CYBAGE
Delivering Value. Scientifically.

HTML5

Presented by: Dadaram Jadhav

Authored by: Dadaram Jadhav

SECTION-I

Training overview
Training Agenda
Software setup

Agenda

- HTML5
- Semantic elements
- New input types
- Media
- HTML API
- Canvas
- SVG
- Drag/drop
- Web storage
- indexedDB
- Best practices

Software setup

- Editors : notepad, notepad++, Visual studio Code, sublime text
- Browser: Google chrome, Mozilla, Internet explorer
- Debugging tool: developer tools

SECTION-I ENDS

SECTION-II

Sample HTML page
HTML processing

Sample HTML page

```
<!DOCTYPE html>
<html>
<head>
<title>Web page title</title>
</head>
<body>

<h1>Welcome to training</h1>

</body>
</html>
```

HTML processing

- Html processing: browser engine, gecko(mozilla), webkit(apple), blink(chrome)
- Css processing: Layout and Rendering engine
- Javascript processing : javascript engine, v8 engine, chakra, spidermonkey
- HTML processing
  - Process HTML markup and build DOM tree
  - Process CSS markup and build CSSOM tree.
  - Combine DOM and CSSOM into render tree
  - Compute geometry of each node
  - Paint individual nodes to screen

HTML drawbacks

- Lengthy DOCTYPE
- Less semantic elements
- Less API
- Limited form input types
- No built in support for media
- Lack of client side storage

SECTION-II ENDS

SECTION-III

HTML5
Structure of page
New elements
Semantic elements

HTML5

- It is latest version of HTML
- UTF-8 character set
- Added semantic elements
- Attributes added in form
- Graphics elements canvas and svg added
- Support for multimedia elements
- HTML API
- Drag and drop
- localstorage
- indexeddb

Structure of page

- <!DOCTYPE html> defines HTML5 page
- <html> root
- <head> meta information
- <title> add title
- <body> content of page

Changes in Doctype

- HTML 4 Doctype
  - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
- HTML 5 Doctype
- <!DOCTYPE html>
- Encoding changes
  - HTML4: <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
- HTML5: <meta charset="utf-8">

New elements

- HTML5 offers new elements for better structure
- Semantic elements <header>, <footer>, <article>, and <section>
- New form elements like <datalist>, <output>
- New attributes for form number, date, time, calendar, and range
- Graphic <svg> and <canvas>
- multimedia <audio> and <video>

Semantic elements

- Element describes meaning to browser and developer
- Search engine can identify correct page contents
- <article>
- <aside>
- <figcaption>
- <figure>
- <footer>
- <header>
- <mark>
- <nav>
- <progress>
- <section>

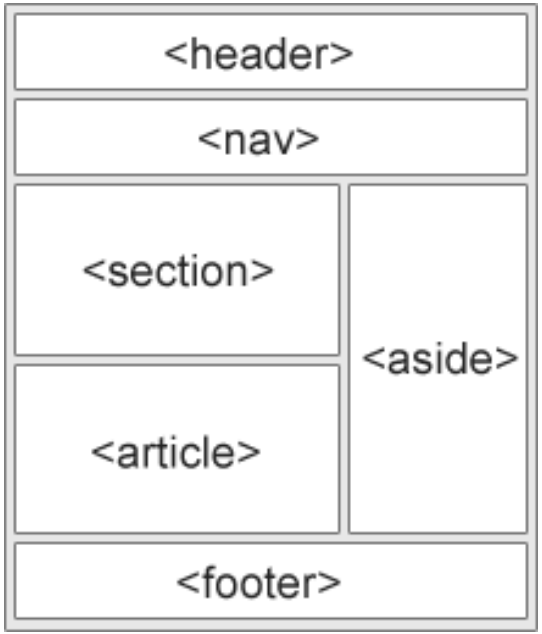
SECTION-III ENDS

SECTION-IV

HTML5 layout
Header, Footer, nav
Section, article

HTML5 Layout

- Uses combination of semantic elements
- Multicolumn layout created using
- HTML table, CSS grid, CSS framework
- CSS float, CSS flexbox



<section>

- <section> defines section in a document
- It is grouping of content, with a heading
- Home page can split into introduction, content, contact information

<article>

- <article> specifies independent, self-contained content
- An article should make sense on its own
- <article> element can be used for
- Forum post
- Blog post
- Newspaper article

<header>

- <header> specifies header for document or section
- <header> should be used as a container for introductory content
- Several <header> elements in one document possible

<footer>

- Specifies footer for document or section
- should contain information about containing element
- Contains author, copyright, terms of use, contact information
- Several <footer> elements possible

<nav> element

- Used to add navigation bar
- <nav></nav>

<aside>

- <aside> defines some content aside from content
- <aside> content should be related to surrounding content

<figure>, <figcaption>

- Purpose of a figure caption is add explanation to image
- Image and caption can be grouped together in <figure>

Migration from HTML4 to 5

Typical HTML4	Typical HTML5
	<header>
	<nav>
	<section>
	<article>
	<footer>

article, section, div

- <section> is defined as block of elements
- <article> is defined as a complete, self-contained block of elements
- is defined as block of children elements

SECTION-IV ENDS

SECTION-V

New Form elements
Responsive web page
Media

New input types

- New input type added in HTML5
- date
- month
- datetime-local
- time
- week
- email
- image
- number
- reset
- color

New input attributes

- Autocomplete, autofocus
- Form, formaction, formenctype, formmethod, formnovalidate, formtarget
- height and width
- list
- min and max
- multiple
- placeholder
- required
- autofocus specifies input field should get focus after page loading

