyboard. In JavaScript, the escape sequence characters must always be enclosed in double quotes.

Explain the multiple escape sequence characters in JavaScript that provide various kind of formatting with the help of table listed on the slides.

Explain table which lists the escape sequence characters and the code snippet.

### Slides 30 to 33

Let us understand built-in functions.

**HTML 5 Built-in Functions 1-4**

- A function is a piece of code that performs some operations on variables to fulfill a specific task.
- It takes one or more input values, processes them, and returns an output value.
- Following table lists the built-in JavaScript functions.

Function	Description	Example
<code>alert()</code>	Displays a dialog box with some information and OK button	<code>alert("Please fill all the fields of the form");</code> Displays a message box with the instruction
<code>confirm()</code>	Displays a dialog box with OK and Cancel buttons. It verifies an action, which a user wants to perform	<code>confirm("Are you sure you want to close the page?");</code> Displays a message box with the question
<code>parseInt()</code>	Converts a string value into a numeric value	<code>parseInt("25 years");</code>
<code>parseFloat()</code>	Converts a string into a number with decimal point	<code>parseFloat("10.33");</code> Returns 10.33

© Aptech Ltd. Introduction to JavaScript / Session 12 30

**HTML 5 Built-in Functions 2-4**

Function	Description	Example
<code>eval()</code>	Evaluates an expression and returns the evaluated result	<code>eval("2+2");</code> Returns 4
<code>isNaN()</code>	Checks whether a value is not a number	<code>isNaN("Hello");</code> Returns true
<code>prompt()</code>	Displays a dialog box that accepts an input value through a text box. It also accepts the default value for the text box	<code>prompt("Enter your name", "Name");</code> Displays the message in the dialog box and Name in the text box.

- An element organizes the content in a Web page hierarchically, which forms the basic HTML structure.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> JavaScript language </title>
    <script>
      var value = "",
```

© Aptech Ltd. Introduction to JavaScript / Session 12 31

**Built-in Functions 3-4**

```

HTML
5 Built-in Functions 3-4

var numone = prompt("enter first value to perform the
                     multiplication operation", value);
var numtwo = prompt("enter second value to perform the
                     multiplication operation", value);
var result = eval(numone * numtwo);
document.write("The result of multiplying: " + numone +
               "and " + numtwo + " is: " + result + ".");

```

- The code snippet, takes the first value from the user and stores in the numOne variable.
- Then, it takes the second value from the user and stores in the numTwo variable.
- It multiples the values and stores the output in the result variable and then, displays the output on the Web page.

© Aptech Ltd. Introduction to JavaScript / Session 12 32

**Built-in Functions 4-4**

- Following figure displays the input first number.
- Following figure displays the input second number.
- Following figure displays the result.

© Aptech Ltd. Introduction to JavaScript / Session 12 33

Using slides 30 to 33, explain the built-in functions of JavaScript.

A function is a piece of code that performs some operations on variables to fulfil a specific task. It takes one or more input values, processes them, and returns an output value. JavaScript provides built-in functions that are already defined to fulfil a certain task.

Explain table which lists the built-in functions. Then, explain the code snippet using slide 31 and 32.

In the code, it takes the first value from the user and stores in the numOne variable. Then, it takes the second value from the user and stores in the numTwo variable. It multiplies the values and stores the output in the result variable and then displays the output on the Web page.

Using slide 33, explain the output of the code.

Show the `prompt()` and `eval()` functions output.

**Tips:**

The `function` keyword must be written in lowercase letters.

**In-Class Question:**

After you finish explaining built-in functions of JavaScript, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which function is used to display information with an OK button?

**Answer:**

`alert()` function is used to display information with a OK button.

**Slide 34**

Let us understand events in JavaScript.

**HTML 5 Events**

- An event occurs when a user interacts with the Web page.
- Some of the commonly generated events are mouse clicks, key strokes, and so on.
- The process of handling these events is known as event handling.
- Following figure displays the event.

© Aptech Ltd. Introduction to JavaScript / Session 12 34

Using slide 34, explain the concept events.

Consider a scenario where you want to design an Employee registration Web form. This form allows the users to fill in the appropriate details and click the `submit` button. When user clicks the `submit` button, the form data is submitted to the server for validation purposes. In this case, when the user clicks the button, an event is generated. The submission of form refers to the action performed on click of the button. An event occurs when a user interacts with the Web page. Some of the commonly generated events are mouse clicks, key strokes, and so on. The process of handling these events is known as event handling.

Explain the figure displays the event on mouse click.

Discuss more events occurrence noticed by the students while working on Operating System (OS) or accessing the Web pages with forms.

Some more examples of events are:

- An event occurring on page load in the browser.
- While moving from one field to another.

### Slides 35 and 36

Let us understand event handling.

**HTML 5 Event Handling 1-2**

- Event handling is a process of specifying actions to be performed when an event occurs. This is done by using an event handler.
- An event handler is a scripting code or a function that defines the actions to be performed when the event is triggered.
- When an event occurs, an event handler function that is associated with the specific event is invoked.
- The information about this generated event is updated on the event object.
- The event object is a built-in object, which can be accessed through the window object.
- It specifies the event state, including information such as the location of mouse cursor, element on which an event occurred, and state of the keys in a keyboard.

© Aptech Ltd. Introduction to JavaScript / Session 12 35

**HTML 5 Event Handling 2-2**

- Following figure displays event handling.

The diagram illustrates the event handling process. A person sitting at a computer is shown with a thought bubble above them containing the text "User Presses a Key". An arrow labeled "Event" points from the person to a box labeled "Event Handler". Another thought bubble above the "Event Handler" box contains the text "Handles the Event".

© Aptech Ltd. Introduction to JavaScript / Session 12 36

Using slides 35 and 36, explain the event handling.

Mention, event handling is a process of specifying actions to be performed when an event occurs. This is done by using an event handler. An event handler is a scripting code or a function that defines the actions to be performed when the event is triggered.

When an event occurs, an event handler function that is associated with the specific event is invoked. The information about this generated event is updated on the event object. The `event` object is a built-in object, which can be accessed through the `window` object.

It specifies the event state, which includes information such as the location of mouse cursor, element on which an event occurred, and state of the keys in a keyboard.

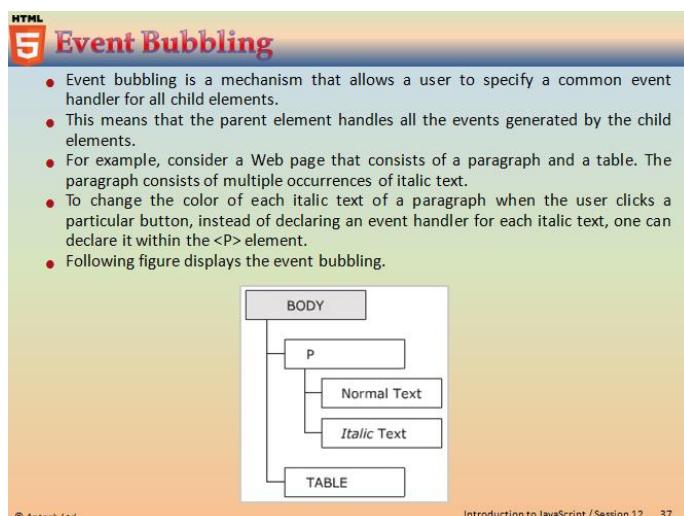
Explain the figure displays the event handling.

**Tips:**

You will find most of the times in JavaScript, an event handler is a function invoked in an `event` attribute.

**Slide 37**

Let us understand event bubbling.

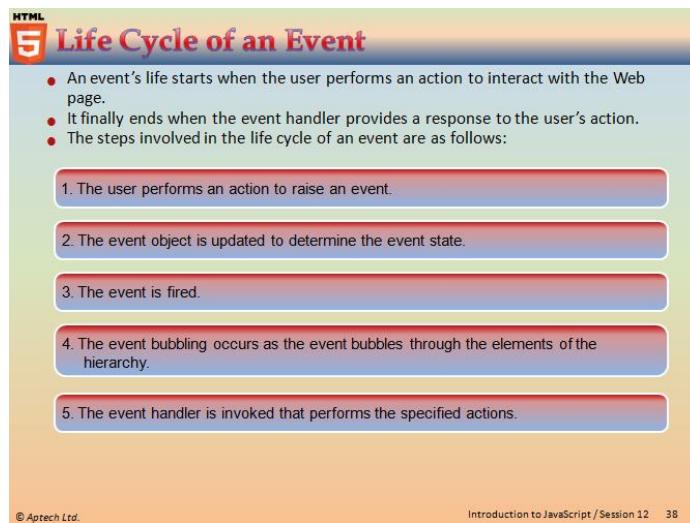


Using slide 37, explain concept of event bubbling.

Explain the event bubbling mechanism in JavaScript along with the figure displaying the event bubbling.

## Slide 38

Let us understand lifecycle of an event.



**HTML5** **Life Cycle of an Event**

- An event's life starts when the user performs an action to interact with the Web page.
- It finally ends when the event handler provides a response to the user's action.
- The steps involved in the life cycle of an event are as follows:

1. The user performs an action to raise an event.
2. The event object is updated to determine the event state.
3. The event is fired.
4. The event bubbling occurs as the event bubbles through the elements of the hierarchy.
5. The event handler is invoked that performs the specified actions.

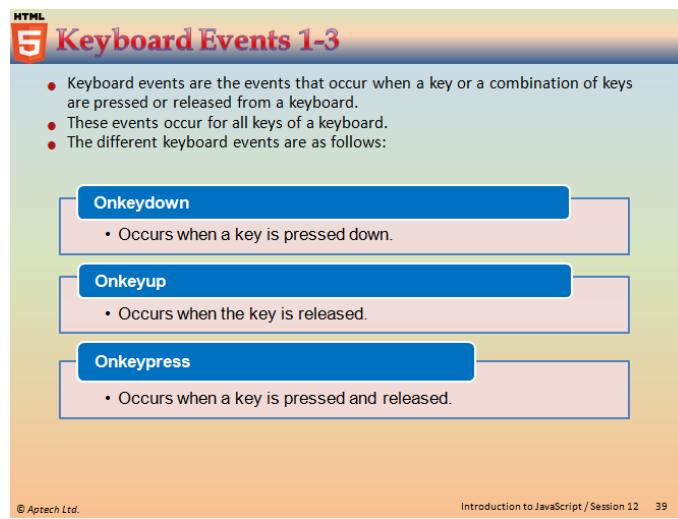
© Aptech Ltd. Introduction to JavaScript / Session 12 38

Using slide 38, explain life cycle of an event.

An event's life starts when the user performs an action to interact with the Web page. It finally ends when the event handler provides a response to the user's action. Explain the steps involved in the life cycle of an event as specified in the slide.

## Slides 39 to 41

Let us understand keyboard events.



**HTML5** **Keyboard Events 1-3**

- Keyboard events are the events that occur when a key or a combination of keys are pressed or released from a keyboard.
- These events occur for all keys of a keyboard.
- The different keyboard events are as follows:

<b>Onkeydown</b>	• Occurs when a key is pressed down.
<b>Onkeyup</b>	• Occurs when the key is released.
<b>Onkeypress</b>	• Occurs when a key is pressed and released.

© Aptech Ltd. Introduction to JavaScript / Session 12 39

**HTML** **5** **Keyboard Events 2-3**

- The Code Snippet demonstrates how to create a JavaScript code that defines the event handlers.

```
function numericonly()
{
    if(!event.keyCode >=48 && event.keyCode<=57)
        event.returnValue=false;
}

function countWords()
{
    var message = document.getElementById('txtMessage').value;
    message= message.replace(/\s+/g, ' ');
    var numberOfWords = message.split(' ').length;
    document.getElementById('txtTrack').value = words
    Remaining: ' + eval(50 - numberOfWords);
    if(numberOfWords > 50)
        alert("too many words.");
}
```

© Aptech Ltd. Introduction to JavaScript / Session 12 40

**HTML** **5** **Keyboard Events 3-3**

- In the code snippet, the function `numericOnly()` declares an event handler function, `numericOnly()`.
- The `event.keyCode` checks if the Unicode character of the entered key is greater than 48 and less than 57.
- This checks that only numeric values are entered. It also declares an event handler function, `countWords()`.
- It retrieves the text specified in the `txtMessage` control. `split()` function splits the specified string when a space is encountered and returns the length after splitting.
- It also calculates and displays the number of remaining words to complete the count of 50 words.
- If the number of words is greater than 50, an alert box is displayed.

© Aptech Ltd. Introduction to JavaScript / Session 12 41

Using slides 39 to 41, explain the different keyboard events.

Keyboard events are the events that occur when a key or a combination of keys are pressed or released from a keyboard. These events occur for all keys of a keyboard. The different keyboard events are as follows:

- **Onkeydown:** Occurs when a key is pressed down.
- **Onkeyup:** Occurs when the key is released.
- **Onkeypress:** Occurs when a key is pressed and released.

Using slides 40 and 41 explain the code snippet demonstrating a keyboard event. Explain the code snippet steps mentioned on the slide.

## Slides 42 to 45

Let us understand mouse events.

**HTML 5 Mouse Events 1-4**

- Mouse events occur when the user clicks the mouse button.
- Following table lists the mouse events.

Event	Description
onmousedown	Occurs when the mouse button is pressed
onmouseup	Occurs when the mouse button is released
onclick	Occurs when the mouse button is pressed and released
ondblclick	Occurs when the mouse button is double-clicked
onmousemove	Occurs when the mouse pointer is moved from one location to other
onmouseover	Occurs when the mouse pointer is moved over the element
onmouseout	Occurs when the mouse pointer is moved out of the element

© Aptech Ltd.

Introduction to JavaScript / Session 12 42

**HTML 5 Mouse Events 2-4**

- The Code Snippet demonstrates the use of different mouse events.

```
<!DOCTYPE HTML>
<html>
<head>
<title> Reservation </title>
<script src="form.js">
</script>
</head>
<body>
<h2> Hotel Reservation Form</h2>
<form id="frmreservation">
<table>
<tr>
<td>
<label for="txtName">Name:</label>
</td>
<td>
<input id="txtName" type="text" />
</td>
</tr>
```

© Aptech Ltd.

Introduction to JavaScript / Session 12 43

43 (a)

**HTML 5 Mouse Events 2-4**

```
<tr>
<td> Arrival Date: </td>
<td> <input id="txtArrival" type="text" /></td>
</tr>
<tr>
<td> Departure Date: </td>
<td> <input id="txtDeparture" type="text" /></td>
</tr>
<tr>
<td> Number of Person: </td>
<td> <input id="txtPerson" type="text" maxlength="3"
size="3"></td>
</tr>
<tr>
<td>

</td>
</tr>
```

© Aptech Ltd.

Introduction to JavaScript / Session 12 43

43 (b)

**HTML 5 Mouse Events 3-4**

```
<td>
    
</td>
</tr>
</table>
</form>
</body>
</html>
```

- In the code snippet, an image is displayed when Submit button is clicked.
- It displays the submit.jpg image when the mouse is released from Submit button.
- It submits the form data when the Submit button is clicked.
- It displays the image when Reset button is clicked and it displays the reset.jpg image when the mouse is released from Reset button.
- It resets the form data when the Reset button is clicked.

© Aptech Ltd. Introduction to JavaScript / Session 12 44

**HTML 5 Mouse Events 4-4**

- The Code Snippet demonstrates the loading of images in a JavaScript file.

```
function showImage(object,url)
{
    object.src=url;
}
```

- Following figure displays the output of mouseup.
- Following figure displays the output on mousedown.

© Aptech Ltd. Introduction to JavaScript / Session 12 45

Using slides 42 to 45, explain the mouse events.

Mouse events occur when the user clicks the mouse button. Explain table which lists the mouse events.

Using slides 43 (a), 43 (b), and 44 explain the code snippet which demonstrate the use of mouse event.

In the code, an image is displayed when Submit button is clicked.

Using slide 45, explain the code snippet and output for loading a image in a JavaScript file.

## Slides 46 to 50

Let us understand focus and selection events.

**HTML Focus and Selection Events 1-4**

- The focus events determine the activation of various elements that uses the element.
- It allows a user to set or reset focus for different elements.
- The selection events occur when an element or a part of an element within a Web page is selected.
- Following table lists the focus and selection events.

Data Type	Description
onfocus	Occurs when an element receives focus
onblur	Occurs when an element loses focus
onselectstart	Occurs when the selection of an element starts
onselect	Occurs when the present selection changes
ondragstart	Occurs when the selected element is moved

© Aptech Ltd. Introduction to JavaScript / Session 12 46

**HTML Focus and Selection Events 2-4**

- The Code Snippet demonstrates the use of focus and selection events.

```
<!DOCTYPE HTML>
<html>
<head>
<title> Reservation </title>
<script>
    function showStyle(field)
    {
        field.style.backgroundColor = '#FFFFCC';
    }
    function hideStyle(field)
    {
        field.style.backgroundColor = '#FFFFFF';
    }
    function setFontStyle(field)
    {
        field.style.fontWeight = 'bold';
        field.style.fontFamily = 'Arial';
    }
</script>
```

© Aptech Ltd. Introduction to JavaScript / Session 12 47

47 (a)

**Focus and Selection Events 2-4**

```
<!DOCTYPE HTML>
<html>
<head>
<title> Reservation </title>
<script>
    function showStyle(field)
    {
        field.style.backgroundColor = '#FFFFCC';
    }
    function hideStyle(field)
    {
        field.style.backgroundColor = '#FFFFFF';
    }
    function setFontStyle(field)
    {
        field.style.fontWeight = 'bold';
        field.style.fontFamily = 'Arial';
    }
</script>
```

© Aptech Ltd. Introduction to JavaScript / Session 12 47

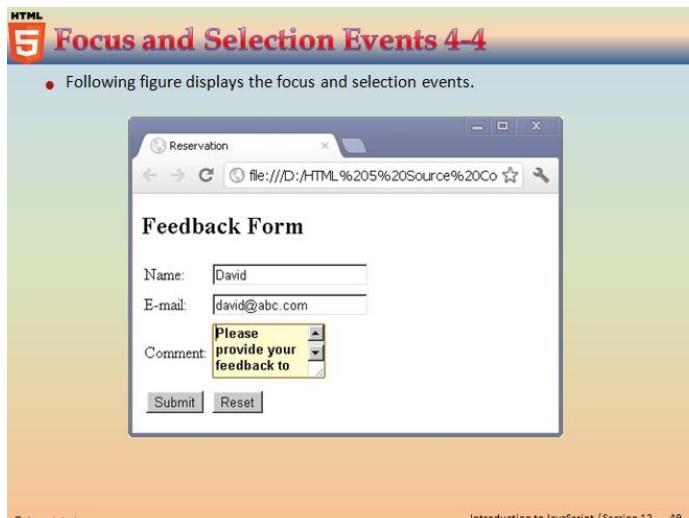
**47 (b)**

**Focus and Selection Events 3-4**

```
<tr>
<td> <label for="txtComment">Comment:</label></td>
<td> <textarea id="txtComment" cols="15" rows="3"
onfocus="showStyle(this);"
onblur="hideStyle(this);"
onselect=setFontStyle(this);> </textarea>
</td>
</tr>
<tr>
<td> <input id="btnSubmit" type="button" type="button"
value="Submit" /></td>
<td> <input id="btnReset" type="reset" /></td>
</tr>
</table>
</form>
</body>
</html>
```

- In the code snippet, a specified style is displayed when the element receives and loses focus.
- It displays the specified font style when the element is selected.
- It declares an event handler function and specifies the background color for the field.
- It sets the font style for text to bold and the text should appear in Arial font.

© Aptech Ltd. Introduction to JavaScript / Session 12 48



Using slides 46 to 49, explain the different focus and selection events.

The focus events determine the activation of various elements that uses the input element. It allows you to set or reset focus for different input elements. The selection events occur when an element or a part of an element within a Web page is selected. Explain table which lists the focus and selection events.

Using slides 47 (a), 47 (b), and 48, explain the code snippet which demonstrates the use of focus and selection events.

Mention, a specified style is displayed when the element receives and loses focus. It also displays the specified font style when the element is selected. It also declares an event handler function and specifies the background color for the field. It sets the font style for text to bold and the text should appear in Arial font.

Using slide 49, explain the output for the code snippet. Tell the students there are many other important and useful events like `onload()` which is called on page load.

#### In-Class Question:

After you finish explaining the focus and selection events, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



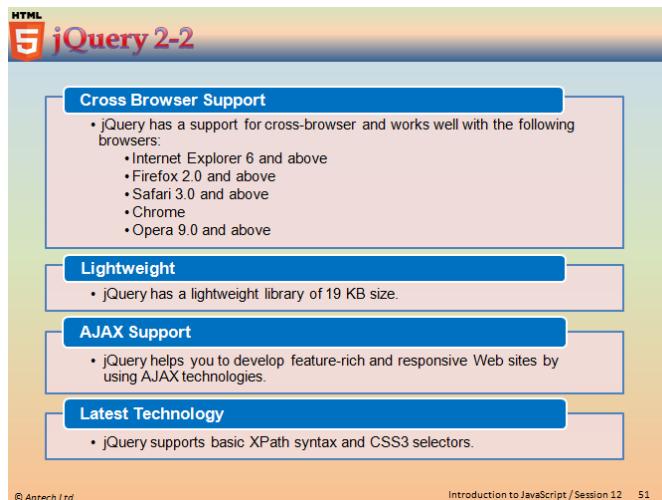
Which event occurs when an element receives focus?

#### Answer:

`onfocus` event occurs when an element receives focus.

#### Slides 50 and 51

Let us understand jQuery.



Using slides 50 and 51, explain the jQuery.

jQuery is a short and fast JavaScript library developed by John Resig in 2006 with a wonderful slogan: Write less and do more. It simplified the client side scripting of HTML. jQuery also simplifies HTML files animation, event handling, traversing, and developing AJAX based Web applications. It helps in rapid Web application development. jQuery is designed for simplifying several tasks by writing lesser code. Explain the features of jQuery specified on the slide.

**Tips:**

1. Some of the features supported in the jQuery library are:

- HTML DOM manipulation
- CSS manipulation
- Adding effects and animations
- AJAX support
- Plugins

2. The jQuery script runs similar in all major browsers, which means it solves cross browser issues.

## Slide 52

Let us understand how to use jQuery library.

**Using jQuery Library**

- To work with jQuery perform the following steps:
  - Download the jQuery library from the <http://jquery.com/> Web site.
  - Place the jquery-1.7.2.min.js file in the current directory of the Web site.
- The user can include jQuery library in their file.
- The Code Snippet shows how to use a jQuery library.

```
<!DOCTYPE HTML>
<html>
<head>
<title>The jQuery Example</title>
// Using jQuery library
<script src="jquery-1.7.2.min.js">
  // The user can add our JavaScript code here
</script>
</head>
<body>
</body>
</html>
```

© Aptech Ltd. Introduction to JavaScript / Session 12 52

Using slide 52, explain the how to use JQuery library.

There is an easy way to use jQuery library. To work with jQuery perform the following steps:

- Download the jQuery library from the <http://jquery.com/> Web site
- Place the jquery-1.7.2.min.js file in the current directory of the Web site. The user can include jQuery library in their file.

Code Snippet shows how to use a jQuery library.

## Slides 53 and 54

Let us understand calling jQuery library functions.

**Calling jQuery Library Functions 1-2**

- While jQuery is reading or manipulating the Document Object Model (DOM) object, the users can add the events when the DOM object is ready.
- If the user wants the event on their page then the user has to call the event in the `$(document).ready()` function.
- The users also register the ready event for the document.
- Place the jquery-1.7.2.min.js file in the current directory and specify the location of this file in the src attribute.
- The Code Snippet shows how to call jQuery library function and ready event in DOM.

```
<!DOCTYPE HTML>
<html>
<head>
<title>The jQuery Example</title>
<script src=" jquery-1.7.2.min.js">
</script>
<script>
$(document).ready(function() {
  $("div").click(function() {
    alert("Welcome to the jQuery world!");
  });
});
</script>
```

© Aptech Ltd. Introduction to JavaScript / Session 12 53

The screenshot displays a presentation slide with the title "Calling jQuery Library Functions 2-2". The slide contains the following content:

```

</head>
<body>
  <div id="firstdiv">
    Click on the text to view a dialog box.
  </div>
</body>
</html>

```

- The code snippet includes the jQuery library and also registers the ready event for the document.
- The ready event contains the click function that calls the click event.
- Following figure displays the output of jQuery.

The screenshot shows a browser window titled "The jQuery Example" with the URL "file:///D:/HTML%205%20Source%20Co". Inside the browser, there is a div with the id "firstdiv" containing the text "Click on the text to view a dialog box.". A JavaScript alert box is overlaid on the page, displaying the message "Welcome to the jQuery world!" with an "OK" button.

Using slides 53 and 54, explain the process of calling a jQuery library function.

Users can do many tasks while jQuery is reading or manipulating the DOM object. The users can add the events only when the DOM object is ready. If the user wants the event on their page then the user has to call the event in the `$(document).ready()` function. All the content inside the event will be loaded as soon as the DOM is loaded, but before the contents of the page are loaded. The users also register the ready event for the document.

Place the `jquery-1.7.2.min.js` file in the current directory and specify the location of this file in the `src` attribute.

#### Tips:

To use jQuery, you first select an HTML element which is referred as **selector**. Then perform some task on it called as **action**.

Thus, syntax is as follows:

`$(selector).action()`

In the syntax, \$ symbol means to access the element.

Some of the examples are:

`$(“p”).hide` – hides all `<p>` elements on the page.  
`$(“p”).show` – shows all the `<p>` elements on the page.  
`$("#div1").animate({height:"200px";})` – animates the height.

Also, all jQuery methods are invoked inside a document. Thus, the syntax is:

`$(document).ready(function(){`

```
// jQuery methods go here...
});
```

Then, explain code snippet which shows how to call jQuery library function and ready event in DOM. The code includes the jQuery library and also registers the ready event for the document. The ready event contains the click function that calls the click event.

## Slides 55 to 60

Let us understand jQuery mobile.

**jQuery Mobile 1-6**

- jQuery mobile is a Web User Interface (UI) development framework that allows the user to build mobile Web applications that work on tablets and smartphones.
- The jQuery mobile framework provides many facilities that include XML DOM and HTML manipulation and traversing, performing server communication, handling events, image effects, and animation for Web pages.
- The basic features of jQuery mobile are as follows:

<b>Simplicity</b>	<ul style="list-style-type: none"> <li>This framework is easy to use and allows developing Web pages by using markup driven with minimum or no JavaScript.</li> </ul>
<b>Accessibility</b>	<ul style="list-style-type: none"> <li>The framework supports Accessible Rich Internet Applications (ARIA) that helps to develop Web pages accessible to visitors with disabilities.</li> </ul>
<b>Enhancements and Degradation</b>	<ul style="list-style-type: none"> <li>The jQuery mobile is influenced by the latest HTML5, JavaScript, and CSS3.</li> </ul>

© Aptech Ltd. Introduction to JavaScript / Session 12 55

**jQuery Mobile 2-6**

- Themes**
  - This framework provides themes that allow the user to provide their own styling.
- Smaller Size**
  - The size for jQuery mobile framework is smaller for CSS it is 6KB and for JavaScript library it is 12KB.

• The Code Snippet shows an example of a jQuery mobile.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
<link rel="stylesheet" href="jquery.mobile-1.0a3.min.css" />
<script src="jquery-1.5.min.js"></script>
<script src="jquery.mobile-1.0a3.min.js"></script>
</head>
<body>
<div data-role="page">
<div data-role="header">
<h1>Car Rental</h1>
</div>
```

© Aptech Ltd. Introduction to JavaScript / Session 12 56

**HTML** **jQuery Mobile 3-6**

```
<div data-role="content">
<p>Choose from the listed car models</p>
<ul data-role="listview" data-inset="true">
<li><a href="#">Ford</a></li>
<li><a href="#">Ferrari</a></li>
<li><a href="#">BMW</a></li>
<li><a href="#">Toyota</a></li>
<li><a href="#">Mercedes-Benz</a></li>
</ul>
</div>
<div data-role="footer">
<h4>©copy; DriveCars 2012.</h4>
</div>
</body>
</html>
```

- The jQuery mobile application should have the following three files:
  - CSS file
  - jQuery library
  - jQuery Mobile library

© Aptech Ltd. Introduction to JavaScript / Session 12 57

**HTML** **jQuery Mobile 4-6**

- In this code snippet, three files are included, the CSS (`jquery.mobile-1.0a3.min.css`), jQuery library (`jquery-1.5.min.js`), and the jQuery mobile library (`jquery.mobile-1.0a3.min.js`).
- A user can also download the jQuery libraries from <http://code.jquery.com/> Web site.

The jQuery Mobile takes HTML tags and renders them on mobile devices. To work with this, HTML has to make use of data attributes.

jQuery uses these attributes as indicators for rendering it on the Web pages.

jQuery also looks for div using a particular data-role values such as page, content, header, and footer.

There are multiple div blocks added to the code for page, content, header, and footer.

To display the different car models a data-role listview is added to enhance the look and feel of the mobile Web page.

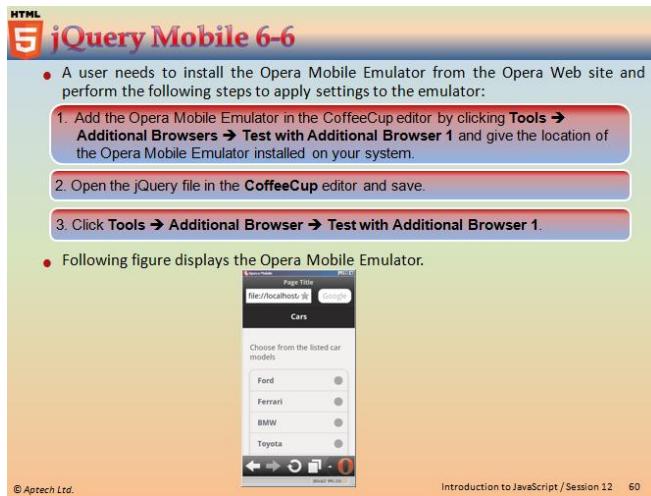
© Aptech Ltd. Introduction to JavaScript / Session 12 58

**HTML** **jQuery Mobile 5-6**

- A user need to install the Opera Mobile Emulator from the Opera Web site.
- After installing the **Opera Mobile Emulator**, perform the following steps to apply settings to the emulator:

1. Select All Programs → **Opera Mobile Emulator** → **Opera Mobile Emulator**. The **Opera Mobile Emulator** dialog box will be displayed.
2. In the **Profile** tab, select the **Samsung Galaxy Tab**.
3. In the **Resolution** drop-down, select the **WVGA Portrait(480x800)**.
4. Click **Update**.
5. Click **Launch**. The **Samsung Galaxy** tab is displayed.

© Aptech Ltd. Introduction to JavaScript / Session 12 59



Using slides 55 to 60, explain the jQuery mobile and its basic functions.

jQuery mobile is a Web User Interface (UI) development framework that allows the user to build mobile Web applications that work on tablets and smartphones. The jQuery mobile framework provides many facilities that include XML DOM and HTML manipulation and traversing, performing server communication, handling events, image effects, and animation for Web pages. The basic features of jQuery mobile are as follows:

- Simplicity:** This framework is easy to use and allows developing Web pages by using markup driven with minimum or no JavaScript.
- Accessibility:** The framework supports Accessible Rich Internet Applications (ARIA) that helps to develop Web pages accessible to visitors with disabilities.
- Enhancements and Degradation:** The jQuery mobile is influenced by the latest HTML5, JavaScript, and CSS3.
- Themes:** This framework provides themes that allow the user to provide their own styling.
- Smaller Size:** The size for jQuery mobile framework is smaller for CSS it is 6KB and for JavaScript library it is 12KB.

Using slide 57, explain the code snippet which shows an example of a jQuery mobile.

The jQuery mobile application should have the following three files:

- CSS file
- jQuery library
- jQuery Mobile library

**Tips:**

jQuery Mobile is built on top of the jQuery library, which makes it easy to learn if you already know jQuery. It uses HTML5, CSS3, JavaScript, and AJAX to accomplish its work for laying out pages with minimal scripting.

Using slide 58, explain further about jQuery mobile requirements.

In the code, three files are included the CSS (`jquery.mobile-1.0a3.min.css`), jQuery library (`jquery-1.5.min.js`), and the jQuery mobile library (`jquery.mobile-1.0a3.min.js`). A user can also download the jQuery libraries from <http://code.jquery.com/> Web site.

**Tips:**

Place the downloaded file in the same directory as the pages where you wish to use it.

The jQuery Mobile takes HTML tags and renders them on mobile devices. To work with this, HTML has to make use of data attributes. jQuery use these attributes as indicators for rendering it on the Web pages. jQuery also looks for div using a particular data-role values such as `page`, `content`, `header`, and `footer` are used in this code. There are multiple div blocks added to the code for `page`, `content`, `header`, and `footer`. Similarly, to display the different car models a data-role listview is added to enhance the look and feel of the mobile Web page.

A user need to install the Opera Mobile Emulator from the Opera Web site.

Using slide 59, explain for the process of installing Opera Mobile Emulator and steps to apply the setting to the emulator.

Using slide 60, explain the steps for executing the code of the CoffeeCup editor and execute page in mobile emulator.

**In-Class Question:**

After you finish explaining the jQuery mobile, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which three files a mobile application should have?

**Answer:**

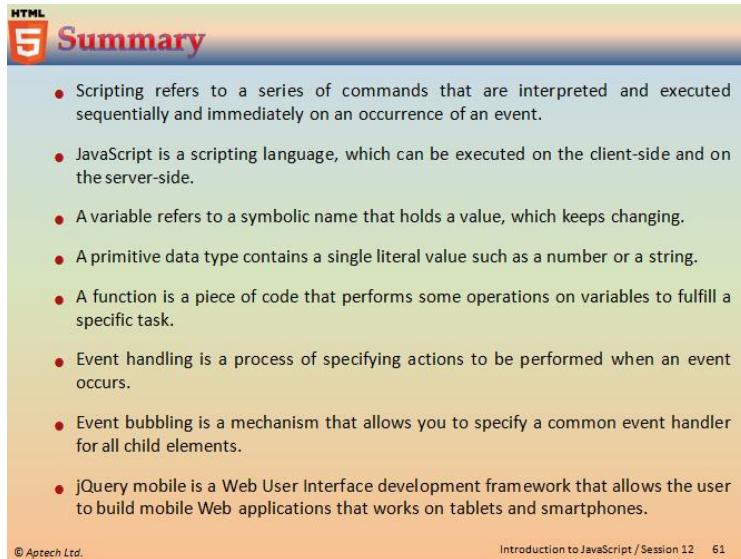
The jQuery mobile application should have the following three files:

- CSS file (`jquery.mobile-1.0a3.min.css`)

- jQuery library (jquery-1.5.min.js)
- jQuery Mobile library (jquery.mobile-1.0a3.min.js)

## Slide 61

Let us summarize the session.



The slide has a header 'HTML 5 Summary'. The main content is a bulleted list of 11 points about scripting, JavaScript, variables, data types, functions, event handling, event bubbling, and jQuery mobile. At the bottom left is the copyright notice '© Aptech Ltd.' and at the bottom right is 'Introduction to JavaScript / Session 12 61'.

- Scripting refers to a series of commands that are interpreted and executed sequentially and immediately on an occurrence of an event.
- JavaScript is a scripting language, which can be executed on the client-side and on the server-side.
- A variable refers to a symbolic name that holds a value, which keeps changing.
- A primitive data type contains a single literal value such as a number or a string.
- A function is a piece of code that performs some operations on variables to fulfill a specific task.
- Event handling is a process of specifying actions to be performed when an event occurs.
- Event bubbling is a mechanism that allows you to specify a common event handler for all child elements.
- JQuery mobile is a Web User Interface development framework that allows the user to build mobile Web applications that works on tablets and smartphones.

In slide 61, you will summarize the session. You will end the session, with a brief summary of what has been taught in the session.

### 12.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session. You should also explore the Operators and Statements that are offered with the next session.

#### Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 13 – Operators and Statements

---

## 13.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 13.1.1 Objectives

By the end of this session, the learners will be able to:

- Explain operators and their types in JavaScript
- Explain regular expressions in JavaScript
- Explain decision-making statements in JavaScript

### 13.1.2 Teaching Skills

To teach this session, you should be well-versed with the different type of operators supported by JavaScript. Also how to use pattern matching functions on text using regular expressions should be known. You should be also aware of decision-making statements in JavaScript. You can refer this site while preparing for regular expressions: [https://developer.mozilla.org/en/docs/Web/JavaScript/Guide/Regular\\_Expressions](https://developer.mozilla.org/en/docs/Web/JavaScript/Guide/Regular_Expressions)

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

#### In-Class Activities:

Follow the order given here during In-Class activities.

## Overview of the Session:

Then give the students the overview of the current session in the form of session objectives.  
Show the students slide 2 of the presentation.

**HTML5 Objectives**

- Explain operators and their types in JavaScript
- Explain regular expressions in JavaScript
- Explain decision-making statements in JavaScript

© Aptech Ltd. Operators and Statements / Session 13 2

Tell the students that this session introduces operators and their types in JavaScript. They will learn about regular expressions in JavaScript. They will also learn various decision-making statements used in JavaScript.

## 13.2 In-Class Explanations

### Slide 3

Let us understand operators in JavaScript.

**HTML5 Introduction**

An operator specifies the type of operation to be performed on the values of variables and expressions.

JavaScript provides different types of operators to perform simple to complex calculations and evaluations.

Certain operators are also used to construct relational and logical statements. These statements allow implementing decision and looping constructs.

© Aptech Ltd. Operators and Statements / Session 13 3

Using slide 3, explain the importance of operators.

An operator specifies the type of operation to be performed on the values of variables and expressions. JavaScript provides different types of operators to perform simple to complex calculations and evaluations.

Certain operators are also used to construct relational and logical statements. These statements allow implementing decision and looping constructs.

## Slides 4 and 5

Let us understand the basics of operators.

**HTML 5 Basics of Operators 1-2**

- An operation is an action performed on one or more values stored in variables.
- The specified action either changes the value of the variable or generates a new value.
- An operation requires minimum one symbol and some value.
- Symbol is called an operator and it specifies the type of action to be performed on the value.
- Value or variable on which the operation is performed is called an operand.

© Aptech Ltd. Operators and Statements / Session 13 4

**HTML 5 Basics of Operators 2-2**

- Three main types of operators are as follows:
  - Unary operators - Operates on a single operand.
  - Binary operators - Operates on two operands.
  - Ternary operators - Operates on three operands.

© Aptech Ltd. Operators and Statements / Session 13 5

Using slides 4 and 5, explain the basics of operators in JavaScript.

An operation is an action performed on one or more values stored in variables. The specified action either changes the value of the variable or generates a new value. Here, the symbol is called an operator and it specifies the type of action to be performed on the value.

The value or variable on which the operation is performed is called an operand.

For example,  $X * 2$  is an expression, where  $X$  and  $2$  are operands and  $*$  is an operator.

There are three main types of operators, which are as follows:

- **Unary operators** - Operates on a single operand. For example, the expression  $y = -x$ .

- **Binary operators** - Operates on two operands. For example, the expression `sum = y + x.`
- **Ternary operators** - Operates on three operands. For example, the expression `age >= 18 ? "Eligible" : "Not Eligible".`

## Slide 6

Let us understand operators in JavaScript and their types.

The slide has a blue header bar with the title 'Operators and their Types'. Below the header, there are three main bullet points in colored boxes:

- Red box:** Operators help in simplifying expressions.
- Green box:** JavaScript provides a predefined set of operators that allow performing different operations.
- Purple box:** JavaScript operators are classified into six categories based on the type of action they perform on operands.

Under the purple box, there is a sub-list:

- Six categories of operators are as follows:
  - Arithmetic operators
  - Relational operators
  - Logical operators
  - Assignment operators
  - Bitwise operators
  - Special operators

At the bottom left is the copyright notice '© Aptech Ltd.', at the bottom center is 'Operators and Statements / Session 13', and at the bottom right is the number '6'.

Using slide 6, explain the operators and its types.

Operators help in simplifying expressions. JavaScript provides a predefined set of operators that allow performing different operations. JavaScript operators are classified into six categories based on the type of action they perform on operands. These six categories of operators are as follows:

- Arithmetic operators
- Relational operators
- Logical operators
- Assignment operators
- Bitwise operators
- Special operators

## Slides 7 and 8

Let us understand arithmetic operators.

**HTML 5 Arithmetic Operators 1-2**

- Are binary operators.
- Perform basic arithmetic operations on two operands.
- Operator appears in between the two operands, which allow you to perform computations on numeric and string values.

- Following table lists arithmetic operators.

Arithmetic Operator	Description	Example
+(Addition)	Performs addition. In case of string values, it behaves as a string concatenation operator and appends a string at the end of the other	45 + 56
-(Subtraction)	Performs subtraction. If a larger value is subtracted from a smaller value, it returns a negative numeric value	76 - 78
/(Division)	Divides the first operand by the second operand and returns the quotient	24 / 8
% (Modulo)	Divides the first operand by the second operand and returns the remainder	90 % 20
*(Multiplication)	Multiples the two operands	98 * 10

© Aptech Ltd. Operators and Statements / Session 13 7

**HTML 5 Arithmetic Operators 2-2**

- The Code Snippet demonstrates the use of arithmetic operators.

```
<SCRIPT>
var loanAmount = 34500;
var interest = 8;
var interestAmount, totalAmount;
interestAmount = loanAmount * (interest / 100);
totalAmount = loanAmount + interestAmount;
document.write("<B>Total amount to be paid ($):</B>" +
totalAmount + "<BR />");
</SCRIPT>
```

© Aptech Ltd. Operators and Statements / Session 13 8

Using slides 7 and 8, explain the arithmetic operators.

Arithmetic operators are binary operators, as they perform basic arithmetic operations on two operands. The operator appears in between the two operands, which allow you to perform computations on numeric and string values. These computations include addition, subtraction, multiplication, and division.

Explain table which lists the arithmetic operators with their descriptions and an example of each type.

Using slide 8, explain the code snippet which demonstrates the use of arithmetic operators.

Mention, code snippet calculates the loan interest and the total loan amount to be repaid by using the arithmetic operators. The code uses the arithmetic operators \* and / to calculate the interest amount. This interest amount is then added to the total amount by

using the + operator. The `write()` method displays the total amount to be repaid on the loan.

### In-Class Question:

After you finish explaining arithmetic operators, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What is the modulo (%) operator used for?

### Answer:

Modulo (%) operator is for finding remainder in a division operation.

### Tips:

The + operator when used with string values, concatenates the values.

Example: `var str1="Welcome";`

```
var str2="To JavaScript";
```

```
str3 = str1 + str2;
```

This will result in Welcome To JavaScript

## Slide 9

Let us understand increment and decrement operators.

**HTML5 Increment and Decrement Operators 1-2**

- Increment and decrement operators are unary operators.
- Increment operator (++) increases the value by 1, while the decrement operator (--) decreases the value by 1.
- These operators can be placed either before or after the operand.
- Operator if placed before the operand, expression is called pre-increment or pre-decrement. Operator if placed after the operand, expression is called post-increment or post-decrement.
  - Following table lists arithmetic operators.

Expressions	Type	Result
<code>numTwo = ++numOne;</code>	Pre-increment	<code>numTwo = 3</code>
<code>numTwo = numOne++;</code>	Post-increment	<code>numTwo = 2</code>
<code>numTwo = --numOne;</code>	Pre-decrement	<code>numTwo = 1</code>
<code>numTwo = numOne--;</code>	Post-decrement	<code>90 % 20</code>

© Aptech Ltd. Operators and Statements / Session 13 9

Using slide 9, explain the increment and decrement operators of JavaScript.

The increment and decrement operators are unary operators, as they operate only on a single operand. The increment operator (++) increases the value by 1, while the decrement operator (--) decreases the value by 1. These operators can be placed either before or after the operand. If the operator is placed before the operand, the expression is called pre-increment or pre-decrement. If the operator is placed after the operand, the expression is called post-increment or post-decrement.

Explain table which demonstrates the use of increment and decrement operators by assuming that the numOne variable's value is 2.

### Slide 10

Let us understand code snippet on increment and decrement operator.

**HTML**

## 5 Increment and Decrement Operators 2-2

- The Code Snippet demonstrates the use of unary operators in JavaScript.

```
<SCRIPT>
  var number = 3;
  alert('Number after increment = ' + ++number);
  alert('Number after decrement = ' + number--);
</SCRIPT>
```

© Aptech Ltd. Operators and Statements / Session 13 10

Using slide 10, explain the code snippet which demonstrate use of increment and decrement operators.

The first `alert()` function will display the incremented value of the `number` variable. This is because in the first statement, `++` operator is evaluated first, and then the incremented value is substituted in the variable `number`.

The second `alert()` function will not display the decremented value of the `number` variable. This is because the current value is first assigned to the variable, and then the `--` operator is evaluated.

## Slides 11 to 13

Let us understand relational operators.

**HTML 5 Relational Operators 1-3**

- Are binary operators that make a comparison between two operands.
- After making a comparison, they return a boolean value namely, true or false.
- Expression consisting of a relational operator is called as the relational expression or conditional expression.

- Following table lists the relational operators.

Relational Operators	Description	Example
<code>== (Equal)</code>	Verifies whether the two operands are equal	<code>90 == 91</code>
<code>!= (Not Equal)</code>	Verifies whether the two operands are unequal	<code>99 != 98</code>
<code>=== (Strict Equal)</code>	Verifies whether the two operands are equal and are of the same type	<code>numTwo = 1</code>
<code>!== (Strict Not Equal)</code>	Verifies whether the two operands are unequal and whether are not of the same type	<code>90 % 20</code>

© Aptech Ltd. Operators and Statements / Session 13 11

**HTML 5 Relational Operators 2-3**

- Following table lists some more relational operators.

Relational Operators	Description	Example
<code>&gt; (Greater Than)</code>	Verifies whether the left operand is greater than the right operand	<code>97 &gt; 95</code>
<code>&lt; (Less Than)</code>	Verifies whether the left operand is less than the right operand	<code>94 &lt; 96</code>
<code>&gt;= (Greater Than or Equal)</code>	Verifies whether the left operand is greater than or equal to the right operand	<code>92 &gt;= 93</code>
<code>&lt;= (Less Than or Equal)</code>	Verifies whether the left operand is less than or equal to the right operand	<code>99 &lt;= 100</code>

© Aptech Ltd. Operators and Statements / Session 13 12

**HTML 5 Relational Operators 3-3**

- The Code Snippet demonstrates the use of relational operators in JavaScript.

```
<SCRIPT>
var firstNumber = 3;
var secondNumber = 4;
document.write('First number is greater than the second
number: ' + (firstNumber > secondNumber));
document.write('<br/>First number is less than the
second number: ' + (firstNumber < secondNumber));
document.write('<br/>First number is equal to the second
number: ' + (firstNumber == secondNumber));
</SCRIPT>
```

© Aptech Ltd. Operators and Statements / Session 13 13

Using slides 11 to 13, explain the relational operators in JavaScript.

