

HTML5 : An Introduction

Anjan

anjan.mitra@gmail.com

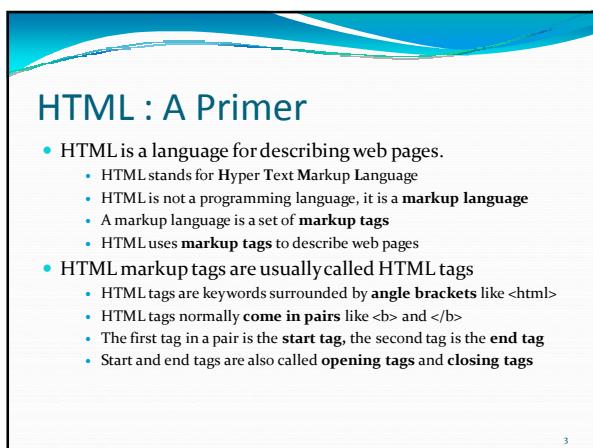
1



Agenda

- History, Vision And Future Of HTML5
- Getting Started With HTML5
- Structure Of A Web Page
- Forms
- Audio And Video
- Canvas

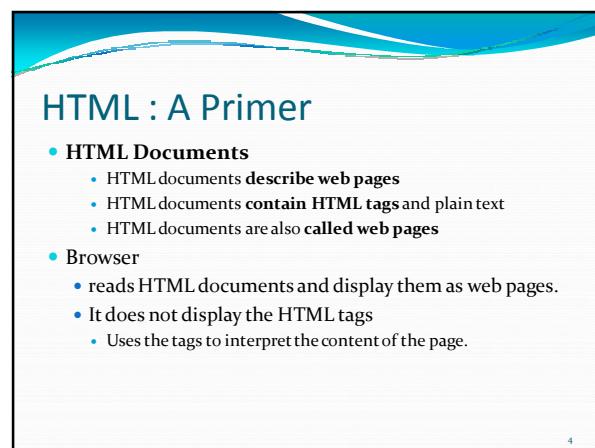
2



HTML : A Primer

- HTML is a language for describing web pages.
 - HTML stands for Hyper Text Markup Language
 - HTML is not a programming language, it is a **markup language**
 - A markup language is a set of **markup tags**
 - HTML uses **markup tags** to describe web pages
- HTML markup tags are usually called HTML tags
 - HTML tags are keywords surrounded by **angle brackets** like <html>
 - HTML tags normally **come in pairs** like and
 - The first tag in a pair is the **start tag**, the second tag is the **end tag**
 - Start and end tags are also called **opening tags** and **closing tags**

3



HTML : A Primer

- **HTML Documents**
 - HTML documents **describe web pages**
 - HTML documents **contain HTML tags** and plain text
 - HTML documents are also **called web pages**
- **Browser**
 - reads HTML documents and display them as web pages.
 - It does not display the HTML tags
 - Uses the tags to interpret the content of the page.

