HTML 5

Objectives

- *History, Vision And Future Of HTML5*
- *Getting Started With HTML5*
- *Structure Of A Web Page*
- *Forms*
- *Styling HTML5 pages using CSS3*
- *Understanding JavaScript*

What is HTML5

- Successor of HTML 4.01 and XHTML 1.1
- Has new tags, features and APIs
  - New structural elements (<header>,<footer>…)
  - Forms 2.0 and client-side validation

History

- **December 1997**: HTML 4.0 is published by the W3C
- **March 1998**: XML 1.0 is published, Mozilla.org is launched
- **January 2000**: XHTML 1.0 (HTML tags in XML) and HTML 4.01 published
- **May 2001**: XHTML 1.1 is published
- **August 2002**: XHTML 2.0 released
- **January 2008**: First W3C working draft of HTML5 is published!!

W3C vs WHATWG

- World Wide Web Consortium
- Founded and lead by Tim Berners-Lee
- Internationally main responsible of standards for the World Wide Web

- Web Hypertext Application Technology Working Group
- Founded in 2004 by individuals from Apple, The Mozilla Foundation, and Opera Software

Vision & Philosophy Behind HTML5

- **Compatibility**
  - Support existing content
    - User-agents should be able to deal with old content containing deprecated tags and attributes
  - Degrade gracefully
    - For instance, there are ways to test if a HTML5 feature is available or not. If not, provide a fallback
  - Do not reinvent the wheel
    - When possible, use features that exist and work rather than spending time implementing new ones
  - Evolution not revolution
    - Old content should be able to use new features without the need to make unrelated changes

Utility

- Solve real problems
  - When possible, specifications must be aimed to solve existing problems that the web is facing
- Secure by design
- Separation of concerns
  - HTML should be used for structuring content. Use CSS to present the content
- DOM consistency
  - The produced DOM tree of a web page should be the same on different browser. It is very important for scripts that work against the DOM tree

TeamLease® Degree Apprenticeship

Interoperability

- Well-defined behavior
  - Behavior of tags should be clearly defined in order to run the same way in different user-agents
- Avoid needless complexity
  - Features should be as simple as possible, easy to be implemented and easy for web developers to understand.
- Handle errors
  - Do not want end-users to be exposed to hard failure in case of errors from web developers. These web developers should prefer graceful error recovery to hard failure

Universal Access

- Media independence
  - Features should, when possible, work across different platforms, devices, and media.
- Support world languages
  - Authors should be able to publish their web site in any languages
- Accessibility
  - Features should be designed in a accessible way to users with disabilities. For instance, blind users cannot see <img> this is why there is the alt attribute that can be read by browser with assistive technology

TeamLease® Degree Apprenticeship

Future Of HTML5

- The current HTML5 Working Draft is from April 11th, 2011
- Candidate Recommendation is planned for 2012 while Proposed Recommendation is planned for 2022

Getting Started With HTML5

TeamLease®
Degree Apprenticeship

The State Of Browser Support

- Most part of the specification is implemented in latest browsers
- Because of legacy browsers and also the fact that all latest browsers do not implement the same features and sometimes partially, you want to test features and provide alternatives

Browsers In Mobile Devices

- HTML5 has a lot of potential for creating mobile web applications.
- Below are some of the key features that will be critical to web applications:
  - Offline Support
  - Canvas and Video
  - GeoLocation API
  - Advanced Forms

TeamLease®
Degree Apprenticeship

Feature Detection

- At present we cannot say a browser supports HTML5 but we can say a browser supports HTML5 features
  - Checking for video codec support:

```
function isCodecSupported(codec) {
if(!!document.createElement('video').canPlayType) {
var v = document.createElement("video");
return v.canPlayType(codec);
}
return false;
}
```

Support For Legacy Browsers

- **Graceful Degradation**
  - Features that, if not supported in a browser, will not break your web application.
    - New input types that are not supported automatically fallback to <input type="text">
  - When the <audio> or <video> element are not supported, can provide a fallback to flash.

TeamLease®
Degree Apprenticeship

Support For Legacy Browsers

- **Emulation**
  - Features that are not supported yet on browsers can be emulated
  - You can use polyfills to emulate features that are not supported
    - Polyfills /shims are pieces of code or plugins that emulate a non existing feature
  - A non exhaustive list of these polyfills:
    - Google Gears can emulate many features such as SQL Storage, Geolocation, Offline applications and Workers.
    - Usage of JavaScript libraries
    - Third party plugins such as Flash, Silverlight, etc… (mostly for audio/video content)

Developer Tools

- An HTML5 web application is composed of
  - HTML
  - CSS
  - JavaScript
  - Some images
  - Manifest files (for offline caching purpose)
- Browsers development tools
  - Chrome developer console
  - Safari web inspector
  - Opera dragonfly
  - Firebug plugin for Firefox for live coding and debugging purposes.