Relational operators are binary operators that make a comparison between two operands. After making a comparison, they return a boolean value namely, true or false.

The expression consisting of a relational operator is called as the relational expression or conditional expression.

Explain table which lists the relational operators along with their descriptions and an example of each type.

Using slide 13, explain the code snippet which demonstrates use of relational operators in JavaScript.

Mention, the code snippet compares the value of the `firstNumber` variable with the value of the `secondNumber` variable using relational operators.

In the code, each condition is evaluated to return a boolean value. The `alert()` function displays the boolean value as true or false.

#### Slide 14

Let us understand logical operators.

**HTML 5 Logical Operators 1-2**

Are binary operators that perform logical operations on two operands.

They belong to the category of relational operators, as they return a boolean value.

- Following table lists the logical operators.

Logical Operators	Description	Example
<code>&amp;&amp; (AND)</code>	Returns true, if either of the operands are evaluated to true. If first operand evaluates to true, it will ignore the second operand	<code>(x == 2) &amp;&amp; (y == 5)</code> Returns false
<code>!(NOT)</code>	Returns false, if the expression is true and vice-versa	<code>!(x == 3)</code> Returns true
<code>  (OR)</code>	Returns true, if either of the operands are evaluated to true. If first operand evaluates to true, it will ignore the second operand	<code>(x == 2)    (y == 5)</code> Returns true

© Aptech Ltd. Operators and Statements / Session 13 14

Using slide 14, explain the logical operators.

Logical operators are binary operators that perform logical operations on two operands. They belong to the category of relational operators, as they return a boolean value.

Explain table which lists the various logical operators and an example of each type, assuming that `x` is 2 and `y` is 2.

## Slide 15

Let us understand the use of logical operators.

The slide title is "Logical Operators 2-2". A bullet point states: "The Code Snippet demonstrates the use of logical AND operator in JavaScript." Below is the code:

```
<SCRIPT>
var name = "John";
var age = 23;
alert('John\'s age is greater than or equal to 23 years : ' +
((name=="John") && (age >= 23)));
</SCRIPT>
```

At the bottom, it says "Operators and Statements / Session 13" and "15".

Using slide 15, explain the code snippet which demonstrates the use of logical operators.

The code declares and initializes two variables namely, name and age to John and 23 respectively. In the `alert()` function, the logical AND operator checks whether the value of the name variable is John and the value of the age variable is 23. Here, both the expressions are true and therefore, the operator will return true, which will be displayed in the message box to the user.

## Slide 16

Let us understand assignment operators.

The slide title is "Assignment Operators". It contains the following points:

- Assignment operators assign the value of the right side operand to the operand on the left side by using the equal to operator (=).
- Simple assignment operator - Is the '=' operator which is used to assign a value or result of an expression to a variable.
- Compound assignment operator - Is formed by combining the simple assignment operator with the arithmetic operators.

A bullet point states: "Following table lists the assignment operators."

Expressions	Description	Example
<code>numOne += 6;</code>	<code>numOne = numOne + 6</code>	<code>numOne = 12</code>
<code>numOne -= 6;</code>	<code>numOne = numOne - 6</code>	<code>numOne = 0</code>
<code>numOne *= 6;</code>	<code>numOne = numOne * 6</code>	<code>numOne = 36</code>
<code>numOne %= 6;</code>	<code>numOne = numOne % 6</code>	<code>numOne = 0</code>
<code>numOne /= 6;</code>	<code>numOne = numOne / 6</code>	<code>numOne = 1</code>

At the bottom, it says "Operators and Statements / Session 13" and "16".

Using slide 16, explain the assignment operators.

Assignment operators assign the value of the right side operand to the operand on the left side by using the equal to operator (=). The assignment operator is divided into two categories in JavaScript that is as follows:

Simple assignment operator - Is the '=' operator which is used to assign a value or result of an expression to a variable. For example, `result = numOne + numTwo;`

Compound assignment operator – Is formed by combining the simple assignment operator with the arithmetic operators. For example, `salary -= eval(salary * tax / 100);`

Explain table which demonstrates the use of assignment operator by assuming the value of the variable numOne as 6.

## Slides 17 and 18

Let us understand bitwise operators.

HTML 5 Bitwise Operators 1-2

- Represent their operands in bits (zeros and ones) and perform operations on them.
- They return standard decimal values.
- Compound assignment operator - Is formed by combining the simple assignment operator with the arithmetic operators.

© Aptech Ltd. Operators and Statements / Session 13 17

HTML 5 Bitwise Operators 2-2

- Following table lists the bitwise operators in JavaScript.
- The Code Snippet demonstrates the use of bitwise operators.

Bitwise Operators	Description	Example
& (Bitwise AND)	Compares two bits and returns 1 if both of them are 1 or else returns 0	00111000 Returns 00011000
~ (Bitwise NOT)	Inverts every bits of the operand and is a unary operator	-00010101 Returns 11101010
(Bitwise OR)	Compares two bits and returns 1 if the corresponding bits of either or both the operands is 1	00111000 Returns 00111100

```
// (56 = 00111000 and 28 = 00011100)
alert("56" + ' & ' + "28" + ' = ' + (56 & 28));
//(56 = 00111000 and 28 = 00011100)
alert("56" + ' | ' + "28" + ' = ' + (56 | 28));
```

© Aptech Ltd. Operators and Statements / Session 13 18

Using slides 17 and 18, explain the bitwise operator.

Bitwise operators represent their operands in bits (zeros and ones) and perform operations on them. However, they return standard decimal values.

**Tips:**

1. Bitwise operators work on 32 bits containing 0's and 1's, instead of decimal, hexadecimal, or octal numbers.

2. Refer this link to understand on bitwise operators:

[https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Operators/Bitwise\\_Operators](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Operators/Bitwise_Operators)

**Slides 19 and 20**

Let us understand special operators.

The screenshot shows a slide titled "Special Operators 1-2". A red callout box says: "There are some operators in JavaScript which do not belong to any of the categories of JavaScript operators." A green callout box says: "Such operators are referred to as the special operators." A bulleted list says: "Following table lists the special operators in JavaScript." A table follows:

Special Operators	Description
, (comma)	Combines multiple expressions into a single expression, operates on them in the left to right order and returns the value of the expression on the right.
? : (conditional)	Operates on three operands where the result depends on a condition. It is also called as ternary operator and has the form condition ? value1:value2, if the condition is true, the operator obtains value1 or else obtains value2.
typeof	Returns a string that indicates the type of the operand. The operand can be a string, variable, keyword, or an object.

© Aptech Ltd. Operators and Statements / Session 13 19

The screenshot shows a slide titled "Special Operators 2-2". A bulleted list says: "The Code Snippet demonstrates the use of special operator." Below is a code snippet and its execution:

```
<SCRIPT>
var age = parseInt(prompt("Enter age", "Age"))
status = ((typeof(age) == "number" && (age >= 18))
? "eligible" : "not eligible";
document.write('You are ' + age + ' years old, so you are '
+status + ' to vote.');
</SCRIPT>
```

A screenshot of a browser window titled "Special Operators" shows a prompt dialog box asking "Enter age" with the value "19" entered. Below the dialog, the browser output area displays: "You are 19 years old, so you are eligible to vote."

© Aptech Ltd. Operators and Statements / Session 13 20

Using slides 19 and 20, explain special operators.

There are some operators in JavaScript which do not belong to any of the categories of JavaScript operators. Such operators are referred to as the special operators.

Explain table which lists the most commonly used special operators in JavaScript.

Using slide 20, explain the code snippet which demonstrates use of special operators and output for the same.

In the code, the `prompt()` function accepts the value from the user. The value is converted into an integer and is stored in the `age` variable. The conditional operator specifies a condition before `?` symbol. The condition checks whether the value of the variable `age` is a number and whether it is greater than equal to 18. If both these expressions return `true`, the value `eligible` is assigned to the `status` variable. Otherwise, the value `not eligible` is assigned to the `status` variable. The `alert()` function displays the final output as whether the user is `eligible` for voting or not `eligible`.

Also explain the figure which shows output for the code.

### In-Class Question:

After you finish explaining special operators, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which operator returns the type of an operand?

### Answer:

`typeof` operator return the type of a operand.

### Slide 21

Let us understand operator precedence.

Operator Precedence			
Precedence Order	Operator	Description	Associativity
1	<code>()</code>	Parentheses	Left to Right
2	<code>++</code> , <code>--</code>	Post-increment and Post-decrement operators	Not Applicable
3	<code>typeof</code> , <code>++</code> , <code>--</code> , <code>,</code> , <code>-</code> , <code>-</code> , <code>!</code>	Pre-increment and Pre-decrement operators, Logical NOT, Bitwise NOT, and Unary negation	Right to Left
4	<code>*</code> , <code>/</code> , <code>-</code>	Multiplication, Division, and Modulo	Left to Right
5	<code>+</code> , <code>-</code>	Addition and Subtraction	Left to Right
6	<code>&lt;</code> , <code>&lt;=</code> , <code>&gt;</code> , <code>&gt;=</code>	Less than, Less than or equal, Greater than, and Greater than or equal	Left to Right
7	<code>==</code> , <code>==</code> , <code>!=</code>	Equal to, Strict equal to, Not equal to, and Strict not equal to	Left to Right
8	<code>&amp;</code> , <code> </code> , <code>^</code> , <code>&amp;&amp;</code> , <code>  </code>	Bitwise AND, Bitwise OR, Bitwise XOR, Logical AND, and Logical OR	Left to Right
9	<code>?:</code>	Conditional operator	Right to Left
10	<code>=</code> , <code>+=</code> , <code>-=</code> , <code>*=</code> , <code>/=</code> , <code>%=</code>	Assignment operators	Right to Left
11	<code>,</code>	Comma	Left to Right

© Aptech Ltd. Operators and Statements / Session 13 21

Using slide 21, explain the operator precedence.

Operators in JavaScript have certain priority levels based on which their execution sequence is determined. For example, the division operator `(/)` has a higher priority than the

subtraction (-) operator. Therefore, the division operator will be carried out first, if an expression involves both these operators.

Further, an execution order is also defined for the operators within expression. This order is referred to as the associativity, which is either from left to right or vice-versa depending upon the operators.

Explain table which lists the precedence of the operators from the highest to the lowest and their associativity.

## Slide 22

Let us understand regular expression.

**HTML 5 Regular Expressions**

- Is a pattern that is composed of set of strings, which is to be matched to a particular textual content.
- Allow handling of textual data effectively as it allows searching and replacing strings.
- Allows handling of complex manipulation and validation, which could otherwise be implemented through lengthy scripts.

● There are two ways to create regular expressions which are as follows:

- Literal Syntax:**
  - Refers to a static value
  - Allows specifying a fixed pattern, which is stored in a variable
  - Syntax is as follows:  
• var variable\_name = /regular\_expression\_pattern/;
- RegExp() Constructor:**
  - Is useful when the Web page designer does not know the pattern at the time of scripting
  - Method dynamically constructs a regular expression when the script is executed
  - Syntax is as follows:  
• var variable\_name = new RegExp("regular\_expression\_pattern", "flag");

© Aptech Ltd. Operators and Statements / Session 13 22

Using slide 22, explain the regular expression.

A regular expression is a pattern that is composed of set of strings, which is to be matched to a particular textual content. For example, you can specify a pattern for US postal code that the code will not contain more than five digits. When the user enters the postal code, the digits entered by the user will be verified against the pattern to ensure that the postal code is valid.

Regular expressions allow handling textual data effectively, as it allows searching and replacing strings. They allow handling complex manipulation and validation that could otherwise be implemented through lengthy scripts.

In JavaScript, there are two ways to create regular expressions which are as follows:

- **Literal Syntax**

A literal refers to a static value. Therefore, a literal syntax allows specifying a fixed pattern, which is stored in a variable. This method of specifying patterns is useful when the Web page designer knows the pattern at the time of scripting. The syntax to specify the literal syntax is as follows:

**Syntax:**

```
var variable_name = /regular_expression_pattern/;
```

where, regular\_expression\_pattern: Is a string pattern.

variable\_name: Is the name of the variable which stores the pattern.

- **RegExp () Constructor**

The RegExp () constructor is useful when the Web page designer does not know the pattern at the time of scripting. This means that the method dynamically constructs a regular expression when the script is executed. The RegExp () constructor is a function that returns a reference to the built-in RegExp object. The syntax to use the RegExp () constructor is as follows:

**Syntax:**

```
var variable_name = new RegExp("regular_expression_
pattern","flag");
```

where, new: Is a keyword and a special operator that creates the RegExp object.

variable\_name: Is the name of the variable which refers to the RegExp object that holds the pattern.

flag: Is a letter that specifies whether to search for patterns in the complete string and to consider the casing of characters in the string.

## Slides 23 to 25

Let us understand methods and properties of regular expressions.

**HTML 5 RegExp Methods and Properties**

- RegExp object supports methods that are used for searching the pattern in a string. They are as follows:

**test(string)** - Tests a string for matching a pattern and returns a Boolean value of true or false. The Boolean value indicates whether the pattern exists or not in the string. The method is commonly used for validation.

**exec(string)** - Executes a string to search the matching pattern within it. The method returns a null value, if pattern is not found. In case of multiple matches, it returns the matched result set.

© Aptech Ltd. Operators and Statements / Session 13 23

**HTML 5 RegExp Methods and Properties**

- RegExp object also supports properties that are used to get information regarding the string.
- Following table lists the properties of the RegExp object:

Audio Attributes	Description
\$n	Represents the number from 1 to 9. It stores the recently handled parts of a parenthesized pattern of a regular expression.
aif	Indicates whether the given regular expression contain a g flag. The g flag specifies that all the occurrences of a pattern will be searched globally, instead of just searching for the first occurrence.
aifc	Indicates whether the given regular expression contains an i flag.
aiff	Stores the location of the starting character of the last match found in the string. In case of no match, the value of the property is -1.
asc	Stores the copy of the pattern.

© Aptech Ltd. Operators and Statements / Session 13 24

**HTML 5 RegExp Methods and Properties**

- The Code Snippet displays the use of a regular expression.

```
<SCRIPT>
var zipcodepattern = /\d{5}$/;
var zipcode = zipcodepattern.exec(prompt('Enter ZIP Code:'));
if(zipcode != null)
{
    alert('Valid ZIP Code.');
    alert('Regular Expression Pattern: ' + zipcodepattern.source);
}
else
{
    alert('Invalid ZIP Code - Format xxxxx.');
}
</SCRIPT>
```

© Aptech Ltd. Operators and Statements / Session 13 25

Using slides 23 to 25, explain the RegExp methods and properties.

Mention, `RegExp` object supports methods that are used for searching the pattern in a string. These methods are as follows:

- `test (string)` – Tests a string for matching a pattern and returns a boolean value of `true` or `false`. The boolean value indicates whether the pattern exists in the string. This method is commonly used for validation.
- `exec (string)` – Executes a string to search the matching pattern within it. The method returns a `null` value, if pattern is not found. In case of multiple matches, it returns the matched result set.

Using slide 24, explain the properties of the `RegExp` method. Apart from these, `RegExp` object also supports properties that are used to get information regarding the string.

Explain table which lists the properties of the `RegExp` object.

Using slide 25, explain the code snippet which demonstrates the use of regular expression.

The script verifies whether the entered zip code is valid or not using a regular expression. In the code, `zipcodepattern` variable stores a pattern that states that the search string must contain only 5 digits. The zip code is accepted from the user and the pattern is matched with the code using the `exec ()` method. The `if` condition checks whether the entered zip code is not null. If it is not null, the Valid ZIP Code message and the pattern is displayed or else the Invalid ZIP Code – Format xxxx message is displayed.

## Slide 26

Let us understand categories of pattern matching.

The slide has a header 'HTML5 Categories of Pattern Matching'. The content includes a bulleted list: 'Different categories of pattern matching character that are required to create a regular expression pattern are as follows:' followed by a secondary list: 'Position Matching', 'Character Classes', 'Repetition', 'Alternation and Grouping', and 'Back Reference'.

At the bottom, there is footer text: '© Aptech Ltd.', 'Operators and Statements / Session 13', and '26'.

Using slide 26, explain the categories in pattern matching.

There are different categories of pattern matching character that are required to create a regular expression pattern. Mention the categories displayed on the slide.

## Slide 27

Let us understand the position matching.

**Position Matching**

Characters or symbols in this category allow matching a substring that exists at a specific position within a string.

- Following table lists the various position matching symbols.

Symbol	Description	Example
<code>^</code>	Denotes the start of a string	<code>/Good/</code> matches "Good" in "Good night", but not in "A Good Eyesight"
<code>\$</code>	Denotes the end of a string	<code>/art\$/</code> matches "art" in "Cart" but not in "artist"
<code>\b</code>	Matches a word boundary. A word boundary includes the position between a word and the space	<code>/ry\b/</code> matches "ry" in "She is very good"
<code>\B</code>	Matches a non-word boundary	<code>\B\an/</code> matches "an" in "operand" but not in "anomaly"

© Aptech Ltd. Operators and Statements / Session 13 27

Using slide 27, explain the position matching concept.

Mention, characters or symbols in this category allow matching a substring that exists at a specific position within a string. Explain list of various position matching symbols.

## Slides 28 and 29

Let us understand character classes.

**Character Classes 1-2**

Characters or symbols in this category are combined to form character classes for specifying patterns.

These classes are formed by placing a set of characters within the square brackets.

- Following table lists the various character classes symbols.

Symbol	Description	Example
<code>[xyz]</code>	Matches one of the characters specified within the character set	<code>^Good/</code> matches "Good" in "Good night", but not in "A Good Eyesight"
<code>[^xyz]</code>	Matches one of the characters not specified within the character set	<code>/[^BC]RT/</code> Matches "RRT" but, not "BRT" or "CRT"
<code>.</code>	Denotes a character except for the new line and line terminator	<code>/s.t/</code> Matches "sat", "sit", "set", and so on
<code>\w</code>	Matches alphabets and digits along with the underscore	<code>\w/</code> Matches "600" in "600%"

© Aptech Ltd. Operators and Statements / Session 13 28

The slide title is 'Character Classes 2-2'. A bullet point states: 'Following table lists some more character classes symbols.' The table has three columns: 'Symbol', 'Description', and 'Example'.

Symbol	Description	Example
\W	Matches a non-word character	/\W/ Matches "%" in "800%"
\d	Matches a digit between 0 to 9	/\d/ Matches "4" in "A4"
\D	Searches for a non-digit	/\D/ Matches "ID" in "ID 2248"
\s	Searches any single space character including space, tab, form feed, and line feed	/\s/w/ Matches " bar" in "scroll bar"
\S	Searches a non-space character	/\S/w/ Matches "scroll" in "scroll bar"

© Aptech Ltd. Operators and Statements / Session 13 29

Using slides 28 and 29, explain the character classes in JavaScript.

Mention, characters or symbols in this category are combined to form character classes for specifying patterns. These classes are formed by placing a set of characters within the square brackets. For example, the / [ a-zA-Z0-9 ] / pattern matches all alphabets and digits.

Explain, table which lists the various character classes symbols.

### Slide 30

Let us understand repetition.

The slide title is 'Repetition'. A callout box says: 'Characters or symbols in this category allow matching characters that reappear frequently in a string.' A bullet point states: 'Following table lists the various repetition matching symbols.'

Symbol	Description	Example
{x}	Matches x number of occurrences of a regular expression	/\d{6}/ Matches exactly 6 digits"
{x,}	Matches either x or additional number of occurrences of a regular expression	/\s{4,}/ Matches minimum 4 whitespace characters
{x,y}	Matches minimum x to maximum y occurrences of a regular expression	/\d{6,8}/ Matches minimum 6 to maximum 8 digits
?	Matches minimum zero to maximum one occurrences of a regular expression	/\s?m/ Matches "Im" or "I m"
*	Matches minimum zero to multiple occurrences of a regular expression	/\s*/ Matches "i" in "Ice" and "imm" in "immaculate", but nothing in "good"

© Aptech Ltd. Operators and Statements / Session 13 30

Using slide 30, explain the repetition.

Characters or symbols in this category allow matching characters that reappear frequently in a string.

Explain table which lists the various repetition matching symbols.

## Slide 31

Let us understand alternation and grouping.

Symbol	Description	Example
( )	Organizes characters together in a group to specify a set of characters in a string	<code>/(xyz)+(uvw)/</code> Matches one or more number of occurrences of "xyz" followed by one occurrence of "uvw"
	Combines sets of characters into a single regular expression and then matches any of the character set	<code>/(xy) (uv) (st)/</code> Matches "xy" or "uv" or "st"

© Aptech Ltd. Operators and Statements / Session 13 31

Using slide 31, explain the concept of alteration and grouping.

Characters or symbols in this category allow grouping characters as an individual entity or adding the 'OR' logic for pattern matching.

Explain table which lists the various alternation and grouping character symbols.

## Slide 32

Let us understand back reference.

Symbol	Description	Example
(\n)	Matches a parenthesized set within the pattern, where n is the number of the parenthesized set to the left	<code>/(w+)\s+\1/</code> Matches any word occurring twice in a line, such as "hello hello". The \1 specifies that the word following the space should match the string, which already matched the pattern in the parentheses to the left of the pattern. To refer to more than one set of parentheses in the pattern, you would use \2 or \3 to match the appropriate parenthesized clauses to the left. You can have maximum 9 back references in the pattern

© Aptech Ltd. Operators and Statements / Session 13 32

Using slide 32, explain the back references.

Characters or symbols in this category allow referring back to a sub-expression in the same regular expression. This is useful when matching the remaining sub-expression of a regular expression is based upon the result of matching the previous sub-expression.

Explain table which lists the back reference matching symbol.

Explain the code which uses the `replace()` method to switch the words in the string. For the replacement text, the script uses the `$1` and `$2` in the replacement to denote the first and second parenthesized substring matches.

```
var re = /(\w+) \s (\w+)/;
var str = "John Smith";
var newstr = str.replace(re, "$2, $1");
console.log(newstr);
```

The code prints "Smith, John".

### Slide 33

Let us understand decision-making statements.



**HTML5 Decision-making Statements**

- Statements are referred to as a logical collection of variables, operators, and keywords that perform a specific action to fulfill a required task.
- Statements help you build a logical flow of the script.
- In JavaScript, a statement ends with a semicolon.
- JavaScript is written with multiple statements, wherein the related statements are grouped together and referred to as block of code and are enclosed in curly braces.
- Decision-making statements allow implementing logical decisions for executing different blocks to obtain the desired output.
- They execute a block of statements depending upon a Boolean condition that returns either true or false.

© Aptech Ltd. Operators and Statements / Session 13 33

Using slide 33, explain the decision making statements in JavaScript.

Statements are referred to as a logical collection of variables, operators, and keywords that perform a specific action to fulfil a required task. For example, the line of code that declares a variable is a statement. Statements help you build a logical flow of the script. In JavaScript, a statement ends with a semicolon. JavaScript is written with multiple statements, wherein the related statements are grouped together. Such a group of statements is referred to as a block of code and the statements within it are enclosed in curly braces.

Mention, decision-making statements allow implementing logical decisions for executing different blocks to obtain the desired output. They execute a block of statements depending upon a Boolean condition. This condition is an expression that returns either true or false.

### Slide 34

Let us understand different types of decision making statements.

The screenshot shows a presentation slide with a blue header bar containing the text 'HTML 5 Decision-making Statements'. The main content area has a light blue background with a yellow-to-orange gradient at the bottom. It contains a bulleted list: 'JavaScript supports four decision-making statements, which are as follows:' followed by 'if', 'if-else', 'if-else if', and 'switch'. At the bottom of the slide, there is a footer bar with the text '@ Aptech Ltd.', 'Operators and Statements / Session 13', and '34'.

Using slide 34, explain the types of decision making statement supported in JavaScript.

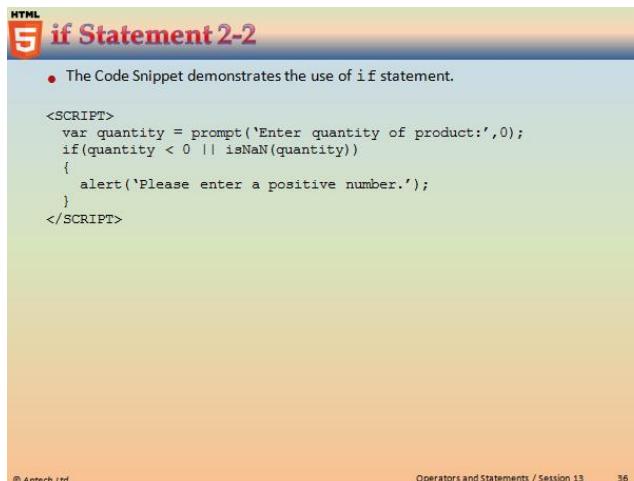
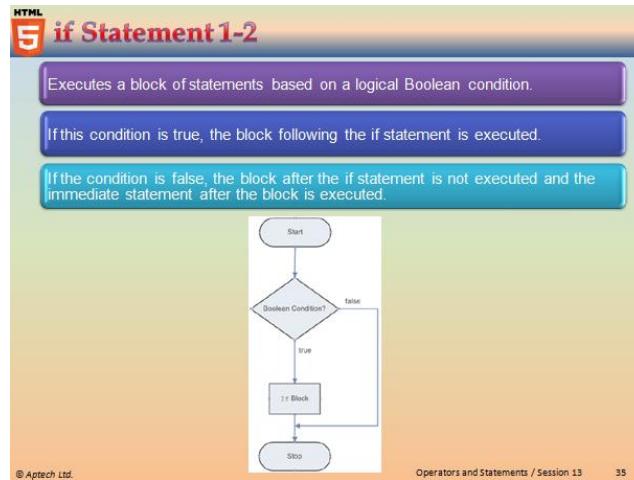
The decision making statements are used when there are branches in the required or when a block code is to be executed based on some condition.

JavaScript supports four decision-making statements which are as follows:

- if
- if-else
- if-else if
- switch

## Slides 35 and 36

Let us understand the if statement.



Using slides 35 and 36, explain the flow of if statement.

The if statement executes a block of statements based on a logical Boolean condition. If this condition is true, the block following the if statement is executed. If the condition is false, the block after the if statement is not executed and the immediate statement after the block is executed.

Figure shows the flow of execution for the 'if' statement.

The syntax to use the if statement is as follows:

### Syntax:

```
if (condition) {
// one or more statements;
}
```

where,

**condition:** Is the boolean expression.

**statements:** Consists of instructions to be executed when the boolean expression is true.

Using slide 36, explain the code snippet which demonstrates the use of if statement.

Code snippet checks whether the quantity value is a number and whether it is greater than 0 using the if statement.

The code accepts a quantity value from the user using the `prompt()` function and stores it in the quantity variable. The if statement is executed and the value of variable quantity is checked whether it is less than 0 and whether it is not a number. If value provided by the user is less than 0 or is a value other than a number, the condition evaluates to true and the output in `alert` box "Please enter a positive number" is displayed to the user.

## Slides 37 and 38

Let us understand the if-else statement.

if statement specifies a block of statement to be executed when the condition in the if statement is true.

Sometimes it is required to define a block of statements to be executed when a condition is evaluated to false.

if-else statement begins with the if block, which is followed by the else block.

The else block begins with the else keyword followed by a block of statements to be executed upon the false condition.

- The Code Snippet demonstrates the use of if-else statement.

```
<SCRIPT>
var firstNumber = prompt('Enter first number:',0);
var secondNumber = prompt('Enter second number',0);
var result = 0;
if (secondNumber == 0)
{
    alert('ERROR Message: Cannot divide by zero.');
}
else
{
    result = firstNumber/secondNumber;
    alert("Result: " + result);
}
</SCRIPT>
```

Using slides 37 and 38, explain the if-else statement.

The `if` statement specifies a block of statement to be executed when the condition in the `if` statement is `true`. However, sometimes it is required to define a block of statements to be executed when a condition is evaluated to `false`. This is done using the `if-else` statement.

The `if-else` statement begins with the `if` block, which is followed by the `else` block. The `else` block begins with the `else` keyword followed by a block of statements to be executed upon the false condition.

The syntax to use the `if-else` statement is as follows:

**Syntax:**

```
if (condition) {  
    // one or more statements;  
}  
else {  
    // one or more statements;  
}
```

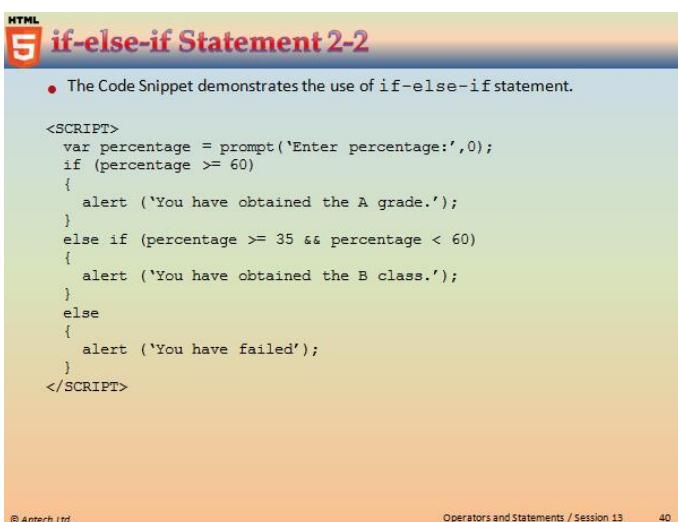
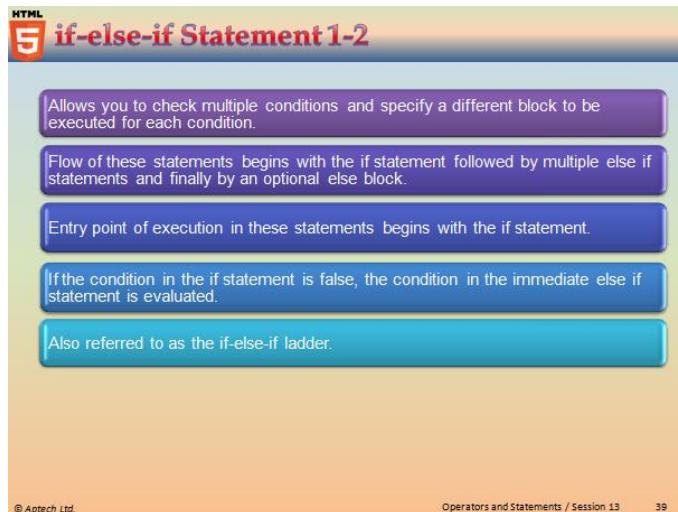
Using slide 38, explain the code snippet for the if-else statement.

Code Snippet performs the division operation and validates that the divisor is not equal to 0 using the `if-else` statement.

Mention, code accepts two numbers for the division operation and stores them in the variables `firstNumber` and `secondNumber` respectively. The `if` statement checks whether the value of the variable `secondNumber` is 0. If it is 0, the `alert()` function displays the error message to the user. If the value is not 0, the `else` block is executed, which performs the division operation. The quotient is stored in the `result` variable and is displayed to the user.

## Slides 39 and 40

Let us understand if-else-if statement.



Using slide 39, explain the if-else-if statement.

The if-else-if statements allow you to check multiple conditions and specify a different block to be executed for each condition. The flow of these statements begins with the if statement followed by multiple else if statements and finally by an optional else block. The entry point of execution in these statements begins with the if statement. If the condition in the if statement is false, the condition in the immediate else if statement is evaluated. The if-else-if statements are also referred to as the if-else-if ladder. The syntax to use the if-else-if statements are as follows:

### Syntax:

```
if (condition) {
// one or more statements;
```

```

} else if (condition) {

// one or more statements;

} else {

// one or more statements;

}

```

Using slide 40, explain the code snippet for `if-else-if` statement.

Code Snippet displays the grades according to the percentage value entered by the user using the `if-else-if` statements.

The code accepts the percentage value from the user and stores it in the `percentage` variable. The `if` statement checks whether the value of the `percentage` variable is greater than or equal to 60. If this is true, the user has obtained the A grade. If the condition is false, the execution control is passed to the `else if` block. Here, the value of the `percentage` variable is checked as to whether it is greater than or equal to 35 and less than 60. If this is true, the user has obtained the B grade. If the condition is false, the `else` block is executed.

## Slides 41 and 42

Let us understand nested if statement.

**Nested if Statement 1-2**

- Comprises of multiple if statements within an if statement.
- Flow of the nested-if statements starts with the if statement, which is referred to as the outer if statement.
- Outer if statement consists of multiple if statements, which are referred to as the inner if statements.
- Inner if statements are executed only if the condition in the outer if statement is true.
- Each of the inner if statements is executed but, only if the condition in its previous inner if statement is true.

© Aptech Ltd. Operators and Statements / Session 13 41

The slide has a blue header with the title 'Nested if Statement 2-2'. Below the title is a bulleted list: 'The Code Snippet demonstrates the use of nested if statement.' The main content is a block of JavaScript code:

```

<SCRIPT>
var username = prompt('Enter Username:');
var password = prompt('Enter Password:');
if (username != "" && password != "") {
    if (username == "admin" && password == "admin123")
    {
        alert('Login Successful');
    }
    else
    {
        alert ('Login Failed');
    }
}
</SCRIPT>

```

At the bottom of the slide, there is footer text: '© Aptech Ltd.', 'Operators and Statements / Session 13', and '42'.

Using slides 41 and 42, explain the nested-if statement.

The nested-if statements comprises multiple if statements within an if statement. The flow of the nested-if statements starts with the `if` statement, which is referred to as the outer `if` statement. This outer `if` statement consists of multiple `if` statements, which are referred to as the inner `if` statements.

The inner `if` statements are executed only if the condition in the outer `if` statement is true. Further, each of the inner `if` statements is executed, but only if the condition in its previous inner `if` statement is true.

The syntax to use the nested-if statements.

#### Syntax:

```

if (condition) {
    // one or more statements;

    if (condition) {
        // one or more statements;

        if (condition) {
            // one or more statements;
        }
    }
}

```

Using slide 42, explain the code snippet for nested-if statement. It validates the username and password using the nested-if statements.

The code accepts the username and password and stores them in the `username` and `password` variables. The `if` statement checks whether the values of both the variables are not empty. If they are not empty, the inner `if` statement is executed. The inner `if` statement checks whether the value of the `username` variable is `admin` and the value of the `password` variable is `admin123`. If this condition is `true`, the `Login Successful` message is displayed to the user. If the condition is `false`, the `else` block is executed.

### Slides 43 and 44

Let us understand switch-case statement.

**HTML 5 switch-case Statement 1-2**

- A program becomes quite difficult to understand when there are multiple if statements.
- To simplify coding and to avoid using multiple if statements, switch-case statement can be used.
- switch-case statement allows comparing a variable or expression with multiple values.

© Aptech Ltd. Operators and Statements / Session 13 43

**HTML 5 switch-case Statement 2-2**

- The Code Snippet demonstrates the use of `switch-case` statement.

```
<SCRIPT>
var designation = prompt('Enter designation:');
switch (designation)
{
    case 'Manager':
        alert ('Salary: $21000');
        break;
    case 'Developer':
        alert ('Salary: $16000');
        break;
    default:
        alert ('Enter proper designation.');
        break;
}
</SCRIPT>
```

© Aptech Ltd. Operators and Statements / Session 13 44

Using slides 43 and 44, explain the switch-case statement.

A program becomes quite difficult to understand when there are multiple if statements. To simplify coding and to avoid using multiple if statements, switch-case statement can be used as a different approach to code the same logic. The switch-case statement allows comparing a variable or expression with multiple values.

The syntax to use the switch-case statement is as follows:

**Syntax:**

```
switch(expression/variable) {  
    case value1:          // statements;  
    break;  
  
    case value2:          // statements;  
    break;  
  
    . . . case valueN:  
    // statements;  
  
    break;  
  
    default:              // default statement  
}  
where,
```

**switch:** Executes a specific case statement that holds the value of the expression or the variable.

**case:** A value and a colon follow the case keyword. The block of a specific case statement is executed when the value of switch expression and the case value are the same. Each case block must end with the break keyword.

**break:** Passes the execution control to the statement existing immediately out of the switch-case statement. If there is no break statement, the next case statement is executed.

**default:** The execution control passes to the default block when none of the case values matches with the switch expression. The default block is the same as the else block of the if-else-if statements.

Using slide 44, explain the code snippet for switch-case statement.

Code snippet displays the salary of an employee according to the designation by using the switch-case statement.

The code uses the variable designation to store the designation of an employee, which is accepted from the user. The switch statement takes the value of the designation variable and this value is matched with the different case statements. If the value matches, the particular case block is executed, which displays the respective salary. If none of the case values matches with the switch variable, the default block is executed.

### In-Class Question:

After you finish explaining switch-case statement, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What is the use of break statement in the switch-case block?

### Answer:

Break statement passes the execution control to the statement existing immediately out of the switch-case statement. If there is no break statement, the next case statement will be executed.

### Slide 45

Let us summarize the session.

**HTML 5 Summary**

- An operator specifies the type of operation to be performed on the values of variables and expressions.
- JavaScript operators are classified into six categories based on the type of action they perform on operands.
- There are six categories of operators namely, Arithmetic, Relational, Logical, Assignment, Bitwise, and Special operators.
- Operators in JavaScript have certain priority levels based on which their execution sequence is determined.
- A regular expression is a pattern that is composed of set of strings, which is to be matched to a particular textual content.
- In JavaScript, there are two ways to create regular expressions namely, literal syntax and RegExp() constructor.
- Decision-making statements allow implementing logical decisions for executing different blocks to obtain the desired output.

© Aptech Ltd.      Operators and Statements / Session 13      45

In slide 45, you will summarize the session. You will end the session, with a brief summary of what has been taught in the session.

### 13.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session. You should also explore the Loops and Arrays that are offered with the next session.

**Tips:**

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 14 – Loops and Arrays

---

## 14.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 14.1.1 Objectives

By the end of this session, the learners will be able to:

- Explain while loop
- Explain for loop
- Explain do..while loop
- Explain break and continue statement
- Explain single-dimensional arrays
- Explain multi-dimensional arrays
- Explain for..in loop

### 14.1.2 Teaching Skills

To teach this session, you should be well-versed with concept of loops and arrays in JavaScript. Along with this, you should prepare yourself how to store collection of values using arrays and for...in loop which is an extension of for loop.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

#### In-Class Activities:

Follow the order given here during In-Class activities.

### Overview of the Session:

Give the students the overview of the current session in the form of session objectives.

Show the students slide 2 of the presentation.

**HTML5 Objectives**

- Explain while loop
- Explain for loop
- Explain do..while loop
- Explain break and continue statement
- Explain single-dimensional arrays
- Explain multi-dimensional arrays
- Explain for..in loop

© Aptech Ltd. Loops and Arrays / Session 14 2

Tell the students that this session introduces them to loops and arrays. They will learn about different types of loops and how to store collection of values using arrays. They will also know about for...in loop which is an extension of for loop.

## 14.2 In-Class Explanations

### Slide 3

Let us understand introduction to loops in JavaScript.

**HTML5 Introduction**

- Loops allow you to execute a single statement or a block of statements multiple times.
- They are widely used when you want to display a series of numbers and accept repetitive input.
- A loop construct consists of a condition that instructs the compiler the number of times a specific block of code will be executed.
- If the condition is not specified within the construct, the loop continues infinitely. Such loop constructs are referred to as infinite loops.
- JavaScript supports three types of loops that are as follows:
  - while Loop
  - for Loop
  - do-while Loop

© Aptech Ltd. Loops and Arrays / Session 14 3

Using slide 3, explain introduction to loops.

Consider a scenario where you want to accept and display ten numbers to the user. Instead of writing the same lines of code again and again for 10 times, you can use loops. Loops allow you to execute a single statement or a block of statements multiple times. They can be used when you want to display a series of numbers and accept repetitive input.

Loop helps you to execute a particular block for a specified number of times.

Explain the types of loops supported in JavaScript.

## Slides 4 to 7

Let us understand `while` loop.

**HTML 5 while Loop 1-4**

- The `while` loop executes a block of code as long as the given condition remains `true`.
- The `while` loop begins with the `while` keyword, which is followed by parentheses containing a boolean condition.
- If this condition returns `true`, the block of statements within the `while` loop are executed.
- Once the condition becomes `false`, the `while` statement stops the execution of loop and transfers the control to next statement appearing after the block.

- Following figure shows the flow of execution of the `while` loop.

```

graph TD
    A{Condition Expression?} -- true --> B[Execute Body of Loop]
    B --> A
    A -- false --> C[Exit Loop]
  
```

© Aptech Ltd. Loops and Arrays / Session 14 4

**HTML 5 while Loop 2-4**

- The syntax for the `while` loop is as follows:

**Syntax:**

```
while (condition)
{
  // statements;
}
```

where,

- condition: Is a boolean expression.

- The Code Snippet displays the sum of numbers from 1 to 10 by using the `while` loop.

```
<script>
var i = 0;
var sum = 0;
```

© Aptech Ltd. Loops and Arrays / Session 14 5

**HTML 5 while Loop 3-4**

```
while(i<=10)
{
sum = sum + i;
i = i + 1;
}
alert('Sum of first 10 numbers: ' + sum);
</script>
```

- The code declares two variables, `i` and `sum`, which are initialized to value 0.
- The variable, `i`, is a counter variable, whose value increases for every execution of loop.
- The condition in the while loop checks that the value of the counter variable, `i`, is less than or equal to 10.
- If this condition is true, the value of the `sum` variable is added to the value of `i` variable.
- The value of the variable `i` is incremented by 1.
- Then, the program control is passed to the while statement to check the condition again.
- When the value of `i` becomes 11, the while loop terminates as the loop condition becomes false.

© Aptech Ltd. Loops and Arrays / Session 14 6

**HTML 5 while Loop 4-4**

- Following figure shows the output.

© Aptech Ltd. Loops and Arrays / Session 14 7

Using slides 4 to 7, explain the `while` loop in details.

Mention, `while` loop executes a block of code as long as the given condition remains true. The `while` loop begins with the `while` keyword, which is followed by parentheses containing a boolean condition. If this condition returns `true`, the block of statements within the `while` loop are executed. After every iteration, the program control is transferred back to the `while` statement, where the condition is again checked for another round of execution. This process is continued till the specified condition becomes `false`. Once the condition becomes `false`, the `while` statement stops the execution of loop and transfers the control to next statement appearing after the block.

Figure shows the flow of execution - `while` loop. Explain the syntax of `while` loop and the code snippet which uses the `while` loop.

Mention, code snippet displays the sum of numbers from 1 to 10 by using the `while` loop.

The code declares two variables, `i` and `sum`, which are initialized to value 0. The variable, `i`, is a counter variable, whose value increases for every execution of loop. The condition in the `while` loop checks that the value of the counter variable, `i`, is less than or equal to 10. If this condition is `true`, the value of the `sum` variable is added to the value of `i` variable. The value of the variable `i` is incremented by 1. Then, the program control is passed to the `while` statement to check the condition again. When the value of `i` becomes 11, the `while` loop terminates as the loop condition becomes `false`.

## Slides 8 to 11

Let us understand the `for` loop.

**HTML 5 for Loop 1-4**

The `for` loop is similar to the `while` loop as it executes the statements within the loop as long as the given condition is `true`.

Unlike the `while` loop, the `for` loop specifies the loop control statements at the top instead in the body of the loop.

The `for` loop begins with the `for` keyword, which is followed by parentheses containing three expressions, each of which are separated by a semicolon.

The three expressions are referred to as **initialization expression**, **condition expression**, and **increment/decrement expression** respectively.

- The syntax for the `for` loop is as follows:

**Syntax:**

```
for (initialization; condition; increment/decrement)
{
    // statements;
}
```

© Aptech Ltd. Loops and Arrays / Session 14 8

**HTML 5 for Loop 2-4**

where,

- initialization:** Initializes the variable(s) that will be used in the condition.
- condition:** Comprises the condition that is checked before the statements in the loop are executed.
- increment/decrement:** Comprises the statement that changes the value of the variable(s) on every successful execution of the loop to ensure that the condition specified in the condition section is reached.

- Following figure shows the `for` loop.

<b>Simple for Loop</b>	<b>for Loop Without Expression 1</b>	<b>for Loop Without Expression 2</b>
<pre>for (initialization; condition; increment/decrement) {     //statements; }</pre>	<pre>for ( ; condition; increment/decrement) {     //statements; }</pre>	<pre>for (initialization; ; increment/decrement) {     //statements; }</pre>
<b>for Loop Without Expression 3</b>	<b>for Loop Without Expressions</b>	
<pre>for (initialization; condition; ) {     //statements; }</pre>	<pre>for ( ; ) {     //statements; }</pre>	

© Aptech Ltd. Loops and Arrays / Session 14 9

**HTML 5 for Loop 3-4**

- The Code Snippet demonstrates the script that accepts a number from the user and displays the first ten multiples of that number.

```
<script>
var inputNum = prompt('Enter any number:');
var result = 0;
document.write ('Multiples of: ' + inputNum + '<br />');
for (var i=1; i<=10; i++)
{
    result = inputNum * i ;
    document.write (inputNum + ' * ' + i + ' = ' +
    result + '<br />');
}
</script>
```

- In the code, a variable, `inputNum`, is created and initialized to the value specified by the user in the prompt box.
- The `for` loop declares a variable, `i`, and initializes it to the value 1.
- If the condition is true, the number specified by the user is multiplied to the value of `i` variable and the result is appended to the `result` variable.

© Aptech Ltd. Loops and Arrays / Session 14 10

**HTML 5 for Loop 4-4**

- The program control is again passed to `for` statement, where the value of `i` is incremented.
- The incremented value is again checked with the specified condition and it is multiplied to the number specified by the user.
- This process continues till the value of `i` becomes 11.
- Following figure shows the multiples of a number.

Multiples of: 4  
4 \* 1 = 4  
4 \* 2 = 8  
4 \* 3 = 12  
4 \* 4 = 16  
4 \* 5 = 20  
4 \* 6 = 24  
4 \* 7 = 28  
4 \* 8 = 32  
4 \* 9 = 36  
4 \* 10 = 40

© Aptech Ltd. Loops and Arrays / Session 14 11

Using slides 8 to 11, explain `for` loop.