4

HTML : A Primer

```
<html>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

5

History, Vision And Future Of HTML5

6

What Is HTML5?

- Successor of HTML 4.01.
- It comes with new tags, features and APIs
 - New structural elements (<header>, <footer>, <nav> and more)
 - Forms and client-side validation
 - Native browser support for audio and video
 - Canvas API
 - Web storage
 - Geolocation
 - Web Workers
 - New communications API - Web Sockets.

7

What Is HTML5?

- It brings more clarification.
- It standardizes many features although they existed before.
 - HTML5 is not so new.
 - It has a lot of background history.

8

A Little Retrospective

- **December 1997:**
 - HTML 4.0 is published by the W3C.
- **February - March 1998:**
 - XML 1.0 is published.
- **May 1998:**
 - A W3C workshop, "Shaping the Future of HTML"
 - next generation of HTML initiated.
- **December 1999 - January 2000:**
 - HTML 4.01 recommendations are published

9

A Little Retrospective

- **October 2006:**
 - W3C to work with the WHAT Working Group.
- **January 2008:**
 - First W3C working draft of HTML5 is published.

10

What Is The W3C?

- Stands for World Wide Web Consortium.
- Founded and lead by Tim Berners-Lee.
- Internationally responsible for standards of the World Wide Web.
- Any W3C Document goes through the following maturity levels until becoming a recommendation:
 - Working Draft (WD)
 - Last Call Working Draft
 - Candidate Recommendation (CR)
 - Proposed Recommendation (PR)
 - W3C Recommendation (REC)

11

What Is The WHATWG?

- Stands for Web Hypertext Application Technology Working Group
- Founded in 2004 by individuals from Apple, The Mozilla Foundation, and Opera Software, due to a concern about W3C vision of the web
- Anyone can contribute for free.
- Until they joined their force with the W3C in 2006, they were the only one working on what we know as the HTML5 specification

12

Vision & Philosophy @ HTML5

- **Compatibility**

- Support existing content - deprecated tags and attributes.
- Degrade gracefully (fallback)
- Do not reinvent the wheel
- Evolution not revolution.

- **Utility**

- Solve real problems
- Priority of constituencies
 - Users -> implementers -> theoretical purity.

13

Vision & Philosophy @ HTML5

- **Utility (cont'd)**

- Secure by design (e.g. concept of "same origin").
- Separation of concerns
 - Use of HTML & CSS
- DOM consistency
 - Same DOM tree of a web page on different browsers.

14

Vision & Philosophy @ HTML5

- **Interoperability**

- Well-defined behavior of tags.
- Avoid needless complexity (i.e. ease of use)
- Handle errors
 - graceful error recovery wrt failure.

- **Universal Access**

- Media independence - different platforms, devices.
- Support world languages.
- Accessibility
 - Those with sight disabilities can use browser with assistive technology.

15

Future Of HTML5

- HTML5 Working Draft - April,
- Candidate Recommendation is planned for 2012.
- Proposed Recommendation is planned for 2022 !!

16

Timeline

- HTML 4 :
 - There is no proper test suite
 - Some parts of the specification do not even have real implementation
 - Some parts of the specification are not fully documented
 - Big parts are not interoperable
 - Lots of errors but have not been fixed yet.

17

Getting Started With Html5

18

The State Of Browser Support

- Among all latest browsers on the market, Internet Explorer (IE) has least support for HTML5 features.
- Check the status at :
 - www.caniuse.com

19

Browsers In Mobile Devices

- Key features:
 - Canvas and Video
 - Offline Support
 - GeoLocation API
 - Advanced Forms

20

Using Web Apps rather than Native Apps

Pros	Cons
<ul style="list-style-type: none"> ■ Available on all platforms / connected devices ■ One single place to find the web app ■ Low development cost / high ROI: <ul style="list-style-type: none"> ■ Simpler language (HTML / JavaScript) ■ Shorter / cheaper debugging process ■ Much larger potential reach ■ Low maintenance cost ■ Free from carrier / device manufacturer / approval control / revenue share ■ Users always have the latest app 	<ul style="list-style-type: none"> ■ Lower usability due to UI design limitations ■ Lower performance due to added browser layer ■ Harder to discover cool apps without store front / user reviews ■ Limited HTML5-compliant browser support (depends on the smartphone)

21

Feature Detection

- Browser's support for HTML5 features
- Need to check for support in the code :
 - `!navigator.geolocation`
 - `!document.createElement('canvas').getContext`
 - `!document.createElement('video').canPlayType`

22

Support For Legacy Browsers

- **Graceful Degradation**
 - New input types that are not supported
 - automatically fallback to `<input type="text">`
 - When the `<audio>` or `<video>` element are not supported
 - the inner content is evaluated.
- **Emulation**
 - use of polyfills / shims to emulate features that are not supported e.g. canvas, Geolocation.

23

Developer Tools

- Components of an web app:
 - HTML
 - CSS
 - JavaScript
 - Images
 - Manifest files (for offline caching purpose).
- Browsers development tools:
 - Chrome developer console
 - Safari web inspector
 - Firebug for Firefox.

24

HTML5 : Structure Of A Web Page

25

HTML5 DOCTYPE

- The release of IE5 on Mac lead to a major problem
 - Many web content based on bad authoring no longer rendered on IE5 for Mac, since this browser respected the standards too much.
 - Microsoft later decided to introduce this DOCTYPE declaration to let web developers choose how the web page should be rendered (by activating a certain browser mode: quirks and standards modes).
 - Pages with no DOCTYPE used to be rendered the same way as before (quirks mode).

26

HTML5 DOCTYPE

- An HTML page first starts with the DOCTYPE declaration.
- There is only one DOCTYPE in HTML5.
 - <!DOCTYPE html>
 - This triggers the standard mode in a browser.

27

Page Encoding

- Is the mapping between what you can see on the browser and what is stored in the disk.
- Character encoding can be specified in two places:
 - At server level on the HTTP headers
 - In the <head> element of a web page using the <meta> tag - has higher precedence.

28

Page Encoding

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

Character encoding must be specified in the first 512 bytes of the document.

29

HTML5 Markup

- HTML (earlier):
 - has structural AND presentational tags.
 - HTML semantics :
 - Overuse of id and class attributes
- Improvement in the HTML5 language made
 - HTML5 introduces 28 new elements: E.g.
 <section>, <article>, <aside>, <hgroup>, <header>, <footer>, <nav>, <figure>, <video>, <audio>, <canvas>, etc.

30

New Sectioning HTML5 elements

Sectioning Element	Description
header	Header content (for a page or a section of the page)
footer	Footer content (for a page or a section of the page)
section	A section in a web page
article	Independent article content
aside	Related content or pull quotes
nav	Navigational aids

31

Sample HTML5 Page (Partial)

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" >
  <title>HTML5</title>
  <link rel="stylesheet" href="html5.css">
</head>
<body>

<header>
  <h1>Header</h1>
  <h2>SubTitle</h2>
  <h4>HTML5 Rocks!</h4>
</header>

<div id="container">
```