TeamLease® Degree Apprenticeship

Structure of A Web Page

HTML5 DOCTYPE

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

Page Encoding

- Is the mapping between what you can see on the browser and what is stored in the disk (from 0 and 1 to real characters)
- 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

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

New and Updated Elements

- 28 new elements:
  - <section>, <article>, <aside>, <hgroup>, <header>,<footer>, <nav>, <figure>, <figcaption>, <video>, <audio>, <source>, <embed>, <mark>, <progress>, <meter>, <time>, <ruby>, <rt>, <rp>, <wbr>, <canvas>, <command>, <details>,<summary>, <datalist>, <keygen> and <output>
- HTML5 also update some of the previous existing elements to better reflect how they are used on the Web or to make them more useful such as:
  - The <a> element can now also contain flow content instead of just phrasing content
  - The <hr> element is now representing a paragraph-level thematic break
  - The <cite> element only represent the title of a work
  - The <strong> element is now representing importance rather than strong emphasis

Tag - Description

Tag	Description
<article>	Specifies independent, self-contained content, could be a news-article, blog post, forum post, or other articles which can be distributed independently from the rest of the site.
<aside>	For content aside from the content it is placed in. The aside content should be related to the surrounding content
<bdi>	For text that should not be bound to the text-direction of its parent elements
<command>	A button, or a radiobutton, or a checkbox
<details>	For describing details about a document, or parts of a document
<summary>	A caption, or summary, inside the details element
<figure>	For grouping a section of stand-alone content, could be a video
<figcaption>	The caption of the figure section
<footer>	For a footer of a document or section, could include the name of the author, the date of the document, contact information, or copyright information
<header>	For an introduction of a document or section, could include navigation
<hgroup>	For a section of headings, using <h1> to <h6>, where the largest is the main heading of the section, and the others are sub-headings
<mark>	For text that should be highlighted
<meter>	For a measurement, used only if the maximum and minimum values are known
<nav>	For a section of navigation
<progress>	The state of a work in progress
<ruby>	For ruby annotation (Chinese notes or characters)
<rt>	For explanation of the ruby annotation
<rp>	What to show browsers that do not support the ruby element
<section>	For a section in a document. Such as chapters, headers, footers, or any other sections of the document
<time>	For defining a time or a date, or both
<wbr>	Word break. For defining a line-break opportunity.

Structural Elements

- **<header>**
  - Used as an introductory element or navigational aids
  - <header> can contain a table of contents, a search form, logos, and navigation blocks (<nav> element).
  - More than one <header> element per page is possible.

- **<hgroup>**
  - Represents heading of a section.
  - Used to group a set of <h1> ... <h6> elements
  - Useful when the headings contains multiple levels such as sub-headings, alternative titles, or taglines.
  - For the purpose of document summaries or outlines, the text of the <hgroup> element is the text of the highest ranked child element (<h1>...<h6>)

- **<section>**
  - Represents a generic section of a page.
  - It is a thematic grouping of content with generally a heading.
  - A web site can be split into major sections, such as the news section, the contact section, etc…
  - <section> elements can contain <article> elements.

- **<nav>**
  - Represents a major navigation block.
  - It groups links to other pages or to parts of the current page.
  - <nav> does not have to be used in every place you can find links. For instance, footers often contains links to terms of service, copyright page and such, the <footer> element would be sufficient in that case

- **<article>**
  - <article> element is a part of the page that can be independently distributed or reused.
  - This could be a forum post, a magazine or newspaper article, a blog entry, etc…
  - <article> ………………….. </article>

- **<aside>**
  - Represents content that is tangentially related to the content around the aside element, and which could be considered separate from that content.

- **<footer>**
  - Represents a footer for its nearest ancestor sectioning content or sectioning root element.
  - Like the <header> element, it can be nested in <nav>, <section>,<article> and <aside> elements.

New Attributes

- Most new features are
  - Forms input type
  - Audio and video
  - Offline application
- Following HTML4 now apply to all elements in HTML5
  - class, dir, id, lang, style, tabindex and title

TeamLease®
Degree Apprenticeship

Deprecated Elements And Attributes

- Elements that are deprecated in HTML5 are -
  - <basefont>, <big>, <center>, <font>, <strike>, <tt>, <u>, <frame>, <frameset>, <noframes>, <acronym>, <applet>, <isindex> and <dir>

Forms

Form

- Previously called Web Forms 2.0
- HTML5 forms are going to make the creation and validation of forms easier than before.