Mention that the `for` loop is similar to the `while` loop in functionality. It executes the statements within the loop as long as the given condition is `true`. Unlike the `while` loop, the `for` loop specifies the loop control statements at the top instead in the body of the loop.

The `for` loop begins with the `for` keyword, which is followed by parentheses containing three expressions, each of which are separated by a semicolon. The three expressions are referred to as **initialization expression**, **condition expression**, and **increment/decrement expression** respectively. These three expressions are optional.

The syntax for the for loop is as follows:

**Syntax:**

```
for (initialization; condition; increment/decrement)
{
    // statements;
}
```

where,

**initialization**: Initializes the variable(s) that will be used in the condition.

**condition**: Comprises the condition that is checked, before the statements in the loop are executed.

**increment/decrement**: Comprises the statement that changes the value of the variable(s) on every successful execution of the loop to ensure that the condition specified in the condition section is reached. The increment and decrement operators, such as `++`, `--`, and shortcut operators: `+=` or `-=` are used in this section.

Explain code snippet in slide 10 which demonstrates the script that accepts a number from the user and displays the first ten multiples of that number.

In the code, a variable, `inputNum`, is created and initialized to the value specified by the user in the `prompt` box. The `for` loop declares a variable, `i`, and initializes it to the value 1. If the condition is `true`, the number specified by the user is multiplied to the value of `i` variable and the result is appended to the `result` variable. The program control is again passed to `for` statement, where the value of `i` is incremented. The incremented value is again checked with the specified condition and it is multiplied to the number specified by the user. This process continues till the value of `i` becomes 11.

Figure shows the multiples of a number.

Then explain the variants of for loop.

Explain the following example which does not contain initialization condition:

```
var i = 2;
var len = cars.length;
var text = "";
for (; i < len; i++) {
    text += cars[i] + "<br>";
}
```

**In-Class Question:**

After you finish explaining `for` loop, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What are the expressions of the `for` loop?

**Answer:**

Initialization, Condition, and Increment/Decrement.

**Slides 12 to 15**

Let us understand the `do-while` loop.

**HTML do-while Loop 1-4**

The do-while loop is similar to the while loop. This is because both the do-while and while loops execute until the condition becomes false.

However, the do-while loop differs by executing the body of the loop at least once before evaluating the condition even if the condition is false.

The do-while loop starts with the `do` keyword and is followed by a block of statements.

At the end of the block, the `while` keyword is specified that is followed by parentheses containing the condition.

When the condition returns false, the block of statements after the `do` keyword are ignored and the next statement following the `while` statement is executed.

© Aptech Ltd. Loops and Arrays / Session 14 12

**HTML do-while Loop 2-4**

- Following figure shows the do-while loop.

```

graph TD
    do[do] --> body[Execute Body of Loop]
    body --> cond{Condition Expression?}
    cond -- true --> body
    cond -- false --> exit[Exit Loop]
  
```

© Aptech Ltd. Loops and Arrays / Session 14 13

**HTML 5 do-while Loop 3-4**

- The syntax for the do-while loop is as follows:

**Syntax:**

```
do
{
...
statements;
...
}while(condition);
where,
• condition: Is a boolean expression.
```

- The Code Snippet demonstrates the script to accept the capital of United States from the user using the do-while loop.

```
<script>
    var answer = '';
    do
    {
        answer = prompt('Capital of United States:', '');
    }while(answer != 'Washington');
```

© Aptech Ltd. Loops and Arrays / Session 14 14

**HTML 5 do-while Loop 4-4**

```
alert('Capital of United States: ' + answer);
</script>
```

- The code declares a variable, answer, which stores the string entered by the user.
- The do block displays a prompt box without checking any condition.
- The prompt box accepts the capital of United States and stores this string in the variable, answer.
- The condition is specified in the while block that checks if the user has entered the string Washington.
- If this condition is true, prompt box is closed; else the prompt box is again displayed to accept the user input.
- Following figure shows the output of capital of United States.



© Aptech Ltd. Loops and Arrays / Session 14 15

Using slides 12 to 15, explain the do-while loop.

Mention do-while loop is similar to the while loop. This is because both the do-while and while loops execute until the condition becomes false. However, the do-while loop differs by executing the body of the loop at least once before evaluating the condition. Thus, even if the condition is false, the do-while loop executes at least once.

The do-while loop starts with the do keyword and is followed by a block of statements. At the end of the block, the while keyword is specified that is followed parentheses containing the condition. When the specified condition returns false, the block of statements after the do keyword are ignored and the next statement following the while statement is executed.

Figure shows the do-while loop.

The syntax for the `do-while` loop is as follows:

**Syntax:**

```
do {  
    ...  
    statements;  
    ...  
} while (condition);
```

where,

`condition`: Is a boolean expression.

Explain code snippet which demonstrates the script to accept the capital of United States from the user using the `do-while` loop.

The code declares a variable, `answer`, which stores the string entered by the user. The `do` block displays a `prompt` box without checking any condition. The `prompt` box accepts the capital of United States and stores this string in the variable, `answer`. The condition is specified in the `while` block that checks, if the user has entered the string `Washington`. If this condition is `true`, `prompt` box is closed; else the `prompt` box is again displayed to accept the user input.

Figure shows the output of capital of United States.

**In-Class Question:**

After you finish explaining `do-while` loop, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What is difference in `do-while` loop as compared to `while` loop?

**Answer:**

The `do-while` loop differs by executing the body of the loop at least once before evaluating the condition.

## Slides 16 to 18

Let us understand the `break` statement.

**HTML 5 break Statement 1-3**

The `break` statement can be used with decision-making such as switch-case and loop constructs such as `for` and `while` loops.

The `break` statement is denoted by using the `break` keyword. It is used to exit the loop without evaluating the specified condition.

The control is then passed to the next statement immediately after the loop.

- Following figure shows the flow of execution of the `break` statement.

```

for (initialization; condition; increment/decrement)
{
    ...
    if (true condition)
        break;
    ...
}
  
```

© Aptech Ltd. Loops and Arrays / Session 14 16

**HTML 5 break Statement 2-3**

- The Code Snippet demonstrates the script that accepts a number from the user and determines if it is a prime number or not.

```

<script>
var inputNum = parseInt(prompt('Enter number: ',''));
var num = 2;
while(num <= inputNum-1)
{
    if(inputNum % num == 0)
    {
        alert(inputNum + ' is not a Prime Number');
        break;
    }
    num++;
}
if(num == inputNum)
{
    alert(inputNum + ' is a Prime Number');
}
</script>
  
```

© Aptech Ltd. Loops and Arrays / Session 14 17

**HTML** **break Statement 3-3**

- The code creates a variable, `inputNum`, which is initialized to the number entered by the user.
- The variable `num` is declared and initialized to 2.
- If the while condition returns true, the inner if statement is checked.
- If this condition returns true, an alert box is displayed stating that the number is not a prime number.
- The `break` statement is used to exit the entire while loop.
- If the condition evaluates to false, the program control is passed to if statement outside the while loop.
- Following figure shows the output of the prime number on accepting number, 6 from the user in the prompt box.

**JavaScript Alert**

6 is not a Prime Number

OK

© Aptech Ltd. Loops and Arrays / Session 14 18

Using slides 16 to 18, explain the `break` statement.

Mention, `break` statement can be used with decision-making statements, such as `switch-case` and loop constructs, such as `for` and `while` loops. It is denoted by using the `break` keyword. It is used to exit the loop without evaluating the specified condition. The control is then passed to the next statement immediately after the loop.

Figure shows the flow of execution of the `break` statement.

Explain code snippet which demonstrates the script that accepts a number from the user and determines if it is a prime number or not. The code creates a variable, `inputNum`, which is initialized to the number entered by the user.

The variable `num` is declared and initialized to 2. If the `while` condition returns `true`, the inner `if` statement is checked. If this condition returns `true`, an alert box is displayed stating that the number is not a prime number. The `break` statement is used to exit the entire `while` loop. If the condition evaluates to `false`, the program control is passed to `if` statement outside the `while` loop.

Figure shows the output of the prime number on accepting number, 6 from the user in the prompt box.

#### Tips:

JavaScript statements can be labeled. Normally, a `break` statement is used inside a loop or `switch` to break the control from the loop or decision statement. However, you can provide a label reference that can be used to jump to any code statement.

Mention label in JavaScript. To label JavaScript statements you precede the statements with a label name and a colon.

Syntax for the label is:

label:

statements;

And using it to come out of loop or a block and to go to a specific block of code is as follows:

`break labelname;`

## Slides 19 to 21

Let us understand continue statement.

**HTML 5 continue Statement 1-3**

The `continue` statement is mostly used in the loop constructs and is denoted by the `continue` keyword.

It is used to terminate the current execution of the loop and continue with the next repetition by returning the control to the beginning of the loop.

This means, the `continue` statement will not terminate the loop entirely, but terminates the current execution.

- Following figure shows the flow of execution of the `continue` statement.

```

for (initialization; condition; increment/decrement)
{
    ...
    If (true condition)
        continue;
    ...
}
  
```

© Aptech Ltd. Loops and Arrays / Session 14 19

**HTML 5 continue Statement 2-3**

- The Code Snippet displays even numbers from 0 to 15.

```

<script>
    var result = '';
    for (var i = 0; i <= 15; i++)
    {
        if((i%2) != 0)
        {
            continue;
        }
        result = result + i + '\n';
    }
    alert('Even Numbers:\n' + result);
</script>
  
```

- The code declares a variable, `i`, in the for loop definition and initializes it to value 1.
- When the value of `i` is divided by zero, the `if` statement checks whether the remainder is equal to zero.

© Aptech Ltd. Loops and Arrays / Session 14 20

**HTML 5 continue Statement 3-3**

- If the remainder is zero, the value of `i` is displayed as the value is an even number.
- If the remainder is not equal to 0, the `continue` statement is executed.
- It transfers the program control to the beginning of the `for` loop.
- Following figure shows the output of the `continue` statement.

**JavaScript Alert**

Even Numbers:

```
0
2
4
6
8
10
12
14
```

OK

© Aptech Ltd. Loops and Arrays / Session 14 21

Using slides 19 to 20, explain `continue` statement.

Mention, `continue` statement is mostly used in the loop constructs. The `continue` statement is denoted by the `continue` keyword. It is used to terminate the current execution of the loop and continue with the next repetition by returning the control to the beginning of the loop. This means, the `continue` statement will not terminate the loop entirely, but terminates the current execution.

Figure shows the flow of execution of the `continue` statement.

Explain code snippet which displays even numbers from 0 to 15.

The code declares a variable, `i`, in the `for` loop definition and initializes it to value 1. When the value of `i` is divided by zero, the if statement checks whether the remainder is equal to zero. If the remainder is zero, the value of `i` is displayed as the value is an even number. If the remainder is not equal to 0, the `continue` statement is executed. It transfers the program control to the beginning of the `for` loop.

Figure shows the output of the `continue` statement.

#### Tips:

For skipping iteration in a loop `continue` statement can be used.

Also, `label` can be used with `continue` statement based on the requirements like `break` statement.

## Slide 22

Let us understand arrays in JavaScript.

An array is a collection of values stored in adjacent memory locations.

These array values are referenced using a common array name. The values of an array variable must be of the same data type.

These values that are also referred to as elements can be accessed by using subscript or index numbers.

The subscript number determines the position of an element in an array list.

- Following figure shows the effective usage of memory achieved using an array.

Array of 100 Names			
Steve	David	John	Steffen
...			

Proper Utilization of Memory

100 variables storing names of students	
var studentOne	Steve
var studentTwo	David
var studentThree	John
var studentFour	Steffen
...	...
... Till 100 variables	

Inefficient Memory Utilization

© Aptech Ltd. Loops and Arrays / Session 14 22

Using slide 22, explain arrays in JavaScript.

Consider a scenario where you want to store the names of 100 employees within an IT department. This can be done by creating 100 variables and storing the names. However, keeping track of 100 variables is a tedious task and it results in inefficient memory utilization. The solution to this problem is to create an array variable to store the names of 100 employees.

Explain the figure that shows the effective usage of memory achieved using an array.

An array is a collection of values stored in adjacent memory locations. These array values are referenced using a common array name. The values of an array variable must be of the same data type. These values that are also referred to as elements and can be accessed by using subscript or index numbers. The subscript number determines the position of an element in an array list.

### In-Class Question:

After you finish explaining arrays in HTML 5, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What is an array?

### Answer:

An array is a collection of values stored in adjacent memory locations.

## Slides 23 to 25

Let us understand single-dimensional array.

**HTML Single-dimensional Array 1-3**

- JavaScript supports two types of arrays that are as follows:
  - Single-dimensional array
  - Multi-dimensional array
- In a single-dimensional array, the elements are stored in a single row in the allocated memory.
- Following figure shows the allocation of single-dimensional array.

As shown in the figure, the first array element has the index number zero.  
The last array element has an index number one less than the total number of elements.  
This arrangement helps in efficient storage of data.  
It also helps to sort data easily and track the data length.

© Aptech Ltd. Loops and Arrays / Session 14 23

**HTML Single-dimensional Array 2-3**

- The array variable can be created using the `Array` object and `new` keyword along with the size of the array element.
- The syntax to declare and initialize a single-dimensional array is as follows:

**Syntax:**

```
var variable_name = new Array(size); //Declaration
variable_name[index] = 'value';
```

where,

- variable\_name: Is the name of the variable.
- size: Is the number of elements to be declared in the array.
- index: Is the index position.

© Aptech Ltd. Loops and Arrays / Session 14 24

**HTML 5 Single-dimensional Array 3-3**

- The Code Snippet shows the different ways to declare and initialize a single-dimensional array.

```
<script>

//Declaration using Array Object and then Initialization
var marital_status = new Array(3);
marital_status[0] = 'Single';
marital_status[1] = 'Married';
marital_status[2] = 'Divorced';

//Declaration and Initialization
var marital_status = new
    Array('Single','Married','Divorced');

//Declaration and Initialization Without Array
var marital_status = ['Single','Married','Divorced'];

</script>
```

© Aptech Ltd. Loops and Arrays / Session 14 25

Using slides 23 to 25, explain a single-dimensional array. In single-dimensional array, the elements are stored in a single row in the allocated memory. Explain the figure that shows the allocation of single-dimensional array.

As shown in figure, the first array element has the index number zero and the last array element has an index number one less than the total number of elements.

The array variable is created using the `Array` object and `new` keyword along with the size of the array element.

Explain the syntax and code snippet that shows how to declare and initialize a single-dimensional array.

## Slides 26 to 29

Let us understand how to access single-dimensional arrays.

**HTML 5 Accessing Single-dimensional Arrays 1-4**

- Array elements can be accessed by using the array name followed by the index number specified in square brackets.

**Accessing Array Elements Without Loops**

- An array element can be accessed without using loops by specifying the array name followed by the square brackets containing the index number.
- The Code Snippet demonstrates a script that stores and displays names of the students using a single-dimensional array.

```
<script>
    var names = new Array("John", "David", "Kevin");
    alert('List of Student Names:\n' + names[0] + ',' + '
        ' + names[1] + ',' + ' ' + names[2]);
</script>
```

- As shown in the code, `var names = new Array("John", "David", "Kevin");` declares and initializes an array.
- The `names[0]` accesses the first array element, which is John.
- The `names[1]` accesses the second array element, which is David.
- The `names[2]` accesses the third array element, which is Kevin.

© Aptech Ltd. Loops and Arrays / Session 14 26

**HTML 5 Accessing Single-dimensional Arrays 2-4**

- Following figure displays the names of the students.

The screenshot shows a JavaScript Alert dialog box with the title "JavaScript Alert". Inside the box, the text "List of Student Names:" is followed by a list: "John, David, Kevin". At the bottom right is an "OK" button.

**> Accessing Array Elements With Loops**

- JavaScript allows you to access array elements by using different loops.
- Thus, you can access each array element by putting a counter variable of the loop as the index of an element.
- However, this requires the count of the elements in an array.
- The length property can be used to determine the number of elements in an array.

© Aptech Ltd. Loops and Arrays / Session 14 27

**HTML 5 Accessing Single-dimensional Arrays 3-4**

- The Code Snippet demonstrates the script that creates an array to accept the marks of five subjects and display the average.

```
<script>
    var sum = 0;
    var marks = new Array(5);
    for(var i=0; i<marks.length; i++)
    {
        marks[i] = parseInt(prompt('Enter Marks:', ''));
        sum = sum + marks[i];
    }
    alert('Average of Marks: ' + (sum/marks.length));
</script>
```

- In the code, `var marks = new Array(5);` declares an array of size 5.
- It displays a prompt box that accepts the marks for a subject in each iteration.
- Then, the code calculates and displays the average marks.

© Aptech Ltd. Loops and Arrays / Session 14 28

**HTML 5 Accessing Single-dimensional Arrays 4-4**

- Following figure displays the average of the marks, 90, 75, 85, 95, and 82 accepted from the user in the prompt box.

The screenshot shows a JavaScript Alert dialog box with the title "JavaScript Alert". Inside the box, the text "Average of Marks: 85.4" is displayed. At the bottom right is an "OK" button.

© Aptech Ltd. Loops and Arrays / Session 14 29

Using slides 26 to 29, explain the process of accessing single-dimensional arrays.

Array elements can be accessed by using the array name followed by the index number specified in square brackets.

## Access Array Elements without Loops

Mention, an array element can be accessed without using loops by specifying the array name followed by the square brackets containing the index number. Code Snippet demonstrates a script that stores and displays names of the students using a single-dimensional array.

In the code, `var names = new Array("John", "David", "Kevin");` declares and initializes an array. The `names [0]` accesses the first array element which is John. The `names [1]` accesses the second array element which is David.

The `names [2]` accesses the third array element, which is Kevin.

Figure displays the names of the students.

## Access Array Elements with Loops

JavaScript allows to access array elements by using different loops. Thus, you can access each array element by putting a counter variable of the loop as the index of an element. However, this requires the count of the elements in an array. So, the length property can be used to determine the number of elements in an array. Code Snippet demonstrates the script that creates an array to accept the marks of five subjects and display the average.

In the code, `var marks = new Array(5);` declares an array of size 5. It displays a prompt box that accepts the marks for a subject in each iteration. Then, the code calculates and displays the average marks.

Figure displays the average of the marks, 90, 75, 85, 95, and 82 accepted from the user in the prompt box.

## Slides 30 and 31

Let us understand multi-dimensional arrays.

**HTML 5 Multi-dimensional Array 1-2**

- A multi-dimensional array stores a combination of values of a single type in two or more dimensions.
- These dimensions are represented as rows and columns similar to those of a Microsoft Excel sheet.
- A two-dimensional array is an example of the multi-dimensional array.
- Following figure shows the representation of a multi-dimensional array.

Employee Salaries	0 BASIC	1 HRA	2 ALLOWANCE	3 TOTAL
0	14350	10500	1500	26350
1	34350	4050	1000	39400
2	6150	4500	3250	13900
3	4920	4500	2250	11670
4	12300	9000	2000	23300

- A two-dimensional array is an array of arrays.
- This means, for a two-dimensional array, first a main array is declared and then, an array is created for each element of the main array.

© Aptech Ltd. Loops and Arrays / Session 14 30

**HTML 5 Multi-dimensional Array 2-2**

- The syntax to declare a two-dimensional array is as follows:

**Syntax:**

```
var variable_name = new Array(size); //Declaration
variable_name[index] = new Array('value1','value2'..);
```

where,

- variable\_name: Is the name of the array.
- index: Is the index position.
- value1: Is the value at the first column.
- value2: Is the value at the second column.

- Following figure shows the declaration of a two-dimensional array.

© Aptech Ltd. Loops and Arrays / Session 14 31

Using slides 30 and 31, explain the multi-dimensional arrays.

Consider a scenario to store the employee IDs of 100 employees and their salary structure. The salary structure will include the basic salary, allowances, HRA, and the total gross salary. Now, if a single-dimensional array is used, then two separate arrays need to be created for storing employee IDs and salaries. However, using a multi-dimensional array, both IDs and salaries are stored in just one array. A multi-dimensional array stores a combination of values of a single type in two or more dimensions. These dimensions are represented as rows and columns similar to those of a Microsoft Excel sheet. A two-dimensional array is an example of the multi-dimensional array. Figure 14.15 shows the representation of a multi-dimensional array.

A two-dimensional array is an array of arrays. This means, for a two-dimensional array, first a main array is declared and then, an array is created for each element of the main array.

Explain declaration and initialization of multi-dimensional array.

## Slides 32 to 35

Let us understand how to access two-dimensional arrays.

**HTML 5 Accessing Two-dimensional Arrays 1-4**

- Multi-dimensional arrays can be accessed by using the index of main array variable along with index of sub-array.

**➤ Accessing Array Elements Without Loops**

- The Code Snippet creates a two-dimensional array that displays the employee details.

```
<script>
    var employees = new Array(3);
    employees[0] = new Array('John', '25', 'New Jersey');
    employees[1] = new Array('David', '21', 'California');
    document.write('<H3> Employee Details </H3>');
    document.write('<P><B>Name: </B>' + employees[0][0] +
    '</P>');
    document.write('<P><B>Location: </B>' + employees[0][2]
    + '</P>');
    document.write('<P><B>Name: </B>' + employees[1][0]
    + '</P>');
    document.write('<P><B>Location: </B>' + employees[1][2]
    + '</P>');
</script>
```

© Aptech Ltd. Loops and Arrays / Session 14 32

**HTML 5 Accessing Two-dimensional Arrays 2-4**

- In the code, `var employees = new Array(3)` creates an array of size 3.
- The declaration `employees[0] = new Array('John', '23', 'New Jersey')` creates an array at the 0th row of the employees array.
- Similarly, `employees[1] = new Array('David', '21', 'California')` creates an array at the first row of the employees array.
- Following figure displays the employee details.

```
for (initialization; condition; increment/decrement)
{
    ...
    If (true condition)
        break;
    ...
}
```

© Aptech Ltd. Loops and Arrays / Session 14 33

**HTML** Accessing Two-dimensional Arrays 3-4

➤ Accessing Array Elements With Loops

- The Code Snippet creates a two-dimensional array to display the product details.

```
<script>
var products = new Array(2);
products[0] = new Array('Monitor', '236.75');
products[1] = new Array('Keyboard', '45.50');
document.write('<TABLE border=1><TR><TH>Name</TH><TH>Price</TH></TR>');
for(var i=0; i<products.length; i++)
{
    document.write('<TR>');
    for(var j=0; j<products[i].length; j++)
    {
        document.write('<TD>' + products[i][j] + '</TD>');
    }
    document.write('</TR>');
}
document.write('</TABLE>');
</script>
```

© Aptech Ltd. Loops and Arrays / Session 14 34

**HTML** Accessing Two-dimensional Arrays 4-4

- In the code, `products[0] = new Array('Monitor', '236.75')` creates an array at the 0th row of the products array.
- Similarly, `products[1] = new Array('Keyboard', '45.50')` creates an array at the first row of the products array.
- The condition, `i < products.length`, specifies that the counter variable, i, should be less than the number of rows in the array variable, products.
- For each row in the array, the condition, `j < products[i].length` specifies that the counter variable, j, should be less than the number of columns specified in the ith row of the array variable, products.
- Finally, `document.write("<TD>" + products[i][j] + "</TD>")` displays the values at the ith row and jth column of array variable, products.
- Following figure displays the product details in a table.

Name	Price
Monitor	236.75
Keyboard	45.50

© Aptech Ltd. Loops and Arrays / Session 14 35

Using slides 32 to 35, explain the process of accessing multi-dimensional arrays.

Mention, multi-dimensional arrays can be accessed by using the index of main array variable along with index of sub-array.

Code Snippet creates a two-dimensional array that displays the employee details.

In the code, `var employees = new Array(3)` creates an array of size 3. The declaration `employees[0] = new Array('John', '23', 'New Jersey')` creates an array at the 0th row of the employees array. Similarly, `employees[1] = new Array('David', '21', 'California')` creates an array at the first row of the employees array.

Figure displays the employee details.

Explain code snippet which creates a two-dimensional array to display the product details.

In the code, `products[0] = new Array('Monitor', '236.75')` creates an array at the 0th row of the products array. Similarly, `products[1] = new Array('Keyboard', '45.50')` creates an array at the first row of the products array. The condition, `i < products.length`, specifies that the counter variable `i` should be less than the number of rows in the array variable, `products`. For each row in the array, the condition, `j < products[i].length` specifies that the counter variable `j`, should be less than the number of columns specified the `i`th row of the array variable, `products`. Finally, `document.write("<TD>" + products[i][j] + "</TD>")` displays the values at the `i`th row and `j`th column of array variable, `products`.

Figure displays the product details in a table.

## Slides 36 to 38

Let us understand the array methods.

**HTML 5 Array Methods 1-3**

- An array is a set of values grouped together and identified by a single name. In JavaScript, the `Array` object allows you to create arrays.
- It provides the `length` property that allows you to determine the number of elements in an array.
- The various methods of the `Array` object allow to access and manipulate the array elements.

- Following table lists the most commonly used methods of the object.

Data Type	Description
<code>concat</code>	Combines one or more array variables.
<code>join</code>	Joins all the array elements into a string.
<code>pop</code>	Retrieves the last element of an array.
<code>push</code>	Appends one or more elements to the end of an array.
<code>sort</code>	Sorts the array elements in an alphabetical order.

© Aptech Ltd. Loops and Arrays / Session 14 36

**HTML 5 Array Methods 2-3**

- The Code Snippet demonstrates how to access and manipulate the array elements.

```
<script>
    var flowers = new Array('Rose', 'Sunflower', 'Daisy');
    document.write('Number of flowers: ' + flowers.length +
    '<br/>');
    document.write('Flowers: ' + flowers.join(', ') +
    '<br/>');
    document.write('Orchid and Lily are Added: ' +
    flowers.push("Orchid", "Lily") + '<br/>');
    document.write('Flowers (In Ascending Order): ' +
    flowers.sort() + '<br/>');
    document.write('Flowers Removed: ' + flowers.pop() +
    '<br/>');
</script>
```

© Aptech Ltd. Loops and Arrays / Session 14 37

- Following figure displays the corresponding output of the script.

Manipulating Array Elements  
file:///H:/HTML5/Source%20Code/session%2014/cc

Number of flowers: 3  
Flowers: Rose, Sunflower, Daisy  
Orchid and Lily are Added: 5  
Flowers (In Ascending Order): Daisy,Lily,Orchid,Rose,Sunflower  
Flowers Removed: Sunflower

- In the code, an array variable, `flowers`, is created that stores the names of three flowers.
- The `length` property is used to display the number of flowers in the array variable.
- The `join()` method joins the flower names and separates them with a comma.
- The `push()` method adds orchid and lily at the end of the array and the total number of flowers in the array list is displayed as 5.
- The `sort()` method sorts the flowers in alphabetical order.
- The `pop()` method retrieves the last element that is `Sunflower`, from the array list.

© Aptech Ltd. Loops and Arrays / Session 14 38

Using slides 36 to 38, explain the array methods.

In JavaScript, the `Array` object allows you to create arrays. It provides the `length` property that allows you to determine the number of elements in an array. The various methods of the `Array` object allow to access and manipulate the array elements.

Table lists the most commonly used methods of the `Array` object.

Explain code snippet which demonstrates how to access and manipulate the array elements.

In the code, an array variable `flowers` is created, that stores the names of three flowers. The `length` property is used to display the number of flowers in the array variable. The `join()` method joins the flower names and separates them with a comma. The `push()` method adds orchid and lily at the end of the array and the total number of flowers in the array list is displayed as 5. The `sort()` method sorts the flowers in alphabetical order. The `pop()` method retrieves the last element that is `Sunflower`, from the array list.

### Tips:

Explain the `concat()` and `indexOf()` are the other commonly used methods. Mention `concat()` method is used to join two or more arrays. This method does not change the existing arrays, but returns a new array, containing the values of the joined arrays.

The `indexOf()` method searches the array for the specified item, and returns its position. The search will start at the specified position, or at the beginning if no start position is specified, and end the search at the end of the array. Returns `-1` if the item is not found. If the item is present more than once, the `indexOf` method returns the position of the first occurrence.

Some of the other array methods are:

1. If you want to search from end to start in an array, use the `lastIndexOf()` method
2. If you want to delete elements from an array, use the `delete()` method.
3. The `splice()` method can be used to add new elements into an array.
4. The `shift()` method removes the first element and shifts other elements by one place.

## Slides 39 to 41

Let us understand the use of `for..in` loop.

**for..in Loop 1-3**

The `for..in` loop is an extension of the `for` loop. It enables to perform specific actions on the arrays of objects.

The loop reads every element in the specified array and executes a block of code only once for each element in the array.

**Syntax:**

```
for (variable_name in array_name)
{
    //statements;
}
```

where,

- `variable_name`: Is the name of the variable.
- `array_name`: Is the array name.

© Aptech Ltd. Loops and Arrays / Session 14 39

**for..in Loop 2-3**

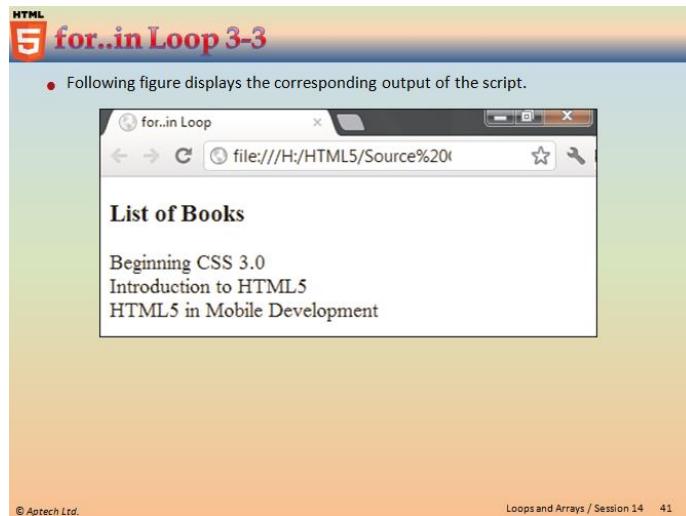
- The Code Snippet demonstrates how to display elements from an array using the `for..in` loop.

```
<script>
var books = new Array('Beginning CSS 3.0',
    'Introduction to HTML5', 'HTML5 in Mobile
    Development');

document.write('<H3> List of Books </H3>');

for(var i in books)
{
    document.write(books[i] + '<br/>');
}
</script>
```

© Aptech Ltd. Loops and Arrays / Session 14 40



Using slides 39 to 41, explain the `for..in` loop.

Mention, `for...in` loop is an extension of the `for` loop. It enables to perform specific actions on the arrays of objects. The loop reads every element in the specified array and executes a block of code only once for each element in the array.

Explain code snippet that demonstrates how to display elements from an array using the `for...in` loop. The figure displays the corresponding output of the script.

#### **Tips:**

1. The `for..in` loop can be used to traverse through the properties of an object in JavaScript.
2. For more information, you can visit this link:  
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/for...in>

## Slide 42

Let us summarize the session.

**HTML5 Summary**

- A loop construct consists of a condition that instructs the compiler the number of times a specific block of code will be executed.
- JavaScript supports three types of loops that include: while loop, for loop, and do-while loop.
- The break statement is used to exit the loop without evaluating the specified condition.
- The continue statement terminates the current execution of the loop and continue with the next repetition by returning the control to the beginning of the loop.
- JavaScript supports two types of arrays namely, Single-dimensional array and Multi-dimensional array.
- The for..in loop is an extension of the for loop that enables to perform specific actions on the arrays of objects.

© Aptech Ltd. Loops and Arrays / Session 14 42

In slide 42, you will summarize the session. You will end the session with a brief summary of what has been taught in the session.

### 14.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session. You should also explore the Functions and Objects that are offered with the next session.

#### Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 15 – Functions and Objects

---

## 15.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 15.1.1 Objectives

By the end of this session, the learners will be able to:

- Explain functions
- Explain parameterized functions
- Explain return statement
- Describe objects
- Explain different browser objects
- Describe DOM

### 15.1.2 Teaching Skills

To teach this session, you should be well-versed with functions, which are independent reusable blocks of code, executed on the occurrence of an event. Along with this, you should prepare yourself with concept of objects used for storing and manipulating entities in JavaScript. You should also be aware with the different types of built-in and browser objects supported by JavaScript.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

#### In-Class Activities:

Follow the order given here during In-Class activities.

## Overview of the Session:

Give the students the overview of the current session in the form of session objectives.

Show the students slide 2 of the presentation.

**Objectives**

- Explain functions
- Explain parameterized functions
- Explain return statement
- Describe objects
- Explain different browser objects
- Describe Document Object Model (DOM)

© Aptech Ltd. Functions and Objects / Session 15 2

Tell the students that this session introduces functions, which are independent reusable blocks of code, executed on the occurrence of an event. They will learn concept of objects used for storing and manipulating entities in JavaScript. They will also learn about different types of built-in and browser objects supported by JavaScript as well as DOM objects.

## 15.2 In-Class Explanations

### Slide 3

Let us understand functions and their need in programming.

**Introduction**

To make the code more task-oriented and manageable, JavaScript allows to group statements before they are actually invoked.

This can be achieved by using functions.

A function is a reusable block of code that is executed on the occurrence of an event.

Event can be a user action on the page or a call within the script.

© Aptech Ltd. Functions and Objects / Session 15 3

Using slide 3, explain the introduction of functions.

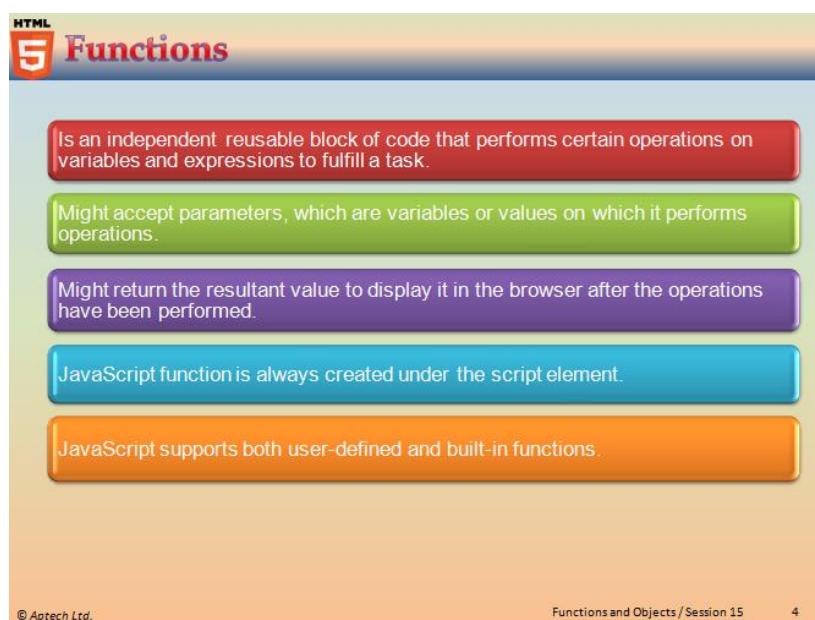
Consider a scenario where a Web page has been designed to greet the user with his/her name on the click of a button. A code can be used here to accomplish this task, but may result in the same output on repetitive execution.

However, writing these statements each time for the same action is tedious, time consuming, and error prone. To make the code more task-oriented and manageable, JavaScript allows to group statements before they are actually invoked. This can be achieved by using the concept of functions.

A function is a reusable block of code that is executed on the occurrence of an event. The event can be a user action on the page or a call within the script.

#### Slide 4

Let us understand the functions in JavaScript.



Using slide 4, explain the functions in JavaScript.

A function is an independent reusable block of code that performs certain operations on variables and expressions to fulfill a task. A function might take parameters, which are variables or values on which it performs operations. After performing operations, a function might return the resultant value to display it in the browser. For example, a function named `add()` might take two numbers on which the addition operation will be performed and will return the result of addition.

A JavaScript function is always created under the `script` element. JavaScript supports both user-defined and built-in functions.

**Tips:**

Some of the places in the code where a function can be invoked:

- On an event. For example, on the click of a button, a function can be invoked.
- A function can invoke another function in JavaScript code.
- A function can be invoked itself.

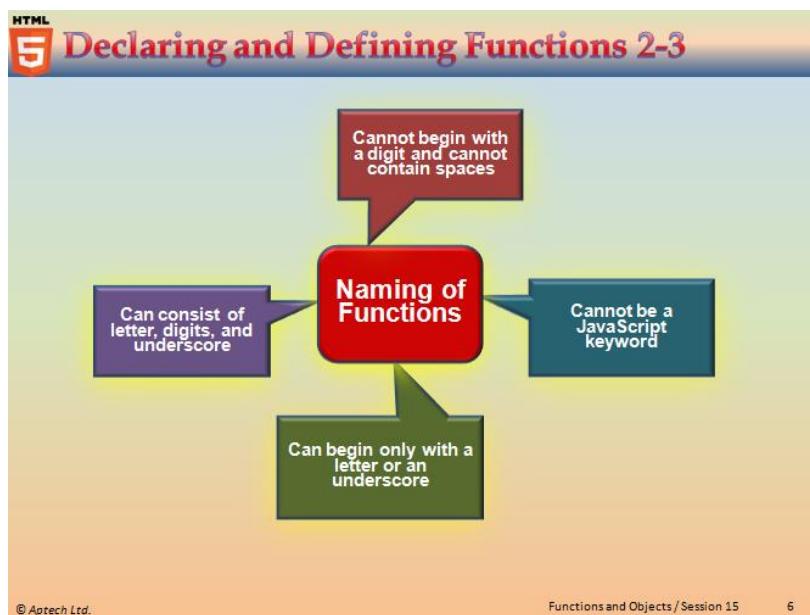
**Slides 5 to 7**

Let us understand how to declare and define functions.

**HTML** **Declaring and Defining Functions 1-3**

- JavaScript allows declaring a function using the function keyword.
- Keyword is followed by the name of the function and the parameters enclosed within the parenthesis.
- If the function does not accept any parameters, then it must be specified with an empty parenthesis.
- Once the function is declared, you need to define the function by specifying the operations or instructions within the curly braces "{" and "}".
- Curly braces indicate the start and end of the function block, which is collectively referred to as the body of the function.
- A function must be defined before it can be invoked in the script and multiple functions can be defined within the script element.

© Aptech Ltd. Functions and Objects / Session 15 5



**HTML5 Declaring and Defining Functions 3-3**

- Syntax to create a function in JavaScript is as follows:

```
function function_name(list of parameters)
{
    // Body of the function
}
```

Annotations:

- Keyword**: Points to the word **function**.
- Name of the Function**: Points to the identifier **add()**.
- No Parameters**: Points to the absence of parentheses after the function name.
- Body of the Function**: Points to the code block enclosed in curly braces {}.

© Aptech Ltd. Functions and Objects / Session 15 7

Using slides 5 to 7, explain how to declare and define a function.

Mention, JavaScript allows declaring a function using the `function` keyword. The keyword is followed by the name of the function and parameters enclosed within the parenthesis `()`. If the function does not take any parameters, then it must be specified with the empty parenthesis.

Once the function is declared, you need to define the function by specifying the operations or instructions within the curly braces `{ and }`. These curly braces indicate the start and end of the function block, which is collectively referred to as the body of the function.

There are certain conventions that must be followed for naming functions. They are as follows:

- Can consist of letter, digits, and underscore
- Can begin only with a letter or an underscore
- Cannot be a JavaScript keyword
- Cannot begin with a digit
- Cannot contain spaces

Explain the syntax of creating a function.

A function must be defined, before it can be invoked in the script. Also, there can be multiple functions defined within the script element.

Explain the figure shows the declaration and definition of a function.

**In-Class Question:**

After you finish explaining declaring and defining functions, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



How to define a function?

**Answer:**

The function by specifying the operations or instructions within the curly braces { and }.

**Slides 8 and 9**

Let us understand process of invoking functions.

### HTML 5 Invoking Functions 1-2

- A function need to be invoked or called to execute it in the browser.
- To invoke a function, specify the function name followed by parenthesis outside the function block.
- A function can be defined and invoked even in an external JavaScript file.
- A function can be called from another function in JavaScript.
- A function that invokes another function is called the calling function; whereas the function that is called is referred to as the called function.
- Functions provide the benefit of code reusability by allowing the user to call a function multiple times.

© Aptech Ltd. Functions and Objects / Session 15 8

### HTML 5 Invoking Functions 2-2

```
<script>
function add()
{
    var num1 = parseInt(prompt("Enter the
        first number for addition"));
    var num2 = parseInt(prompt("Enter the
        second number for addition"));
    var result = num1 + num2;
    alert ("Addition Result :" + result);
}

function calling_add()
{
    add();
}

calling_add();
</script>
```

Diagram illustrating the function invocation process:

- Invoking the calling\_add() Function**: The outermost function call.
- Called Function**: The function being called, indicated by a blue arrow pointing from the calling function to the add() function.
- Calling Function**: The function that is being called, indicated by a blue arrow pointing from the calling function to the add() function.

© Aptech Ltd. Functions and Objects / Session 15 9

Using slides 8 and 9, explain process of invoking function.

A function need to be invoked or called to execute it in the browser. To invoke a function, specify the function name followed by parenthesis outside the function block.

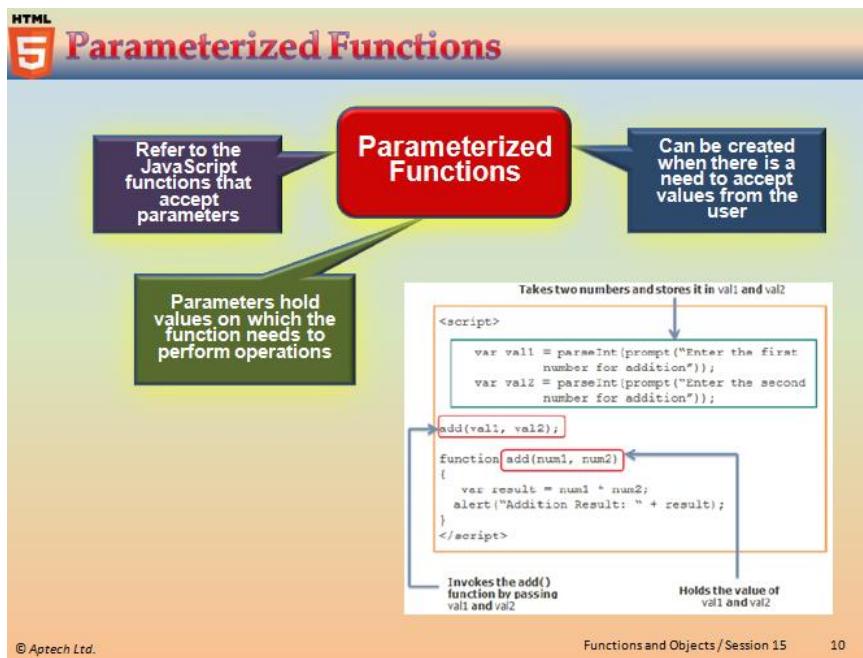
A function can be defined and invoked even in an external JavaScript file. Also, a function can be called from another function in JavaScript. The function that invokes another function is called the **calling** function; whereas the function that is called is referred to as the **called** function.

Functions provide the benefit of code reusability by allowing the user to call a function multiple times.

Figure shows invoking of function.

## Slide 10

Let us understand the parameterized functions.



Using slide 10, explain parameterized functions.

Parameterized functions refer to JavaScript functions that take parameters. These parameters hold values on which the function needs to perform operations. Parameterized functions can be created to accept values for performing operations.

Figure shows the parameterized functions.

As shown in figure, the `num1` and `num2` parameters will hold the values of `val1` and `val2` arguments to perform the operations.

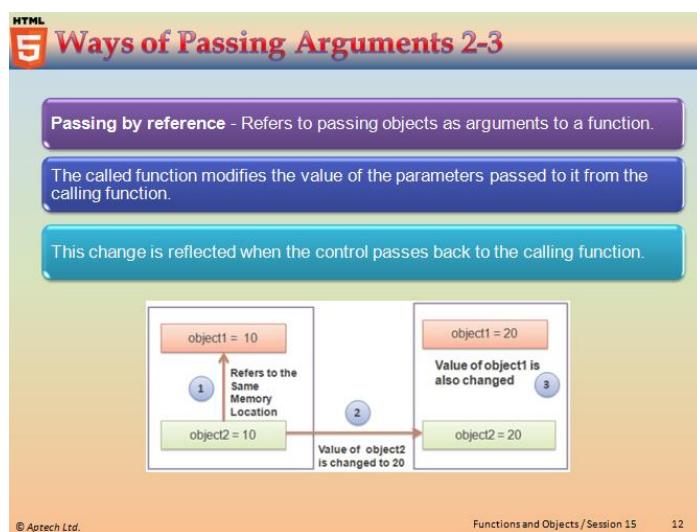
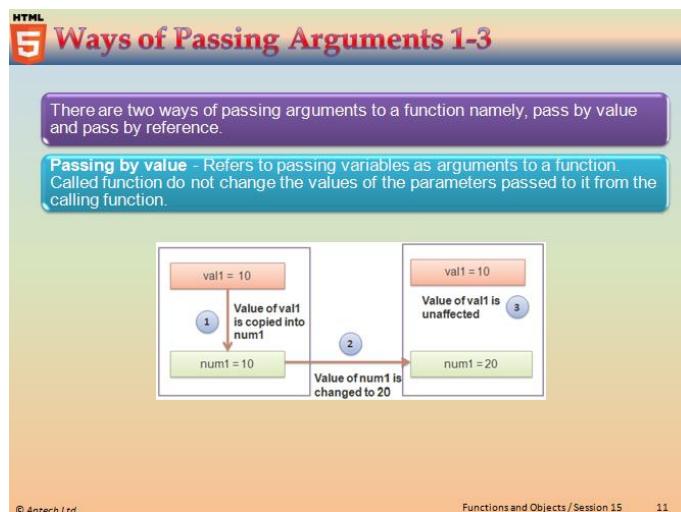
The `num1` and `num2` parameters are only accessible within the function. The parameters of a function are variables that are declared in the function declaration. Here, `num1` and `num2` are the parameters of the function.

Similarly, `val1` and `val2` are the arguments whose values are passed to the parameters, `num1` and `num2`, while invoking the function. Both the set of variables, that is, arguments and parameters of the function will occupy different memory space.

Alternatively, one can use same variable names for arguments and parameters while creating and invoking functions. In either of the case, the variables will occupy different memory space.

### Slides 11 to 13

Let us understand ways of passing arguments to the functions.