32

Java Script Methods		
Function	Description	Example
getElementById()	Returns the element with the specified id attribute value	<div id="foo"> getElementById("foo");
getElementsByName()	Returns all elements whose name attribute has the specified value	<input type="text" name="foo"> getElementsByName("foo");
getElementsByTagName()	Return all elements whose tag name matches the specified value	<input type="text"> getElementsByTagName("input");

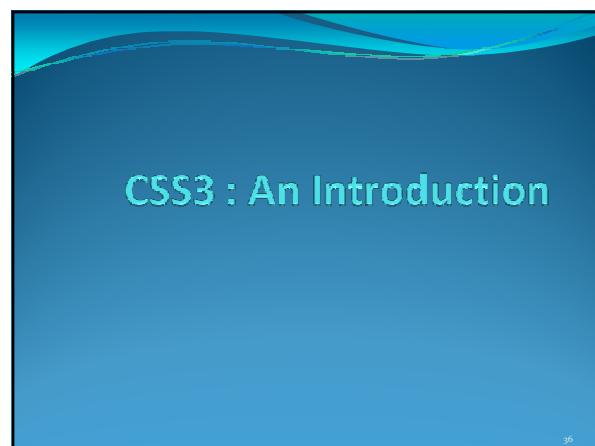
33

New Methods			
Function	Description	Example	Result
querySelector()	Return the first element in the page which matches the specified selector rule(s)	querySelector("input.error");	Return the first input field with a style class of "error"
querySelectorAll()	Returns all elements which match the specified rule or rules	querySelectorAll("#results td");	Return any table cells inside the element with id results

34

Java Script Engine		
Browser	Engine Name	Notes
Apple Safari 5	Nitro (otherwise known as SquirrelFish Extreme)	Released in Safari 4 and refined in version 5, it introduces byte code optimizations and a context-threaded native compiler.
Google Chrome 5	V8	Since Chrome 2, it uses generational garbage collection for high memory scalability without interruptions.
Microsoft Internet Explorer 9	Chakra	This focuses on background compilation and an efficient type system and demonstrates a tenfold improvement over IE8.
Mozilla Firefox 4	JägerMonkey	Refined from version 3.5, this combines fast interpretation with native compilation from trace trees.
Opera 10.60	Carakan	This one uses register-based byte code and selective native compilation and claims improvements of 75% on version 10.50.