What Are The Needs For Web Applications

- Web Application use forms to gather input from users
- A form allow users to:
  - Leave a comment on a blog article
  - Book a flight
  - Sign in to a web site
- Controls that most of Web Applications require are: Email, Date, Phone number, URL, Sliders
- Web Application also require:
  - 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 to prevent sending the form to the server if there are errors on the form

Current Solutions

- There are a lot of limitations with HTML4, it does not fit Web Applications need today
- Custom controls that does not exist natively such as date picker or slider are provided using third-party libraries (JQuery, MooTools, etc…)
  - That adds libraries to learn
  - JavaScript might be disabled on some client browsers
- Validation on the client side is quite complicated for some controls
  - Usage of regular expressions for many common used information (email, url, phone number)

TeamLease®
Degree Apprenticeship

13 New Input Types

<input type="color" />	To choose a color
<input type="date" />	To choose a date (year, month and day) with no timezone
<input type="datetime" />	Same as choosing a date but include the time (hour, minute, second, fraction of second) but the user cannot change the time zone offset
<input type="datetime-local" />	Same as datetime except that the user can change the time zone offset
<input type="month" />	To choose only a date and a month
<input type="time" />	To choose a time (hour, minute, second, fraction of second)
<input type="week" />	To choose a year with a week number for this year
<input type="email" />	To enter an email

TeamLease®
Degree Apprenticeship

New Input Types

<input type="number" />	To enter a number. Can be combined with other attributes such as **min** to define min value, **max** to define max value and **step** to define the increment
<input type="range" />	Displays a slider
<input type="search" />	No real difference with a text input except that the display might look different (rounded corners in Safari on the Mac)
<input type="tel" />	Looks like a text input except that the purpose is for entering phone number
<input type="url" />	Looks like a text input except that the purpose is for providing a URL

TeamLease®
Degree Apprenticeship

New Attributes

- autofocus
  - Boolean that give automatically the focus on a control. There should be only one control focused per page
- placeholder
  - This attribute allows developers to give a hint to the users on what information to provide
    - <input type="text" id="firstname“ placeholder="Type your first name here" />

TeamLease® Degree Apprenticeship

- required
  - Boolean that specify if a user input is mandatory on a control.
- multiple
  - <input type="file" multiple> allows a user to send multiple files
- pattern
  - Used with mail type or url input.
  - Allows the developer to use a regular expression and value entered by a user will have to match this pattern

- autocomplete
  - Have two values: on and off.
  - Can be applied on the entire form (on by default) and on controls. It can be good to remember previous values entered by users. Developers should consider this attribute carefully (sensitive data such as credit card number should not be remembered ⇒ set to off).

- min and max
  - Respectively define minimum and maximum values for controls. That works for number type but can also be applied to other input types such as date
- step
  - Define the increment number for number input type.
  - <input type="number" min="0" max="10" step="2" value="0" />

- list
  - To reference a <datalist> element by its id attribute. the <datalist> element contains a list of values.
  -
    ```
    <input type="text" list="fruit" />
    <datalist id="fruit">
    <option value="Apple">
    <option value="Banana">
    <option value="Cherry">
    </datalist>
    ```

Form Validation

- Form validation can happen in many places:
  - On the client-side using JavaScript to avoid unnecessary round trip between the client and the server
  - On the server-side with any server side language (Java, PHP, ASP.NET and others)

HTML5 and CSS3

- CSS3 is built upon its previous version but also comes with loads of new features - new selectors, rounded corners, box and text shadows, transitions, animations, transformations, etc…
- HTML5 is still in development
  - Most of the browsers are fine with elements they do not recognize but CSS by default assumes that they are display:inline
  - To be on the safe side until the standard is implemented everywhere, we need to precise the following:
    - article, aside, footer, header, hgroup, nav, section {
          display: block;
      }

TeamLease®
Degree Apprenticeship

Fonts and Measurements

- CSS3 font and text properties support:
  - External fonts

```
@font-face  {
  font-family: newGroovyFont;
  src: url('CandaraPlus.ttf')
}
```

  - Absolute text sizes

```
font-size : 16pt;
line-height : 0.5in;
letter-spacing : 12mm;
```

  - Relative text sizes

```
font-size : 1em;
border-width : 300px;
padding : 16rem;
```

TeamLease®
Degree Apprenticeship

Implementing Text Effects

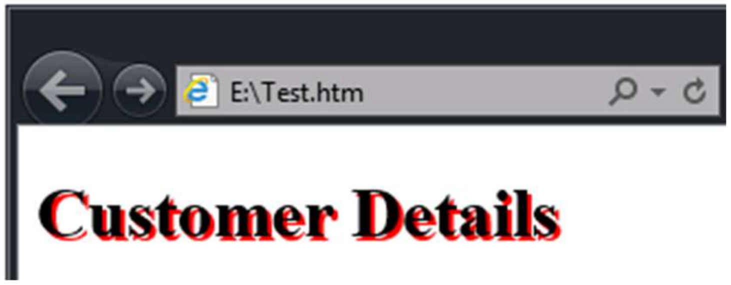
- CSS3 includes further text styling support for:
  - Paragraph indentation
  - Line wrapping
  - Text spacing
  - Shadow effects