The screenshot shows a presentation slide with the title "Ways of Passing Arguments 3-3". The slide content includes a bullet point stating "The Code Snippet demonstrates passing of Array object as a parameter to a function." Below this is an `<script>` block containing JavaScript code. The code defines an array `names` with three elements: 'James', 'Kevin', and 'Brad'. It contains two functions: `change_names` which changes the first element to 'Stuart', and `display_names` which prints the names in an `UL` list. The `change_names` function is called within the `display_names` function to demonstrate how changes made in one function affect the other.

```

HTML
Ways of Passing Arguments 3-3

• The Code Snippet demonstrates passing of Array object as a parameter to a function.

<script>
var names =new Array('James', 'Kevin', 'Brad');
function change_names(names) {
    names[0]= 'Stuart';
}
function display_names(){
    document.writeln('<H3> List of Student Names:</H3>');
    document.write('<UL>');
    for(vari=0; i<names.length; i++) {
        document.write('<LI>' + names[i]+ '</LI>');
    }
    document.write('</UL>');
    change_names(names);
    document.write('<H3> List of Changed Students Names:</H3>');
    document.write('<UL>');
    for(vari=0; i<names.length; i++) {
        document.write('<LI>' + names[i]+ '</LI>');
    }
    document.write('</UL>');
}
display_names(names);
</script>

```

© Aptech Ltd. Functions and Objects / Session 15 13

Using slides 11 to 13, explain the ways of passing arguments.

There are two ways of passing arguments to a function namely, pass by value and pass by reference. The description about these is as follows:

**Passing by value** - Refers to passing variables as arguments to a function. In the pass by value method, the called function do not change the values of the parameters passed to it from the calling function.

This is because each parameter occupies different memory locations. This is because each parameter occupies different memory locations.

Figure in slide 11 shows the pass by value method.

**Passing by reference** - Refers to passing objects as arguments to a function. In the pass by reference method, the called function modifies the value of parameters passed to it from the calling function. This change is reflected when the control passes back to the calling function.

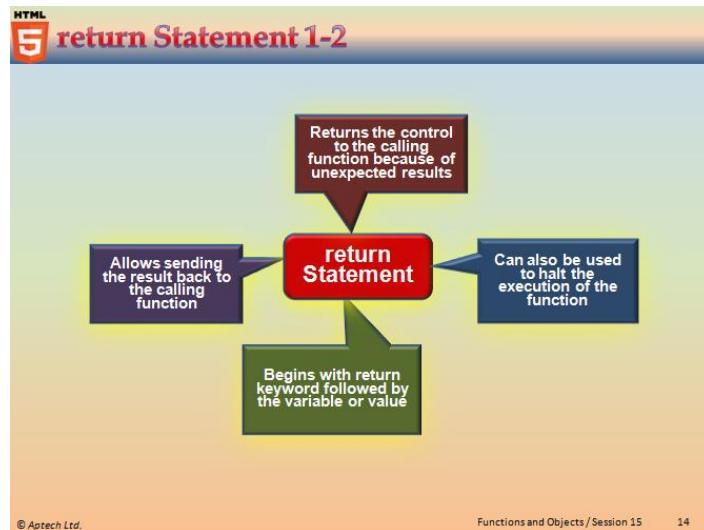
Figure in slide 12 shows the pass by reference method.

Explain code snippet which demonstrates the code to pass `Array` object as a parameter to a function.

In the code, the function `change_names (names)` takes the `names` array as parameter. It changes the value at the 0th position in the array. The function is further invoked in the `display_names ()` function. The `display_names ()` function displays the values from the array before and after the value is changed at the 0th position in the array.

## Slides 14 and 15

Let us understand the `return` statement.



The Code Snippet demonstrates the use of `return` statement.

```

<script>
    function factorial(num) {
        if(num==0)
            return0;
        elseif(num==1)
            return1;
        else
        {
            var result =num;
            while(num>1)
            {
                result = result *(num-1);
                num--;
            }
            return result;
        }
    }
    varnum=prompt('Enter number:','');
    var result = factorial(num);
    alert('Factorial of ' +num+ ' is ' + result + '.');
</script>
  
```

© Aptech Ltd.

Functions and Objects / Session 15 15

Using slides 14 and 15, explain the `return` statement.

JavaScript allows sending the result back to the calling function by using the `return` statement.

The `return` statement begins with `return` keyword followed by the variable or value, which needs to be returned to the calling function. The `return` statement can also be used to halt the execution of the function and to return the control to the calling function. This is required when a particular condition is false or when there are chances of unexpected results during the code execution.

Explain code snippet which demonstrates the script that calculates the factorial of a number using a function and display the output to the user.

The code defines a function named `factorial()` which takes the `num` variable as the parameter. The execution of the script starts from the `prompt()` function, which takes the number from the user and stores it in the `num` variable. Next, the `factorial()` function is invoked by passing the `num` argument. If the user enters the value as 0 or 1, the function returns the value as 0 or 1 respectively. For any other number, the function calculates the factorial and returns the output value by using the return statement. The output is stored in the `result` variable, which is displayed to the user.

Similarly, the `return` statement can be used to return a collection of values stored in arrays. Figure in slide 14 shows a `display_list()` function which declares and initializes an array named, `languages` to store the languages. The function returns an array whose values are then displayed to the user.

## Slides 16 and 17

Let us understand the objects in JavaScript.

**HTML 5 Objects 1-2**

- Are entities with properties and methods and resemble to real life objects.
- Properties specify the characteristics or attributes of an object.
- Methods identify the behavior of an object.

<b>Object: Car</b> 	<b>Properties</b> Make - ford Color - green Wheels - four
	<b>Methods</b> <code>run()</code> <code>stop()</code>
<b>Object: Bird</b> 	<b>Properties</b> Type - pigeon Color - gray Wings - two
	<b>Methods</b> <code>eat()</code> <code>fly()</code>

© Aptech Ltd. Functions and Objects / Session 15 16

**HTML 5 Objects 2-2**

- JavaScript provides built-in objects and allows creating user-defined objects.

<b>Built-in Objects</b> - Are pre-defined objects which are already defined. Their properties and methods need to be called to fulfill a task.
<b>Custom Objects</b> - Are user-defined objects, which the developer explicitly creates in the script and defines their properties and methods.

© Aptech Ltd. Functions and Objects / Session 15 17

Using slides 16 and 17, explain the objects.

Objects are entities with properties and methods and resemble to real life objects. Properties specify the characteristics or attributes of an object, while methods identify the behavior of an object. For example, consider a real life object namely, Car.

The attributes of the Car object can include color, car number, and model. The methods of the car could be `run()` that specifies the running behavior of the car. Similarly, in JavaScript, objects have their own properties and methods. Figure shows objects with their properties and methods.

JavaScript provides built-in objects and allows creating user-defined objects. The description of the object is as follows:

**Built-in Objects** - Are pre-defined objects which are already defined. Their properties and methods need to be called to fulfil a task. An example of a pre-defined object is the Array object.

**Custom Objects** - Are user-defined objects, which the developer explicitly creates in the script and defines their properties and methods. For example, to store the doctor details, such as name, age, hospital name, and so on, an object named doctor can be created.

#### Tips:

Object are mutable: They are addressed by reference, not by value.

#### Slides 18 to 22

Let us understand creating custom objects.

The screenshot shows a presentation slide with a blue header bar containing the text "HTML 5 Creating Custom Objects 1-5". Below the header, there are three purple callout boxes with white text, each containing a statement about custom objects. The first box says: "Object object is the parent object from which all the JavaScript objects are derived." The second box says: "Custom objects can be derived from this object by using the `new` keyword." The third box says: "Two main methods to create a custom object namely, direct method and by defining a template and initializing it with the `new` keyword." At the bottom of the slide, there is a footer bar with the text "© Aptech Ltd.", "Functions and Objects / Session 15", and the number "18".

**HTML 5 Creating Custom Objects 2-5**

- Syntax to create object using these methods is as follows:

- Direct Method**

```
var object_name = new Object();
```

where,

`object_name`: Is the name of the object.

`new`: Is the keyword that allocates memory to the custom object. This is known as instantiation of an object.

`Object`: Is the built-in JavaScript object that allows creating custom objects.

© Aptech Ltd. Functions and Objects / Session 15 19

**HTML 5 Creating Custom Objects 3-5**

- Template Method**

An object's template refers to a structure that specifies the custom properties and methods of an object.

First, the object type is declared using a constructor function.

Second, you specify the object of the declared object type by using the `new` keyword.

JavaScript allows creating a reusable template without having to redefine properties and methods repeatedly to fulfill a particular object's requirements.

This template is known as the constructor function.

© Aptech Ltd. Functions and Objects / Session 15 20

**HTML 5 Creating Custom Objects 4-5**

- Template Method**

A constructor function is a reusable block that specifies the type of object, its properties, and its methods.

It might or might not take any parameters.

After creating the constructor function, you specify an object of the declared object type using the `new` keyword.

`new` keyword allocates memory to the object and invokes the constructor function.

- Syntax to create a constructor function is as follows:

```
function object_type(list of parameters)
{
    // Body specifying properties and methods
}
```

© Aptech Ltd. Functions and Objects / Session 15 21

The slide is titled "Creating Custom Objects 5-5" and includes the following content:

- Syntax to create a object using the new keyword is as follows:
 

```
object_name = new object_type(optional list of arguments);
```
- The Code Snippet shows the creation of objects using direct and template method is as follows:
 

```
<script>
  //create an object using direct method
  var doctor_details=new Object();
  //create an object using new keyword
  studOne = new student_info ('James', '23', 'New Jersey');
</script>
```

© Aptech Ltd. Functions and Objects / Session 15 22

Using slides 18 to 22, explain creating custom objects.

The `Object` object is the parent object from which all JavaScript objects are derived. The custom objects can be derived from this object by using the `new` keyword.

There are two main methods to create a custom object namely, direct method and template method.

Using slides 19 and 20 explain the direct method and template method. In the first method, an object can be created by using the built-in `Object` object which is also known as the direct method. In the second method, an object can be created by defining a template and initializing it with the `new` keyword.

Then, explain the syntax of creating objects through direct as well as template methods.

Explain the syntax in slides 21 and 22 for creating objects.

Explain code snippet shows the creation of objects using direct method and template method.

In the code, `doctor_details` object is created using the `Object` object. After creating the object, properties and methods can be specified. Similarly, `student_info` object is created using `new` keyword. The values 'James', '23', and 'New Jersey' are the properties of the `student_info` which are initialized by constructor function during creation.

Mention about `this` keyword. In JavaScript, the thing called `this`, is the object that "owns" the JavaScript code. The value of `this`, when used in a function, is the object that "owns" the function. The value of `this`, when used in an object, is the object itself.

The `this` keyword in an object constructor does not have a value. It is only a substitute for the new object. The value of `this` will become the new object when the constructor is used to create an object.

## Slides 23 and 24

Let us understand the process of creating properties of custom objects.

**Properties specify the characteristics of an object created through Object or template method.**

To access a property of an object created using Object object, specify the object name followed by a period and the property name.

- The Code Snippet demonstrates the script that creates the student\_details object.

```
<script>
var student_details=new Object();
student_details.first_name= 'John';
student_details.last_name= 'Fernando';
student_details.age= '15';
alert('Student\'s name: ' +student_details.first_name+ ' ' +
+student_details.last_name);
</script>
```

© Aptech Ltd. Functions and Objects / Session 15 23

**The Code Snippet creates the employee\_info object and adds properties in the constructor function.**

```
<script>
// To define the object type
function employee_info(name, age, experience)
{
    this.name = name;
    this.age= age;
    this.experience= experience;
}
// Creates an object using new keyword
empMary=new employee_info('Mary', '34', '5 years');
alert("Name: "+ empMary.name + '\n' +"Age: "+empMary.age+ '\n' +
+"Experience: "+empMary.experience);
</script>
```

© Aptech Ltd. Functions and Objects / Session 15 24

Using slides 23 and 24, explain the process of creating properties of custom objects.

Mention that, properties specify the characteristics of an object. They can be specified for objects created through `Object` or template method.

To create and access a property of an object created using `Object` object, specify the object name followed by a period and the property name.

Explain code snippet which demonstrates the script that creates the `student_details` object and adds properties namely, `first_name`, `last_name`, and `age` along with their values.

The code specifies three properties of the `student_details` object namely, `first_name`, `last_name`, and `age` along with their values. The values of these properties are displayed in the browser using the `write()` method.

Figure on slide 23 shows the output of the `student_details` object.

Explain code snippet on slide 24.

The code specifies three properties namely, `name`, `age`, and `experience` along with their values in the constructor function. The object named `empMary` is created, which passes the values as the arguments. This invokes the constructor function and initializes the properties to their values. The `this` keyword is a reference to the current object whose properties are being initialized. It is used in the constructor to resolve conflict between the property and the parameter, both of which have the same name.

The `this` keyword marks the distinction between the two, while assigning the value to the properties of an object. Figure on slide 24 displays the output of the `employee_info` object.

## Slide 25

Let us understand creating custom methods for custom objects.

The slide has a header 'HTML5 Creating Methods for Custom Objects'. Below the header are four bullet points in red boxes:

- Methods are similar to JavaScript functions.
- A method is associated with an object and is executed by referring to that object but a function is not associated with any object and is executed independently.
- One or more methods can be specified after creating an object using the Object object or while creating the template.
- To invoke a method, they must be specified with the object name followed by a period, method name, and parenthesis with parameters, if any.

Below the bullet points is a list item with a red dot:

- The Code Snippet demonstrates the script that defines a custom method.

```

<script>
    var square =new Object();
    square.length=parseInt("5");
    square.cal_area=function() {
        var area_=(parseInt(square.length)*parseInt("4"));
        return area;
    }
    alert("Area: "+square.cal_area());
</script>

```

At the bottom left is the copyright notice '© Aptech Ltd.' and at the bottom right are the page numbers 'Functions and Objects / Session 15' and '25'.

Using slide 25, explain how to create methods for custom objects.

Methods are similar to JavaScript functions, but there is a slight difference between them. A method is always associated with an object and is executed by referring to that object. On the other hand, a function is not associated with any object and is executed independently. Similar to functions, the custom methods can also take parameters.

One or more methods can be specified after creating an object using the `Object` object or while creating the template. To invoke a method, they must be specified with the object name followed by a period, method name, and parenthesis with parameters, if any.

Explain code snippet which demonstrates the code that defines a custom method to calculate the area of a square.

The code defines a custom object named `square` whose `length` property is set to a numeric value 5. It also defines a custom function named `cal_area()`, which calculates the area of the square and returns the result.

### Tips:

Methods are similar to functions that are defined as object properties.

### Slide 26

Let us understand built-in objects.

**Built-in Objects**

- Object model of JavaScript language forms the foundation of the language.
- These objects help to provide custom functionalities in the script.
- JavaScript treats the primitive data types as objects and provide equivalent object for each of them.
- JavaScript objects are categorized as built-in objects, browser objects, and HTML objects.
- Built-in objects are static objects which can be used to extend the functionality in the script.
- Browser objects, such as `window`, `history`, and `navigator` are used to work with the browser window.
- HTML objects, such as `form`, `anchor`, and so on are used to access elements on the Web page.

© Aptech Ltd. Functions and Objects / Session 15 26

Using slide 26, explain the built-in object in JavaScript.

Mention object model of JavaScript language forms the foundation of the language. These objects help to provide custom functionalities in the script.

JavaScript treats the primitive data types as objects and provide equivalent object for each of them. For example, if a variable contains a string of characters, then it is treated as `String` object by JavaScript. Similarly, if a variable contains the value `true` or `false`, it is treated as `Boolean` object.

JavaScript objects are categorized as built-in objects, browser objects, and HTML objects.

The built-in objects are static objects which can be used to extend the functionality in the script. Some of these objects are: `String`, `Math`, and `Date`. The browser objects, such as `window`, `history`, and `navigator` are used to work with the browser window, whereas the HTML objects, such as `form`, `anchor`, and so on are used to access elements on a Web page.

## Slides 27 to 29

Let us understand String object.

**HTML5 String Object 1-3**

- Strings in JavaScript are a set of characters that are surrounded by single or double quotes.
- Built-in String object allows you to perform different text operations on them.
- String object is instantiated with the **new** keyword, which invokes the predefined constructor function of the String object.
  - Syntax to initialize the String object is as follows:  
`var object_name = new String("Set of characters") ;`
  - Following table lists properties of the String object.

Property	Description
length	Retrieves the number of characters in a string.
prototype	Adds user-defined properties and methods to the String instance.

© Aptech Ltd. Functions and Objects / Session 15 27

**HTML5 String Object 2-3**

- Following table lists the methods of the String object.

Method	Description
charAt()	Retrieves a character from a particular position within a string.
concat()	Merges characters from one string with the characters from another string and retrieves a single new string.
indexOf()	Retrieves the position at which the specified string value first occurred in the string.
lastIndexOf()	Retrieves the position at which the specified string value last occurred in the string.
replace()	Matches a regular expression with the string and replaces it with a new string.
search()	Searches for a match where the string is in the same format as specified by a regular expression.
split()	Divides the string into substrings and defines an array of these substrings.
substring()	Retrieves a part of a string between the specified positions of a string.
toLowerCase()	Specifies the lowercase display of the string.

© Aptech Ltd. Functions and Objects / Session 15 28

The Code Snippet demonstrates the script that creates a `String` object and test various methods.

```
<script>
    var full_name=new String('David James Taylor');
    document.write('Number of Characters are: ' +full_name.length+
    '<BR/>');
    document.write('Character at Position 6 is: ' +
    full_name.charAt(6)+ '<BR/>');
    document.write('Student\'s Name and their Father\'s name are:
    '+full_name.split(' ',2)+ '<BR/>');
    document.write('Student\'s Full Name is: ' +
    full_name.toUpperCase());
</script>
```

Using slides 27 to 29, explain the `String` object.

Strings in JavaScript are a set of characters that are surrounded by single or double quotes. The built-in `String` object allows you to perform different text operations on them. Some of the examples of these operations include: searching for a specific character occurrence, retrieving a substring, merging two set of characters, and so on.

The `String` object is instantiated with the `new` keyword, which invokes the predefined constructor function of the `String` object.

Explain the syntax to initialize `String` object. Also explain the properties and methods of the `String` object.

Explain the code snippet demonstrates the script that creates a `String` object and test various methods on it.

In the code, property `length` of `String` objects and methods `split()`, `charAt()`, and `toUppercase()` are used. Explain the output in figure in slide 29.

### In-Class Question:

After you finish explaining `String` object, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which method of `String` object is used to divide a string into substrings?

### Answer:

`split()` method of `String` object is used to divide a string into substrings.

## Slides 30 and 31

Let us understand Math object.

**HTML 5 Math Object 1-2**

- Math object allows the user to perform mathematical operations on numeric values.
- Math object is a pre-defined object that provides static properties and methods to perform mathematical operations.
- Properties and methods are declared as static, thus they can be invoked directly with the object name.

- Syntax to access the properties of the Math object is as follows:  
`var variable_name = Math.PropertyName;`
- Syntax to invoke the methods of the Math object is as follows:  
`var variable_name = Math.MethodName(optional list of parameters);`

© Aptech Ltd. Functions and Objects / Session 15 30

**HTML 5 Math Object 2-2**

- The Code Snippet demonstrates the script that creates a Math object.

```
<script>
    var full_name=new String('David James Taylor');
    document.write('Number of Characters are: ' +full_name.length+
    '<BR/>');
    var area=Math.floor(tempArea);
    return area;
}
alert('Area of circle is: ' +area_circle(5));
</script>
```

A screenshot of a JavaScript Alert dialog box showing the output: "Area of circle is: 78".

© Aptech Ltd. Functions and Objects / Session 15 31

Using slides 30 and 31, explain the Math object.

The Math object allows the user to perform mathematical operations on numeric values. The Math object is a pre-defined object that provides static properties and methods to perform mathematical operations.

The properties and methods are declared as static, thus they can be invoked directly with the object name. In other words, no object instantiation is required for the Math object.

Explain the syntax for the property and methods of Math object. Explain code snippet which demonstrates the script that calculates area of a circle using the Math object.

In the code, `Math.PI` is a property that stores the value 3.142 into the variable, `pi`. The `Math.pow(radius, 2)` method calculates the radius raised to the power 2. Similarly, `Math.floor(tempArea)` method rounds the resultant value to a number less than or equal to the resultant value. Figure shows the area of circle with the radius 5.

### Tips:

`Math` class does not have constructor. All properties and methods of `Math` can be called without creating object of it.

### Slides 32 to 34

Let us understand `Date` Object.

**HTML Date Object 1-3**

- Date object allows you to define and manipulate the date and time values programmatically.
- It supports both the Universal Time Coordinated (UTC) and Greenwich Mean Time (GMT) conventions.
- Date object calculates dates in milliseconds from 01 January, 1970.

- Syntax to instantiate the `Date` object is as follows:

```
var object_name = new Date();
var object_name = new Date(milliseconds);
var object_name = new Date(year, month, day, hour, minutes,
seconds, milliseconds);
var object_name = new Date("dateString");
```

© Aptech Ltd. Functions and Objects / Session 15 32

**HTML Date Object 2-3**

- Following table lists the methods of the `Date` object.

Method	Description
<code>getDate()</code>	Retrieves a numeric value between 1 and 31, which indicates the day of the month.
<code>getDay()</code>	Retrieves a numeric value between 0 and 6, which indicates the day of the week.
<code>getTime()</code>	Retrieves a numeric value, which indicates the time passed in milliseconds since midnight 01/01/1970.
<code>getFullYear()</code>	Retrieves a four digit numeric value, which indicates the year in the given date.

© Aptech Ltd. Functions and Objects / Session 15 33

The screenshot shows a presentation slide with the title "Date Object 3-3". A bullet point states: "The Code Snippet demonstrates the script that displays the current date in the mm/dd/yyyy format." Below the text is a `<script>` block:

```

<script>
    function display_date()
    {
        var today =new Date();
        var date =today.getDate();
        var month =today.getMonth();
        month++;
        var year =today.getFullYear();
        alert('Today\'s date is: ' + month + '/' + date + '/' + year);
    }
    display_date();
</script>

```

To the right of the code is a screenshot of a JavaScript alert dialog box. The dialog box has the title "JavaScript Alert" and the message "Today's date is: 6/26/2012". There is an "OK" button at the bottom of the dialog.

At the bottom left of the slide is the copyright notice "© Aptech Ltd.". At the bottom right are the page numbers "Functions and Objects / Session 15" and "34".

Using slides 32 to 34, explain Date object.

The Date object allows you to define and manipulate the date and time values programmatically. It supports both the Universal Time Coordinated (UTC) and Greenwich Mean Time (GMT) conventions. GMT is the standard time zone that includes Greenwich and divides the globe into 24 zones, each with a difference of an hour in time. UTC splits time into days, hours, minutes, and seconds and approximates GMT. The Date object calculates dates in milliseconds from 01 January, 1970. The date and time can be specified by creating an instance of the Date object.

Explain the various syntax for instantiating the Date object. Then, explain the list of the methods of the Date object.

Explain code snippet on slide 34 which demonstrates the script that displays the current date in the mm/dd/yyyy format.

## Slide 35

Let us understand `with` statement.

**with Statement**

- with** statement allows to remove the object reference for each JavaScript statement.
- with** statement starts with the **with** keyword followed by the open and close brackets, which holds the statements that refer to a common object.
- with** statement increases the readability of the code and also reduces time required in writing each object reference in every related statement.

● Syntax to declare the **with** statement is as follows:

```
with(object_name)
{
//Statements
}
```

The diagram illustrates the usage of the `with` statement. It shows a function `displayCarInformation()` that creates a `car` object and sets its `price` and `mileage` properties. Inside the function, a `with (car)` block is used to refer to the `car` object. This allows the code to directly access the `price` and `mileage` properties without prefixing them with `car.`. Annotations explain that the `with` block refers to the `car` object and that the properties inside refer to the `car` object's properties.

© Aptech Ltd. Functions and Objects / Session 15 35

Using slide 35, explain `with` statement.

The `with` statement allows to remove the object reference for each JavaScript statement. This is done by referring to the common object only once for a set of statements.

## Slide 36

Let us understand `browser` object.

**Browser Objects**

- JavaScript also provides objects to access and manipulate various aspects of the Web browser.
- These objects are called as browser objects.
- They exist on all pages displayed in the browser and correspond to elements of a page.

A diagram shows the `window` object with four properties: `document`, `history`, `location`, and `navigator`. Arrows point from the `window` box to each of these four boxes.

© Aptech Ltd. Functions and Objects / Session 15 36

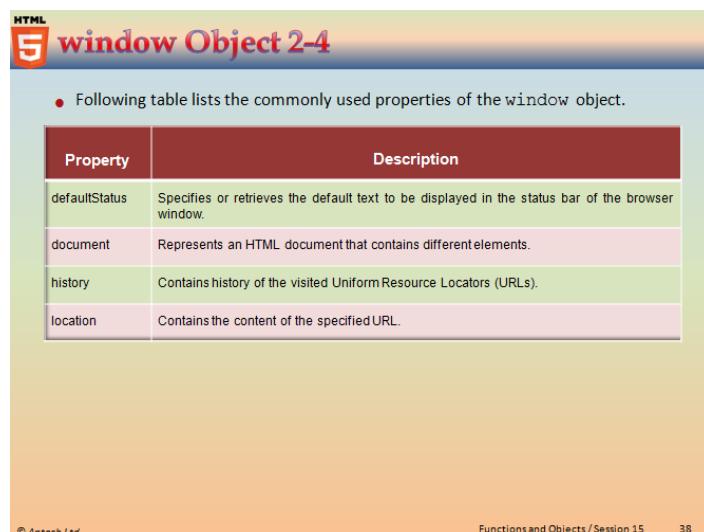
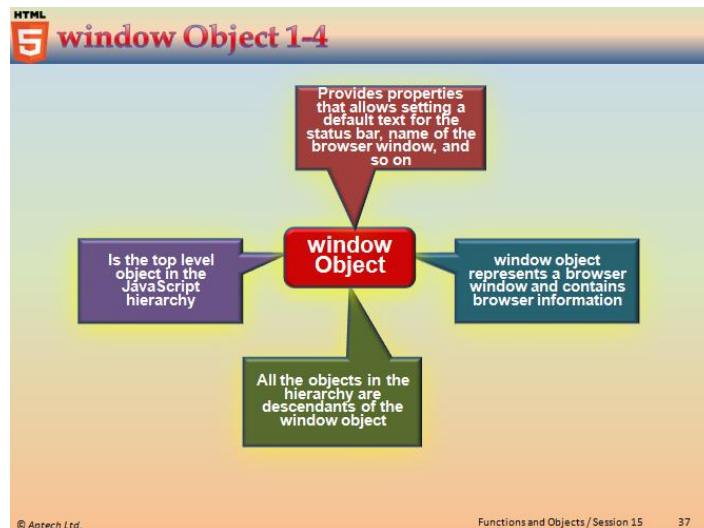
Using slide 36, explain `browser` objects.

Apart from built-in objects, JavaScript also provides objects to access and manipulate various aspects of a Web browser. These objects are called as browser objects. They exist on all pages displayed in the browser and correspond to elements on a page.

For example, browser objects allow accessing various characteristics of the browser, such as browser window itself, browser history, changing current URL, and moving backward and forward in the browser.

### Slides 37 to 40

Let us understand `window` object.



**HTML5 window Object 3-4**

- Following table lists the methods of the window object.

Method	Description
alert()	Displays an alert box that states the message and an OK button.
confirm()	Prompts a dialog box that displays a message with the OK and Cancel buttons.
createPopup()	Creates a pop-up window.
focus()	Focuses the current window.
open()	Opens the specified file in a new browser window.
prompt()	Prompts a dialog box that accepts input from the user.

© Aptech Ltd.

Functions and Objects / Session 15 39

**HTML5 window Object 4-4**

- The Code Snippet demonstrates the methods of the window object.

```
<!DOCTYPE html>
<head>
    <title> window Object </title>
    <script>
        function new_window() {
            if(confirm('Do you want to open a new page?')) {
                window.open('http://www.aptech-education.com/pages/about-us/about-aptech.html', '_parent');
            }
            else {
                window.alert('In the Current Window');
            }
        }
    </script>
</head>
<body>
    <input type="button" value="Click to move to the next page" onclick="new_window();"/>
</body>
</html>
```

© Aptech Ltd.

Functions and Objects / Session 15 40

Using slides 37 to 40, explain window Object.

Mention, window object is the top level object in JavaScript hierarchy. This means that all the objects in the hierarchy are descendants of the window object. The window object represents a browser window. It contains browser information, such as the look and feel of the browser, its version, and so on.

The window object provides properties that allows setting a default text for the status bar, name of the browser window, retrieve history, and so on.

Explain the list some of the commonly used properties of the window object.

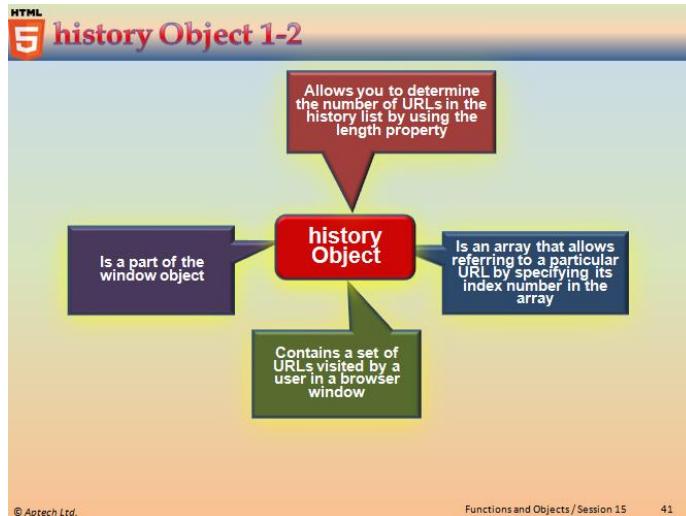
The window object provides methods that allow displaying error messages, confirmation boxes, and so on. Explain the list some of the commonly used methods of the window object.

Explain code snippet which demonstrates the methods of the window object.

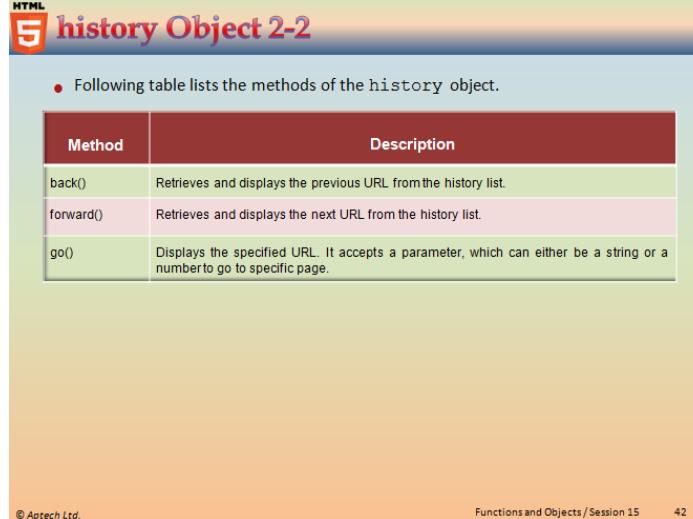
The code invokes the function `new_window()` on a button click. This function asks the user for opening the new page. If user clicks OK, then `about-aptech.html` page is opened in the current window. Otherwise, displays a message to the user.

## Slides 41 and 42

Let us understand `history` object.



© Aptech Ltd. Functions and Objects / Session 15 41



© Aptech Ltd. Functions and Objects / Session 15 42

Using slides 41 and 42, explain `history` object.

The `history` object is a part of the `window` object. It contains a set of URLs visited by a user in a browser window. The `history` object is an array that allows referring to a particular URL by specifying its index number in the array. The `length` property allows you to determine the number of URLs in the history list.

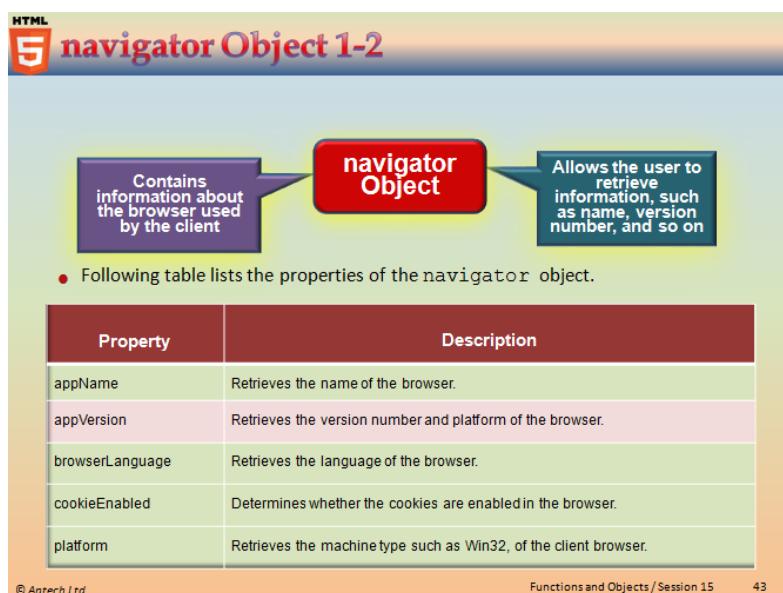
Explain the list of methods of the `history` object.

Explain the code snippet which uses `history.back()` method to load the previous URL in the history list.

```
<script>
function goBack()
{
    window.history.back()
}
</script>
```

## Slides 43 and 44

Let us understand navigator object.



The diagram illustrates the **navigator Object 2-2**. A bulleted list states:

- The Code Snippet demonstrates the use of navigator object to retrieve information of the browser.

```
<!DOCTYPE html>
<head>
    <title> navigator Object </title>
    <script>
        function display_browser() {
            document.write('Browser name: ' +navigator.appName+ '<BR/>');
            document.write('Browser version: ' +navigator.appVersion+
            '<BR/>');
            document.write('Browser language: ' +navigator.browserLanguage+
            '<BR/>');
            document.write('Platform: ' +navigator.platform+ '<BR/>');
            if(navigator.cookieEnabled) {
                document.write('Cookie is enabled in the browser.');
            }
        }
    </script>
</head>
<body>
    <input type="button" value="Browser Information"
    onclick="display_browser()"/>
</body>
</html>
```

© Aptech Ltd. Functions and Objects / Session 15 44

Using slides 43 and 44, explain navigator object. The navigator object contains information about a browser used by the client. It allows the user to retrieve browser specific information, such as name, version number, and language.

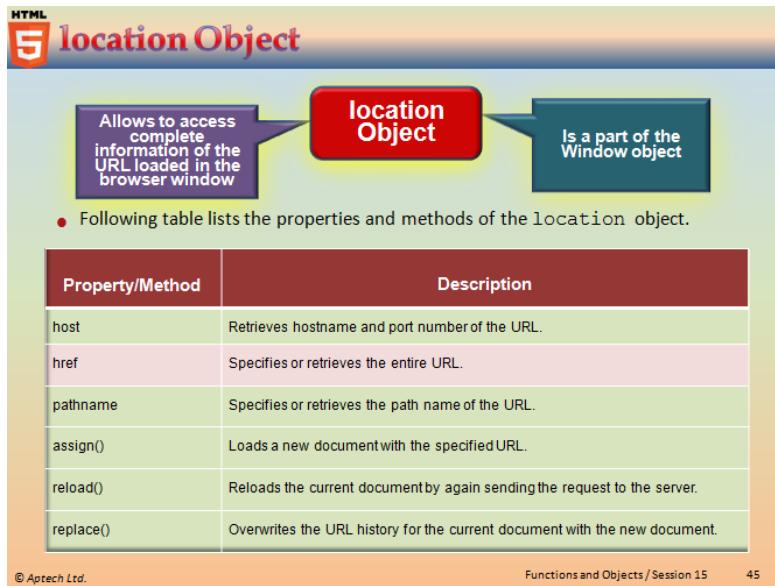
Explain the list of properties of the `navigator` object. Explain code snippet which demonstrates the use of `navigator` object to retrieve information about the browser.

### Tips:

The method `javaEnabled()` returns true if Java is enabled.

### Slide 45

Let us understand `location` object.



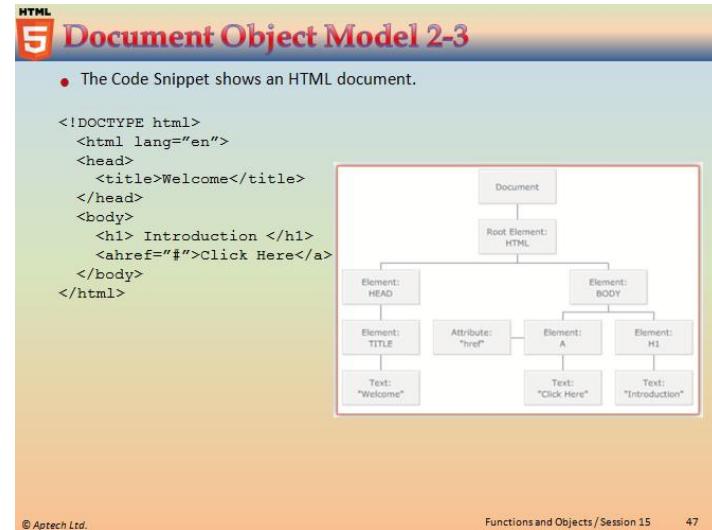
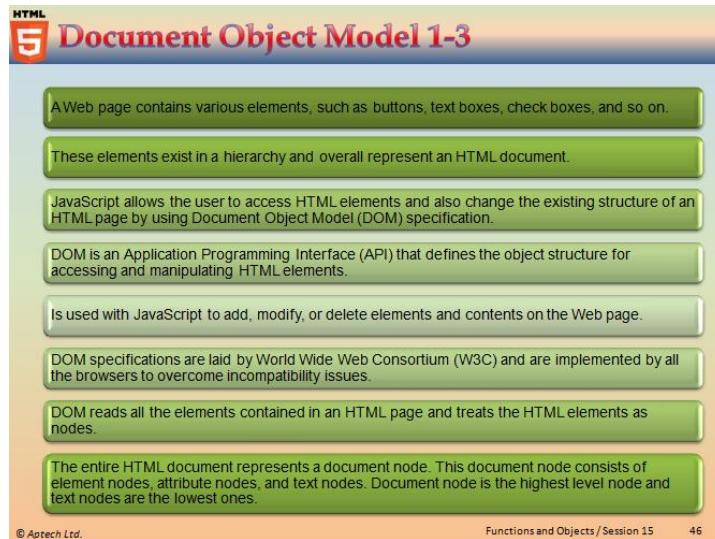
Using slide 45, explain `location` object.

The `location` object allows to access complete information of the URL loaded in the browser window. It is a part of the `Window` object. A single URL is composed of different portions, such as the host name, port number, and so on which can be accessed through the `location` object.

Explain the list of properties and methods of the `location` object.

## Slides 46 to 48

Let us understand Document Object Model (DOM).



The slide has a red header bar with the title 'Document Object Model 3-3'. The main content area has a yellow-to-orange gradient background. It contains several bullet points and some callout boxes:

- All the nodes present in the node hierarchy contain certain properties.
- These properties provide information about the node. The different node properties are as follows:
  - nodeName** - Represents the name of the node. It contains the tag name of the HTML element in upper case.
  - nodeValue** - Represents the text contained within the node. This property is only available for attribute nodes and not for document and element nodes.
  - nodeType** - Represents the type of the node. For example, the document node, element node, and so on.
- HTML DOM provides standard objects for HTML documents and some of these objects are as follows:
  - Document object
  - Form object
  - Link object
  - Table object

At the bottom left is the copyright notice '© Aptech Ltd.' and at the bottom right are the page numbers 'Functions and Objects / Session 15' and '48'.

Using slides 46 to 48, explain DOM. A Web page contains various elements, such as buttons, text boxes, check boxes, and so on. These elements exist in a hierarchy and overall represent an HTML document.

JavaScript allows the user to access HTML elements and also change the existing structure of an HTML document. This can be done by using DOM specification. The DOM is an API that defines the object structure for accessing and manipulating HTML elements. It is used with JavaScript to add, modify, or delete elements and contents on a Web page.

DOM specifications are laid by W3C and are implemented by all the browsers to overcome incompatibility issues. W3C DOM specifications are divided into levels. The level 1 specification of DOM was first defined in 1998. The current DOM specification is level 3.

The DOM reads all the elements contained in an HTML page. It treats the HTML elements as nodes. According to DOM specification, the entire HTML document represents a document node. This document node consists of element nodes, attribute nodes, and text nodes. Thus, the document node is the highest level node and text nodes are the lowest ones. Every node in the node hierarchy has a parent node, which consists of multiple child nodes. For example, `<head>` and `<body>` are the child nodes of `<html>`. All these nodes together form up a node tree and are related to each other.

## Slides 49 to 51

Let us understand document object.

**Document Object 1-3**

- Is used within the JavaScript to access all HTML elements presented on the page.
- Represents the entire HTML document and provides access to other elements, such as links, anchors, and so on.
- Has only one document object which is created when the BODY element is loaded on the Web page.
- Is also the part of the window object and is accessed as, `window.document`.
- Provides properties that allow the user to specify or retrieve the information about the elements and its content.

© Aptech Ltd.

Functions and Objects / Session 15

49

**Document Object 2-3**

- Following table lists the commonly used methods of the document object.

Method	Description
<code>close()</code>	Closes a data stream and displays the data collected using the <code>open()</code> method.
<code>getElementById()</code>	Retrieves a collection of HTML elements by using the specified ID.
<code>getElementsByName()</code>	Retrieves a collection of HTML elements by using the specified name.
<code>getElementsByTagName()</code>	Retrieves a collection of HTML elements with the specified tag name.
<code>open()</code>	Opens a stream to accept the output from <code>write()</code> or <code>writeln()</code> methods.
<code>write()</code>	Writes the text or HTML expression to a document.

© Aptech Ltd.

Functions and Objects / Session 15

50

**50 (a)**

**HTML5 Document Object 2-3**

- Following table lists the commonly used properties of the document object.

Property	Description
body	Provides access to the BODY element.
title	Specifies or retrieves the title of the document.
anchors	Retrieves the collection containing all the anchors contained in a document.
forms	Retrieves the collection containing all the forms contained in a document.
images	Retrieves the collection containing all the images contained in a document.
links	Retrieves the collection containing all the links contained in a document.

© Aptech Ltd. Functions and Objects / Session 15 50

**50 (b)**

**HTML5 Document Object 3-3**

- The Code Snippet demonstrates how to use the document object.

```
<!DOCTYPE html>
<head> <title> Document Object </title>
<script>
function change_image() {
    var imgText=document.getElementById('myImg').alt;
    if(imgText=="ford") {
        document.getElementById('myImg').src="ferrari.jpg";
        document.getElementById('myImg').alt ="ferrari";
        document.getElementById('mytext').value ="Ferrari Car";
    } else {
        document.getElementById('myImg').src="ford.jpg";
        document.getElementById('myImg').alt ="ford";
        document.getElementById('mytext').value ="Ford Car";
    }
}</script> </head>
<body>
<imgid="myImg" src="ford.jpg" width="300" height="300" alt="ford"
/> <br/>
Model: <input type="text" id="mytext" value="Ford Car"
readonly="readonly"/><br/><br/>
<input type="button" value="Change Image" onclick="change_image()"/>
</body>
```

© Aptech Ltd. Functions and Objects / Session 15 51

Using slides 49 to 51, explain the document object.

The HTML DOM provides a document object which is used within JavaScript to access all HTML elements presented on the page. The document object is one of the basic JavaScript objects. It represents the entire HTML document. It provides access to other elements, such as links, anchors, and so on.

Each HTML page has only one document object. This object is created when the BODY element is loaded on the Web page. The document object is also part of the window object and is accessed as `window.document`.

The document object provides properties that allow the user to specify or retrieve the information about the elements and its content.

Explain the list of properties and methods of the document object using slide 50 (a) and 50 (b).

Explain code snippet which demonstrates the use of the document object to change the image on the click of a button.

In the code, image and text elements on the page are accessed using `document.getElementById()` method. The method retrieves the elements based on the specified ids and sets new values for their properties. This is all done at runtime, and not through markup. The use of `document.getElementById()` enables to access the elements within JavaScript function and change them dynamically.

## Slides 52 and 53

Let us understand `form` object.

**HTML 5 Form Object 1-2**

- Accepts input from the user and sends the user data for validation.
- A single HTML document can contain multiple forms.
- DOM specification provides a form object that represents an HTML form which is created for each <form> tag in an HTML document.

© Aptech Ltd. Functions and Objects / Session 15 52

**HTML 5 Form Object 2-2**

- The Code Snippet demonstrates how to use the `form` object.

```
<!DOCTYPE html>
<head>
    <title> Form Object </title>
    <script>
        function display_length() {
            var count = document.getElementById("form1").length;
            alert("Number of controls on the form: " + count);
        }
    </script>
</head>
<body>
    <form id="form1" action="welcome.php">
        First name: <input type="text" name="firstname" value="John" /><br />
        Last name: <input type="text" name="lastname" value="Smith" /><br />
        Age : <input type="text" name="age" value="40" /><br />
        <input type="button" value="Controls" onClick="display_length()" />
    </form>
</body>
</html>
```

© Aptech Ltd. Functions and Objects / Session 15 53

Using slides 52 and 53, explain `form` object.

Form accepts input from the user and sends data for validation. JavaScript allows you to process and validate the form data. A single HTML document can contain multiple forms. The DOM specification provides a `form` object that represents an HTML form. A `form` object is created for each `<form>` tag in an HTML document.

Explain code snippet which demonstrates the use of the `form` object that counts number of controls in a form.

In the code, a Web page contains a form with input elements, such as text and a button. The form is accessed in JavaScript using the `id` attribute which is set to `form1`. Then, the `length` property of the `form` object is used to retrieve the number of elements in a form.

Thus, the statement `form1.length` returns the value 4, which is stored in the variable `count`. Finally, the value of variable `count` is displayed in the alert window.

**In-Class Question:**

After you finish explaining `form` object, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which property is used to retrieve number of elements in form?

**Answer:**

`length` property is used to retrieve number of elements in form.

## Slide 54

Let us summarize the session.

The slide has a header 'HTML5 Summary' with a small orange icon. The content area contains a bulleted list of nine items:

- A function is reusable piece of code, which performs calculations on parameters and other variables.
- The return statement passes the resultant output to the calling function after the execution of the called function.
- Objects are entities with properties and methods and resemble to real life objects.
- There are two ways to create a custom object namely, by directly instantiating the Object object or by creating a constructor function.
- JavaScript provides various built-in objects, such as String, Math, and Date.
- JavaScript also provides browser objects, such as window, history, location, and navigator.
- DOM is a standard technique for dynamically accessing and manipulating HTML elements. The DOM provides a document object which is used within the JavaScript to access all HTML elements presented on the page.