35



36

Overview

- Used to control the style and layout of Web pages.
- CSS3 is the latest standard for CSS.
- Completely backwards compatible.
- CSS3 is split up into "modules".
- CSS3 specification is still under development by W3C.

37

Overview

- Some of the most important CSS3 modules are:
 - Selectors
 - Box Model
 - Backgrounds and Borders
 - Text Effects
 - 2D Transformations
 - Animations
 - Multiple Column Layout

38

Important Changes

- **CSS3 Borders**
 - Rounded borders


```
div
{
  border:2px solid;
  border-radius:25px;
  -moz-border-radius:25px;
}
```
 - Use an image as a border
 - `border-image:url(border.png) 30 30 round;`

39

Important Changes

- **CSS3 Box Shadow**
- **CSS3 Text Effects**
 - Contains several new text features.
 - `text-shadow`
 - `word-wrap`
 - `p {word-wrap:break-word;}`

40

Important Changes

- **CSS3 Transforms**
 - It lets an element change shape, size and position.



41

Important Changes

- **CSS3 2D Transforms**
 - translate() : the element moves from its current position.
 - rotate()
 - scale() : width & height changes.
 - skew() : the element turns in a given angle.

42

Important Changes

- **CSS3 Transitions**
 - Adds an effect when changing from one style to another
 - Without using Flash animations or Java Scripts.
 - To do this, you must specify :
 - Specify the CSS property you want to add an effect to.
 - Specify the duration of the effect (non-zero).

43

Important Changes

- **CSS3 Animations**
 - Can create animations & thus replace animated images, Flash animations, etc.
 - @keyframes rule
 - Bind the animation to a selector :
 - Specify the name of the animation
 - Specify the duration of the animation

44

Important Changes

CSS3 Animations

- An animation is an effect that lets an element gradually change from one style to another.
- You can change as many styles you want, as many times you want.
 - Specify when the change will happen in percent, or the keywords "from" and "to", which is the same as 0% and 100%.
 - 0% is the beginning of the animation, 100% is when the animation is complete.
 - For best browser support, you should always define both the 0% and the 100% selectors.

45

Html5 : Forms

46

Needs of Web Applications

- Gather input from users
 - Leave a comment on a blog article
 - Book a flight
 - Sign in to a web site
- Controls used by many Web Applications
 - Email
 - Date
 - Phone number
 - Sliders

47

Forms

- Other usage:
 - Auto focus on an element of the form
 - Give hint to user on what information they should give and how
 - Handling minimum/maximum values for some controls
 - Force the user to enter a value on a form
- Display error messages to their users
 - When they do not provide required information
 - When the provided information is in the wrong format
- Validate the form on the client side (JS enabled)

48

New Input Types

- HTML5 comes with new controls to ease the creation of forms.
- 13 new values for the type attribute:
 - <input type="color" /> To choose a color
 - Date, datetime, month, time, week, ...
 - email, tel
 - number , Range (displays slider)
 - url

49

New HTML5 Form Elements

Type	Purpose
tel	Telephone number
email	Email address text field
url	Web location URL
search	Term to supply to a search engine. For example, the search bar atop a browser.
range	Numeric selector within a range of values, typically visualized as a slider

50

HTML Form

- The specification deals with functional behavior and semantics, not appearances or displays.
 - it separates styling from semantics
- Browsers free to innovate.
 - Unsupported feature : graceful degradation.
 - Users need to handle fallback scenarios.
- New functionality :
 - New input types
 - New functions and attributes.

51

Onscreen KB (iPhone)

52

<input type="range">



53

New Attributes

Some of the commonly used ones are:

- Autofocus
- Placeholder (i.e. hint)
- Required (i.e. mandatory)
- Autocomplete
- Min & max (e.g. for number, date, etc.)
- Step (for number)
- list

54

The *placeholder* Attribute

```
<label>Runner: <input name="name"
placeholder="First and last name" required>
</label>
```

Runner: First and last name

55

The *autocomplete* Attribute

```
<input type="text" name="creditcard"
autocomplete="off">
```

Type	Purpose
on	The field is not secure, and its value can be saved and restored.
off	The field is secure, and its value should not be saved.
unspecified	Default to the setting on the containing <form>. If not contained in a form, or no value is set on the form, then behave as if on.