New Form attributes

- autocomplete
- novalidate
- novalidate means form data should not be validated when submitted

Responsive

- HTML page can be make responsive using
- <meta name="viewport" content="width=device-width, initial-scale=1.0">

Media

- Media contents can be easily added with HTML5
- Add video with <video>
- Add audio with <audio>
- Attributes width, height, controls, autoplay, muted, loop, volume
- Add youtube contents

```
<iframe width="500" height="500" src="https://www.youtube.com/embed/j6muwUGdvXw?autoplay=1&&loop=1">
```

SECTION-V ENDS

SECTION-VI

Web storage

Web storage

- Used as alternative to cookies
- It is more secure
- Large data can be stored upto 5mb
- Information never transferred to server
- localstorage for storing data without expiry date
- sessionstorage data stored for one session

SECTION-VI ENDS

SECTION-VII

indexDB

IndexedDB

- It is about storing data inside user's browser
- More powerful than localstorage
- Useful when require large storage
- Earlier option was Web SQL
- Stores data as key:value pair

Advantages of indexeddb

- It allows several databases and several storages for each one
- It is NOT SQL
- API is asynchronous

IndexedDB example

```
var db
var req = window.indexedDB.open('product')
req.onsuccess = function (evt) {
    console.log('opened database successfully')
    db = req.result
}
var emp = [{ id: 101, name: "dmjadhav", address: "pune" }]

req.onupgradeneeded = (evt) => {
    var db = event.target.result;
    var objectStore = db.createObjectStore("product", { keyPath: "id" });
    for (var i in emp) {
        objectStore.add(emp[i]);
    }
}
```

SECTION-VII ENDS

SECTION-VIII

Canvas
SVG

HTML canvas

- Useful for drawing graphics using javascript
- Canvas has methods for drawing paths, boxes, circles, text
- Canvas provides high-performance element best suited for rendering faster graphics like image editing, an application that requires pixel manipulation

```
function drawShape() {
    var canvas = document.getElementById('mycanvas');
    var ctx = canvas.getContext('2d');
    ctx.beginPath();
    ctx.moveTo(25, 25);
    ctx.lineTo(105, 100);
    ctx.lineTo(25, 105);
    ctx.fill();
}
```

SVG

- It is Scalable Vector Graphics
- Used to define graphics for web using xml
- SVG does not depend on the resolution, means it is resolution independent
- If we enlarge the image, it will not lose its shape
- Canvas is resolution dependent. If the image is enlarged, it will start reflecting the pixels of the image
- SVG is mainly recommended for use in full-screen user interfaces
- <svg> is container for SVG graphics

```
<svg width="400" height="100">
  <rect width="400" height="100" />
</svg>
```

SVG vs Canvas

- SVG is for 2d graphics
- Canvas used for 2d graphics using Javascript
- SVG is slow for rendering
- SVG is faster when rendering really large objects, but slower when rendering many objects
- Map will perform better with SVG
- A game would probably be faster in Canvas
- Canvas would be better for faster things and heavy bitmap manipulation
- SVG refers as shape based whereas Canvas refers as pixel based
- SVG images cannot be saved in other formats. In Canvas, you can save the resulting images in .png and .jpg format.

SECTION-VIII ENDS

SECTION-IX

Drag and drop API

Drag/drop

- It is used to grab and drop object
- Any element can be draggable
- Element can be made draggable using draggable="true"

Drag and drop example

```
    <script>
        function allowDrop(ev) { ev.preventDefault(); }

        function drag(ev) { ev.dataTransfer.setData("text/html", ev.target.id); }

        function drop(ev) {
            ev.preventDefault();
            var data = ev.dataTransfer.getData("text/html");
            ev.target.appendChild(document.getElementById(data));
        }
    </script>
<div id="div1" style="width:350px;height:100px;padding:10px;border:1px solid #aaaaaa;"
        ondragover="allowDrop(event)"          ondrop="drop(event)">
    </div>
    <br>
    <button id="drag1" draggable="true" ondragstart="drag(event)">Drag</button>
```

SECTION-IX ENDS

SECTION-X

Code checking
Best practices

Code checking

- Linter can be used for code checking
- VS code have vscode-linthtml extension
- Can use Beautify for code alignment

Best practices

- A consistent use of style makes it easier for others to understand your HTML
- Using a well-formed syntax can be smart
- Always keep your code tidy, clean and well-formed
- Use Correct Document Type: <!DOCTYPE html>
- Use Lower Case Element Names
- Mixing uppercase and lowercase names is bad
- Developers normally use lowercase names
- Lowercase look cleaner
- Lowercase are easier to write

Cont...

- Close All HTML Elements
- Close Empty HTML Elements
- Use Lower Case Attribute Names
- Quoted values are easier to read
- Always add the alt attribute to images
- HTML5 allows spaces around equal signs
- Avoid Long Code Lines

Cont...

- For readability, add blank lines for large blocks
- For readability, add two spaces of indentation
- Do not use unnecessary blank lines and indentation
- Don’t omit <html> and <body>
- Don’t omit <head> tag
- The <title> element is required in HTML5

Cont...

- Make title as meaningful as possible
- Short comments should be written on one line
- Use simple syntax for linking to style sheets
- Use simple syntax for loading external scripts
- HTML files should have a .html or .htm extension
- CSS files should have a .css extension
- JavaScript files should have a .js extension

SECTION-X ENDS

Thank You!

www.cybage.com