At the bottom left is a copyright notice '© Aptech Ltd.' and at the bottom right are 'Functions and Objects / Session 15' and '54'.

In slide 54, you will summarize the session. You will end the session, with a brief summary of what has been taught in the session.

### 15.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session. You should also explore the Building a Mobile Web Application that are offered with the next session.

#### Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 16 – Building a Mobile Web Application

---

## 16.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 16.1.1 Objectives

By the end of this session, the learners will be able to:

- Describe the features of different mobile devices
- List the different types of platforms available for mobile devices
- Explain the design and architectural aspects of a mobile Web site
- Explain the requirements for developing and testing of a mobile Web site
- Explain HTML5 support for a mobile Web site
- List the best practices for optimizing a mobile Web site

### 16.1.2 Teaching Skills

To teach this session, you should be well-versed with different categories and platforms for mobile devices. Along with this, you should prepare yourself about the architectural and design needs of mobile devices and use of HTML5 to design Web pages for mobiles.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

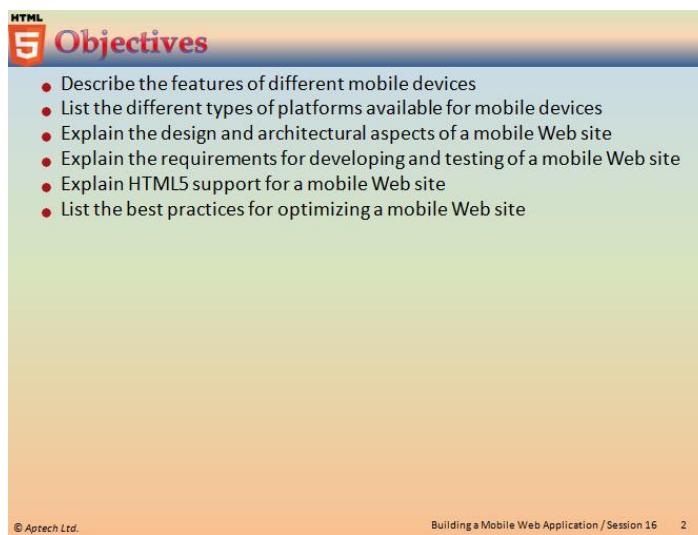
**In-Class Activities:**

Follow the order given here during In-Class activities.

**Overview of the Session:**

Then give the students the overview of the current session in the form of session objectives.

Show the students slide 2 of the presentation.



Tell the students that this session introduces them to different types of mobile devices and their operating systems. They will learn about the architectural and design needs of a mobile device and understand the HTML5 tags used for designing Web pages for mobile devices.

## 16.2 In-Class Explanations

### Slide 3

Let us understand mobile application environment.

The slide has a header 'Mobile Application Environment' with a small icon. It contains four bullet points in colored boxes:

- Today, access to the Web is not limited to only desktop systems, but is also available on portable and wireless devices, such as mobile devices.
- A mobile device, also known as a handheld device, is a small portable computing device with a small display screen and keyboard.
- A mobile device has an operating system on which various types of application software are executed.
- These application software are also known as apps.
- The most commonly used apps are mobile browsers that display the Web pages.

At the bottom left is the copyright notice '© Aptech Ltd.' and at the bottom right is 'Building a Mobile Web Application / Session 16 3'

Using slide 3 explain mobile application environment.

Today, mobile devices such as smartphones and tablets are equipped with browsers and network access that provide a better Web experience to their users.

As the number of mobile users has increased, so the need for a mobile Web experience has increased, that is identical to a desktop user experience. In other words, mobile users now look for applications targeting their mobiles that are similar to the ones on their desktops. This has led to the emergence of mobile Web application development.

Different companies follow distinctive strategies for developing mobile applications depending on the need of their users.

Then, explain the students about mobile devices and their apps as mentioned on slide 3.

#### Tips:

Apps are referred to as stand-alone programs designed for specific devices, platforms, or operating systems.

## Slides 4 to 6

Let us understand types of mobile devices.

**HTML 5 Types of Mobile Devices 1-6**

- The different categories of mobile phones available in the market are as follows:

**➤ Basic Mobile Devices**

- Very basic models with only call and Short Message Service (SMS) facility.
- Do not provide support for Web browsers or network access.

**➤ Low-end Mobile Devices**

- Provide more features than a basic mobile device, typically Web support
- Preferred by users who do not need heavy Internet usage.
- Include a basic camera and a basic music player.
- Manufacturers, such as Nokia, Motorola, Sony Ericsson, Samsung, and so forth have gained popularity for offering low cost handsets in the global market.

© Aptech Ltd. Building a Mobile Web Application / Session 16 4

**HTML 5 Types of Mobile Devices 2-6**

- Following figure shows the low-end mobile devices available from different manufacturers.



**➤ Mid-end Mobile Devices**

- These types of mobile devices have gained popularity due to their increased user experience and moderate cost.
- Some key features of these devices include: medium sized-screen, HTML supported browser, a decent camera, games, and support for applications.
- They have a proprietary Operating System (OS) that is not well-known and is also not portable across various platforms.

© Aptech Ltd. Building a Mobile Web Application / Session 16 5

**HTML 5 Types of Mobile Devices 3-6**

- Following figure shows the mid-end mobile devices available from different manufacturers.



**➤ High-end Mobile Devices**

- These types of mobile devices have advanced features, such as an accelerometer, advanced camera features, and Bluetooth.
- They have a better look and feel as compared to mid-end mobile devices.

© Aptech Ltd. Building a Mobile Web Application / Session 16 6

Using slides 4 to 6, explain the different types of mobile devices.

Explain the different types of mobile device models available in the market based on their features, manufacturers, and cost.

Tell the students that with the advent of mobile devices, the format and designing of Web sites is changing keeping users in the mind.

### Slides 7 to 9

Let us understand the features of smartphones and tablets.

**HTML 5 Types of Mobile Devices 4-6**

- Following figure shows the high-end mobile devices.



**➤ Smartphones**

- These are mobile devices with multitasking capabilities.
- These devices have a full browser support similar to desktop browsers with wireless LAN and 3G connection.

© Aptech Ltd. Building a Mobile Web Application / Session 16 7

**HTML 5 Types of Mobile Devices 5-6**

- They have several advanced features that are as follows:
  - Digital Compass
  - Global Positioning System (GPS)
  - Touch screen
  - Camera with video recording
  - TV out
  - Bluetooth
  - Accelerometer
- Following figure shows the various smartphone devices.



© Aptech Ltd. Building a Mobile Web Application / Session 16 8

**HTML 5 Types of Mobile Devices 6-6**

**➤ Tablets and Notebooks**

- These devices are larger than mobile phones.
- They are mobile computers with a touch screen virtual keyboard and stylus or digital pen.
- Features of tablets include: multi-touch display, better user experience, high quality screen resolution, better Web support, and multitasking OS with high speed.
- Some of the tablets available in the market are BlackBerry PlayBook Tablet PC, Samsung Galaxy Tab, and HCL Me Tab.
- Following figure shows different types of tablet devices.



© Aptech Ltd. Building a Mobile Web Application / Session 16 9

Using slides 7 to 9, explain different features of smartphones, notebooks, and tablets devices.

Tell the students that the main concern of the Web designer should be the seamless access to the Web pages on the smartphones and tablet devices.

### Slides 10 to 12

Let us understand the mobile platforms.

**HTML 5 Mobile Platforms 1-3**

- A mobile device platform is similar to a software platform.
- It is basically responsible to interact with the device hardware and run software/services on the mobile device.
- The mobile platforms are categorized as proprietary and open source.
- Proprietary platforms are those which are designed and developed by the mobile device manufacturers.
- These platforms are developed for specific devices and are not supported on all platforms.
- Open source platforms are those which are freely available to the users.
- The users can download the source code and alter them as per their requirements.

© Aptech Ltd. Building a Mobile Web Application / Session 16 10

**HTML 5 Mobile Platforms 2-3**

- The brief description of the platforms available on mobile devices is as follows:

<b>Palm OS</b>	<ul style="list-style-type: none"> <li>It is a proprietary mobile OS developed by Palm Inc. and was used for Personal Digital Assistants (PDAs).</li> <li>Currently, Palm Inc. has developed webOS, which is based on the Linux kernel.</li> </ul>
<b>Blackberry OS</b>	<ul style="list-style-type: none"> <li>It is a proprietary mobile OS developed by Research in Motion (RIM) and is based on Java platform.</li> <li>It is primarily used by Blackberry smartphone devices.</li> </ul>
<b>iOS</b>	<ul style="list-style-type: none"> <li>It is a mobile OS developed by Apple Inc. and was initially referred to as iPhone OS.</li> <li>It is derived from Mac OS X, which is based on the UNIX platform.</li> </ul>
<b>Symbian</b>	<ul style="list-style-type: none"> <li>It is an open source mobile OS developed for mobile phones.</li> <li>It includes a user interface framework, libraries, and component tools.</li> </ul>

© Aptech Ltd. Building a Mobile Web Application / Session 16 11

**HTML 5 Mobile Platforms 3-3**

<b>Windows Mobile</b>	<ul style="list-style-type: none"> <li>It is a mobile OS that runs on top of the Windows Mobile platform.</li> </ul>
<b>Linux</b>	<ul style="list-style-type: none"> <li>It is an open source OS and is supported by smartphones that are manufactured by Motorola.</li> </ul>
<b>Android</b>	<ul style="list-style-type: none"> <li>Android is an open source OS developed by Google. It is currently used by smartphones and tablet computers.</li> <li>It is a mobile OS developed by Apple Inc. and was initially referred to as iPhone OS.</li> </ul>

© Aptech Ltd. Building a Mobile Web Application / Session 16 12

Using slides 10 to 12, explain different types of mobile platforms and their features.

**Tips:**

Refer the link <http://mobileplatforms.wikidot.com/> to get more information on mobile platforms.

**In-Class Question:**

After you finish explaining different mobile platforms, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What are the features of Android OS?

**Answer:**

Android is a free and open-source OS, developed by Google. It can be customized by device manufacturers and is currently used by many smartphones and tablet devices.

## Slides 13 to 18

Let us understand the design aspects of mobile Web site.

**HTML 5 Design Aspects of Mobile Web Site 1-6**

- Some of the basic considerations needed for designing a Web sites for intended mobile device are as follows:
  - Resolution and Physical Dimension
  - Page Orientation
  - Input methods

**➤ Resolution and Physical Dimension**

- The resolution means the number of pixels (width and height) on the screen of the mobile device.
- Following table lists the resolutions of mobile devices based on their categories.

Category	Resolutions (in pixels)
Low-end mobile devices	128 x 160 or 128 x 128
Mid-end mobile devices	176 x 220 or 176 x 208
High-end devices	240 x 320
Smartphones	240 x 480, 480 x 320, 640 x 480, or 960 x 640

© Aptech Ltd. Building a Mobile Web Application / Session 16 13

**HTML 5 Design Aspects of Mobile Web Site 2-6**

The resolution of mobile devices is measured in terms of the physical dimensions of the screen.

The screen dimensions are either measured diagonally in terms of inches/centimeters or in terms of width and height.

The relation between the physical dimension and resolution is termed as Pixels per Inch (PPI) or Dots per Inch (DPI).

The higher DPI results in good print-quality graphics on the mobile device.

© Aptech Ltd. Building a Mobile Web Application / Session 16 14

**HTML 5 Design Aspects of Mobile Web Site 3-6**

- Following table lists the resolution and display sizes of different mobile devices.

Manufacturer	Model	Screen Size	Resolution (in pixels)	Type
Apple	iPad3	9.7"	2048x1536	Tablet
Apple	iPhone 3GS	3.5"	480x320	Smartphone
Apple	iPhone 4S	3.5"	960x640	Smartphone
Blackberry	Torch 9810	3.2"	640x480	Smartphone
HP	Touchpad	9.7"	768x1024	Tablet
Samsung	Galaxy S 4G	4"	480x800	Smartphone
Samsung	Galaxy S II	4.52"	800x480	Smartphone
Nokia	Lumia 800	3.7"	480x800	Smartphone

© Aptech Ltd. Building a Mobile Web Application / Session 16 15

**HTML5 Design Aspects of Mobile Web Site 4-6**

### ➤ Page Orientation

- The mobile devices are also categorized based on their orientation, vertical and horizontal.
- The vertical orientation devices are also referred to as portrait devices with taller display.
- Similarly, the horizontal orientation devices are referred as landscape devices with wider display.
- Following figure shows the mobile devices with vertical and horizontal screens.



© Aptech Ltd. Building a Mobile Web Application / Session 16 16

**HTML5 Design Aspects of Mobile Web Site 5-6**

Smartphones and tablets can switch between landscape and portrait views to present the better viewing of a Web page.

This rotation capability of changing the view from landscape to portrait or vice-versa is due to the hardware accelerators available in the phones.

A mobile Web site must be aware of these rotations and should provide a good user experience in both the orientations.

### ➤ Input Methods

- Some of the possible input methods for a mobile device are as follows:
  - Numeric keypad
  - Alphanumeric keypad (Simple or QWERTY)
  - Virtual keypad on screen
  - Multi-touch
  - External keypad
  - Voice and handwriting recognition

© Aptech Ltd. Building a Mobile Web Application / Session 16 17

**HTML5 Design Aspects of Mobile Web Site 6-6**

- Following figure shows a mobile device with QWERTY keyboard and touch screen.



© Aptech Ltd. Building a Mobile Web Application / Session 16 18

Using slides 13 to 18, explain the design aspects of mobile Web site.

The design process of a mobile Web site is similar to that of a traditional Web site developed for desktop Web browsers. Still, there are some differences between them that need to be taken into consideration.

An ideal mobile Web site is supported as well as rendered properly by maximum possible browsers and operating system.

Some of the basic considerations needed for designing Web sites for intended mobile device are as follows:

- **Resolution and Physical Dimension** - The resolution means the number of pixels (width and height) on the screen of the mobile device. As there are no standards defined for screen resolution, it varies depending on its model and manufacturer.
- **Page Orientation** - The mobile devices are also categorized based on their orientation, vertical and horizontal. The vertical orientation devices are also referred to as portrait devices with taller display. Similarly, the horizontal orientation devices are referred as landscape devices with wider display.
- **Input Methods** - There are different methods to input data on the mobile devices. A mobile device can support more than one input method.

#### In-Class Question:

After you finish explaining design aspects of mobile Web site, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which device takes care of the screen rotation while changing the view from landscape to portrait and vice-versa?

#### Answer:

Hardware accelerators availability on the phones

## Slides 19 to 23

Let us understand the architectural aspects.

**HTML 5 Architectural Aspects 1-5**

- The Web site developed for a mobile device is a collection of Web pages.
- Thus, it is essential to understand a few architectural concepts that can help to create meaningful mobile services.
- Some of the concepts that relate to its architecture are as follows:

**> Navigation**

- Navigation is the path followed by a user to travel in a Web site.
- As compared to the navigation tree of a desktop site, almost 80% of the information of a desktop site will not be useful to a mobile Web site.
- Thus, the main focus should be on 20%.

Design Web pages based on the use cases.

Arrange Web pages depending on the frequent requirements of the mobile users.

Restrict the depth of a mobile page to three clicks for a specific use case.

© Aptech Ltd. Building a Mobile Web Application / Session 16 19

**HTML 5 Architectural Aspects 2-5**

Design minimum input controls for the form pages.

Desktop Web site normally has a welcome screen. In case of mobile Web sites, avoid developing welcome screens.

While designing a service, decide its usability.

Approximate the number of mobile pages required to separate services, after the home page.

**> Perspective**

- The perspective of a mobile user is different from a desktop user in terms of needs and accessibility.
- Hence, a user-centric design approach should be followed for designing mobile Web sites.
- This ensures that a user completes the task easily and successfully.

© Aptech Ltd. Building a Mobile Web Application / Session 16 20

**HTML 5 Architectural Aspects 3-5**

Some of the possible users' contexts are as follows:

What is the location of the user?

Why is a mobile Web site accessed by the user?

What are the needs of the user?

What solution is offered by a mobile application to solve the user's problem?

Where is the user present while accessing a Web site?

**> Enhancement**

- Enhancement is a simple and powerful technique that can be adopted while designing a mobile Web site.
- This technique defines compatibility of Web site and allows access to basic content, services, and functionality on all type of mobile devices.
- Also, it provides a better Web experience on devices with higher standards.

© Aptech Ltd. Building a Mobile Web Application / Session 16 21

**HTML 5 Architectural Aspects 4-5**

Some of the core principles for enhancing mobile Web sites are as follows:

- Basic content and functionality are accessible in all browsers.
- Enhanced layout and behavior must be provided through external style sheets and JavaScript that are linked with the Web pages.
- Markup elements used on the pages must have proper semantic.
- Web browser settings on a user's device should be considered.

➤ **Use of Web Standards**

- The Web standards, such as HTML, CSS, and JavaScript followed in the mobile Web site design must be correctly used.
- This increases the possibility of displaying pages on large number of devices.
- The well-formedness of the markup tags used on a page can be achieved by validating them.
- Also, the use of certain HTML elements can be avoided while designing the Web pages for mobile devices.

© Aptech Ltd. Building a Mobile Web Application / Session 16 22

**HTML 5 Architectural Aspects 5-5**

• The brief description of these elements is as follows:

- Use of HTML tables**
  - As the screen size of mobile devices is small, so the use of tables in layouts should be avoided.
  - It makes the scrolling difficult and also slows down the page loading in the browser.
- Pop-up windows**
  - The Web sites with pop-up windows makes the site impractical to work with.
  - Also, all mobile browsers do not provide support for them.
- Use of graphics**
  - The use of graphics increases download time of the pages.
  - Also, they can obstruct the layout of the old mobile browsers, resulting in incorrect display of the page.
- Use of frames**
  - Many mobile devices do not provide the support for frames due to usability problems.
  - Also, the HTML5 new specification does not provide the support for frames.

© Aptech Ltd. Building a Mobile Web Application / Session 16 23

Using slides 19 to 23, explain the architectural aspects of developing mobile apps.

The architecture for designing a mobile Web site begins with the understanding of the information and service that is offered to the mobile user.

Explain the concepts that relate to the mobile architecture, such as:

- Navigation
- Perspective
- Enhancement
- Use of Web Standards

### Slides 24 and 25

Let us understand the process of setting up environment for developing mobile apps.

**Setting Up the Environment 1-2**

- Mobile Web applications are developed to be run on different mobile devices.
- Hence, they need to be tested in several different environments.
- The tools required to develop a mobile Web application are namely, Integrated Development Environment (IDE) and emulators.
- These are described as follows:

**IDE**

- An IDE is a tool used for coding the markup, JavaScript, and CSS.
- Some of these tools are as follows:
  - Adobe Dreamweaver
  - Microsoft Expression Web
  - Aptana Studio
  - Eclipse
  - Editplus (text editor)
- Latest versions of these tools provide better support for mobile markups.
- They also provide support for validating pages against mobile Web standards.

© Aptech Ltd. Building a Mobile Web Application / Session 16 24

**Setting Up the Environment 2-2**

**Emulators**

- An emulator is a software that translates the compiled code to the native platform on which the application is executed.
- The emulator runs as a desktop application that allows testing and debugging of a mobile application.
- It offers the environment similar to a real mobile device on which an application will be executed.
- Emulators are developed by manufacturers and are often offered free to users.
- They are either standalone applications or bundled with a Software Development Kit (SDK) for native development.
- Some of the popular emulators are as follows:
  - Android
  - iOS
  - webOS
  - Blackberry
  - Windows Phone
  - Opera Mobile

© Aptech Ltd. Building a Mobile Web Application / Session 16 25

Using slides 24 and 25, explain the process of setting up environment for developing mobile apps.

Desktop Web applications are created and tested in the environment for which they are developed. Mobile Web applications are developed to be run on different mobile devices. Hence, they need to be tested in several different environments. The tools required to develop a mobile Web application are namely, Integrated Development Environment (IDE) and emulators.

Explain that an IDE is a tool used for coding the markup, JavaScript, and CSS. Today, in the market, there are different tools which provide the facility to easily build a mobile Web application. Some of these tools are as follows:

- Adobe Dreamweaver
- Microsoft Expression Web
- Aptana Studio

- Eclipse
- EditPlus (text editor)

Latest versions of these tools provide better support for mobile markups. They also provide support for validating pages against mobile Web standards.

Explain emulators using slide 25. The testing of a mobile Web application can be done using an emulator. An emulator is software that translates the compiled code to the native platform on which the application is executed.

Emulators are developed by manufacturers and are often offered free to users. They are either standalone applications or bundled with a Software Development Kit (SDK) for native development. Some of the popular emulators that either run as standalone applications or in an SDK are as follows:

- Android
- iOS
- webOS
- Blackberry
- Windows Phone
- Opera Mobile

**In-Class Question:**

After you finish explaining setting up environment for developing mobile Web site, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What is an emulator?

**Answer:**

An emulator is software that translates the compiled code to the native platform on which the application is executed.

## Slide 26

Let us understand HTML supports on mobiles.

**HTML5** **HTML Support on Mobiles**

- Today, majority of smartphones and tablets are providing good support for HTML5.
- Most Android and iOS mobile devices as well as tablets use browsers that are based on Webkit.
- The Webkit is a layout engine supported by browsers, such as Google Chrome and Apple Safari to render Web pages.
- The features suited for mobile devices are as follows:

The features suited for mobile devices are as follows:

- Video
- Audio
- Drag and drop
- Accessing browser history
- Geolocation API for accessing location
- Web storage API to save data on mobile devices
- Offline Web applications (Applications with no Internet connection)

© Aptech Ltd. Building a Mobile Web Application / Session 16 26

Using slide 26, explain HTML5 support for mobile devices.

### In-Class Question:

After you finish explaining how HTML5 supports mobile devices, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



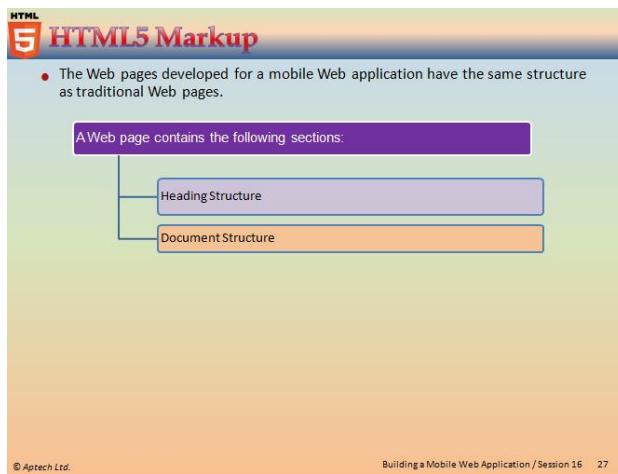
Which layout engine is supported by Google chrome and Apple Safari to render Web pages?

### Answer:

Webkit layout engine

## Slide 27

Let us understand the HTML5 markup.



Using slide 27, explain the HTML5 markup.

The Web pages developed for a mobile Web application have the same structure as traditional Web pages. A Web page contains the following sections:

- Heading Structure
- Document Structure

## Slides 28 to 35

Let us understand the heading structure.

The diagram focuses on the "Meta Tag" section of the heading structure. It lists three points about the <meta> tag:
 

- The heading structure is represented by a <head> element defined in an HTML Web page.
- It defines a <meta> tag that is used specifically for mobile browsers.
- The brief description for some of the tags defined under element is as follows:

**➤ Meta Tag**

A <meta> tag indicate that the document is optimized for mobile devices and are used to control the display scale, while displaying HTML content on the device.

It is specific to mobile browsers.

Following table lists some of the variations of <meta> tag used for different mobile browsers.

Meta Tag	Description	Supported Mobile Browser
<meta name="HandheldFriendly" content="true"/>	Indicates that the content is designed for small-screen handheld devices.	BlackBerry and others
<meta name="MobileOptimized" content="width" />	Accept width (in pixels) to place the content and forces the layout to one column in the browser	Windows Mobile and Windows Phone

**HTML 5 Heading Structure 2-8**

Meta Tag	Description	Supported Mobile Browser
<meta name="Apple-mobile-web-app-capable" content="yes"/>	Indicates that the Web application will run in a full-screen mode.	Safari
<meta name="Format-detection" content="telephone=no"/>	Automatic detection of phone numbers is enabled or disabled on Web pages.	Safari running on iOS

- Similarly, a non-standard variation of <meta> tag is specified by giving an alternate <link> tag.
- This tag is mostly used with desktop Web pages and defines an alternative URL for displaying the same content on different medias, such as handheld devices.

```
<link rel="alternate" media="handheld" href="http://mysite.com" />
```

© Aptech Ltd.

Building a Mobile Web Application / Session 16 29

**HTML 5 Heading Structure 3-8**

### Viewport Meta Tag

- This is a new technique used to inform the browser that the Web page is optimized for a mobile device.
- A viewport is the rectangular display area on the screen, where the content of a Web page are displayed by the browser.
- It contains attributes, such as width and height that can be set to larger or smaller values depending on the total visible area on the screen.
- Following table lists the attributes of viewport meta tag.

Attribute	Description	Value
width	Defines the horizontal size of the viewport in pixels	Integer value (in pixels) or constant device-width
height	Defines the vertical size of the viewport in pixels	Integer value (in pixels) or constant device-height
initial-scale	Sets the scale of the page for its initial display	Floating value between 0.1 to n

© Aptech Ltd.

Building a Mobile Web Application / Session 16 30

**HTML 5 Heading Structure 4-8**

Attribute	Description	Value
minimum-scale	Defines the minimum zoom scale of the viewport	Floating value between 0.1 to n
maximum-scale	Defines the maximum zoom scale of the viewport	Floating value between 0.1 to n
user-scalable	Allows scaling of application on the mobile devices. That is, users can zoom in and out in the application	no or yes

- The Code Snippet demonstrates the viewport meta tag to set the device width for a mobile Web page.

```
<!DOCTYPE html>
<head>
  <title>Mobile</title>
  <!-- <meta name="viewport" content="width=device-width,
       user-scalable=no"> -->
</head>
```

© Aptech Ltd.

Building a Mobile Web Application / Session 16 31

**HTML5 Heading Structure 5-8**

```

<body>
  <header> Mobile Design </header>
  <NAV>
    <a href="home.html">Home</a> | <a href="aboutUs.html">
      About Us</a> |
      <a href="contactUs.html">Contact Us</a>
    </NAV>
  <section id="intro">

    <p>This is the introductory text to my mobile Web
      application.
    </p>

    <p> Mobile development is more than cross-browser, it
      should be cross-platform. The vast number of mobile
      devices makes thorough testing a practical
      impossibility, leaving developers nostalgic for the
      days when they only had to support legacy browsers.
    </p>

  </section>
</body>
</html>

```

© Aptech Ltd. Building a Mobile Web Application / Session 16 32

**HTML5 Heading Structure 6-8**

- In the code, the viewport width has been set to “device-width” which sets the device width to 320px.
- As all phones does not support the same width, so setting “device-width” allows the mobile browsers to set the width according to the device width.
- Also, setting the attribute user-scalable=no prevents the user from increasing the display scale of the application.
- The default width taken for the viewport is 980px which is approximately the desktop size.
- Following figure displays the Web page on Opera Mobile Emulator before setting the viewport meta tag.

© Aptech Ltd. Building a Mobile Web Application / Session 16 33

**HTML5 Heading Structure 7-8**

- Following figure displays the Web page on **Opera Mobile Emulator** after removing the comments from the code.
- The code sets the viewport meta tag.

© Aptech Ltd. Building a Mobile Web Application / Session 16 34

**Title Tag**

- Apart from `<meta>` tag, `<head>` element also contains a `<title>` tag.
- The text selected for `<title>` tag should be meaningful, short, and precise.
- It should be between four and eight words, as some old mobile devices truncate the long titles after 10 or 12 words.

**Icons**

- To add icons to a mobile Web page, images in or format could be used.
- These formats are compatible with mobile devices, as they are easy to export and are optimized in size.
- For example,

```
<link rel="icon" type="image/png" href="mobile.png" />
```

- From HTML5 onwards, Android supports the `apple-touch-icon-precomposed` meta tag in order to display high-resolution icons.

Using slides 28 to 35, explain the heading structure.

The heading structure is represented by a `<head>` element defined in an HTML Web page. It provides information about the Web page such as keywords and language used, which is not displayed on the Web page. Keywords are important terms existing in a Web page used by the search engines to identify the Web page with respect to the search criterion.

It defines a `<meta>` tag that is used specifically for mobile browsers.

A `<meta>` tag has two factors that need to be considered while designing a mobile Web application are its initial display (zoom) scale and orientation. Thus, it is necessary to inform mobile browsers to consider these factors while displaying a Web page.

A `<meta>` tag indicates that the document is optimized for mobile devices and is used to control the display scale, while displaying HTML content on the device. It is specific to mobile browsers.

Explain some of the variations of `<meta>` tag used for different mobile browsers listed on slides 28 and 29.

Similarly, a non-standard variation of `<meta>` tag is specified by giving an alternate `<link>` tag. This tag is mostly used with desktop Web pages and defines an alternative URL for displaying the same content on different media such as handheld devices.

```
<link rel="alternate" media="handheld" href="http://mysite.com" />
```

Then, explain viewport meta tag using slide 30. It is supported on many smartphones, such as iPhone, Android based phones, and browsers such as Internet Explorer Mobile, Opera Mini, and Opera Mobile.

A viewport is the rectangular display area on the screen, where the content of a Web page are displayed by the browser. It contains attributes, such as width and height that can be set to larger or smaller values depending on the total visible area on the screen.

Explain the syntax of viewport tag: `<meta name="viewport" content="width=device-width, user-scalable=no">`. Then, explain the list of attributes of the viewport meta tag.

Explain code snippet on slides 31 and 32 which demonstrates the viewport meta tag, to set device width for a mobile Web page.

Figure on slide 33 displays the Web page on Opera Mobile Emulator, before setting the viewport meta tag.

Figure on slide 34 displays the Web page on Opera Mobile Emulator, after removing the comments from code snippet. The code sets the viewport meta tag.

**Tips:**

You can go through this link to get the viewport sizes for various mobiles and tablets:  
<http://i-skool.co.uk/mobile-development/web-design-for-mobiles-and-tablets-viewport-sizes/>

Explain the `<title>` tag should be meaningful, short, and precise. It should be between four to eight words, as some old mobile devices truncate the long titles after 10 to 12 words.

Icons are used to add images of type GIF or PNG format on the mobile Web page. These formats are compatible with mobile devices, as they are easy to export and are optimized in size.

For example, `<link rel="icon" type="image/png" href="mobile.png" />`

From HTML5 onwards, Android supports the `rel="apple-touch-icon-precomposed"` and iOS supports the `rel="apple-touch-icon"` meta tag in order to display high-resolution icons. These create a flashy app-like icon on the user's home screen when they bookmark your site.

## Slides 36 to 38

Let us understand how to design layouts and images for mobiles.

**HTML 5 Document Structure 1-10**

- The document structure is represented by a <body> element in the HTML Web page.
- The <body> element of a mobile Web application defines the content that are displayed to the user.
- Some of the elements used in the <body> element of a mobile Web page are as follows:

**> Layouts**

- The HTML5 new tags that provide semantics for the layout of an HTML document are as follows:
  - <article> - An independent portion of the document or site
  - <aside> - Content that is tangential to the main part of the page or site
  - <figcaption> - Caption for a figure
  - <figure> - A figure or quotation pulled out of the flow of text
  - <footer> - The footer of a document or section
  - <header> - The header of a document or section
  - <hgroup> - A group of headings
  - <nav> - A navigation section
  - <section> - Identifies a block of content

© Aptech Ltd. Building a Mobile Web Application / Session 16 36

**HTML 5 Document Structure 2-10**

**> Images**

- Images can be used in mobile Web applications for pictorial representation.
- Almost all mobile browsers understand formats, such as GIF, JPEG, and PNG.
- The <img> tag is used to display image on a Web page.
- The attributes of <img> tag, such as width, height, and alt should be specified, as it reduces the rendering time of the image.
- The Code Snippet demonstrates a mobile Web page with an <img> tag.

```
<!DOCTYPE html>
<head>
<title>Images</title>
<meta name="viewport" content="width=device-width,
user-scalable=no"/>
</head>
<body>
<article>
<h2> Gift Basket </h2>

</article>
</body>
</html>
```

© Aptech Ltd. Building a Mobile Web Application / Session 16 37

**HTML 5 Document Structure 3-10**

- Following figure displays image on a mobile Web page.

A screenshot of a mobile web browser window titled "Opera Mobile - Samsung Galaxy S II". The address bar shows "file://localhost/H/". The main content area displays a heading "Gift Basket" above an image of a gift basket filled with various items. Below the image are standard mobile browser navigation buttons (back, forward, refresh, etc.). At the bottom of the screen, there is a status bar with the text "Samsung Galaxy S II" and "320x480 PPI 256".

© Aptech Ltd. Building a Mobile Web Application / Session 16 38

Using slides 36 to 38, explain the document structure and placing of images on the mobile Web page design.

The document structure is represented by a <body> element in an HTML Web page. The <body> element of a mobile Web application defines the content that is displayed to the user. Some of the elements used in the <body> element of a mobile Web page are as follows:

## Layouts

Tell the students that in the earlier sessions, they have learnt about the HTML5 new tags that provide semantics for the layout of an HTML document. Describe the tags listed on slide 36 which are used to design the layout of the mobile application.

## Images

After explaining the image formats, explain them that Images should not be used for setting background, buttons, links, or presenting titles on a mobile Web page. This is because images increase the number of request to the Web server and also load time of the Web page. The <img> tag is used to display image on a Web page.

Also, the attributes of <img> tag such as width, height, and alt should be specified, as it reduces the rendering time of the image.

Finally, explain the code snippet which demonstrates a mobile Web page with an <img> tag.

## Slides 39 to 41

Let us understand the use of lists in mobile Web pages.

The screenshot shows a presentation slide with the following content:

- HTML5 Document Structure 4-10**
- > Lists**
- A mobile Web application supports different types of lists.
- The various list types are as follows:
  - Ordered lists**
    - Used for navigational menus and are defined using <ol> tag on a Web page.
  - Unordered lists**
    - Used for presenting objects of same type and are defined using <ul> tag on a Web page.
  - Definition lists**
    - Used for presenting information as key/value pairs and are defined using <dl> tag on a Web page.

© Aptech Ltd. Building a Mobile Web Application / Session 16 39

The screenshot shows a mobile web application titled "Document Structure 5-10". The page displays the following HTML code:

```
<!DOCTYPE html>
<head>
    <title>List on Mobile</title>
    <meta name="viewport" content="width=device-width, user-scalable=no"/>
    <link rel="apple-touch-icon-precomposed" href="mobile.png" />
</head>
<body>
    <header>
        <h5> Hardware Components Capacity List </h5>
    </header>
    <section>
```

Below the code, there is a note: "The Code Snippet demonstrates the use of a definition list to present the capacity list of different hardware components on a mobile Web page."

At the bottom of the slide, it says "© Aptech Ltd." and "Building a Mobile Web Application / Session 16 40".

**Slide 40 (a)**

The screenshot shows a mobile web application titled "Document Structure 5-10". The page displays the following HTML code:

```
<dl>
    <dt>RAM Memory</dt>
    <dd>4.00GB</dd>
    <dt>Hard Disk</dt>
    <dd>500GB</dd>
    <dt>LAN</dt>
    <dd>Wifi, Bluetooth</dd>
    <dt>CPU</dt>
    <dd>2.93GHz</dd>
</dl>
</section>
</body>
</html>
```

Below the code, there is a note: "In the code, the <dl> tag is used to represent the information as key and value pair format."

At the bottom of the slide, it says "© Aptech Ltd." and "Building a Mobile Web Application / Session 16 40".

**Slide 40 (b)**

The screenshot shows a mobile web application titled "Document Structure 6-10". The page displays a list of hardware components under the heading "Hardware Components List". The list includes:

- RAM Memory 4.00GB
- Hard Disk 500GB
- LAN Wifi, Bluetooth
- CPU 2.93GHz

The application is running on a Samsung Galaxy S II device, as indicated by the status bar at the bottom.

At the bottom of the slide, it says "© Aptech Ltd." and "Building a Mobile Web Application / Session 16 41".

Using slides 39 to 41, explain the use of lists in the mobile Web page design.

A mobile Web application supports different types of lists. The various list types are as follows:

- **Ordered lists** - Used for navigational menus and is defined using `<ol>` tag on a Web page.
- **Unordered lists** - Used for presenting objects of same type and are defined using `<ul>` tag on a Web page.
- **Definition lists** - Used for presenting information as key/value pairs and are defined using `<dl>` tag on a Web page.

Explain the code snippet which demonstrates the use of a definition list to present the capacity list of different hardware components on a mobile Web page.

In the code, the `<dl>` tag is used to represent the information as key and value pair format. Figure displays the definition list on a mobile Web page.

## Slides 42 to 45

Let us understand use of links in mobile Web pages.

**Document Structure 7-10**

➤ Links

- Hyperlinks are used to link pages in a Web application.
- A hyperlink is defined using `<a>` tag with `href` attribute.
- The `href` attribute is set to the URL of a resource.
- The `<a>` tag should also have `accesskey` attribute specified with it.
- The `accesskey` attribute is a keyboard shortcut and is useful for mobile devices that have support for access keys.
- The Code Snippet demonstrates the use of `<a>` and `<ul>` tag to create a navigation list on a mobile Web page.

```
<!DOCTYPE html>
<head>
<title>Navigation list</title>
<meta name="viewport" content="width=device-width, user-scalable=no"/>
<link rel="apple-touch-icon-precomposed" href="mobile.png" />
</head>
<body>
<header>
<h5> Main Menu </h5>
</header>
<NAV>
```

© Aptech Ltd. Building a Mobile Web Application / Session 16 42

Slide 42 (a)

**HTML Document Structure 7-10**

**Links**

- Hyperlinks are used to link pages in a Web application.
- A hyperlink is defined using `<a>` tag with `href` attribute.
- The `href` attribute is set to the URL of a resource.
- The `<a>` tag should also have `accesskey` attribute specified with it.
- The `accesskey` attribute is a keyboard shortcut and is useful for mobile devices that have support for access keys.
- The Code Snippet demonstrates the use of `<a>` and `<ul>` tag to create a navigation list on a mobile Web page.

```

<ul>
  <li><a title="Comprehensive Animation"
    href="comprehensive.html" accesskey="1">Comprehensive
    Animation Pro</a></li>
  <li><a title="2D and 3D" href="animation_3d.html"
    accesskey="2">2D & 3D Animation </a></li>
  <li><a title="3D Animation" href="animation_2d.html"
    accesskey="3">Animation & 3D</a></li>
</ul>
</NAV>
</body> </html>

```

© Aptech Ltd. Building a Mobile Web Application / Session 16 42

**Slide 42 (b)**

**HTML Document Structure 8-10**

- Following figure displays the navigation list on a mobile Web page.

**Links**

- As mobile devices are basically phones, hence, links can be created to perform phone call actions.
- This is achieved using the `tel:<phone number>` scheme embedded with a hyperlink.
- The `tel` scheme is useful in situations, such as accessing helpdesk systems or voicemail systems.

© Aptech Ltd. Building a Mobile Web Application / Session 16 43

**HTML Document Structure 9-10**

- The Code Snippet demonstrates a `tel` scheme defined on a mobile Web page.

```

<!DOCTYPE html>
<head>
  <title> Mobile Application </title>
  <meta name="viewport" content="width=device-width,minimum-
  scale=1.0,maximum-scale=1.0"/>
</head>
<body>
  <HEADER>
    <SECTION> <b> Animation |<br/>
      Multimedia </b>
    </SECTION>
    <NAV><ul>
      <li><a title="Comprehensive Animation"
        href="#">Comprehensive Animation Pro</a></li>
      <li><a title="2D and 3D" href="#">2D & 3D Animation
        </a></li>
      <li><a title="3D Animation" href="#">Animation & 3D</a>
        </li>
    </ul>
  </NAV>

```

© Aptech Ltd. Building a Mobile Web Application / Session 16 44

**Slide 44 (a)**

The screenshot shows a mobile web application titled "Document Structure 9-10". It displays an HTML code snippet:

```

</NAV>
</HEADER>
<SECTION id="intro">
<p> Arena Animation is leader in animation and multimedia education with the widest network of centers across the country. Over a span of 14 years, 250,000 students and professionals have, through Arena, found their calling in animation, graphics, print publishing, web designing & films.
</p>
</SECTION>
<FOOTER>
<p> <a href="tel:+91 22 2827 2300">Contact us </a> <br/>
Copyright © 2012 Aptech Ltd.</p>
</FOOTER>
</body>
</html>

```

At the bottom, it says "© Aptech Ltd." and "Building a Mobile Web Application / Session 16 44".

### Slide 44 (b)

The screenshot shows a mobile web application titled "Document Structure 10-10". It displays a screenshot of a tablet device showing a website for "Arena Animation". The website has a header "Arena Animation | Multimedia" and a list of services: "Comprehensive Animation Pro", "2D & 3D Animation", and "Animation & 3D". Below this is a paragraph about Arena Animation's history and services. At the bottom, there is a "Contact us" link and a copyright notice "Copyright © 2012 Aptech Ltd.". The screenshot is labeled "Samsung Galaxy Tab".

At the bottom, it says "© Aptech Ltd." and "Building a Mobile Web Application / Session 16 45".

Using slides 42 to 45, explain the use of links in the mobile Web page design.

Hyperlinks are used to link pages in a Web application. They are one of the important elements on Web pages and this holds true in case of mobile Web applications too. A hyperlink is defined using `<a>` tag with `href` attribute. The `href` attribute is set to the URL of a resource. Apart from `href`, the `<a>` tag should also have `accesskey` attribute specified with it. The `accesskey` attribute is a keyboard shortcut and is useful for mobile devices that have support for access keys. Code Snippet demonstrates the use of `<a>` and `<ul>` tag to create a navigation list on a mobile Web page.

Figure displays the navigation list on a mobile Web page.

Tel scheme as mobile devices are basically phones, hence, links can be created to perform phone call actions. This is achieved using the `tel:<phone number>` scheme embedded with a hyperlink. The `tel` scheme is useful in situations, such as accessing helpdesk systems or voicemail systems.

Explain the code snippet which demonstrates a `tel` scheme defined on a mobile Web page.

Figure on slide 45 displays the output of a Web page in a tablet selected in an Opera Mobile Emulator.

### In-Class Question:

After you finish explaining the use of the links on the mobile pages, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which attribute is useful for mobile devices that have support for access keys?

### Answer:

<li> tag is used to define the item in a list

## Slides 46 to 51

Let us understand CSS for mobile Web applications.

**HTML 5 CSS for Mobile 1-8**

- CSS3 provides properties for adding colors, selectors, borders, backgrounds, and so on for effective appearance of a Web page.

Most modern mobile browsers support following features of CSS3:

- Rounded corners
- Images with borders
- Adding shadow effect on text and boxes
- Animations
- Transitions
- Multi-column layout

© Aptech Ltd. Building a Mobile Web Application / Session 16 46

**HTML 5 CSS for Mobile 2-8**

- Modern browsers have provided new styles for CSS3 that are specific to each browser.
- To add these styles on a Web page, the relevant properties need to be prefixed with the browser specific keyword.
- The property prefixed with the keyword represents the browser on which it is supported.
- Following table lists the keywords with their supported browsers.