56

Form Validation

- On the client-side using JavaScript
 - An optimization.
- On the server-side with any server side language (Java, PHP, RoR, etc.) – a must.

57

Form Validation

- **ValidityState** object
 - It's a live object.
 - Can be accessed from any form control in a browser.
- ```
var valCheck = document.myForm.myInput.validity;
```

58

## Validity Constraints

- valueMissing (e.g. login name)
- typeMismatch (e.g. email)
- patternMismatch (e.g. credit card number)
- tooLong (e.g. user name)
- rangeUnderflow (e.g. < range\_min)
- rangeOverflow (e.g. > range\_max)
- stepMismatch

59

## Form Validation

- <input> have some properties and methods :
    - willValidate – simulates “submit”.
  - Form validation (entire) can be disabled :
    - the *formnovalidate* on a <input type="submit"/>
- ```
<input type="submit" formnovalidate name="save" value="Save current progress">
<input type="submit" name="process" value="Process order">
```

60

Html5 : Audio And Video

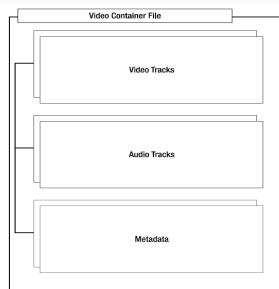
61

Current Scenario

- Based On Plug-ins e.g. Silverlight or Flash
- Browser uses the <object> tag.
 - Browser cannot differentiate wrt video / audio
- An end-user might not have the plug-in installed / not permitted (e.g. in corporate).
- Plug-ins are not cross platform
 - Flash : not readily supported on iOS.

62

An Quick Overview



63

File Formats

- Some of the popular video container formats include the following:
 - Audio Video Interleave (.avi)
 - Flash Video (.flv)
 - MPEG 4 (.mp4)
 - Matroska (.mkv)
 - Ogg (.ogg)

64

Some examples

- Audio codecs :

- AAC
- MPEG-3
- Ogg Vorbis

- Video codecs :

- H.264
- VP8
- Ogg Theora

65

<audio>/<video> element

```
<!DOCTYPE HTML>
<html>
<head> ... </head>
<body>
  <video>
    <source src="clip.mp4">
  </video>
</body>
</html>
```

- Use of <source> tag
- fallback section

66

Attributes

- **autoplay**: Can be true or false.
- **controls**: (play/pause/volume/etc) - Can be true or false.
- **loop**: Can be true or false.
- **muted**: Can be true or false.
- **playbackRate**: Numeric value (double).
- **Src**

67

Video Attributes

Attribute	Value
poster	The URL of an image file used to represent the video content before it has loaded. Think "movie poster." This attribute can be read or altered to change the poster.
width, height	Read or set the visual display size. This may cause centering, letterboxing, or pillarizing if the set width does not match the size of the video itself.
videoWidth, videoHeight	Return the intrinsic or natural width and height of the video. They cannot be set.

68

Control Functions

Function	Behavior
load()	Loads the media file and prepares it for playback. Normally does not need to be called unless the element itself is dynamically created. Useful for loading in advance of actual playback.
play()	Loads (if necessary) and plays the media file. Plays from the beginning unless the media is already paused at another position.
pause()	Pauses playback if currently active.
canPlayType(type)	Tests to see whether the video element can play a hypothetical file of the given MIME type.

69

Events

- Play
- Pause
- Timeupdate
- Volumechange
- Ratechange
- Ended
- Seeking

70

HTML5 : Canvas

71

Overview Of Graphics @ Browser

- Static images :
- dynamic graphics? no native support
- So, current solutions :
 - Use plug-ins such as Flash, Silverlight, etc.
 - Now, HTML5 has <canvas> element.

72

Canvas Vs. SVG

- One main alternative to Canvas : SVG (Scalable Vector Graphics).
- Both Canvas and SVG allow graphic manipulation in the browser
 - different techniques used.

73

Canvas Vs. SVG

	Canvas	SVG
Advantages	<ul style="list-style-type: none"> High performance graphics Pixel-level manipulation Constant performance depending on the resolution used Canvas drawing surface can be saved as an image file 	<ul style="list-style-type: none"> Vector-based, scalable to any resolution Good support for animations DOM manipulated elements
Drawbacks	<ul style="list-style-type: none"> No API for animation, You have to redraw every time Pixel-manipulation: impossible for shape you create to respond to events Not scalable Not suited for user interfaces 	<ul style="list-style-type: none"> Works with the DOM so with a lot of elements it gets slower Not suited for gaming applications

74

Accessibility

- When the canvas is not supported in the browser, use the fallback option:

```
<canvas>
  Text which is displayed if your browser does not support
  the canvas element...
</canvas>
```

Note:

If you want to access your canvas and draw on it, you have to use JavaScript.

75

Browser Support Check

```
try {
  document.createElement("canvas").getContext("2d");
  document.getElementById("support").innerHTML =
    "HTML5 Canvas is supported in your browser.";
} catch (e) {
  document.getElementById("support").innerHTML = "HTML5 Canvas is not supported
  in your browser.";
}
```

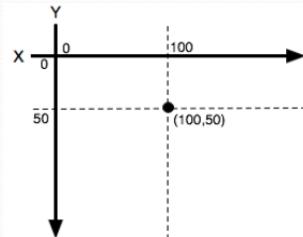
76

Context And Coordinates

- All drawing operations on a canvas happen in the canvas's context:
 - its 2D context.
 - Use JavaScript to get the canvas's context:
 - `var ctx = document.getElementById("mycanvas").getContext("2d");`

77

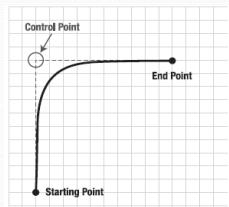
Context And Coordinates



Coordinates : in pixels

78

Quadratic Curve



79

Drawing Shapes

- Canvas supports only one basic shape: Rectangle
- Filled rectangle:


```
var ctx=document.getElementById("mycanvas").getContext('2d');
ctx.fillRect(50,50,100,200);
```
- Default color of the rectangle is black.

80

Drawing Shapes

- Stroked rectangle:
- ```
var ctx=document.getElementById("mycanvas").getContext('2d');
ctx.strokeRect(50,50,100,200);
```
- Default color of the rectangle is black.

81

## Working With Paths

- To draw shapes that are more complex.
- Path => a collection of pixels.
- A path can also be composed of subpaths.
- Methods :
  - beginPath()
  - closePath()
  - stroke()
  - fill()

82

## Drawing Lines

- Straight Lines
  - moveTo(x,y)
  - lineTo(x,y)
- Circular Lines
  - arc(x, y, radius, startAngle, endAngle)

83

## Drawing Text

- Drawing text is still drawing a path like other shapes
  - filled text
  - stroked text

```
var ctx=document.getElementById("mycanvas").getContext('2d');
ctx.beginPath();
ctx.textAlign="center";
ctx.font="italic 50px verdana";
ctx.fillText("Hello World!", 265, 150);
```

84

## Drawing Images

- To draw an image on a canvas, we have three methods available:
  - `drawImage(image, dx, dy)`: no resizing the image
  - `drawImage(image, dx, dy, dw, dh)`: resizes
  - `drawImage(image, sx, sy, sw, sh, dx, dy, dw, dh)`: select a particular region of the image.

85

## Working With Pixels

- It is possible to get the pixels data from a canvas.
  - `context.getImageData(sx,sy,sw,sh)`

86

## Understanding Transforms

- Wrt the context's **state**:
  - `Save()`
  - `Restore()`
  - The state of the canvas contains the current style and transformations applied.
- Translation & Rotation:**
  - `translate(x,y)` : moves the origin.
  - `rotate(angle)`

87

## Q & A

Thank You !!

88