Keyword	Browser
-moz	Firefox
-ms	Internet Explorer
-o	Opera
-webkit	Google Chrome and Safari

© Aptech Ltd. Building a Mobile Web Application / Session 16 47

```

      ● The Code Snippet demonstrates the CSS3 properties for a Web page that was developed in the code.

      <!DOCTYPE html>
      <head>
        <title> Mobile Application </title>
        <meta name="viewport" content="width=device-width"/>
        <style>
          html, body {
            margin: 0;
            padding: 0;
            border: 0;
            font-size: 100%;
            font-weight: normal;
            vertical-align: baseline;
            background: transparent;
          }
          body {
            line-height: inherit;
          }
          #nav {
            width:500px;
            height:60px;
          }
        </style>
      </head>
      <body>
        <div id="nav"></div>
      </body>
    
```

© Aptech Ltd. Building a Mobile Web Application / Session 16 48

### Slide 48 (a)

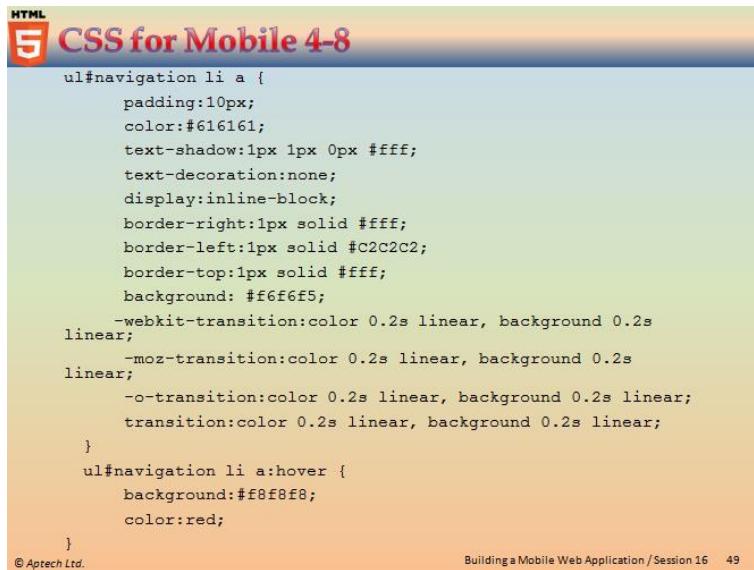
```

      ● The Code Snippet demonstrates the CSS3 properties for a Web page that was developed in the code.

      background-color:#A4A4FF;
      margin:0px;
      margin-top:5px;
      padding:0px;
    }
    ul#navigation {
      margin:0px;
      border-left:1px solid #c4bbe7;
      border-right:1px solid #c4dbe7;
      padding-top:5px;
    }
    ul#navigation li {
      display:inline;
      font-size:12px;
      font-weight:bold;
      margin:0;
      padding:0;
      float:left;
      position:relative;
      border-top:1px solid #c4dbe7;
      border-bottom:2px solid #c4dbe7; }
  
```

© Aptech Ltd. Building a Mobile Web Application / Session 16 48

### Slide 48 (b)



```

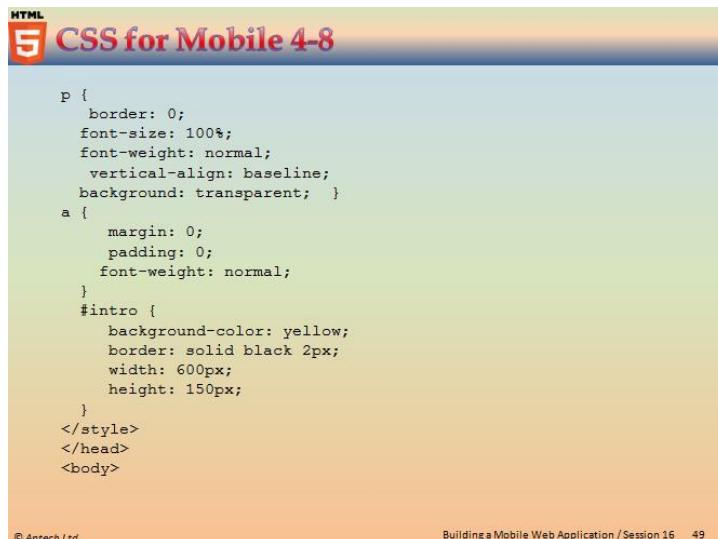
HTML CSS for Mobile 4-8

ul#navigation li a {
    padding:10px;
    color:#616161;
    text-shadow:1px 1px 0px #ffff;
    text-decoration:none;
    display:inline-block;
    border-right:1px solid #ffff;
    border-left:1px solid #c2c2c2;
    border-top:1px solid #ffff;
    background: #f6f6f5;
    -webkit-transition:color 0.2s linear, background 0.2s
linear;
    -moz-transition:color 0.2s linear, background 0.2s
linear;
    -o-transition:color 0.2s linear, background 0.2s linear;
    transition:color 0.2s linear, background 0.2s linear;
}
ul#navigation li a:hover {
    background:#f8f8f8;
    color:red;
}

© Aptech Ltd. Building a Mobile Web Application / Session 16 49

```

Slide 49 (a)



```

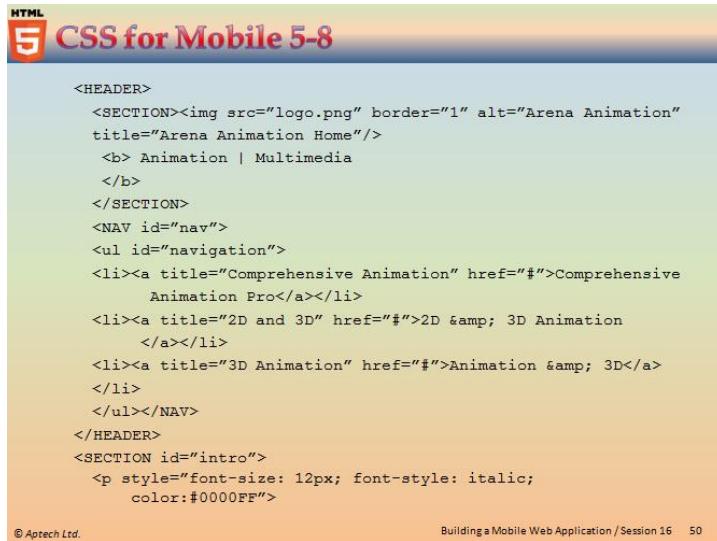
HTML CSS for Mobile 4-8

p {
    border: 0;
    font-size: 100%;
    font-weight: normal;
    vertical-align: baseline;
    background: transparent; }
a {
    margin: 0;
    padding: 0;
    font-weight: normal;
}
#intro {
    background-color: yellow;
    border: solid black 2px;
    width: 600px;
    height: 150px;
}
</style>
</head>
<body>

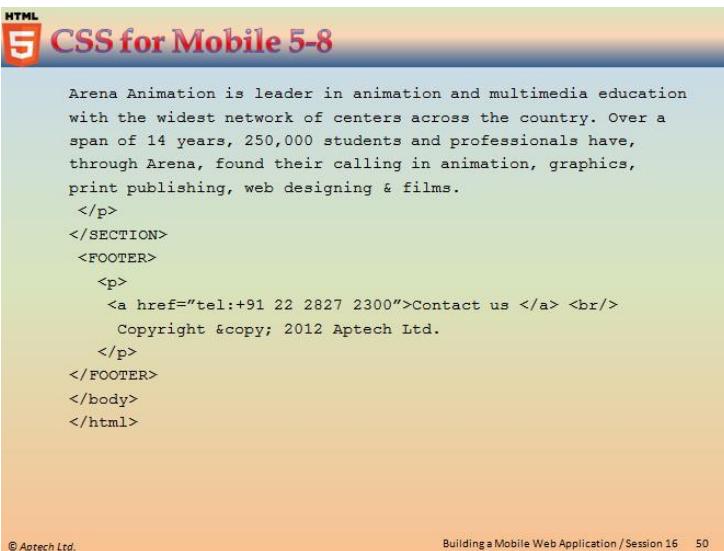
© Aptech Ltd. Building a Mobile Web Application / Session 16 49

```

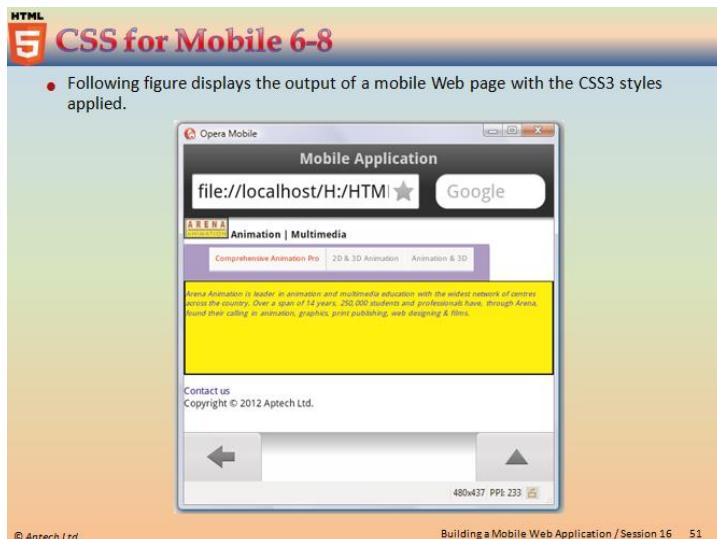
Slide 49 (b)



Slide 50 (a)



Slide 50 (b)



Using slides 46 to 51, explain the CSS for mobile.

CSS3 provides properties for adding colors, selectors, borders, backgrounds, and so on for effective appearance of a Web page. Most modern mobile browsers support following features of CSS3:

- Rounded corners
- Images with borders
- Adding shadow effect on text and boxes
- Animations
- Transitions
- Multi-column layout

Apart from W3C specifications that are laid for CSS3, modern browsers have provided new styles for CSS3. These styles are specific to each browser. Therefore, to add these styles on a Web page, the relevant properties need to be prefixed with the browser specific keyword. The property prefixed with the keyword represents the browser on which it is supported. Table lists the keywords with their supported browsers.

Explain the code snippet using slides 48 to 50 which demonstrates the CSS3 properties. These properties have been explored in the earlier sessions. Then, explain the figure in slide 51 that displays the output of a mobile Web page with the CSS3 styles applied.

## Slides 52 and 53

Let us understand media queries in CSS.

The screenshot shows a presentation slide with the following content:

- Title:** CSS for Mobile 7-8
- Section:** Media Queries for Browser Detection
- List:**
  - Media queries are used to target specific features, such as screen width, orientation, and resolution of the devices.
  - The use of a media query is to display HTML pages on various devices, such as computers and mobile devices with different styles based on their media types.
  - In media queries, expressions are added for specific media type, then checking for condition is done, and finally, respective style sheet is applied to a Web page.
- Text Box:** Media queries are used in two ways that are as follows:
  - Inline within a CSS style sheet
  - In the <link> tag as "media" attribute

The Code Snippet shows the markup to apply a style sheet named `screen.css` to a device with screen and set the viewing-width of the area to 480.

```
<link type="text/css" rel="stylesheet" media="only screen and (max-device-width: 480px)" href="screen.css" />
```

The Code Snippet shows the code to change the background color of a Web page depending on the device width.

```
@media only screen and (max-device-width: 480px) {  
    body {  
        background-color: #666;  
    }  
}
```

The Code Snippet shows the markups to serve style sheets based on the orientation of the device.

```
<link rel="stylesheet" media="all and (orientation: portrait)" href="portrait_orientation.css" />  
<link rel="stylesheet" media="all and (orientation: landscape)" href="landscape_orientation.css" />
```

© Aptech Ltd. Building a Mobile Web Application / Session 16 53

Using slides 52 and 53, explain media queries for browser detection.

Explain the students with a scenario which describes the use of properties that are used to display a Web page for a particular media which can be a screen or print. To provide the CSS for different media, media queries are used.

Media queries are used to target specific features such as screen width, orientation, and resolution of the devices. The use of a media query is to display HTML pages on various devices, such as computers and mobile devices with different styles based on their media types.

Inform the students that earlier CSS2 allowed creating style sheets for specific media types, such as screen and print. However, CSS3 has been enhanced further with the features of media queries.

Media queries are used in two ways that are as follows:

- Inline within a CSS style sheet
- In the `<link>` tag as “`media`” attribute

Explain the code snippet which shows the markup to apply a style sheet named `screen.css` to a device with screen and set the viewing-width of the area to 480. Some browsers can also interpret the CSS media queries placed inside a `.css` file.

Explain the code snippet which shows the code to change the background color of a Web page depending on the device width.

CSS media queries can also be used to define style sheets for different orientations.

Explain the code snippet in slide 53 which shows the markups to serve style sheets based on the orientation of the device.

**Tips:**

Some of the media types other than screen and print are as follows:

all	Used for all media type devices
aural	Used for speech and sound synthesizers
braille	Used for braille tactile feedback devices
embossed	Used for paged braille printers
handheld	Used for small or handheld devices
print	Used for printers
projection	Used for projected presentations, like slides
screen	Used for computer screens
tty	Used for media using a fixed-pitch character grid, such as teletypes and terminals
tv	Used for television-type devices

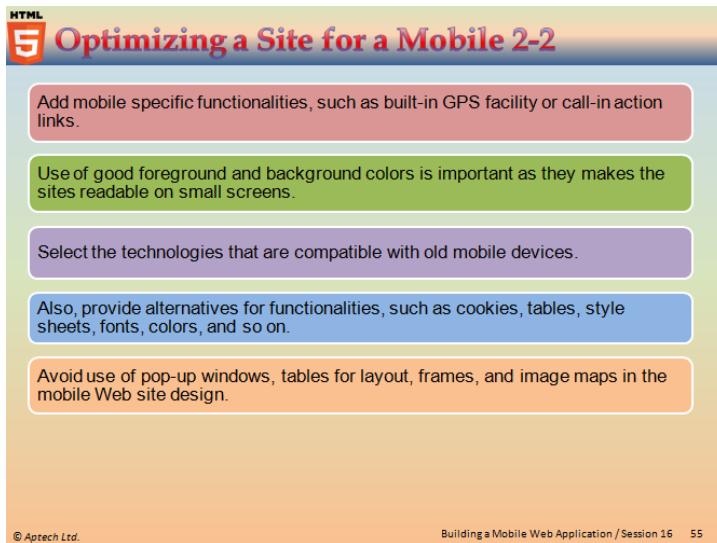
**Slides 54 and 55**

Let us understand how to optimize a Site for a mobile.

The slide has a header 'HTML5 Optimizing a Site for a Mobile 1-2'. Below the header is a bulleted list: 'Mobile Web sites should be optimized for better performance.' and 'Some of the best practices that can be followed for mobile applications are as follows:'. The following points are listed in colored boxes:

- Design of a mobile Web site should be simple to fit on small screens.
- Avoid horizontal scrolling as some phones do not support horizontal scrolling and hide the content on the screen.
- Use buttons, instead of providing many tiny links, as this can annoy the mobile users.
- Create cookies to store the user's choice for viewing the full version of the site.
- Avoid creating complex forms with many input fields, as data entry can be difficult on mobile devices compared to the desktops.
- Limit the use of images due to bandwidth restrictions on mobile devices.

At the bottom left is the copyright notice '© Aptech Ltd.' and at the bottom right is the page number 'Building a Mobile Web Application / Session 16 54'.



**HTML5 Optimizing a Site for a Mobile 2-2**

- Add mobile specific functionalities, such as built-in GPS facility or call-in action links.
- Use of good foreground and background colors is important as they makes the sites readable on small screens.
- Select the technologies that are compatible with old mobile devices.
- Also, provide alternatives for functionalities, such as cookies, tables, style sheets, fonts, colors, and so on.
- Avoid use of pop-up windows, tables for layout, frames, and image maps in the mobile Web site design.

© Aptech Ltd. Building a Mobile Web Application / Session 16 55

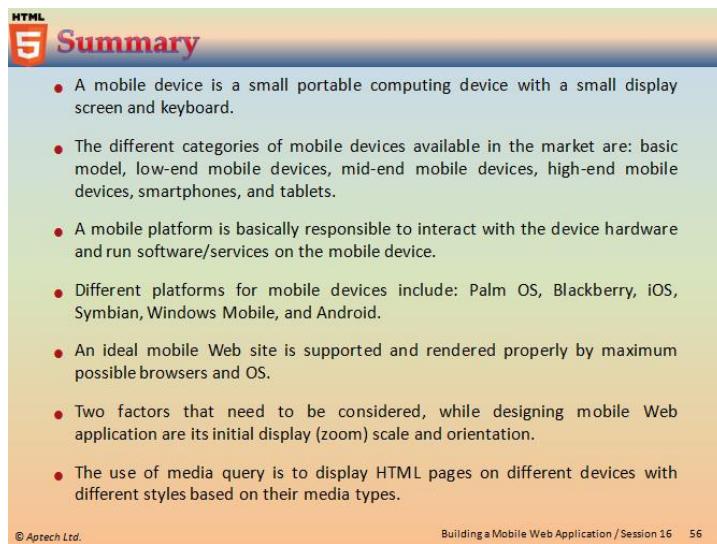
Using slides 54 and 55, explain how to optimize a site for mobile.

Mention the needs of mobile users are different from desktop users, in terms of screen size and connectivity issues. Thus, mobile Web sites should be optimized for better performance.

Explain some of the best practices that can be followed for mobile applications as listed on the slides.

## Slide 56

Let us summarize the session.



**HTML5 Summary**

- A mobile device is a small portable computing device with a small display screen and keyboard.
- The different categories of mobile devices available in the market are: basic model, low-end mobile devices, mid-end mobile devices, high-end mobile devices, smartphones, and tablets.
- A mobile platform is basically responsible to interact with the device hardware and run software/services on the mobile device.
- Different platforms for mobile devices include: Palm OS, Blackberry, iOS, Symbian, Windows Mobile, and Android.
- An ideal mobile Web site is supported and rendered properly by maximum possible browsers and OS.
- Two factors that need to be considered, while designing mobile Web application are its initial display (zoom) scale and orientation.
- The use of media query is to display HTML pages on different devices with different styles based on their media types.

© Aptech Ltd. Building a Mobile Web Application / Session 16 56

Using slide 56, summarize the session. End the session with a brief summary of what has been taught in the session.

### **16.3 Post Class Activities for Faculty**

You should familiarize yourself with the topics of the next session. You should also explore the Canvas and JavaScript that are offered with the next session.

**Tips:**

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 17 – Canvas and JavaScript

---

## 17.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 17.1.1 Objectives

By the end of this session, the learners will be able to:

- Describe Canvas in HTML5
- Explain the procedure to draw lines
- Explain the procedure to use color and transparency
- Explain the procedure to work with various drawing objects
- Describe working with images and text
- Describe the procedure to create Web page events with JavaScript and jQuery
- Describe the process of including external content in Web pages

### 17.1.2 Teaching Skills

To teach this session, you should be well-versed with `<canvas>` element in HTML5. Along with this, you should prepare yourself procedure to draw lines, use color, and transparency in Web pages. You should also aware yourself with drawing objects, images, text, and create Web page events with JavaScript and jQuery.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

#### In-Class Activities:

Follow the order given here during In-Class activities.

## Overview of the Session:

Then give the students the overview of the current session in the form of session objectives.  
Show the students slide 2 of the presentation.

**Objectives**

- Describe Canvas in HTML5
- Explain the procedure to draw lines
- Explain the procedure to use color and transparency
- Explain the procedure to work with various drawing objects
- Describe working with images and text
- Describe the procedure to create Web page events with JavaScript and jQuery
- Describe the process of including external content in Web pages

© Aptech Ltd. Canvas and JavaScript / Session 17 2

Tell the students that this session introduces `<canvas>` element in HTML5. They will learn about the procedure to draw lines, use color, and transparency in Web pages. They will also know about how to work with drawing objects, images, text, and create Web page events with JavaScript and jQuery.

## 17.2 In-Class Explanations

### Slides 3 to 8

Let us understand `<canvas>` element.

**Canvas Element 1-6**

- The `<canvas>` element in HTML5 can be used to draw shapes on Web sites as well as to dynamically draw graphics using JavaScript.
- The `<canvas>` element is represented like a rectangle on a page and allows the user to draw arcs, text, shapes, gradients, and patterns.
- The `<canvas>` in HTML5 is like the `<div>`, `<table>`, or `<a>` tag except that the content used in it is rendered through JavaScript.
- The `<canvas>` element does not contain any drawing abilities, instead, the drawing is done using a JavaScript code.
- To make use of the `<canvas>` element, a user has to add the `<canvas>` tag on the HTML page.
- Using `<canvas>` with JavaScript improves the overall performance of Web sites and avoids the requirement to download images from the sites.

© Aptech Ltd. Canvas and JavaScript / Session 17 3

## HTML5 Canvas Element 2-6

- The Code Snippet demonstrates the use of <canvas> element.

```
<!DOCTYPE HTML>
<html>
<head>
<title> Canvas </title>
<style>
    canvas{border: medium double red; margin: 4px}
</style>
</head>
<body>
<canvas width="278" height="200"></canvas>
</body>
</html>
```

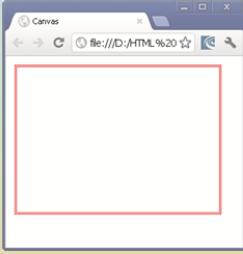
- In the code, the <style> element is used to display the border of the <canvas> element.
- The height and width attributes specify the size of the <canvas> element on the page.

© Aptech Ltd.

Canvas and JavaScript / Session 17 4

## HTML5 Canvas Element 3-6

- Following figure displays the <canvas> element.



To draw a <canvas> element, the user can use a context object.

The context object contains the drawing functions for a specific style of graphics.

Two-Dimensional (2d) context is used to work with 2d operations.

© Aptech Ltd.

Canvas and JavaScript / Session 17 5

## HTML5 Canvas Element 4-6

- The <canvas> element in DOM exposes the HTMLCanvasElement interface.

This interface provides the methods and properties for changing the presentation and layout of canvas elements.

The HTMLCanvasElement has a getContext(context) method that returns the drawing context for the canvas.

- The Code Snippet demonstrates the 2d context object for the canvas.

```
<!DOCTYPE HTML>
<html>
<head>
<title> Canvas </title>
<script>
    window.onload = function()
    {
        var canvas = document.getElementById('mCanvas');
        var ctext = canvas.getContext('2d');
```

© Aptech Ltd.

Canvas and JavaScript / Session 17 6

**HTML5 Canvas Element 5-6**

```

        ctext.beginPath();
        ctext.rect(18, 50, 200, 100);
        ctext.fillStyle = "DarkBlue";
        ctext.fill();
    };
</script>
</head>
<body>
<canvas id="mCanvas" width="578" height="200"></canvas>
</body>
</html>

```

- In the code, the `height` and `width` attributes define the height and width of the `canvas` element respectively.
- In the initializer function, the DOM object is accessed through the `id` attribute and gets a 2d context by using the `getContext()` method.
- The rectangle is created by using the `rect(18, 50, 200, 100)` method with `x`, `y`, `height`, and `width` parameters and is positioned at left corner of the page.

© Aptech Ltd. Canvas and JavaScript / Session 17 7

**HTML5 Canvas Element 6-6**

- Following figure displays the `<canvas>` element.

© Aptech Ltd. Canvas and JavaScript / Session 17 8

Using slides 3 to 8, explain `<canvas>` element.

Canvas is one of the most interesting features added in HTML5. The `<canvas>` element is used to draw graphics on the Web pages. It can also be used to dynamically draw graphics using JavaScript. This improves the overall performance of Web sites and avoids the requirement to download images from the sites.

The `<canvas>` element is represented like a rectangle on a page and allows the user to draw arcs, texts, shapes, gradients, and patterns.

The markup syntax is as follows: `<canvas id="myCanvas" width="200" height="100"></canvas>`

By using `<canvas>`, the user can draw many complex shapes and also apply various effects and transformations.

The `<canvas>` in HTML5 is like the `<div>`, `<table>`, or `<a>` tag except that the content used in it is rendered through JavaScript. The `<canvas>` element is simple and easy to use with JavaScript.

The `<canvas>` element does not contain any drawing abilities; instead, the drawing is done using a JavaScript code. To make use of the `<canvas>` element, a user has to add the `<canvas>` tag on the HTML page.

Explain the code snippet using slide 4 which demonstrates the use of `<canvas>` element.

In the code, the `<style>` element is used to display the border of the `<canvas>` element. The `height` and `width` attributes specify the size of the `<canvas>` element on the page.

The canvas is initially blank. To display something, a script first needs to access the rendering context and draw on it.

Mention that the DOM exposes the `HTMLCanvasElement` interface to work with the canvas element. This interface provides the methods and properties for changing the presentation and layout of canvas elements. The `HTMLCanvasElement` has a `getContext (context)` method that returns the drawing context for the canvas.

Explain the code snippet using slides 6 and 7 which demonstrates the 2D context object for the canvas.

#### **Tips:**

Since some older versions of the browsers such as Internet Explorer, don't support the `<canvas>` element, in that case, you should provide fallback content to be displayed by those browsers.

## Slides 9 to 12

Let us understand how to draw a line in canvas.

**HTML 5 Drawing a Line in Canvas 1-4**

- You can create lines in a canvas using the `stroke()`, `beginPath()`, `lineTo()`, and `moveTo()` methods.
- The following is the syntax to create a line in canvas:

**Syntax:**

```
ctext.beginPath();
ctext.moveTo(x,y);
ctext.lineTo(x,y);
ctext.stroke();
```

where,

- ctext - specifies a context object
- beginPath() - Specifies a new drawing path
- moveTo() - Specifies the creation of new sub path to the given position
- lineTo() - Specifies the drawing of a line from the context position to the given position
- stroke() - Specifies how to assign a color to the line and display it

© Aptech Ltd. Canvas and JavaScript / Session 17 9

**HTML 5 Drawing a Line in Canvas 2-4**

- The Code Snippet demonstrates creating a line in HTML5 canvas.

```
<!DOCTYPE HTML>
<html>
<head>
<title>Line</title>
<style>
body
{
    margin: 0px;
    padding: 0px;
}
#mCanvas
{
    border: 1px solid red;
}
</style>
<script>
window.onload = function() {
    var canvas = document.getElementById("mCanvas");
    var ctext = canvas.getContext("2d");
}
```

© Aptech Ltd. Canvas and JavaScript / Session 17 10

## Slide 10 (a)

**HTML 5 Drawing a Line in Canvas 2-4**

- The Code Snippet demonstrates creating a line in HTML5 canvas.

```

ctext.beginPath();
ctext.moveTo(100, 150);
ctext.lineTo(250, 50);
ctext.lineWidth = 5;
ctext.strokeStyle = "blue";
ctext.stroke();
};

</script>
</head>
<body>
<canvas id="mCanvas" width="360" height="200"></canvas>
</body>
</html>

```

© Aptech Ltd. Canvas and JavaScript / Session 17 10

### Slide 10 (b)

**HTML 5 Drawing a Line in Canvas 3-4**

- In the code, the height and width attributes are defined.
- The initializer function has the DOM object which is accessed through the id attribute and gets a 2d context by using the getContext () method.
- The beginPath () method is called through the context object to draw the path of the line.
- The moveTo (100, 150) method is called that creates a new path for the given point to place the drawing cursor and moves the position of the window to the upper-left corner by giving the x and y coordinates.
- The lineTo (250, 50) method is called to draw the line from the context point to given point.
- The lineWidth property is specified as 5 to define the width of the line on the canvas.
- The strokeStyle property sets the color of the line to blue.
- The stroke () method assigns the color to the line.

© Aptech Ltd. Canvas and JavaScript / Session 17 11

**HTML 5 Drawing a Line in Canvas 4-4**

- Following figure displays a line drawn in a canvas.

© Aptech Ltd. Canvas and JavaScript / Session 17 12

Using slides 9 to 12, explain how to draw line in canvas.

The functions used to draw a line are `stroke()`, `beginPath()`, `lineTo()`, and `moveTo()` methods.

The `stroke()` method actually draws the path you have defined with all those `moveTo()` and `lineTo()` methods. The default color is black.

The `lineTo()` method adds a new point and creates a line from that point to the last specified point in the canvas (this method does not draw the line).

The `moveTo()` method moves the path to the specified point in the canvas, without creating a line.

#### Tips:

Use the `strokeStyle` property to draw with another color/gradient.

Then, explain the code snippet which demonstrates creating a line in HTML5 canvas.

#### In-Class Question:

After you finish explaining how to draw a line in canvas, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which methods are used to define the starting point and ending point of the line?

#### Answer:

`moveTo(x, y)` and `lineTo(x, y)`

#### Slides 13 to 17

Let us understand the process of drawing the rectangle on canvas.

**HTML5 Working with Drawing Objects 1-17**

- HTML5 canvas allows the user to work with different types of drawing objects.
- Following objects can be drawn on a canvas element:

**> Rectangle**

- With HTML5 canvas, the user can create a rectangle using the `rect()` method.
- The HTML5 canvas is placed by using the `x` and `y` parameters and appropriately sized through `height` and `width` properties.
- Following table lists the common properties and methods of various shapes.

Properties and Methods	Description
<code>fillStyle</code>	The values can be gradient, pattern, or a CSS color. The default property style is solid black, but the user can set the color according to the requirements.
<code>fillRect(x, y, width, height)</code>	Enables the user to draw a rectangle with the existing fill style.
<code>strokeStyle()</code>	The values can be gradient, pattern, or a CSS color.

© Aptech Ltd. Canvas and JavaScript / Session 17 13

**HTML 5 Working with Drawing Objects 2-17**

Properties and Methods	Description
<code>strokeRect(x, y, width, height)</code>	Enables the user to draw a rectangle with the existing stroke style. This property is used to draw the edges of the rectangle.
<code>clearRect(x, y, width, height)</code>	Used to clear the pixels in a rectangle.

- The Code Snippet demonstrates how to create a rectangle in HTML5 canvas.

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      #mCanvas {
        border: 1px solid green;
      }
      body {
        margin: 0px;
        padding: 0px;
      }
    </style>
  </head>
  <body>
    <canvas id="mCanvas" width="278" height="200"></canvas>
  </body>
</html>
```

© Aptech Ltd. Canvas and JavaScript / Session 17 14

**HTML 5 Working with Drawing Objects 3-17**

```
<script>
  window.onload = function() {
    var canvas = document.getElementById('mCanvas');
    var ctext = canvas.getContext('2d');
    ctext.beginPath();
    ctext.rect(30, 50, 150, 100);
    ctext.fillStyle = "Magenta";
    ctext.fill();
    ctext.lineWidth = 5;
    ctext.strokeStyle = 'black';
    ctext.stroke();
  };
</script>
</head>
<body>
  <canvas id="mCanvas" width="278" height="200"></canvas>
</body>
</html>
```

© Aptech Ltd. Canvas and JavaScript / Session 17 15

**HTML 5 Working with Drawing Objects 4-17**

- In the code, the `height` and `width` attributes are defined.
- The initializer function has the DOM object which is accessed through the `id` attribute and gets a 2d context by using the `getContext()` method.
- The `beginPath()` method is called through the context object to draw the rectangle.
- The `rect(30, 50, 150, 100)` method takes `x, y, height, and width` as the parameters.
- The `fillStyle` property fills the rectangle with magenta color.
- The `fill()` method is used to paint the rectangle.
- The `lineWidth` property is specified as 5 to define the width of line on the canvas.
- The `strokeStyle` property sets the stroke style of the rectangle to black.
- The `stroke()` method assigns the color to the rectangle.

© Aptech Ltd. Canvas and JavaScript / Session 17 16

The screenshot shows a presentation slide with a red header bar containing the title 'Working with Drawing Objects 5-17'. Below the header, there is a list of bullet points and a screenshot of a browser window displaying a pink rectangle on a white canvas.

Following figure displays a rectangle drawn on the canvas.

With HTML5 canvas, the user can create an arc by using the `arc()` method.

Arcs are represented using a start angle, an end angle, a radius, a center point, and the drawing direction (anticlockwise or clockwise).

© Aptech Ltd. Canvas and JavaScript / Session 17 17

Using slides 13 to 17, explain the working with drawing object, rectangle.

HTML5 canvas allows the user to work with different types of drawing objects.

Explain the rectangle object.

With HTML5 canvas, the user can create a rectangle using the `rect()` method. The HTML5 canvas is placed by using the `x` and `y` parameters and appropriately sized through height and width properties. There is a collection of methods and properties that are used to draw different types of shapes.

Explain the list of common properties and methods of various shapes.

Explain the code snippet using slides 14 to 16 which demonstrates how to create a rectangle in HTML5 canvas.

**Tips:**

Canvas coordinates are represented by the upper-left corner coordinates (0, 0).

## Slides 18 to 20

Let us understand the process of drawing the arc on canvas.

**HTML 5 Working with Drawing Objects 6-17**

- The syntax to draw an arc in HTML5 is as follows:

**Syntax:**

```
arc(x, y, radius, startAngle, endAngle, anticlockwise)
```

where,

- x, y - Specifies the coordinates of the center of an arc
- radius - Specifies the distance from the center to any point on the circle
- startAngle, endAngle - Specifies the start and end points in the arc
- anticlockwise - Draws the arc clockwise or anticlockwise and accepts a boolean value

- The Code Snippet demonstrates how to create an arc in HTML5 canvas.

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      body
      {
        margin: 0px;
        padding: 0px;
      }
    </style>
  <script>
    window.onload = function()
    {
      var canvas = document.getElementById("mCanvas");
      var ctext = canvas.getContext("2d");
      var x = canvas.width / 2;
      var radius = 75;
      var startAngle = 1.1 * Math.PI;
      var endAngle = 1.9 * Math.PI;
      var ctrClockwise = false;
      ctext.beginPath();
      ctext.arc(x, y, radius, startAngle, endAngle, ctrClockwise);
      ctext.lineWidth = 25;
      // line color
      ctext.strokeStyle = "DarkGreen";
      ctext.stroke();
    };
  </script> </head>
  <body>
    <canvas id="mCanvas" width="278" height="250"></canvas>
  </body></html>
```

© Aptech Ltd. Canvas and JavaScript / Session 17 18

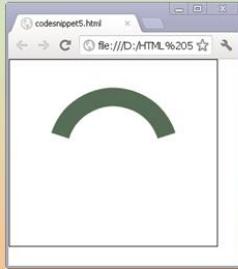
**HTML 5 Working with Drawing Objects 7-17**

```
#mCanvas {
  border: 1px solid black;
}
</style>
<script>
  window.onload = function()
  {
    var canvas = document.getElementById("mCanvas");
    var ctext = canvas.getContext("2d");
    var x = canvas.width / 2;
    var radius = 75;
    var startAngle = 1.1 * Math.PI;
    var endAngle = 1.9 * Math.PI;
    var ctrClockwise = false;
    ctext.beginPath();
    ctext.arc(x, y, radius, startAngle, endAngle, ctrClockwise);
    ctext.lineWidth = 25;
    // line color
    ctext.strokeStyle = "DarkGreen";
    ctext.stroke();
  };
</script> </head>
<body>
  <canvas id="mCanvas" width="278" height="250"></canvas>
</body></html>
```

© Aptech Ltd. Canvas and JavaScript / Session 17 19

**HTML 5 Working with Drawing Objects 8-17**

- In the code, the `beginPath()` method is called through the context object to draw an arc by using the `arc()` method which has `x, y, and radius` as the parameters.
- The `startAngle` and the `endAngle` are the start and end points of the arc.
- The `anticlockwise` specifies the direction of the arc between the two start and end points.
- Following figure displays an arc in HTML5 canvas.



© Aptech Ltd. Canvas and JavaScript / Session 17 20

Using slides 18 to 20, explain working with drawing object, arc.

With HTML5 canvas, the user can create an arc by using the `arc()` method. Arcs are represented using a start angle, an end angle, a radius, a center point, and the drawing direction (anticlockwise or clockwise).

Explain the syntax of the arc method. Explain the code snippet which demonstrates how to create an arc in HTML5 canvas.

### Slides 21 to 23

Let us understand the process of drawing the circle on canvas.

**HTML 5 Working with Drawing Objects 9-17**

**> Circle**

- In HTML5, you can draw a circle using the `arc()` method.
- You have to set the start angle with 0 and the end angle is specified as  $2 * \pi$ .
- Following is the syntax to draw a circle in HTML5 as follows:

**Syntax:**

```
arc(x, y, radius, startAngle, endAngle, anticlockwise)
```

where,

- x, y - Specifies the coordinates of the center of an arc
- radius - Specifies the distance from the center to any point on the circle
- startAngle, endAngle - Specifies the start and end points in the arc
- anticlockwise - Draws the arc clockwise or anticlockwise and accepts a boolean value

© Aptech Ltd. Canvas and JavaScript / Session 17 21

**HTML 5 Working with Drawing Objects 10-17**

- The Code Snippet demonstrates how to create a circle using HTML5.

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      body
      {
        margin: 0px;
        padding: 0px;
      }
      #mCanvas
      {
        border: 1px solid blue;
      }
    </style>
  <script>
    window.onload = function() {
      var canvas = document.getElementById("mCanvas");
      var ctext = canvas.getContext("2d");
      var ctrX = canvas.width / 2;
```

© Aptech Ltd. Canvas and JavaScript / Session 17 22

### Slide 22 (a)

**HTML5 Working with Drawing Objects 10-17**

- The Code Snippet demonstrates how to create a circle using HTML5.

```

var ctrY = canvas.height / 2;
var radius = 70;
ctext.beginPath();
ctext.arc(ctrX, ctrY, radius, 0, 2 * Math.PI, false);
ctext.fillStyle = "DarkOrchid";
ctext.fill();
ctext.lineWidth = 4;
ctext.strokeStyle = "black";
ctext.stroke();
};

</script>
</head>
<body>
<canvas id="mCanvas" width="356" height="150"></canvas>
</body>
</html>

```

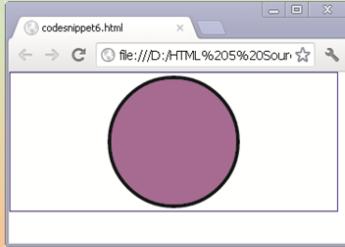
© Aptech Ltd.

Canvas and JavaScript / Session 17 22

**Slide 22 (b)**

**HTML5 Working with Drawing Objects 11-17**

- In this code, a circle is defined by using the `arc()` method which has `ctrX`, `ctrY`, and `radius` as the parameters.
- To define the arc with the points the `startAngle` is set to 0 and the `endAngle` is specified as `2*PI`.
- The `anticlockwise` defines the direction of the path of an arc between the two start and end points.
- Following figure displays a circle in HTML5 canvas.



© Aptech Ltd.

Canvas and JavaScript / Session 17 23

Using slides 21 to 23, explain the working with drawing object, circle.

In HTML5, circle can be drawn using the `arc()` method. You have to set the start angle with 0 and the end angle with `2 * PI`.

Explain the code snippet which demonstrates how to create a circle using HTML5.

## Slides 24 to 29

Let us understand the process of drawing the Bezier and quadratic curves on canvas.

**Working with Drawing Objects 12-17**

### Bezier Curves

- Using HTML5 canvas, you can create a Bezier curve using the `bezierCurveTo()` method.
- Bezier curves are represented with the two control points, context points, and an end point.
- The Code Snippet demonstrates how to create a Bezier curve using HTML5.

```
<!DOCTYPE HTML>
<html>
<head>
<style>
body
{
    margin: 0px;
    padding: 0px;
}
#mCanvas
{
    border: 1px solid maroon;
}
</style>
```

© Aptech Ltd. Canvas and JavaScript / Session 17 24

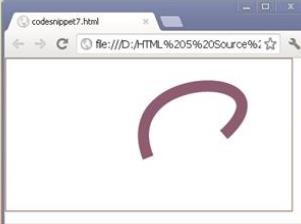
**Working with Drawing Objects 13-17**

```
<script>
window.onload = function()
{
    var canvas = document.getElementById("mCanvas");
    var ctext = canvas.getContext("2d");
    ctext.beginPath();
    ctext.moveTo(188, 130);
    ctext.bezierCurveTo(140, 10, 388, 10, 288, 100);
    ctext.lineWidth = 15;
    // line color
    ctext.strokeStyle = "purple";
    ctext.stroke();
};
</script>
</head>
<body>
<canvas id="mCanvas" width="378" height="200"></canvas>
</body>
</html>
```

© Aptech Ltd. Canvas and JavaScript / Session 17 25

**Working with Drawing Objects 14-17**

- In this code, the Bezier curve uses the `bezierCurveTo()` method.
- This method defines the current context point, two control points, and an end point.
- The context point uses the `moveTo()` method.
- The first portion of the curve is tangential to the imaginary line defined in the context point and first control point.
- The second portion of the curve is tangential to the imaginary line which is defined by the second control point and the ending point.
- Following figure displays a Bezier curve in a canvas.



© Aptech Ltd. Canvas and JavaScript / Session 17 26

**HTML5 Working with Drawing Objects 15-17**

### Quadratic Curves

- HTML5 canvas allows the user to create quadratic curves using the `quadraticCurveTo()` method.
- Quadratic curves are represented through the context point, an end point, and a control point.
- The Code Snippet demonstrates how to create a quadratic curve using HTML5.

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      body
      {
        margin: 0px;
        padding: 0px;
      }
      #mCanvas
      {
        border: 1px solid #9C9898;
      }
    </style>
  </head>
  <body>
    <script>
      window.onload = function()
      {
        var canvas = document.getElementById("mCanvas");
        var ctext = canvas.getContext("2d");
        ctext.beginPath();
        ctext.moveTo(178, 150);
        ctext.quadraticCurveTo(220, 0, 320, 150);
        ctext.lineWidth = 15;
        // line color
        ctext.strokeStyle = "Fuchsia";
        ctext.stroke();
      };
    </script>
  </body>
</html>
```

© Aptech Ltd. Canvas and JavaScript / Session 17 27

**HTML5 Working with Drawing Objects 16-17**

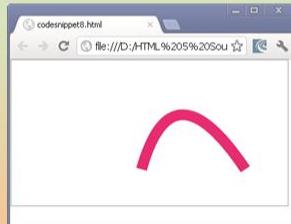
```
window.onload = function()
{
  var canvas = document.getElementById("mCanvas");
  var ctext = canvas.getContext("2d");
  ctext.beginPath();
  ctext.moveTo(178, 150);
  ctext.quadraticCurveTo(220, 0, 320, 150);
  ctext.lineWidth = 15;
  // line color
  ctext.strokeStyle = "Fuchsia";
  ctext.stroke();
};

</script>
</head>
<body>
  <canvas id="mCanvas" width="378" height="200"></canvas>
</body>
</html>
```

© Aptech Ltd. Canvas and JavaScript / Session 17 28

**HTML5 Working with Drawing Objects 17-17**

- In the code, the control point defines the curve of the quadratic by two tangential lines that are connected to both the context point and the end point.
- The context point is represented using the `moveTo()` method.
- This method moves the control point from the context point and the end point to create a sharper curve.
- It also moves the control point close to the context point and end point to create broad curves.
- Following figure displays a quadratic curve in a canvas.



© Aptech Ltd. Canvas and JavaScript / Session 17 29

Using sides 24 to 29, explain the Bezier curves and quadratic curves.

Using HTML5 canvas, you can create a Bezier curve using the `bezierCurveTo()` method. Bezier curves are represented with the two control points, context points and an end point. Explain the code snippet which demonstrates how to create a Bezier curve using HTML5.

A cubic Bezier curve requires three points. The first two points are control points that are used in the cubic Bezier calculation and the last point is the ending point for the curve. The starting point for the curve is the last point in the current path. If a path does not exist, use the `beginPath()` and `moveTo()` methods to define a starting point.

Then, explain the code snippet using slides 24 to 26 which demonstrates the drawing of the Bezier curve.

HTML5 canvas allows the user to create quadratic curves using the `quadraticCurveTo()` method. Quadratic curves are represented through the context point, an end point, and a control point. Code Snippet demonstrates how to create a quadratic curve using HTML5.

A quadratic Bezier curve requires two points. The first point is a control point that is used in the quadratic Bezier calculation and the second point is the ending point for the curve. The starting point for the curve is the last point in the current path. If a path does not exist, use the `beginPath()` and `moveTo()` methods to define a starting point.

Explain the code snippet to draw the quadratic curve.

**Tips:**

`bezierCurveTo()` has two control points while `quadraticCurveTo()` has one control point.

**In-Class Question:**

After you finish explaining working with drawing objects, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which method is used to moves the path to the specified point in the canvas, without creating a line?

**Answer:**

`moveTo()` method is used to moves the path to the specified point in the canvas, without creating a line.

## Slides 30 to 32

Let us understand how to draw an image on the canvas.

**HTML 5 Working with Images 1-3**

- In HTML5, the user can draw image objects on canvas using the `drawImage()` method.
- The `drawImage()` method can also draw parts of an image and increase or reduce the size of the image.
- This method accepts nine parameters, depending on editing that is required on the image.
- The image object can be a video, an image, or another canvas element.
- The Code Snippet demonstrates how to create an image using HTML5.

```
<!DOCTYPE HTML>
<html>
<head>
<style>
body {
margin: 0px;
padding: 0px;
}
#mCanvas {
border: 1px solid #9C9898;
}
</style>

```

© Aptech Ltd. Canvas and JavaScript / Session 17 30

**HTML 5 Working with Images 2-3**

```
<script>
window.onload = function()
{
var canvas = document.getElementById("mCanvas");
var ctext = canvas.getContext("2d");
var imgObj = new Image();
imgObj.onload = function()
{
ctext.drawImage(imgObj, 69, 50);
};
imgObj.src = "bird.jpg";
};
</script>
</head>
<body>
<canvas id="mCanvas" width="368" height="300"></canvas>
</body>
</html>
```

© Aptech Ltd. Canvas and JavaScript / Session 17 31

**HTML 5 Working with Images 3-3**

- In the code, the `onload` property is used.
- The source of the object is defined by using the `src` property.
- The image has to be loaded first and then instantiated with the `drawImage()` method.
- This method takes image object as the parameter with the x and y coordinates of the image.
- Following figure displays an image drawn on a HTML5 canvas.



© Aptech Ltd. Canvas and JavaScript / Session 17 32

Using slides 30 to 32, explain the drawing of images on the canvas.

In HTML5, the user can draw image objects on canvas using the `drawImage()` method. The `drawImage()` method can also draw parts of an image and increase or reduce the

size of the image. This method accepts nine parameters, depending on editing that is required on the image. The image object can be a video, an image, or another canvas element.

Explain the syntax as: `context.drawImage(img, x, y);`

Explain the code snippet which demonstrates how to create an image using HTML5.

### Tips:

To set size of the image by adding two parameters, width and height, the `drawImage(img, x, y, width, height)` method can be used.

## Slides 33 to 37

Let us understand how to work with text on the canvas.

**Working with Text 1-5**

- HTML5 canvas enables you to set the font, style, and size of text by using the font properties.
- The font style can be italic, normal, or bold.
- To set the text color, the fillStyle property of the canvas can be used.
- The Code Snippet demonstrates how to set the font, size, style, and color of the text on a HTML5 canvas.

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      body {
        margin: 0px;
        padding: 0px;
      }
      #mCanvas {
        border: 1px solid blue;
      }
    </style>
  </head>
  <body>
    <script>
```

© Aptech Ltd. Canvas and JavaScript / Session 17 33

**Working with Text 2-5**

```
<script>
  window.onload = function() {
    var canvas = document.getElementById("mCanvas");
    var ctext = canvas.getContext("2d");
    ctext.font = "italic 30pt Calibri";
    ctext.fillStyle = "MediumVioletRed";
    ctext.fillText("Welcome to HTML5!", 40, 100);
  };
</script>
</head>
<body>
  <canvas id="mCanvas" width="380" height="170"></canvas>
</body>
</html>
```

- In this code, the font text is specified as Calibri, style as italic, and size is set to 30pt.
- The `fillStyle` property specifies the text color and the `fillText` property is used to set the text on the canvas.

© Aptech Ltd. Canvas and JavaScript / Session 17 34

**HTML5 Working with Text 3-5**

- Following figure displays working with text in a HTML5 canvas.



In the screenshot, a browser window titled "codesnippet11.html" shows the text "Welcome to HTML5!" centered in a white rectangular box with a thin black border. The browser interface includes standard controls like back, forward, and search.

**HTML5 Working with Text 4-5**

- The Code Snippet demonstrates the use of stroke text in HTML5 canvas.

```
<!DOCTYPE HTML>
<html>
<head>
<style>
body {
    margin: 0px;
    padding: 0px;
}
#mCanvas {
    border: 1px solid black;
}
</style>
<script>
window.onload = function() {
    var canvas = document.getElementById("mCanvas");
    var ctext = canvas.getContext("2d");
}
```

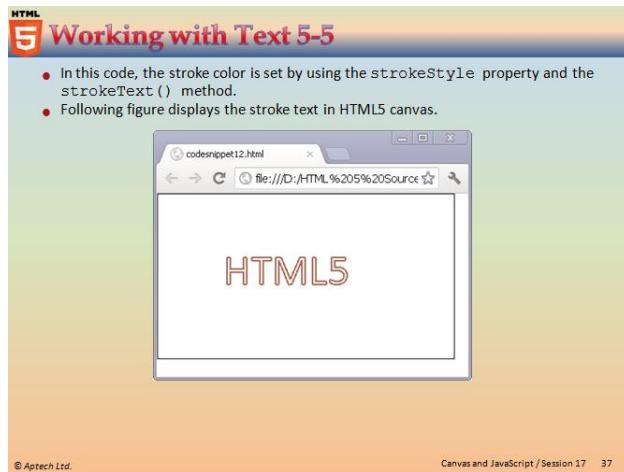
**HTML5 Working with Text 4-5**

- The Code Snippet demonstrates the use of stroke text in HTML5 canvas.

```
var x = 80;
var y = 110;
ctext.font = "40pt Calibri";
ctext.lineWidth = 2;
// stroke color
ctext.strokeStyle = "Brown";
ctext.strokeText("HTML5", x, y);
};

</script>
</head>
<body>
<canvas id="mCanvas" width="360" height="200"></canvas>
</body>
</html>
```

**Slide 36 (a)****Slide 36 (b)**



Using slides 33 to 37, explain how to work with text.

HTML5 canvas enables you to set the font, style, and size of text by using the font properties. The font style can be italic, normal, or bold. For setting the text color, you can use the `fillStyle` property of the canvas.

Explain the code snippet which demonstrates how to set the font, size, style, and color of the text on a HTML5 canvas.

In HTML5 canvas, the user can set the stroke color by using the `strokeText()` method and `strokeStyle` property of the canvas context.

The `strokeText()` method draws text (with no fill) on the canvas. The default color of the text is black.

Explain the code snippet using slides 36 (a) and 36 (b) which demonstrates the use of `strokeText()` in HTML5 canvas. In the code, the stroke color is set by using the `strokeStyle` property and the `strokeText()` method.

## Slides 38 to 40

Let us understand transparency of text in canvas.

**HTML 5 Using Transparency for Text in Canvas 1-3**

- There are two ways to set the transparency for the text and shapes.
- The first method is to use the `strokeStyle` and `fillStyle` by using the `rgb` function.
- The second method is to use `globalAlpha` drawing state property, which can be applied universally.
- The `globalAlpha` property is a value that ranges between 0 (fully transparent) and 1 (fully opaque).
- The Code Snippet demonstrates the use of `globalAlpha` property.

```
<!DOCTYPE HTML>
<html>
<head>
<style>
body {
margin: 0px;
padding: 0px;
}
#mCanvas {
border: 1px solid black;
}
</style>

```

© Aptech Ltd. Canvas and JavaScript / Session 17 38

**HTML 5 Using Transparency for Text in Canvas 2-3**

```
<script>
window.onload = function()
{
    var canvas = document.getElementById("mCanvas");
    var ctext = canvas.getContext("2d");
    ctext.fillStyle = "Indigo";
    ctext.strokeStyle = "black";
    ctext.lineWidth=2;
    ctext.font = "italic 30pt Calibri";
    ctext.fillText("HTML5", 40, 100);
    ctext.strokeText("HTML5", 40, 100);
    ctext.fillStyle="blue";
    ctext.globalAlpha=0.5;
    ctext.fillRect(100, 10, 150, 100);
};
</script>
</head>
<body>
<canvas id="mCanvas" width="350" height="170"></canvas>
</body>
</html>
```

© Aptech Ltd. Canvas and JavaScript / Session 17 39

**HTML 5 Using Transparency for Text in Canvas 3-3**

- In the code, the `fillStyle` and `strokeStyle` is used to color the text.
- The 'HTML5' text `lineWidth` is specified as 2 and the `font-family` is set to `Calibri` with `italic` style and `font-size` to `30pt`.
- The `fillText` property fills the color and `strokeText` property applies the `stroke` color to the `HTML5` text.
- The `fillStyle` is set to `blue` and `globalAlpha` property is set to `0.5`.
- The `fillRect(100, 10, 150, 100)` specifies the `x, y, height, and width` of the rectangle.
- Following figure displays the stroke text in `HTML5` canvas.

© Aptech Ltd. Canvas and JavaScript / Session 17 40

Using slides 38 to 40, explain transparency of text in canvas.

There are two ways to set the transparency for the text and shapes. The first method is to use the `strokeStyle` and `fillStyle` by using the `rgb` function. The second method is to use `globalAlpha` drawing state property, which can be applied universally. The

`globalAlpha` property is a value that ranges between 0 (fully transparent) and 1 (fully opaque).

Explain the code snippet which demonstrates the use of `globalAlpha` property.

In the code, the `fillStyle` and `strokeStyle` is used to color the text. For the 'HTML5' text, `lineWidth` is specified as 2, the `font-family` is set to Calibri with italic style, and `font-size` to 30pt. The `fillText` property fills the color and `strokeText` property applies the stroke color to the HTML5 text. The `fillStyle` is set to blue and `globalAlpha` property is set to 0.5. The `fillRect(100, 10, 150, 100)` specifies the `x`, `y`, `height`, and `width` of the rectangle.

Figure displays the transparency in text.

#### Tips:

You can provide text inside the `<canvas>` element which will be displayed in browsers that do not support `<canvas>`.

#### Slides 41 to 46

Let us understand events with jQuery.

The slide is titled "Using Events with jQuery 1-6". It contains the following text and code:

- jQuery also offers different events to deal with common interactions when the user moves the mouse or switches between two actions while clicking.
- The following are the events:

**➤ hover() event**

- The mouseenter and mouseleave are the two events often used together.
- jQuery provides a `hover()` function that accepts two parameters.
- The first parameter executes when the mouse moves over the element and the second function executes when the mouse moves away from the element.
- The Code Snippet demonstrates the hover event.

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery-1.7.2.min.js"></script>
<script>
$(document).ready(function() {
```

© Aptech Ltd.      Canvas and JavaScript / Session 17      41

**HTML**

## Using Events with jQuery 2-6

```
$("p").hover(function() {
  $("p").css("background-color", "red");
}, function() {
  $("p").css("background-color", "maroon");
});

</script>
</head>
<body>
  <p>Hover the mouse on this line.</p>
</body>
</html>
```

- In the code, the `hover()` method is used.
- When the mouse is placed on the text, then the background color changes to red.
- When the user moves the mouse outside the text, the background color changes to maroon.

© Aptech Ltd. Canvas and JavaScript / Session 17 42

**HTML**

## Using Events with jQuery 3-6

- Following figure displays the mouseenter effect.
- Following figure displays the mouseleave effect.

© Aptech Ltd. Canvas and JavaScript / Session 17 43

**HTML**

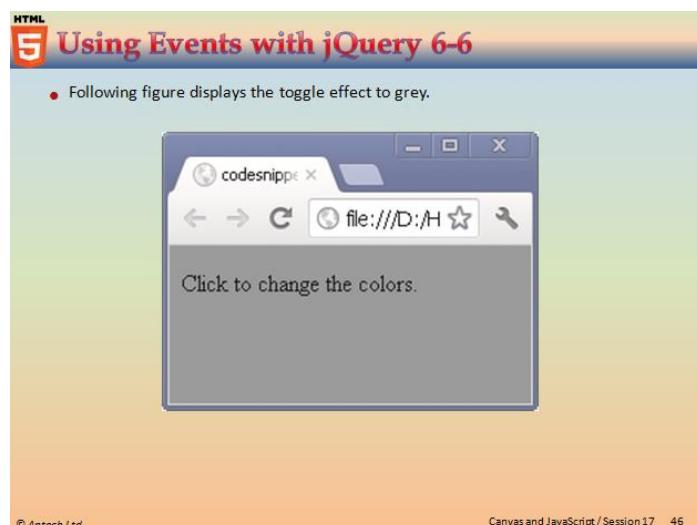
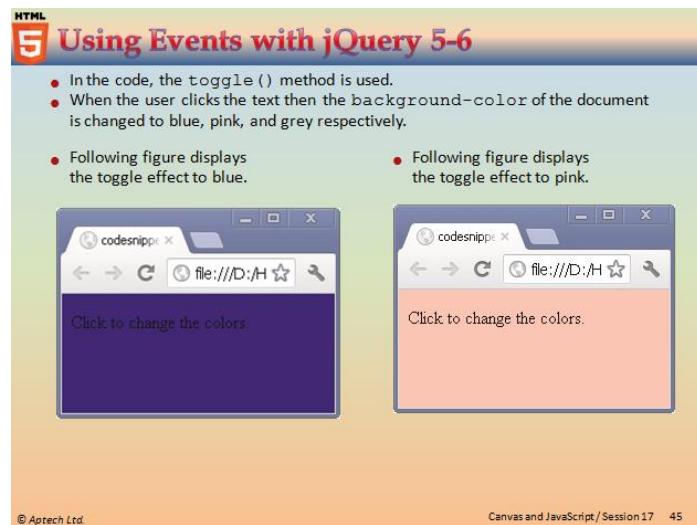
## Using Events with jQuery 4-6

### ➤ `toggle()` event

- The `toggle()` event works in a similar manner as that of the `hover()` event, except that it responds to mouse clicks.
- The `toggle()` function accepts more than two functions as arguments.
- All the functions passed to the `toggle()` event will react to its corresponding click action.
- The Code Snippet demonstrates the `toggle` event.

```
<!DOCTYPE html>
<html>
<head>
  <script src="jquery-1.7.2.min.js"></script>
  <script>
    $(document).ready(function() {
      $("p").toggle(function() {
        $("body").css("background-color", "blue");
      },
      function() {
        $("body").css("background-color", "pink");
      },
    });
  </script>
</head>
<body>
  <p>Hover the mouse on this line.</p>
</body>
</html>
```

© Aptech Ltd. Canvas and JavaScript / Session 17 44



Using slides 41 to 46, explain the events with jQuery.

A Web page can respond to user actions through events. Some of the examples of the events are as follows:

- Moving a mouse over an element
- Selecting a radio button
- Clicking an element on the page

jQuery also offers different events to deal with common interactions when the user moves the mouse or switch between two actions while clicking.

Explain the `hover ()` event.

The `mouseenter` and `mouseleave` are the two events often used together. For example, when a user moves a mouse over a menu, a tooltip appears and when the user moves the mouse off the menu, the tooltip disappears. Combining these two events is very

common, therefore, jQuery provides a `hover()` function that accepts two parameters. The first parameter executes when the mouse moves over the element and the second function executes when the mouse moves away from the element.

Explain the code snippet which demonstrates the `hover` event.

Then, explain the `toggle()` event using slides 44 and 45.

The `toggle()` event works in a similar manner as that of the `hover()` event, except that it responds to mouse clicks. The `toggle()` function accepts more than two functions as arguments. For example, you want to perform some action on the first click, another action on the second click, and one more action on the third click. All the functions passed to the `toggle()` event will react to its corresponding click action.

Explain the code snippet which demonstrates the `toggle` event.

### In-Class Question:

After you finish explaining the events in jQuery, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What are the parameters of the `hover()` event?

### Answer:

The `hover()` method takes two functions and is a combination of the `mouseenter()` and `mouseleave()` methods.

### Slide 47

Let us understand how to include external content in Web pages.

**HTML5** **Inclusion of External Content in Web Pages**

HTML5 introduces the `<eventsource>` tag that allows the user to push external content in the Web page. This model is referred to as push model.

Since the `<event source>` tag is not supported in many browsers, users make use of the `<embed>` tag for this purpose.

The `<embed>` tag is a new element in HTML5 and it is represented as a container for an interactive content or an external application.

The `<embed>` tag is often used to add elements such as image, audio, or video on a Web page.

- The Code Snippet demonstrates the use of `<embed>` tag.  
`<embed src="mymovie.mp3" />`
- In this code, the `src` attribute specifies the path of an external file to embed.

© Aptech Ltd

Canvas and JavaScript / Session 17 47

Using slide 47, explain how to include the external content in Web pages.

HTML5 introduces the `<eventsource>` tag that allows the user to push external content in the Web page. This model is referred to as push model. Since the `<eventsource>` tag is not supported in many browsers, users make use of the `<embed>` tag for this purpose. The `<embed>` tag is a new element in HTML5 and it is represented as a container for an interactive content or an external application. The `<embed>` tag is often used to add elements such as image, audio, or video on a Web page.

In the code, the `src` attribute specifies the path of an external file to embed.

## Slide 48

Let us summarize the session.

**Summary**

- The `<canvas>` element is a drawing area where the user can draw graphics, use images, add animations, and also add text for enhancing the user experience on Web pages.
- To create a line, on a canvas one can use the `stroke()`, `beginPath()`, `lineTo()`, and `moveTo()` methods.
- Arches are represented using a start angle, an end angle, a radius, a center point, and the drawing direction (anticlockwise or clockwise).
- With HTML5 canvas, the user can create a rectangle using the `rect()` method.
- Bézier curves are represented with the two control points, context points, and an end point.
- HTML5 canvas allows the user to create quadratic curves using the `quadraticCurveTo()` method.
- HTML5 canvas enables the user to draw image object on canvas using the `drawImage()` method.

© Aptech Ltd. Canvas and JavaScript / Session 17 48

Using slide 48, summarize the session. End the session with a brief summary of what has been taught in the session.

## 17.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session. You should also explore the HTML5 Web Storage that is offered with the next session.

### Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 18 – HTML5 Web Storage

---

## 18.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 18.1.1 Objectives

By the end of this session, the learners will be able to:

- Explain Web storage in HTML5
- Explain session storage
- Explain local storage
- Explain the IndexedDB API
- Describe a native app
- Explain the difference between native apps and HTML5 apps
- Describe the advantages of native and HTML5 apps
- List the steps to convert HTML5 apps to native apps

### 18.1.2 Teaching Skills

To teach this session, you should be well-versed with the Web storage API that provides functionality for storing data on the client-side. Along with this, you should prepare yourself with IndexedDB API which allows hosting of Web databases locally within the user's browser. Also, you should aware yourself with the differences between the native apps and HTML5 app.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

### In-Class Activities:

Follow the order given here during In-Class activities.

### Overview of the Session:

Then give the students the overview of the current session in the form of session objectives. Show the students slide 2 of the presentation.

**HTML5 Objectives**

- Explain Web storage in HTML5
- Explain session storage
- Explain local storage
- Explain the Indexed DB API
- Describe a native app
- Explain the difference between native apps and HTML5 apps
- Describe the advantages of native and HTML5 apps
- List the steps to convert HTML5 apps to native apps

© Aptech Ltd.      HTML5 Web Storage / Session 18      2

Tell the students that this session introduces them to the storage of date in the Web pages. The session also explains them about indexed API which acts as a local database within the browsers. They will also learn about the difference between the native apps and HTML5 app.

## 18.2 In-Class Explanations

### Slide 3

Let us understand need for Web storage.

**HTML5 Introduction**

Traditionally, over the last few decades, Web applications have been using cookies to store small amounts of information on a user's computer.

A cookie is a file that stores user-related information and may either be temporary or permanent.

A cookie can be created for login details which can be saved for a specified period on a user's computer.

- Drawbacks of cookies are as follows:
  - Cookies slow down the performance of Web application, as they are included with every HTTP request
  - Cookies cannot be considered as safe means for transmission of sensitive data
  - Cookies cannot store large amount of information, as they have a limitation of size of 4 KB
  - W3C has designed a specification named Web Storage API which offer a solution to store data on the client-side

© Aptech Ltd.      HTML5 Web Storage / Session 18      3

Using slide 3, explain the need for Web storage.

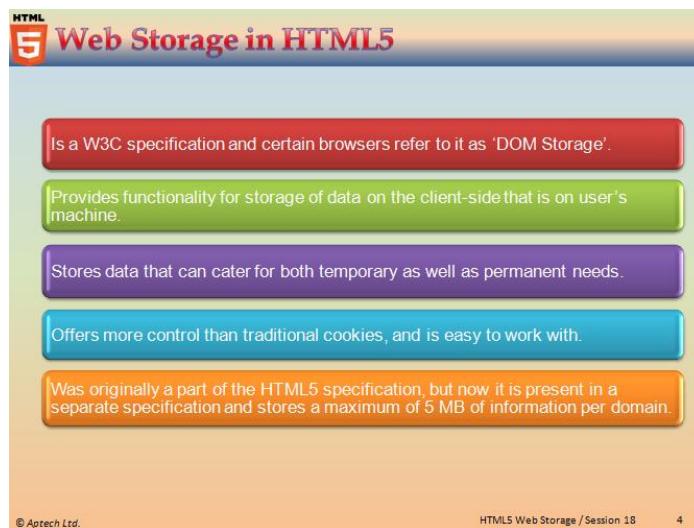
Consider an e-mail client, such as Gmail. To log in to your mail account in Gmail, you need to enter your username and password.

Traditionally, over the last few decades, Web applications have been using cookies to store small amounts of information on a user's computer. A cookie is a file that stores user-related information and may either be temporary or permanent. Thus, in this case, a cookie can be created for login details which can be saved for a specified period on a user's computer.

To overcome these drawbacks and offer a solution to store data on the client-side, W3C has designed a specification named, Web Storage API. The Web storage provides the functionality using which data can be stored on the client-side for a session or beyond the session.

#### **Slide 4**

Let us understand the Web storage in HTML5.



Using slide 4, explain the Web storage in HTML5.

Web storage is a W3C specification. It provides functionality for storage of data on the client-side that is on user's machine. This data can cater for both temporary as well as permanent needs. Certain browsers also refer to it as 'DOM Storage'. The advantage of such storage is that it offers more control than traditional cookies, and is easy to work with.

Web storage was originally a part of the HTML5 specification, but now it is present in a separate specification. It enables to store a maximum of 5 MB of information per domain.

HTML5 Web applications make use of Web storage to implement client-side persistent storage.

**In-Class Question:**

After you finish explaining Web storage in HTML5, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What is Web Storage?

**Answer:**

Web storage is a W3C specification. It provides functionality for storage of data on the client-side that is on user's machine.

**Slide 5**

Let us understand Web storage versus cookies.

The slide has a header 'HTML 5 Web Storage versus Cookies'. Below the header, a bulleted list states: 'Some key differences between cookies and Web storage are as follows:'

- Cookies are meant to be read on the server-side, whereas Web storage is available only on the client-side.
- Cookies are sent along with each HTTP request to the server, whereas Web storage data is not carried over to the server.
- Cookies result in bandwidth overhead and thus lead to high costs, as they are sent with each HTTP request. The Web storage is stored on the user's hard drive, so it costs nothing to use.
- With cookies, the information data that could be stored is 4 KB, whereas with Web storage, a large amount of data can be stored upto 5 MB.

At the bottom left is the copyright notice '© Aptech Ltd.', at the bottom center is 'HTML5 Web Storage / Session 18', and at the bottom right is the page number '5'.

Using slide 9, explain the Web storage versus cookies.

Explain the difference between Web storage and cookies as mentioned on the slide.

## Slide 6

Let us understand the browser specific Web storage.

Browser	Version
IE	8.0+
Firefox	3.6+
Safari	4.0+
Chrome	5.0+
Opera	10.5+

Using slide 6, explain the browser-specific Web storage.

Web storage is browser-specific. If a user visits a site in Google Chrome, any data will be stored to Google Chrome's Web storage store. Similarly, if the user revisits that same site in Firefox, the data saved earlier through Google Chrome will be unavailable. The location where the Web storage data is stored depends on the browser. Each browser's storage is separate and independent, even if it is present on the same machine.

HTML5 Web storage is implemented natively in most Web browsers, so one can use it even when a third-party browser plug-in is not available.

Explain the list of supports of various browsers for HTML5 Web storage.

## Slides 7 and 8

Let us explore Web storage.

Two types of HTML5 Web storage are namely, session storage and local storage.

Both session and local storage enable to store around 5 MB of data per domain.

To check for browser support of HTML5 Web storage, a property named **localStorage** or **sessionStorage** is available as a global variable for the window object.

If there is no support, the **localStorage** or **sessionStorage** property will be undefined.

- Code Snippet demonstrates the script to check the support for HTML5 Web storage in the browser.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Support for Web Storage</title>
    <script>
      function checkSupport() {
        if ('sessionStorage' in window) && window['sessionStorage']
        !== null)
        {
          alert("Your browser supports Web Storage");
          return;
        }
        alert("Your browser does not support Web Storage");
      }
    </script>
  </head>
  <body onload="checkSupport();">
  </body>
</html>
```

**JavaScript Alert**  
Your browser supports Web Storage  
OK

Using slides 7 and 8, explain the Web storage in detail.

The two types of HTML5 Web storage are namely, session storage and local storage.

**localStorage** - The data will be stored in the browser permanently.

**sessionStorage** – The data will be stored within the browser only for a single session.

Both session and local storage enable to store around 5 MB of data per domain.

To check for browser support of HTML5 Web storage, a property named **localStorage** or **sessionStorage** is available as a global variable for the **window** object. If there is no support, the **localStorage** or **sessionStorage** property will be undefined.

Explain the code snippet which demonstrates the script to check the support for HTML5 Web storage in the browser.

In the code, the `if` statement checks whether a property named `sessionStorage` exists in the global `window` object. If the property exists, it means that session storage is supported and an appropriate message is displayed to indicate the same. If however, the property does not exist, it means session storage is not supported on that browser, and an appropriate message is displayed to indicate the same.

### **In-Class Question:**

After you finish explaining how to explore Web storage, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which property is used for checking the support Web storage?

### **Answer:**

The `localStorage` or `sessionStorage` properties are used for checking the support Web storage.

### **Slides 9 to 14**

Let us understand the session storage.

**HTML5 Session Storage 1-6**

- Keeps track of data specific to one window or tab and discards it as soon the user closes the tab (or window) that he/she was working with.
- Lasts for the entire duration of the session and hence, is not persistent.
- Makes use of named key/value pairs which are enclosed within double quotes.
- Stores the data using the named key, whereas the data is retrieved by referring to that key.
- Key is a string, whereas the value stored in the key can be of any data type such as string, boolean, integer, or float. Regardless of the type of data that is stored, it is actually stored internally as a string.
- Storing and retrieving data of other types requires the use of functions to convert them into the appropriate data types.

© Aptech Ltd.      HTML5 Web Storage / Session 18      9

**HTML 5 Session Storage 2-6**

- Following table lists some examples of named key/value pairs belonging to various data types.

Key	Value
Name	Sarah
book	C Programming
Email	info@me.com
car	Toyota Innova
age	28
uservalid	true

© Aptech Ltd.      HTML5 Web Storage / Session 18      10

**HTML 5 Session Storage 3-6**

- Several operations that can be performed with the **sessionStorage** object are as follows:

Storing and retrieving data - **setItem()** and **getItem()** methods are used to store and retrieve data from session storage respectively.

- Syntax to use **setItem()** and **getItem()** methods is as follows:
- To assign data

```
sessionStorage.setItem(key, value);
```

where,

key: Is the named key to refer to the data.

value: Is the data to be stored.

© Aptech Ltd.      HTML5 Web Storage / Session 18      11

**HTML 5 Session Storage 4-6**

- To retrieve data

```
var item = sessionStorage.getItem(key);
```

where,

item: Is the variable into which the data will be saved.

key: Is the named key to refer to the data.

© Aptech Ltd.      HTML5 Web Storage / Session 18      12

**HTML5 Session Storage 5-6**

- Code snippet demonstrates how to set and retrieve a name using sessionStorage object.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Working with Session Storage</title>
    <script>
      function testStorage() {
        if ('sessionStorage' in window) && window['sessionStorage'] !== null) {
          sessionStorage.setItem('name', 'Sarah');
          alert('The name is: ' + sessionStorage.getItem('name'));
        }
      }
    </script>
  </head>
  <body onload="testStorage()">
  </body>
</html>
```



© Aptech Ltd.      HTML5 Web Storage / Session 18      13

**HTML5 Session Storage 6-6**

- Removing data

```
sessionStorage.removeItem(key);
```

where,

key: Is the named key to refer to the data.

- Clearing data

```
sessionStorage.clear();
```

© Aptech Ltd.      HTML5 Web Storage / Session 18      14

Using slides 9 to 14, explain the session storage.

Session storage keeps track of data specific to one window or tab and discards it as soon as the user closes the tab (or window) that he/she was working with. Thus, even if you are visiting the same site in two different windows, each window will have its own individual session storage object. This means that each window contains separate session storage object with distinct data. Session storage lasts for the entire duration of the session and hence, is not persistent.

Session storage makes use of named key/value pairs. The data is stored using the named key, whereas the data is retrieved by referring to that key. Both the key-value pairs are enclosed within double quotes.

The key is a string, whereas the value stored in the key can be of any type of data, such as string, boolean, integer, or float. Regardless of the type of data that is stored, it is actually stored internally as a string.

Therefore, storing and retrieving data of other types requires the use of functions to convert them into the appropriate data types. For example, the function, `parseInt()` is used to convert data into an appropriate JavaScript data type.

Explain some examples of named key/value pairs belonging to various data types given in the table on slide 10.

Using slides 11 and 12, explain various operations to be performed on the session storage.

Explain the code snippet which demonstrates how to set and retrieve a name using `sessionStorage` object.

It is also possible to remove and clear data from the session storage. The `removeItem()` method is used to remove a particular item from the list. To remove the value associated with the key, `username` set in the code snippet, the statement must be specified as shown here: `sessionStorage.removeItem('username');`

The `removeItem(key)` method must cause the key/value pair with the given key to be removed from the list associated with the object, if it exists. If no item with that key exists, the method must do nothing.

Similarly, in order to clear all items present in the session storage, use the `clear()` method as shown here: `sessionStorage.clear();`

The `clear()` method must atomically cause the list associated with the object to be emptied of all key/value pairs, if there are any. If there are none, then the method must do nothing.

Also, the `length` attribute determines the number of key/value pairs present in the storage as shown here: `var itemcount = sessionStorage.length;`

When the `setItem()`, `removeItem()`, and `clear()` methods are invoked, events are fired on the window objects of other documents that can access the newly stored or removed data, as defined in the sections on the `sessionStorage` and `localStorage` attributes.

## Slides 15 to 18

Let us understand the local storage.

**HTML5 Local Storage 1-4**

- Enables to save data for longer periods on the user's computer, through the browser.
- Data is persistent and can be retrieved when a user visits the site again.
- Is used, if data needs to be stored for more than a single session.
- Works in a similar fashion as session storage.
- Uses the same functions, such as `setItem()`, `getItem()`, `removeItem()`, and `clear()`.

© Aptech Ltd.

HTML5 Web Storage / Session 18

15

**HTML5 Local Storage 2-4**

- Code snippet demonstrates the use of local storage to store the value of `username` field and later, retrieve the value in another Web page.

```
<!DOCTYPE html>
<html>
  <title> Local Storage </title>
  <script>
    function store() {
      if ('localStorage' in window) && window['localStorage'] !== null) {
        var username = document.getElementById('username').value;
        localStorage.setItem('username', username);
      } else {
        alert ('your browser does not support storage');
      }
    }

    function cancel_store() {
      if ('localStorage' in window) && window['localStorage'] !== null) {
        localStorage.removeItem('username');
      } else {
        alert ('your browser does not support storage');
      }
    }
  </script>
</html>
```

© Aptech Ltd.

HTML5 Web Storage / Session 18

16

**HTML 5 Local Storage 3-4**

- Code snippet demonstrates the use of local storage to store the value of `username` field and later, retrieve the value in another Web page.

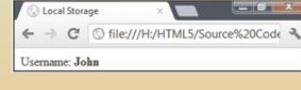
```
</script>
</head>
<body>
<form method="get" action="success.html">
    Username: <input type="text" id="username" value="" size="20" onblur="store()"/>
    <input type="submit" value="Submit"/>
    <input type="reset" Value="Cancel" onclick="cancel_store()"/>
</body>
</html>
```

© Aptech Ltd.      HTML5 Web Storage / Session 18      17

**HTML 5 Local Storage 4-4**

- Code snippet shows the `success.html` page that retrieves value from the local storage and displays it in the browser.

```
<!DOCTYPE html>
<head>
    <title> Local Storage </title>
    <script>
        function print() {
            var username = localStorage.getItem('username');
            document.getElementById('lblMsg').innerHTML = 'Username: is <b>' + username + '</b>';
        }
    </script>
</head>
<body onload="print()">
    <label id="lblMsg"></label><br>
</body>
</html>
```



© Aptech Ltd.      HTML5 Web Storage / Session 18      18

Using slides 15 to 18, explain the local storage.

Unlike session storage, local storage enables to save data for longer periods on the user's computer, through the browser. The data is persistent and can be retrieved when a user visits the site again. In other words, local storage is used, if data needs to be stored for more than a single session.

A simple scenario would be to count the number of times a person has visited a Web page. In terms of methods, local storage works in a similar fashion as session storage. It uses the same functions, such as `setItem()`, `getItem()`, `removeItem()`, and `clear()`.

Explain the code snippet using slides 16 and 17 which demonstrates the use of local storage to store the value of `username` field and later, retrieve the value in another Web page.

Here, in the code, the support of `localStorage` object is checked in the current browser. If it is supported, then the contents of the `username` box are retrieved and stored in a variable named `username`. Then, the content of this variable is assigned to the local storage object with the key set as `username`. If `localStorage` object is not supported, an appropriate message is displayed in the alert window.

Also, the function `cancel_store()` is invoked, when the user clicks **Cancel**. In the `cancel_store()` function, the `removeItem()` method removes the specified key and its value from local storage.

When the **Submit** button is clicked, the user is redirected to the Web page, **success.html**, which displays the value stored with the key, `username`.

Explain the code snippet using slide 18 which shows the **success.html** page that retrieves value from the local storage and displays it in the browser.

Here, in the code, `getItem()` method of local storage retrieves the value from the `username` key and stores in the variable `username`. Then, the value of the variable `username` is displayed in the `<label>` tag.

### Slides 19 to 21

Let us understand the Indexed database API.

The slide has a blue header bar with the text "HTML5" and "Indexed Database API 1-3". Below the header are six green callout boxes with white text:

- A database is an organized collection of data.
- Databases, such as relational database stores the data in the form of tables.
- A table comprises rows and columns that are used to store data.
- The representation of data from a table is in the form of records.
- HTML5 has introduced a new Web Storage API which can host Web databases locally within the user browser.
- Web databases are not like relational databases in terms of functionality.

At the bottom left is the copyright notice "© Aptech Ltd.". At the bottom right are the page numbers "HTML5 Web Storage / Session 18" and "19".

**HTML5 Indexed Database API 2-3**

- Indexed Database API is a specification also known as IndexedDB.
- It is basically an object store that can be used to store and manipulate data on the client-side.
- The object store is the primary storage mechanism that stores the object in the database managed locally within the browser.
- It enables to create an object store of a particular type in which objects can be persisted using JavaScript.
- IndexedDB enables to create Web applications with rich query abilities and which can work both online and offline.
- IndexedDB supports two types of API namely, synchronous and asynchronous.
- The synchronous API can be used with WebWorkers, whereas asynchronous API can be used for Web applications.

© Aptech Ltd.      HTML5 Web Storage / Session 18      20

**HTML5 Indexed Database API 3-3**

- IndexedDB API is implemented using `window.indexedDB` object.
- Browsers implement the IndexedDB object with their own prefixes. For example, Chrome browser uses the `webkit` prefix, whereas Mozilla supports `-moz` prefix.

● Following table lists the browser support for the IndexedDB API.

IE	Firefox	Chrome	Safari	Opera	iOS Safari
6.0	-	-	-	-	3.2
7.0	8.0moz	-	-	-	4.0-4.1
8.0	9.0moz	16.0webkit	5.0	-	4.2-4.3
9.0	10.0moz	17.0webkit	5.1	11.6	5.0
10.0	11.0moz	18.0webkit	6.0	12.0	-
-	12.0moz	19.0webkit	-	-	-

© Aptech Ltd.      HTML5 Web Storage / Session 18      21

Using slides 19 to 21, explain the Indexed database API.

HTML5 has introduced a new Web Storage API which can host Web databases locally within the user browser. However, Web databases are not like relational databases in terms of functionality.

Indexed Database API is a specification also known as IndexedDB. It is basically an object store that can be used to store and manipulate data on the client-side. The object store is the primary storage mechanism that stores the object in the database managed locally within the browser. It enables to create an object store of a particular type in which objects can be persisted using JavaScript. Thus, IndexedDB enables to create Web applications with rich query abilities and which can work both online and offline.

The IndexedDB API is implemented using `window.indexedDB` object. As the current

specification is still in the evolving stage, browsers implement the `IndexedDB` object with their own prefixes. For example, Chrome browser uses the `webkit` prefix, whereas Mozilla supports `-moz-` prefix.

Explain the table that lists the browser support for the IndexedDB API.

#### Tips:

IndexedDB maintains indexes over the records it stores and developers use the IndexedDB JavaScript API to locate records by key or by looking up an index.

#### Slides 22 and 23

Let us understand the common terms in IndexedDB API.

**HTML5** **Indexed DB API 1-2**

- Some of the basic constructs of IndexedDB API are as follows:

- Database** - The IndexedDB database comprises more than one object store. Each database contains a name that identifies the origin of the database and a version number which identifies the lifetime of the database.
- Object Store** - Is the main mechanism to store data in a database. They hold the data stored in the database in the form of records.
- Keys** - Each record stored in the database is identified by a unique key.
- Values** - Are the data stored in the records.
- Key Path** - Is a string that defines how the browser should extract key from a value. The key from a value can be extracted either in the object store or index.
- Index** - Is used when the data from the object store is retrieved based on some other values other than a key.
- Transaction** - Any addition or retrieval of the data in a database is performed by using transaction. Each transaction has a mode, scope, and a request list

© Aptech Ltd.      HTML5 Web Storage / Session 18      22

**HTML5** **Indexed DB API 2-2**

- Some of the other basic constructs of IndexedDB API are as follows:

- Requests** - Operations, such as reading or writing on the database is performed using a request. Each request contain attributes, such as flag, source object, result, and error.
- Cursor** - Is a mechanism used to retrieve multiple records from a database.
- Key Range** - Records from the object stores and indexes are retrieved using keys or key ranges. A key range refers to retrieval of data between specified bounds based on the keys.

© Aptech Ltd.      HTML5 Web Storage / Session 18      23

Using slides 22 and 23, explain the basic constructs of IndexedDB API.

**In-Class Question:**

After you finish explaining IndexedDB API, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which construct is used to retrieve multiple records?

**Answer:**

Cursor is used to retrieve multiple records.

**Slides 24 to 29**

Let us understand how to implement IndexedDB API.

**HTML5 Implementing IndexedDB API 1-6**

- Steps to implement the IndexedDB API in a Web application are as follows:

Open a database

Create an object store

Start a transaction

Perform some database operations, such as add and retrieve

Work with the retrieved results

© Aptech Ltd.      HTML5 Web Storage / Session 18      24

**HTML5 Implementing IndexedDB API 2-6**

- Opening a Database
- Code snippet shows the code to open a database

```

var indexedDB = window.indexedDB || window.webkitIndexedDB ||  

window.mozIndexedDB || window.msIndexedDB;  

var request = indexedDB.open("CompanyDB", 1);  

request.onsuccess = function (event) {  

  ...
};  

request.onerror = function (event) {  

  console.log("IndexedDB error: " + event.target.errorCode);
};

```

© Aptech Ltd.      HTML5 Web Storage / Session 18      25

**HTML5 Implementing IndexedDB API 3-6**

- **Updating Version of a Database**

After the database is opened, it can be structured by providing a version number which helps to set up the database.

- Code snippet shows the code that specifies the version number to the database

```
var setVrequest = db.setVersion("1.99");
setVrequest.onsuccess = function(event) {
    ...
}
```

© Aptech Ltd.      HTML5 Web Storage / Session 18      26

**HTML5 Implementing IndexedDB API 4-6**

- **Creating the Object Store**

Structure of IndexedDB database facilitates the storage of multiple object stores.

Object store is created using `createObjectStore()` method which accepts two arguments namely, the store name and a parameter object.

- Code snippet demonstrates the code to create an object store named employee in the CompanyDB database.

```
var employeeData = [
    { name: "John Smith", email: "john@company.com" },
    { name: "Jill Patrick", email: "jill@company.com" },
    { name: "Rock Ethan", email: "rock@company.com" },
    { name: "Daniel Andrew", email: "daniel@company.com" }
];
var objectStore = db.createObjectStore("employee", {
    keyPath: "id", autoIncrement: true });
for (i in employeeData) {
    objectStore.put(employeeData[i]);
    alert("Record added");
}
```

© Aptech Ltd.      HTML5 Web Storage / Session 18      27

**HTML5 Implementing IndexedDB API 5-6**

- **Creating a Transaction**

To perform database operation, such as retrieving data from the object store, IndexedDB provides a `IDBTransaction` object.

This object can be created in three mode namely, read-only, read-write, and snapshot.

- Code snippet demonstrates the code to retrieve data from the employee object store using the `get()` function of the transaction object.

```
var trans = db.transaction(["employee"], IDBTransaction.READ_WRITE).objectStore("employee");
var request = trans.get(2);
request.onerror = function(event) {
    // Handle errors!
};
request.onsuccess = function(event) {
    // Do something with the request.result!
    alert("Name: " + request.result.name);
};
```

© Aptech Ltd.      HTML5 Web Storage / Session 18      28

**HTML5 Implementing IndexedDB API 6-6**

**Opening a Cursor**

Cursor is used to retrieve multiple records from an object store.

They can be used when the value of key path is not known. They are part of a transaction and are opened for a particular object store.

- Code snippet demonstrates the code to retrieve multiple records from the employee object store.

```
store = db.transaction("employee").objectStore("employee");
store.openCursor().onsuccess = function(event) {
  var cursor = event.target.result;
  if (cursor) {
    alert("Name for id " + cursor.key + " is " + cursor.value.name);
    cursor.continue();
  }
};
```

© Aptech Ltd.      HTML5 Web Storage / Session 18      29

Using slides 24 to 29, explain how to implement IndexedDb API.

The steps to implement the IndexedDB API in a Web application are as follows:

- Open a database
- Create an object store
- Start a transaction
- Perform some database operations, such as add and retrieve
- Work with the retrieved results

Explain the code snippet using slide 25 which shows the code to open a database.

Here, the code detects the support for IndexedDB API in different browsers and creates the indexedDB object. The indexedDB object contains a method named `open()` which opens the CompanyDB database.

In case, if the database exists, then `open()` method simply opens it, otherwise first creates the database. The `open()` method returns an `IDBRequest` object named `request`. The `request` object provides handlers, such as `success` and `error`. These handlers are invoked depending on the success or failure of opening the database. The `onsuccess()` handler contains an event of type `success` as its argument. Similarly, `onerror()` handler is invoked with an `error` event as its argument.

Explain the step for updating version of database using slide 26.

After the database is opened, it can be structured by providing a version number. This helps to set up the database. The version number will be specified to a database in the `onsuccess()` function.

Explain the code snippet which shows the code that specifies the version number to the database CompanyDB.

Explain the step for creating an object store using slide 27.

The structure of IndexedDB database facilitates the storage of multiple object stores. Object store is created using `createObjectStore()` method. The `createObjectStore()` method accepts two arguments namely, the store name and a parameter `object`. The `parameter object` is used for defining an optional property which is important. In this case, a key path is defined that is used for identifying unique objects in the object store. For example, an `employee` store contains the `id` property as key path, which will be unique for each object and must be present for each object.

Explain the code snippet which demonstrates the code to create an object store named `employee` in the CompanyDB database.

The code creates an array named `employeeData` containing name and e-mail values. Then, the `createObjectStore()` method creates an `employee` store with its key path set to `id` attribute. The key path is used with `autoIncrement` option that automatically generates `id` for each of the objects. All the individual objects in the object store are identified based on the `id`. Finally, the `for .. in` loop stores the data in the `employee` object store.

Explain the step for creating a transaction using slide 28.

To perform database operation, such as retrieving data from the object store, IndexedDB provides an `IDBTransaction` object. This object can be created in three modes namely, read-only, read-write, and snapshot. The read-write mode is used to update the object, whereas read-only mode is used for other operations.

Explain the code snippet which demonstrates the code to retrieve data from the employee object store using the `get()` function of the transaction object.

In the code, the `transaction()` method accepts two parameters. The second parameter is optional. The first parameter is the list of the object stores that are extended by the `transaction` object. In this case, there is a single object store named `employee`, created in the database. The optional second parameter specifies the type of the transaction, that is, read-only, read-write, or snapshot. Here, the transaction type is defined as `IDBTransaction.READ_WRITE`. This type allows reading as well as writing in the database.

The `employee` object store is retrieved from the `transaction` object on which operations are performed. Here, `get()` method is invoked on the `employee` object store

which returns the value against the key path 2. Finally, the result of the get () method is stored in the request object on which callback functions, such as `onsuccess` and `onerror` are invoked.

Explain the step for opening a cursor using slide 29.

Cursor is used to retrieve multiple records from an object store. They can be used when the value of key path is not known. They are part of a transaction and are opened for a particular object store.

Explain the code snippet which demonstrates the code to retrieve multiple records from the `employee` object store. Here, in the code, the `transaction` object is created for the `employee` object store. Then, the `openCursor()` function is invoked on the object store. If the cursor is successfully opened, a cursor object is returned which will retrieve the data from the object store.

#### **Tips:**

Visit the Web site to understand more on IndexedDB API

<http://code.tutsplus.com/tutorials/working-with-indexeddb--net-34673>

#### **Slide 30**

Let us understand the limitations of IndexedDB API.

**HTML5** **Limitations of IndexedDB API**

- Design limitations for IndexedDB API used for client-side storage of data are as follows:

Internationalized sorting deals with sorting of string data. As the database does not follow any international order for storing data, internationalized sorting is not supported by the API.

IndexedDB API does not synchronize client-side database with the server-side databases.

IndexedDB API supports querying the client-side database, but does not support the use of operators, such as LIKE that is used by Structured Query Language (SQL).

© Aptech Ltd.      HTML5 Web Storage / Session 18      30

Using slide 30, explain the limitations of IndexedDB API.

The IndexedDB API is used for client-side storage of data, but it has some design limitations. Explain these limitations to the students mentioned on the slide.

## Slide 31

Let us understand how to convert HTML5 apps to native apps.

The slide has a header 'HTML 5 Converting HTML5 apps to Native apps'. It contains three bullet points:

- A native application also known as native app is an application program that is built for using it on a particular device or platform.
- A native app, when compared with Web app is installed on a device and has a faster response, because it has a direct user interface.
- BlackBerry Messenger (BBM) is a native app available on blackberry mobile devices.

At the bottom left is the copyright notice '© Aptech Ltd.', at the bottom center 'HTML5 Web Storage / Session 18', and at the bottom right '31'.

Using slide 31, explain the conversion of HTML5 apps to native apps.

A native application is also known as native app. It is an application program that is built to be used on a particular device or platform.

A native app, when compared with Web app, is installed on a device and has a faster response, because it has a direct user interface.

For example, BlackBerry Messenger (BBM) is a native app available on BlackBerry mobile devices.

Tell them that there are many frameworks that allow you to convert the HTML application into native app.

## Slide 32

Let us understand the difference between native apps and HTML5 apps.

**Difference between Native Apps and HTML5 Apps**

HTML5 Web apps are accessible and used on any devices through Web browser similar to the mobile Web site.

Web apps have the ability of offline access which means that the user need not have a network connection.

- Following table lists differences between the native apps and HTML5 apps.

Native Apps	HTML5 Apps
Native Apps runs on iOS and Android devices that can be downloaded or purchased from the online app stores.	HTML5 Apps runs on a Web server, usually in a Web browser.
Native Apps use programming language, such as Java for Android devices and Objective C for iOS devices.	Web developers use HTML, JavaScript, and CSS. They need to acquire the skills of Java and objective C for writing native applications.

Using slide 32, explain the difference between native apps and HTML5 apps.

With HTML5 release, a discussion started about HTML5 which involved whether HTML5 can develop native mobile apps (that is apps designed especially for Blackberry, iPhone, Android, and so on).

HTML5 is the latest version of HTML language providing a simple building block for Web pages. This is the first version of markup language that supports the use of multimedia without using any additional plug-in and is supported by many devices and computer systems.

HTML5 Web apps are accessible and used on any devices through Web browser similar to the mobile Web site. The Web apps have the ability of offline access which means that the user need not have a network connection.

Explain the list of differences between the native apps and HTML5 apps.

## Slide 33

Let us understand the advantages of HTML5 apps.

The slide has a header 'HTML 5 Advantages of HTML5 Apps'. Below the header is a bulleted list:

- Main advantage of using HTML5 is to create applications that executes on a wide range of devices easily. Some of the reasons to develop HTML5 applications are as follows:

**Users cannot identify the differences** - Cannot identify whether they are working on a hybrid HTML5-native application or a fully native application or an HTML5 application.

**Users adjust styles for devices** - HTML5 apps can be viewed on any devices that contains Web browser.

**Upcoming functionalities** - HTML5 does not support all features on a device, but it is coming up with new functionalities.

**Improving Performance** - Many developers learn new methods to improve the performance of Web.

**Independent device** - If the developers want that their application to be used by a large number of users, then they should design and develop applications for both mobile users as well as desktop users.

**Developers are not locked in app stores** - HTML5 developers are not restricted to an app store. Instead, they can create applications and sell them like any other Web page.

© Aptech Ltd.      HTML5 Web Storage / Session 18      33

Using slide 33, explain the advantages of HTML5 apps.

The main advantage of using HTML5 is to create applications that execute on a wide range of devices easily. App development on HTML5 is cheaper as compared to native app development. Developers do not have to learn any new programming language and the development becomes much easier.

There are many reasons to develop HTML5 applications rather than native applications. Explain these reasons mentioned on the slide.

## Slide 34

Let us understand advantages of native apps.

**Advantages of Native Apps**

- Major advantage of native apps over HTML5 apps is that they are faster than HTML5 apps. Native apps provide more benefits over HTML5 apps. These are as follows :

- Providing access to device hardware** - There are no APIs available for accelerometers, cameras, or any other device hardware for HTML5 apps.
- Uploading Files** - Native apps can access the file system in Android and some files in iOS. However, the HTML5 file API does not work on Android or iOS.
- Push Notifications** - The push notifications are sent always with an open IP connection to applications on the iOS device.
- Accessing device files** - Native apps communicate with files on devices, such as contacts and photos. However, these files cannot be seen from HTML5 app.
- Superior graphics than HTML5** - HTML5 has a canvas element, but it will not create a full 3D experience.
- Offline access** - HTML5 provides access to offline Web applications. However, a native app is stored on local machine, so the users does not require access to the Web to work with the application.

Using slide 34, explain the advantages of native apps.

The major advantage of native apps over HTML5 apps is that they are faster than HTML5 apps. Similar to normal Web pages, HTML5 apps are slow, because these apps work on HTTP that uses a request/response cycle mechanism. When an HTTP request is made, it takes more time for the applications to execute, as it has to wait for the request to go and return back with a response.

Native apps provide many more benefits over HTML5 apps. Explain these benefits mentioned on the slide.

## Slide 35

Let us understand the concept of converting HTML5 apps to native apps.

**HTML5** Converting HTML5 Apps to Native Apps

- Users have a choice of developing their application in HTML5 and convert them into a native app
- Users can use some tools to convert an HTML5 app to Native app and they are as follows:

**PhoneGap** - Is an HTML5 app that allows the user to create native apps with Web technologies and is accessible to app stores and APIs.

**Appcelerator** - Is a cross-platform mobile application development support and allows the users to create Android, iOS, and mobile Web apps.

© Aptech Ltd.      HTML5 Web Storage / Session 18      35

Using slide 35, explain the concept of converting HTML5 apps to native apps.

Users can use tools to convert HTML5 app to a native app. The following are the best tools used for converting an HTML5 app to native app:

- **PhoneGap**

PhoneGap is an HTML5 app that allows the user to create native apps with Web technologies and is accessible to app stores and APIs. PhoneGap controls the Web technologies.

- **Appcelerator**

Appcelerator is a cross-platform mobile application development support. It allows the users to create Android, iOS, and mobile Web apps. Native applications are developed using a JavaScript code base with Eclipse as the IDE.

## Slide 36

Let us summarize the session.

**HTML5 Summary**

- Web Storage is a W3C specification that provides functionality for storing data on the client-side for both temporary as well as permanent needs.
- HTML5 Web applications make use of Web storage to implement client-side persistent storage and they are: session storage and local storage.
- Session storage keeps track of data specific to one window or tab.
- The `setItem()` and `getItem()` methods are used to store and retrieve the data from session storage.
- Local storage enables to save data for longer periods on the user's computer, through the browser.
- IndexedDB API is basically an object store that can be used to store and manipulate data on the client-side.
- A native application also called as native app is an application program that is built for a particular device or platform.

© Aptech Ltd.      HTML5 Web Storage / Session 18      36

Using slide 36, summarize the session. End the session, with a brief summary of what has been taught in the session.

### 18.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session. You should also explore the HTML5 Geolocation and APIs that are offered with the next session.

#### Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.

# Session 19 – HTML5 Geolocation and APIs

---

## 19.1 Pre-Class Activities

Familiarize yourself with the topics of this session in-depth. You should revisit topics of the previous session for a brief review.

Here, you can ask students the key topics they can recall from previous session. Prepare a question or two which will be a key point to relate the current session objectives.

### 19.1.1 Objectives

By the end of this session, the learners will be able to:

- Explain geolocation and its use in HTML5
- Explain the Google Maps API
- Explain the drag-and-drop operations in HTML5
- Explain the concept of Application Cache

### 19.1.2 Teaching Skills

To teach this session, you should be well-versed with the new APIs supported by HTML5 such as Geolocation API and Google Maps API that are used to determine and display the location on a map. Along with this, you should prepare yourself to explain the drag-and-drop mechanism which is used to perform the drag-and-drop operations. Also, learn about the description of Application Cache.

You should teach the concepts in the theory class using the images provided. For teaching in the class, you are expected to use slides and LCD projectors.

#### Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

#### In-Class Activities:

Follow the order given here during In-Class activities.

#### Overview of the Session:

Then give the students the overview of the current session in the form of session objectives. Show the students slide 2 of the presentation.

**HTML5 Objectives**

- Explain geolocation and its use in HTML5
- Explain the Google Maps API
- Explain the drag-and-drop operations in HTML5
- Explain the concept of Application Cache

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      2

Tell the students that this session introduces the new APIs supported by HTML5. They will learn about the Geolocation API and Google Maps API used to determine and display the location on a map. They will also know about the drag-and-drop mechanism which is used to perform the drag-and-drop operations and the description of Application Cache.

## 19.2 In-Class Explanations

### Slides 3 and 4

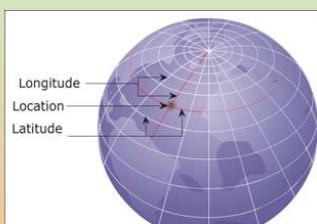
Let us understand the term, Geolocation.

**HTML5 Geolocation 1-2**

Geolocation in computing terminology determines the current location of a user on the devices.

The location of the user is represented as a single point that comprises two components: latitude and longitude.

- Following figure shows the representation of latitude and longitude with respect to a location on the globe.



© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      3

The different sources through which devices can determine the information about the location are as follows:

- Global Positioning System (GPS)**
  - GPS is a satellite navigation system that provides information about the location on any part of the globe.
  - The GPS system is maintained by the government of the United States.
- IP Address**
  - Location information can be derived from IP Address which is assigned to devices, such as desktops, printers, and so on connected on a network.
- GSM/CDMA Cell IDs**
  - These are used by the cell phones.
- WiFi and Bluetooth MAC address**
  - These are used by devices that have wireless network connection.
- User Input**
  - It is a software tool which can be used on any device requesting for location information.
  - The information retrieved by the tool is based on the data provided by the user. For example, a zip code.

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      4

Using slides 3 and 4, explain the term, Geolocation.

Consider a scenario where you are visiting a new city and are unaware of specific locations and routes. You want to get information regarding hotels in your locality, such as their exact address, tariffs, and so on. In such a situation, an application which can provide relevant information about the hotels based on your current location would be useful. A feature that can detect location and list relevant information based on that location is called Geolocation.

Geolocation is a term used to identify the geographic location of a person, place, or an object. Today, modern devices such as computers, smartphones, tablets, and so on provide Internet-enabled browsers through which the geographic locations of a user or an object can be detected.

Geolocation in computing terminology indicates a feature that determines the current location of a user on devices. The location of the user is represented as a single point that comprises two components, latitude and longitude. The components can be used further to retrieve more information for the user, such as businesses in the neighborhood or other users within the same coverage area.

Figure shows the representation of latitude and longitude with respect to a location on the globe.

There are different sources through which devices can determine the information about the location. Explain these with the help of slide 4.

**In-Class Question:**

After you finish explaining Geolocation, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which source is based on satellite navigation to provide information about any location on the globe?

**Answer:**

GPS

**Slides 5 and 6**

Let us understand the Geolocation API.

**HTML5 Geolocation API 1-2**

- In HTML5, the Geolocation API is a specification by W3C for providing a consistent way to develop location-aware Web applications.
- The Geolocation API provides a high-level interface to retrieve location information related to the hosting devices.
- The interface hides the details, such as how the information is gathered or which methods were used to retrieve the information.
- The object that holds implementation of the Geolocation API is the Geolocation object.
- This object is used in JavaScript to retrieve the geographic information about the devices programmatically.
- The browser processes the script and returns the location to the Geolocation API.
- The Geolocation API is supported on most of the modern browsers available on desktop and mobile phones.

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      5

**HTML5 Geolocation API 2-2**

- Following table lists the browsers providing support for Geolocation API.

Browser	Version Support
Safari	5.0+
Chrome	5.0+
Firefox	3.5+
Internet Explorer	9.0+
Opera	10.6+
iOS (Mobile Safari)	3.2+
Android	2.0+
Blackberry	6+

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      6

Using slides 5 and 6, explain the Geolocation API.

The Geolocation API provides a high-level interface that can be used by developers to retrieve location information related to the hosting devices. The interface hides the details, such as how the information is gathered or which methods were used to retrieve the information. This helps the developer to concentrate on geographic information rather than its processing methods.

The object that holds implementation of the Geolocation API is the `Geolocation` object.

`Geolocation` object is used in JavaScript to retrieve the geographic information about the devices programmatically. The browser processes the script and returns the location to the Geolocation API.

Explain the list of the browsers providing support for Geolocation API.

Mention that, it is not necessary that information retrieved by the Geolocation API is the actual location of the device. For example, if the satellites are invisible to GPS, then it may not return the accurate location information. Geolocation is much more accurate for devices with GPS, such as iPhone.

### Slides 7 to 9

Let us understand how to implement Geolocation object.

The screenshot shows a presentation slide with a blue header containing the text "HTML5 Implementing Geolocation Object 1-3". The main content area has a yellow background. It contains several text boxes and a list:

- A red box states: "The Geolocation object is available as a new property of the navigator object."
- A green box states: "The navigator object is a browser object that allows a user to retrieve information about the specific location."
- A bulleted list: "Following syntax shows how to create a Geolocation object in JavaScript."
- A section titled "Syntax:" with the code: "var geolocation = window.navigator.geolocation;"
- An explanatory note: "where," followed by a bullet point: "window: Is the top level object in JavaScript object hierarchy."

At the bottom left is the copyright notice "© Aptech Ltd.", and at the bottom right is the page number "HTML5 Geolocation and APIs / Session 19 7".

**HTML 5 Implementing Geolocation Object 2-3**

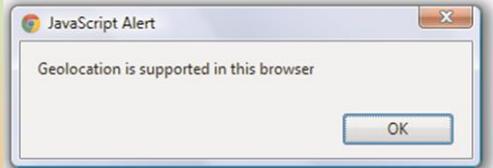
- The Code Snippet demonstrates the script that tests the existence of geolocation object within a browser.

```
<!DOCTYPE html>
<html>
  <head>
    <title> Testing Support for Geolocation in Browsers</title>
    <script>
      function display_location_enabled()
      {
        // Default message
        var str = "Geolocation is not supported in this browser";
        if (window.navigator.geolocation)
        {
          str = "Geolocation is supported in this browser";
        }
        alert (str);
      }
    </script>
  </head>
  <body>
    <input type="button" value="Geolocation Support"
          onClick="display_location_enabled()"></input>
  </body></html>
```

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      8

**HTML 5 Implementing Geolocation Object 3-3**

- In the code, the 'if' statement checks existence of the geolocation property in the browser.
- If browser provides implementation for the property, then an alert window displays the message 'Geolocation is supported in this browser'.
- Otherwise, the default message is displayed.
- Following figure shows the existence of geolocation object in the Chrome browser.



© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      9

Using slides 7 to 9, explain how to implement Geolocation object.

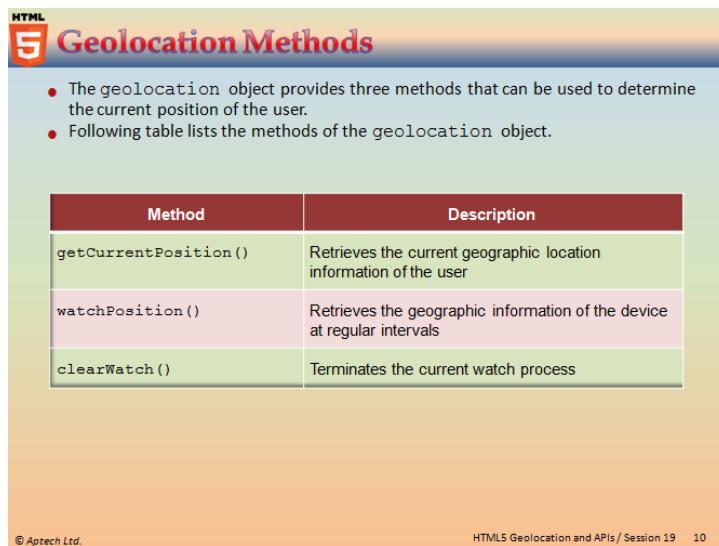
The Geolocation object is available as a new property of the navigator object. The navigator object is a browser object that allows a user to retrieve information about the specific location.

Explain the syntax of creating Geolocation object.

Then, explain the code snippet which demonstrates the script that tests the existence of Geolocation object within a browser.

## Slide 10

Let us understand Geolocation methods.



The slide title is "Geolocation Methods". It contains two bullet points: "The geolocation object provides three methods that can be used to determine the current position of the user." and "Following table lists the methods of the geolocation object." Below this is a table with three rows:

Method	Description
<code>getCurrentPosition()</code>	Retrieves the current geographic location information of the user
<code>watchPosition()</code>	Retrieves the geographic information of the device at regular intervals
<code>clearWatch()</code>	Terminates the current watch process

At the bottom left is the copyright notice "© Aptech Ltd." and at the bottom right is the page number "HTML5 Geolocation and APIs / Session 19 10".

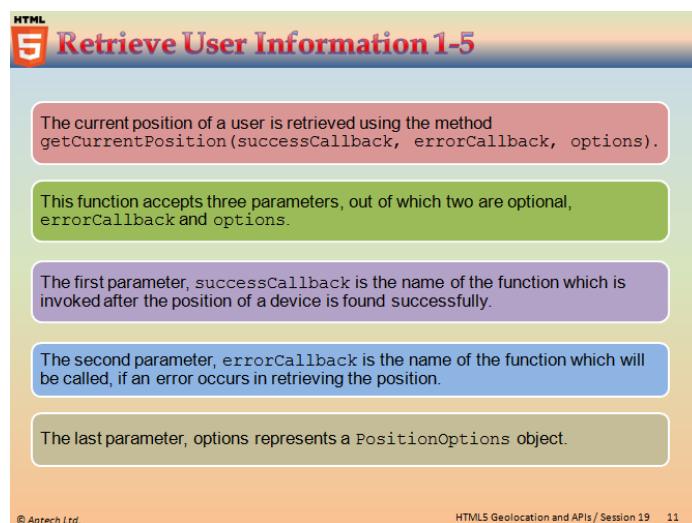
Using slide 10, explain Geolocation methods.

The Geolocation object provides three methods that can be used to determine the current position of the user.

Explain these methods as listed in the table on the slide.

## Slides 11 to 15

Let us understand the process of retrieving user information.



The slide title is "Retrieve User Information 1-5". It contains five callout boxes with the following text:

- The current position of a user is retrieved using the method `getCurrentPosition(successCallback, errorCallback, options)`.
- This function accepts three parameters, out of which two are optional, `errorCallback` and `options`.
- The first parameter, `successCallback` is the name of the function which is invoked after the position of a device is found successfully.
- The second parameter, `errorCallback` is the name of the function which will be called, if an error occurs in retrieving the position.
- The last parameter, `options` represents a `PositionOptions` object.

At the bottom left is the copyright notice "© Aptech Ltd." and at the bottom right is the page number "HTML5 Geolocation and APIs / Session 19 11".

**HTML5 Retrieve User Information 2-5**

- The Code Snippet demonstrates the script that will retrieve the current location of the user.

```
<!DOCTYPE html>
<html>
<head>
<title>Geolocation API</title>
<script>
function getLocation()
{
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(showPosition);
    }
    else{
        alert ("Geolocation is not supported in this browser.");
    }
}
function showPosition(position)
{
    alert('Latitude: ' + position.coords.latitude + '\n' +
          'Longitude: ' + position.coords.longitude);
}
</script>
</head>
```

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19    12

**HTML5 Retrieve User Information 3-5**

```
<body>
<input type="button" value=" Display Location"
       onClick="getLocation()"/>
</body>
</html>
```

- In the code, the `getCurrentPosition()` function obtains the position which is passed as a parameter to the `showPosition()` function.
- The `showPosition()` function obtains the coordinates of a location through `position` object.
- The `position` object is defined in the Geolocation API and holds the current location of the device.
- It contains attribute named `coords` that retrieves the latitude and longitude of the location.
- The values retrieved for latitude and longitude are in decimal degrees.

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19    13

**HTML5 Retrieve User Information 4-5**

- Following table lists the attributes of the `position` object.

Attribute	Description
<code>coords</code>	An object of type <code>Coordinates</code> that provides different properties, such as latitude, longitude, altitude, accuracy, speed, and so on.
<code>timestamp</code>	An object of type <code>DOMTimeStamp</code> .

- Following figure shows the notifications for the Web page containing geolocation code.

- The browser seeks permission from the user to share their location information with the application.

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19    14



Using slides 11 to 15, explain the steps and code snippet to retrieve user information.

The current position of a user is retrieved using the `getCurrentPosition(successCallback, errorCallback, options)` method. This function accepts three parameters, out of which two are optional, `errorCallback` and `options`.

- The first parameter, `successCallback` is the name of the function which is invoked after the position of a device is found successfully.
- The second parameter, `errorCallback` is the name of the function which will be called, if an error occurs in retrieving the position.
- The last parameter, `options` represents a `PositionOptions` object.

Explain the code snippet using slide 12 which demonstrates the markup that will retrieve the current location of the user.

In the code, the `getCurrentPosition()` function obtains the position which is passed as a parameter to the `showPosition()` function. The `showPosition()` function obtains the coordinates of a location through `position` object.

The `position` object is defined in the Geolocation API and holds the current location of the device. It contains attribute named `coords` that retrieves the latitude and longitude of the location. The values retrieved for latitude and longitude are in decimal degrees.

Explain the list of attributes of the `position` object on slide 14.

Figure shows the notifications for the Web page containing geolocation code. The browser seeks permission from the user to share their location information with the application.

Figure shows a message displaying current location of the user, when the **Share My Location** button is clicked.

**In-Class Question:**

After you finish explaining Geolocation methods, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What are the parameters of the `getCurrentPosition()` method?

**Answer:**

It has three parameters, `successCallback`, `errorCallback`, and `options`.

**Slides 16 to 19**

Let us understand error handling in HTML.

**HTML 5 Handling Errors 1-4**

- An application could fail in gathering geographic location information. In that case, the geolocation object calls an `errorCallback()` function.
- The `errorCallback()` function handles errors by obtaining a `PositionError` object from the API.

**➤ HTML**

- The `PositionError` object holds information related to errors occurred while finding the geographic location of the user.
- Following table lists the properties of `PositionError` object.

Property	Description
<code>code</code>	Returns a numeric value for the type of error occurred.
<code>message</code>	Returns a detailed message describing the error encountered. The message can be used for debugging.

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      16

**HTML 5 Handling Errors 2-4**

- Following table lists the different error codes returned by the `code` property of the `PositionError` object.

Code	Constant	Description
1	<code>PERMISSION_DENIED</code>	Application does not have permission to access Geolocation API.
2	<code>POSITION_UNAVAILABLE</code>	Position of the device could not be obtained.
3	<code>TIMEOUT</code>	Unable to retrieve location information within the specified interval.

- The Code Snippet demonstrates the error handling routine for the geolocation code.

```
<!DOCTYPE html>
<html>
<head>
    <title>Handling Error</title>
```

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      17

**Handling Errors 3-4**

```

HTML
<script>
    function getLocation()
    {
        function showPosition(position)
        {
            alert('Latitude: ' + position.coords.latitude + '\n' +
            'Longitude: ' + position.coords.longitude);
        }
        function errorHandler(error) {
            switch (error.code) {
                case error.PERMISSION_DENIED:
                    alert ('You have denied access to your position.');
                    break;
                case error.POSITION_UNAVAILABLE:
                    alert ('There was a problem getting your position.');
                    break;
                case error.TIMEOUT:
                    alert ('The application has timed out attempting to
                           get your position.');
                    break;
            }
        }
    }
</script> </head>

```

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      18

**Handling Errors 4-4**

```

HTML
<body>
    <input type="button" value="Display Location"
           onClick="getLocation()"/>
</body> </html>

```

- In the code, if application fails to find the current location of the user, then the `errorHandler()` function is invoked.
- The function is passed as the second parameter in the `getCurrentPosition()` method and is used to handle the errors occurred in the application.
- It obtains the `error` object which is of type `PositionError` from the API and compares it with the error codes specified in the `switch-case` statement.
- Depending on the error occurred, the appropriate `case` statement is executed and an alert message is displayed to the user.
- Following figure shows the output displaying error message for geolocation application.

- The reason for the error is that the Chrome browser blocks the URL whose file path starts with file:///.

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      19

Using slides 16 to 19, explain error handling.

An application could fail in gathering geographic location information. In that case, the `Geolocation` object calls an `errorCallback()` function.

The `errorCallback()` function handles errors by obtaining a `PositionError` object from the API. The `PositionError` object holds information related to errors occurred while finding the geographic location of the user.

Mention the properties and methods of the `PositionError` object.

Explain the code snippet using slides 17 and 18 which demonstrates the error handling routine for the geolocation code.

Figure shows the output displaying error message for geolocation application. The reason for displaying error is that the Chrome browser blocks the URL whose file path starts with `file:///`.

## Slides 20 to 22

Let us understand the `PositionOptions` object.

**HTML5 PositionOptions Object 1-3**

- `PositionOptions` object is an optional third parameter passed to the `getCurrentPosition()` method.
- This object defines properties that are optional and are used by an application while retrieving the geolocation information.
- Following table lists the attributes of `PositionOptions` object.

Attribute	Description
<code>enableHighAccuracy</code>	Indicates that the application wants to receive the most accurate results for geolocation. The default value of the attribute is false.
<code>maximumAge</code>	Obtains the cached position object whose age is less than the specified <code>maximumAge</code> limit (in milliseconds). If age limit is set to 0, then the application must obtain a new position object.
<code>timeout</code>	Indicates the maximum time length (in milliseconds) for which the application can wait to obtain the position object.

© Aptech Ltd.

HTML5 Geolocation and APIs / Session 19 20

**HTML5 PositionOptions Object 2-3**

- The Code Snippet demonstrates the script to set the attributes of `PositionOptions` object.

```
<script>
    var options = {
        enableHighAccuracy: true,
        maximumAge: 50000,
        timeout: 60000
    };
    function getLocation() {
        if (navigator.geolocation) {
            navigator.geolocation.getCurrentPosition(showPosition,
                errorHandler, options);
        }
        else{
            alert ("Geolocation is not supported in this browser.");
        }
    }
</script>
```

© Aptech Ltd.

HTML5 Geolocation and APIs / Session 19 21

**HTML5 PositionOptions Object 3-3**

- In the code, an object named `options` is set with attributes.
- The attribute, `maximumAge` enables the application to use a cached position object which is not older than 50 seconds.
- Also, the `timeout` limit is set to 60 seconds for an application, before notifying an error.
- The `options` is passed as third parameter to the `getCurrentPosition()` method.

© Aptech Ltd.

HTML5 Geolocation and APIs / Session 19 22

Using slides 20 to 22, explain `PositionOptions` object.

`PositionOptions` object is an optional third parameter passed to the `getCurrentPosition()` method. This object defines properties that are optional and are used by an application while retrieving the location information.

Explain the attributes of the `PositionOptions` object and code snippet that demonstrates the script to set the attributes of `PositionOptions` object.

In the code, an object named `options` is set with attributes. The attribute `maximumAge` enables the application to use a cached `position` object which is not older than 50 seconds. Also, the `timeout` limit is set to 60 seconds for an application, before notifying an error.

The `options` is passed as third parameter to the `getCurrentPosition()` method.

### Slides 23 to 28

Let us understand Google Maps API.

**HTML5 Google Maps API 1-6**

Google Maps API is used to display locations on a map based on the values of their coordinates, latitude and longitude.

The Google Maps API must be configured in JavaScript, before it can be referenced further on the page.

It contains a `Map` object which is instantiated and displayed on a Web page.

- Following syntax shows the configuration of Google Maps API in JavaScript.

**Syntax:**

```
<script src="http://maps.google.com/maps/api/js?sensor=false">
</script>
```

where,

- src: Is the URL of Google Maps API.
- sensor: Parameter sent with the URL. It indicates whether application uses any sensor such as GPS system.

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      23

**HTML5 Google Maps API 2-6**

- The Code Snippet demonstrates how to load and initialize the Google Maps API in the `<script>` tag.
- The code will execute after the page is loaded completely and will invoke a function in response to the `onload` event.

```
<!DOCTYPE html>
<html>
  <head>
    <title> Load and Initialize Google Maps </title>
    <style>
      html { height: 100% }
      body { height: 100%; width: 100%; margin: 10% }
      #map_canvas { height: 50%; width: 50% }
    </style>
  <script
    src="http://maps.google.com/maps/api/js?sensor=false">
  </script>
```

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      24

**HTML5 Google Maps API 3-6**

```

function initialize()
{
    // Loading Google Maps
    var num = new google.maps.LatLng(51.528663,-0.173171);
    var myOptions = {
        zoom: 16,
        center: num,
        mapTypeId: google.maps.MapTypeId.HYBRID
    };
    var mymap = new google.maps.Map(document.getElementById("map_canvas"), myOptions);
    var marker = new google.maps.Marker({
        position: num,
        map: mymap,
        title:"Lord's Cricket Ground, London!"
    });
}
</script>
</head>
<body onload="initialize()">
    <div id="map_canvas"></div>
</body>
</html>

```

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      25

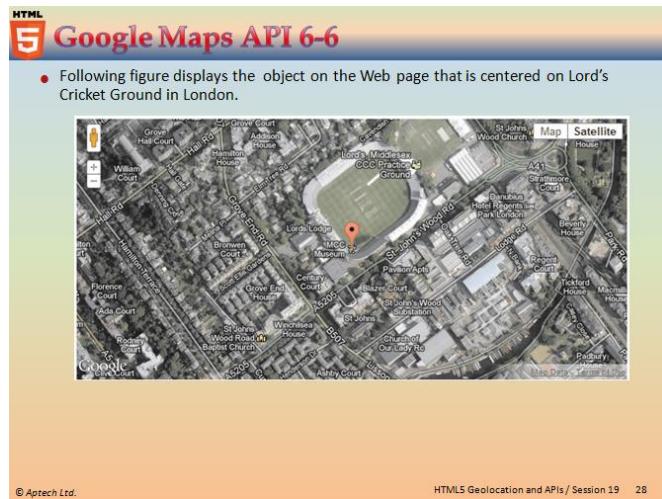
- HTML5 Google Maps API 4-6**
- In the code, the URL 'http://maps.google.com/maps/api/js ? sensor=false' defines all symbols and definitions to be loaded for Google Maps API.
  - Then, the function `initialize()` is invoked after the page is loaded completely.
  - This function creates the object of type `Map` and initializes it with the map initialization variables.
  - In the function, `var myOptions = {};` is an object of type `options` that contains properties, such as `zoom`, `center`, and `mapTypeId`.
  - These properties are used to initialize the map.
  - Then, the statement `new google.maps.Map (document.getElementById ("map_canvas"), myOptions);` creates an instance of `Map` object.
  - The object is displayed in a container on the Web page specified with the `<div>` element.
  - Finally, to display an icon on the identified location on the Google maps, the `Marker` object is created.
  - The `Marker` object's constructor sets the value for the properties, such as `position`, `map`, and `title`.
  - The `position` property is specified with the location of the marker on the map.
  - The `map` property is specified with the `Map` object to attach the marker with the map.
  - The `title` property sets the title to be displayed as a tooltip on the map.
- © Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      26

**HTML5 Google Maps API 5-6**

- Following table lists some of the `myOptions` properties.

Property	Description
<code>zoom</code>	Sets the initial resolution at which map is displayed. A lower zoom value 0 represents a full map of the Earth. Similarly, a higher zoom value displays a map with high resolution.
<code>center</code>	Centers the map on a specific point by creating an object of type <code>LatLng</code> which holds the location coordinates.
<code>mapTypeId</code>	Sets an initial map type. The map types supported are: <code>ROADMAP</code> for normal, <code>SATELLITE</code> for photographic tiles, <code>HYBRID</code> for roads and city names, and <code>TERRAIN</code> for water features.

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      27



Using slides 23 to 28, explain the Google Maps API.

The Google Maps API is used to display locations on a map, based on the values of their coordinates - latitude and longitude. The Google Maps API must be configured in JavaScript, before it can be referenced further on the page. It contains a Map object which is instantiated and displayed on a Web page.

Explain the syntax that shows the configuration of Google Maps API in JavaScript.

Explain the code snippet which demonstrates how to load and initialize the Google Maps API in the `<script>` tag. The code will execute after the page is loaded completely and will invoke a function in response to the `onload` event.

In the code, the URL `http://maps.google.com/maps/api/js?sensor=false` defines symbols and definitions to be loaded for the Google Maps API. Then, the function `initialize()` is invoked after the page is loaded completely. This function creates the object of type `Map` and initializes it with the map initialization variables.

In the function, `var myOptions = {}`, is an object of type `options` that contains properties, such as `zoom`, `center`, and `mapTypeId`. These properties are used to initialize the map.

Then, the statement, `new google.maps.Map (document.getElementById ("map_canvas"), myOptions);` creates an instance of `Map` object. The object is displayed in a container on the Web page specified with the `<div>` element.

Finally, to display an icon on the identified location on the Google maps, the `Marker` object is created. The `Marker` object's constructor sets the value for the properties, such as `position`, `map`, and `title`. The `position` property is specified with the location of the marker on the map. The `map` property is specified with the `Map` object to attach the marker with the map. Also, the `title` property sets the title to be displayed as a tooltip on the map. As mentioned earlier, the `myOptions` object has several properties.

Explain some of its properties listed in the table.

Figure displays the Map object on the Web page that is centered on **Lord's Cricket Ground in London**.

### In-Class Question:

After you finish explaining Google Maps API, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



Which of the following properties are used to initialize the map?

### Answer:

`zoom`, `center`, and `mapTypeId`

### Slides 29 to 31

Let us understand how to track user's location.

**HTML 5 Tracking User's Location 1-3**

- The Geolocation object is used by the Google Maps API to display the geolocation information in the applications.
- The Code Snippet demonstrates the code that displays current location of a user on the map using Geolocation object.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <style>
      html, body {
        width: 100%;
        height: 100%;
        padding: 10%
      }
      #map_canvas {
        height: 50%;
        width: 50%;
      }
    </style>
  <script src="http://maps.google.com/maps/api/js?sensor=false">
  </script>
```

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      29

29 (a)

**HTML5 Tracking User's Location 1-3**

- The Geolocation object is used by the Google Maps API to display the geolocation information in the applications.
- The Code Snippet demonstrates the code that displays current location of a user on the map using Geolocation object.

```
<script>
    // Check support for Geolocation in the browser
    if (navigator.geolocation) {
        // Locate position and invoke function
        navigator.geolocation.getCurrentPosition(displayPosition,
            errorFunction);
    }
    else {
        alert('Geolocation is not enabled in your browser');
    }

```

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19    29

**29 (b)**

**HTML5 Tracking User's Location 2-3**

```
// Success function
function displayPosition(position) {
    var my_lat = position.coords.latitude;
    var my_lng = position.coords.longitude;
    var div_info = document.getElementById('user_location');
    div_info.innerHTML = '<h1> Latitude is : ' + my_lat + ' and
Longitude is ' + my_lng + '</h1>';

// Load Google Maps
var latlng = new google.maps.LatLng(my_lat, my_lng);
var myOptions = {
    zoom: 2, //the initial resolution is set at which map is
              //displayed
    center: latlng, //centers the map
    mapTypeId: google.maps.MapTypeId.ROADMAP //sets the map type
};


```

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19    30

**30 (a)**

**HTML5 Tracking User's Location 2-3**

```
// Creates the Map object
var map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);

// Displays icon on the located position
var marker = new google.maps.Marker({
    position: latlng,

    map: map,
    title:"User location"
});

// Error callback function
function errorFunction(pos) {
    alert('Error!');
}

</script> </head>
<body>
    <div id="map_canvas"></div>
    <div id="user_location"></div>
</body> </html>
```

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19    30

**30 (b)**

**HTML5 Tracking User's Location 3-3**

- The code uses the `getCurrentPosition()` method and retrieves the current position of the user.
- Then, it passes the information to `displayPosition()` function, which retrieves the coordinates, latitude and longitude.
- The retrieved coordinates are set into the properties of the `Options` object named `myOptions` and initialize the `Map` object.
- Finally, the `Map` object is displayed along with the current position information in the `<div>` element.
- Following figure shows the output displaying the current location of the user on the Google Maps.

Latitude is :19.017656 and Longitude is 72.856178

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      31

Using slides 29 to 31, explain the code to track user's location.

The `Geolocation` object is used by the Google Maps API to display the geolocation information in the applications.

Explain the code snippet which demonstrates the code that displays current location of a user on the map using `Geolocation` object.

The code uses the `getCurrentPosition()` method and retrieves the current position of the user. Then, it passes the information to `displayPosition()` function, which retrieves the coordinates, latitude and longitude. The retrieved coordinates are set into the properties of the `options` object named `myOptions` and initialize the `Map` object. Finally, the `Map` object is displayed along with the current position information in the `<div>` element. Figure shows the output displaying the current location of the user on the Google Maps.

## Slide 32

Let us understand drag-and-drop operations.

**HTML5 Drag and Drop**

HTML5 defines drag-and-drop operations that are based on events. Currently, drag-and-drop operations are supported by all major browsers.

The event-based mechanism allow the elements to be copied, reordered, or deleted on a Web page.

The drag-and-drop operation involves the use of a pointing device, such as mouse on a visual medium.

To perform the drag operation, a mousedown event is triggered followed by multiplemousemove events.

Similarly, the drop operation is performed when a user releases the mouse.

The benefit of drag-and-drop mechanism is that it has brought the drag-and-drop operations on the browser level.

This makes the programming easier, thus eliminating the need of complex JavaScript code written in earlier HTML versions.

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      32

Using slide 32, explain drag-and-drop operations.

HTML5 defines drag-and-drop operations that are based on events. The event-based mechanism allows the elements to be copied, reordered, or deleted on a Web page. The drag-and-drop operation involves the use of a pointing device, such as mouse on a visual medium. To perform the drag operation, a mousedown event is triggered followed by multiple mousemove events. Similarly, the drop operation is performed when a user releases the mouse.

Then, explain the benefits of drag-and-drop operations listed on the slide.

Currently, drag-and-drop operations are supported by all major browsers.

### Slide 33

Let us understand the drag operation.

**Drag Operation**

- The steps required to make any element draggable on a Web page are as follows:
  - Set the `draggable` attribute of an element to be dragged.
  - Set an `ondragstart` event on the element which stores the data being dragged.
  - Store the data into the `DataTransfer` object.
- The Code Snippet shows how to set the `draggable` attribute of an image element.
 

```
<!DOCTYPE html>
<html>
  <head>
    <title>Drag and Drop API</title>
  </head>
  <body>
    <div id="div" style="border: 2px solid; height:125px;
      width:75px; padding: 10px">
      
    </div>
  </body> </html>
```

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      33

Using slide 33, explain the drag operation.

The steps required to make any element draggable on a Web page are as follows:

- Set the `draggable` attribute of an element to be dragged
- Set an `ondragstart` event on the element which stores the data being dragged
- Store the data into the `DataTransfer` object

Explain the code snippet which shows how to set the `draggable` attribute of an image element.

In the code, the `<img>` element contains `draggable` attribute that is set to `true`. The value `true` indicates that the element is eligible for dragging.

## Slide 34

Let us understand drag events.

**Drag Events**

- During various stages of the drag-and-drop operation, a number of events are fired.
- These events are mouse-based events.
- Following table lists the various events triggered during the drag operation.

Event	Description
dragstart	Triggers when an element is started to be dragged by the user.
drag	Triggers when an element is being dragged using a mouse.
dragleave	Triggers when the drag and drop operation is completed.

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      34

Using slide 34, explain drag events.

During various stages of the drag-and-drop operation, a number of events are fired. These events are mouse-based events.

Explain the various events triggered during the drag operation.

## Slides 35 and 36

Let us understand the `dataTransfer` object.

**DataTransfer Object 1-2**

- The `dataTransfer` object reveals the **drag data store** that contains the dragged data in the drag-and-drop operation.
- It allows getting and setting of the data being dragged.
- In other words, the `dataTransfer` object holds the data during drag-and-drop operation.
- The `dataTransfer` Object enables to define two types of information.
- These are as follows:
  - The data type of the draggable element
  - The value of the data being stored in the data store
- The Code Snippet demonstrates how to associate an element with `dragstart` event to store the data being dragged.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Drag and Drop API</title>
    <script>
      function drag_image(event)
      {
        event.dataTransfer.setData("image", event.target.id);
      }
    </script>  </head>
```

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      35

**HTML 5 DataTransfer Object 2-2**

- In the code, the <img> element has been set with an event listener for the dragstart event.
- When the image is dragged, then the dragstart event is fired and calls drag\_image() function.
- The function uses the dataTransfer object to store the data during drag-and-drop operation.
- The string 'image' represents the data type and event.target.id represents the value of id attribute of the draggable element.
- Following figure shows the output of the image element to be dragged.

**Drag and Drop API**

file:///H:/HTML5/Source

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      36

Using slides 35 and 36, explain the dataTransfer object.

The dataTransfer object reveals the drag data store that contains the dragged data in the drag-and-drop operation. It allows getting and setting of the data being dragged. In other words, the dataTransfer object holds the data during drag-and-drop operation. The dataTransfer object enables to define two types of information. These are as follows:

1. The data type of the draggable element
2. The value of the data being stored in the data store

Explain the code snippet which demonstrates how to associate an element with dragstart event to store the data being dragged.

#### In-Class Question:

After you finish explaining dataTransfer object, you will ask the students an In-Class question. This will help you in reviewing their understanding of the topic.



What is the dataTransfer object used for?

#### Answer:

The dataTransfer object is used to hold the data that is being dragged during a drag-and-drop operation.

## Slide 37

Let us understand drop operation.

The slide has a blue header bar with the text 'HTML 5 Drop Operation'. Below the header, there are five colored callout boxes containing text:

- Red box: After the element has been set up for dragging, it can be dropped in some element on the Web page.
- Green box: By default, elements on the page are not set up to receive dragged elements.
- Purple box: Thus, the behavior of element acting as a drop element must be changed.
- Orange box: This can be done by creating event listeners for the drop element.
- Blue box: The drop element is also referred to as target element.

At the bottom left is the copyright notice '© Aptech Ltd.' and at the bottom right is 'HTML5 Geolocation and APIs / Session 19 37'.

Using slide 37, explain the drop operation.

After the element has been set up for dragging, it can be dropped on some element on the Web page. By default, elements on the page are not set up to receive dragged elements. Thus, the behavior of element acting as a drop element must be changed. This can be done by creating event listeners for the drop element. The drop element is also referred to as target element.

## Slides 38 to 41

Let us understand the drop events.

The slide has a blue header bar with the text 'HTML 5 Drop Events 1-4'. Below the header, there is a bulleted list and a table:

- For any element to receive the drop operation, it must be associated with the drop events.
- Following table lists the events of the drop operation.

Event	Description
dragenter	Triggers when a draggable element is being dragged on the target element for the first time.
dragleave	Triggers when an element is dragged outside the target element.
dragover	Triggers when an element is dragged inside the target element.
drop	Triggers when an element is dropped in the target element.

At the bottom left is the copyright notice '© Aptech Ltd.' and at the bottom right is 'HTML5 Geolocation and APIs / Session 19 38'.

**HTML** **Drop Events 2-4**

- The Code Snippet demonstrates how to set up event listeners to drop the image element on the target element.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Drag and Drop API</title>
    <script>
      function drag_image(event)
      {
        event.dataTransfer.setData("image", event.target.id);
      }
      function allow_drop(event)
      {
        event.preventDefault();
      }
      function drop_image(event)
      {
        var data=event.dataTransfer.getData("image");
        event.target.appendChild(document.getElementById(data));
      }
    </script> </head>
```

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19    39

**39 (a)**

**HTML** **Drop Events 2-4**

- The Code Snippet demonstrates how to set up event listeners to drop the image element on the target element.

```
<body>
  <div id="div1" style="border: blue 2px solid; height:125px;
  width:75px; padding: 10px">
    
  </div>
  <br/>
  <div id="div2" style="border: red 2px solid; height:125px;
  width:75px; padding: 10px" ondrop="drop_image(event)"
  ondragover="allow_drop(event)">
  </div>
</body>
</html>
```

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19    39

**39 (b)**

**HTML5 Drop Events 4-4**

- Following figure shows the output of the drop operation, after the image is dragged on the target element.

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      41

Using slides 38 to 41, explain the drop events.

For any element to receive the drop operation, it must be associated with the drop events.

Explain the list of events of the drop operation.

Explain the code snippet using slide 39 which demonstrates how to set up event listeners to drop the image element on the target element.

## Slide 42

Let us understand the offline Web application API.

**HTML5 Offline Web Applications API**

- HTML5 supports offline Web applications that allow a user to work with them without being online.
- The offline Web applications works by saving all the Web pages locally on the user's system.
- This concept is also known as **Application Cache**.
- The **Application Cache** enables all resources, such as HTML, JavaScript, images, and CSS pages of a Web application to be stored locally on the system.
- Following are the steps that can be taken to cache resources locally on the system.

1. Create a manifest file to define the resources that need to be saved.
2. Reference the manifest file in each Web page designed to use cached resources.

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      42

Using slide 42, explain the offline Web application API.

Consider a situation where a user is travelling outside the coverage area of Internet Service Provider (ISP). In this case, the user will not be able to access Web applications due to the network connection failure.

HTML5 supports offline Web applications that allow a user to work with them without being online. Offline Web applications work by saving all the Web pages locally on the user's system. This feature is known as the Application Cache.

The Application Cache enables all resources, such as HTML, JavaScript, images, and CSS pages of a Web application to be stored locally on the system.

Following are the steps that can be taken to cache resources locally on the system:

- Create a manifest file to define the resources that need to be saved.
- Reference the manifest file in each Web page designed to use cached resources.

### Slides 43 and 44

Let us understand how to create a manifest file.

**HTML 5 Creating a Manifest File 1-2**

- The manifest file is a text file that defines the caching behavior for resources used by the Web page.
- The file should be saved with the .manifest extension.
- The Code Snippet demonstrates creation of a manifest file.

```

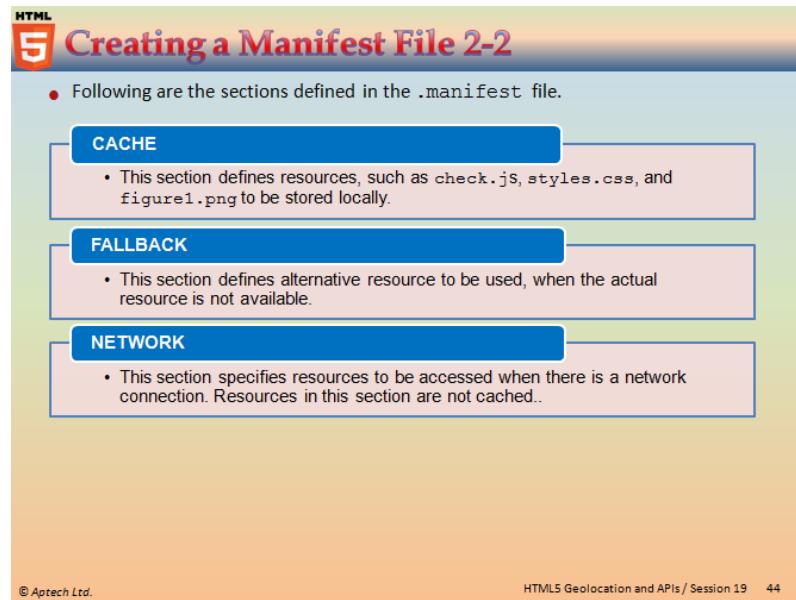
CACHE:
# Defines resources to be cached.
check.js
styles.css
images/figure1.jpg

FALLBACK:
# Defines resources to be used if non-cached resources cannot be
# downloaded
Other_images/ figure2.png

NETWORK:
# Defines resources that will not be cached.
figure3.png

```

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      43



Using slides 43 and 44, explain how to create the manifest file.

The manifest file is a text file that defines the caching behavior for resources used by the Web page. The file should be saved with the `.manifest` extension.

Explain the code snippet which demonstrates how to create a manifest file.

Following are the sections defined in the `.manifest` file:

- **CACHE:** This section defines resources, such as `check.js`, `styles.css`, and `figure1.png` to be stored locally.
- **FALLBACK:** This section defines alternative resource to be used, when the actual resource is not available. For example, `figure2.png` is defined as a fallback image. If a browser cannot access `figure1.jpg` in the images folder, then `figure2.png` will replace the unavailable image at the time of rendering the markup on the Web page. The unavailability of the image can be due to network connection or server problem.
- **NETWORK:** This section specifies resources to be accessed when there is a network connection. Resources in this section are not cached.

## Slides 45 to 47

Let us understand declaring a manifest.

**HTML5 Declaring a Manifest 1-3**

- To associate a manifest with a Web page, assign `.manifest` file to the attribute named `manifest` specified with the `html` element.
- The Code Snippet demonstrates how to add the `.manifest` file in an HTML document.

```
<!doctype html>
<html manifest="appcache.manifest">
<head>
    <title> Web Page </title>
    <link rel="stylesheet" href="styles.css"/>
    <script type="text/javascript" src="check.js"></script>
</head>
<body>
    <input type="button" value="click Here..." onClick="display()"/>
    
</body>
</html>
```

- In the code, the “`appcache.manifest`” is specified with the `<html>` tag.
- The interpretation of the manifest file is similar to any other file reference.
- The document uses a relative file path, as both the manifest file and HTML document are located in the same directory.
- By default, a Web page declaring manifest is cached automatically.

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      45

**HTML5 Declaring a Manifest 2-3**

- The benefit of Application Cache is that it improves the performance of a Web page by reducing the number of requests made to the Web server.
- The Web server hosts the Web application to be accessed on the network.
- Following figure shows how to enable the **Work Offline** mode in the Opera browser.
- This enables to cache the resources of the Web application pages locally.

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      46

**HTML5 Declaring a Manifest 3-3**

- Following figure shows the cached Web page in the Opera browser.

© Aptech Ltd.      HTML5 Geolocation and APIs / Session 19      47

Using slides 45 to 47, explain declaring a manifest.

To associate a manifest with a Web page, assign `.manifest` file to the attribute named `manifest` specified with the `html` element.

Explain the code snippet which demonstrates how to add the `.manifest` file in an HTML document.

In the code, the “appcache.manifest” is specified with the `<html>` tag. The interpretation of the manifest file is similar to any other file reference. The document uses a relative file path, as both the manifest file and HTML document are located in the same directory. By default, a Web page declaring manifest is cached automatically.

Explain the benefits of the Application Cache listed on slide 46.

Figure on slide 46 shows how to enable the Work Offline mode in the Opera browser. This enables to cache the resources of the Web application pages locally.

Slide 48

Let us summarize the session.

- Geolocation determines the current location of a user on devices.
- The location is represented as a single point on a map that comprises two components: latitude and longitude.
- The Geolocation API is a specification provided by the W3C which provides a consistent way to develop location-aware Web applications.
- Google Maps API is used to display the user's location on the map.
- The object of type Map is created in JavaScript, before it can be referenced in an HTML document.
- The drag-and-drop operations defines an event-based mechanism using which elements on a Web page can be copied, reordered, or deleted.
- HTML5 supports offline Web applications that allow a user to work with them without being online.

Using slide 48, summarize the session. End the session, with a brief summary of what has been taught in the session.

### **19.3 Post Class Activities for Faculty**

You can ask queries related to the sessions covered in the course.

## Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the OnlineVarsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the OnlineVarsity site to ask queries related to the sessions.