```
text-indent: 3rem;
```

```
hyphens: manual;
word-wrap: break-word;
```

```
word-spacing: 2rem;
```

```
text-shadow: 2px 2px 0 red;
```



Customer Details

New Block Properties in CSS3

- CSS3 adds new box-level support for:
  - Outlines

```
outline: 2px solid green;
outline-offset: 5rem;
```

  - Presentation

```
border-radius: 50% / 30%;
overflow: hidden;
resize: horizontal;
```

  - Multiple column layouts

```
column-count: 3;
column-gap: 5rem;
column-rule: 1px solid black;
```

Block Layout Models

- CSS3 supports several block layout methods:

Block	`display: block;`
Inline	`display: inline;` `display: inline-block;`
Table	`display: table;`
Positioned	`position: relative;` `position: absolute;` `position: fixed;`
Flexbox	`display: flexbox;`

TeamLease®
Degree Apprenticeship

Text Pseudo-Elements

CSS pseudo-elements enable you to select:

• The first letter of a text element	`p::first-letter`
• The first line of a text element	`p::first-line`
• The space before or after a text element	`p::before` `p::after`
• Text selected by the user	`::selection`

Link and Form Pseudo-Classes

CSS defines two sets of contextual pseudo-classes:

- Link classes

```
a:link
a:visited
a:focus
a:hover
a:active
```

- Form classes

```
input:enabled
input:disabled
input:checked
```

TeamLease® Degree Apprenticeship

DOM-Related Pseudo-Classes

Use positional pseudo-classes to select a single element from a set based on:

- Position

```
p:first-child
p:nth-child(2)
```

- Position and type

```
p:last-of-type
p:nth-last-of-type(4)
```

- Document structure

```
:empty
:root
:not(p, h1)
:target
```

TeamLease® Degree Apprenticeship

Specifying Color Values

CSS3 defines several different sets of color values:

- Keywords

```
color:  red;
color: transparent;
color: currentColor;
```

- RGB \ RGBA model values

```
color: #ff0000;
color: rgb(255,0,0);
color: rgba(100%,0,0,0.5);
```

- HSL \ HSLA model values

```
color: hsl(240, 100%, 50%);
color: hsl(120, 100%, 50%, 0.5);
```

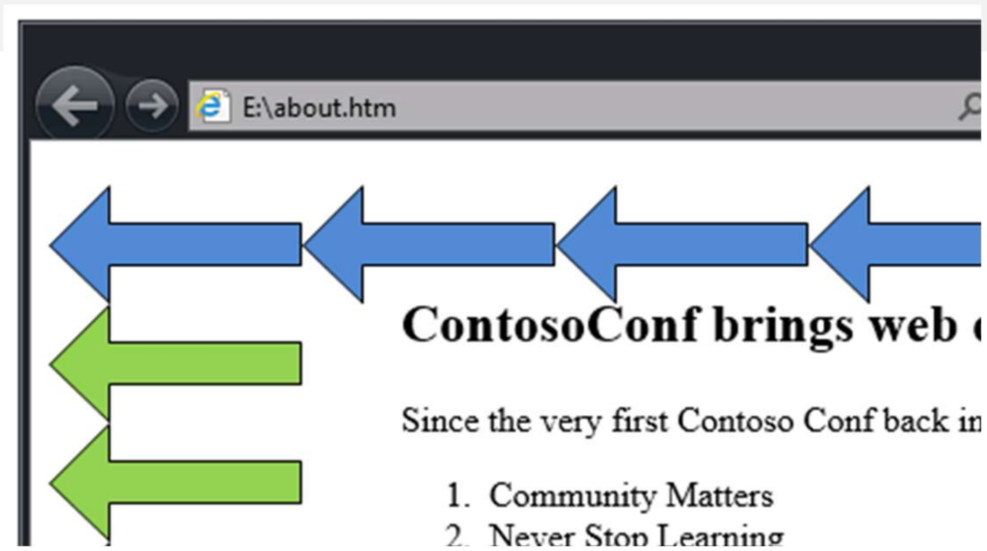
TeamLease® Degree Apprenticeship

Defining Backgrounds and Effects

CSS3 supports:

- Multi-image backgrounds

```
article {
      background-image: url('bluearrow.png'), url('greenarrow.png');
      background-repeat: repeat-x, repeat-y;
}
```



ContosoConf brings web

Since the very first Contoso Conf back in

1. Community Matters
2. Never Stop Learning

- Gradients

```
background: radial-gradient(top right, ellipse, red, blue);
```

TeamLease®
Degree Apprenticeship

Implementing Transformations and Graphics

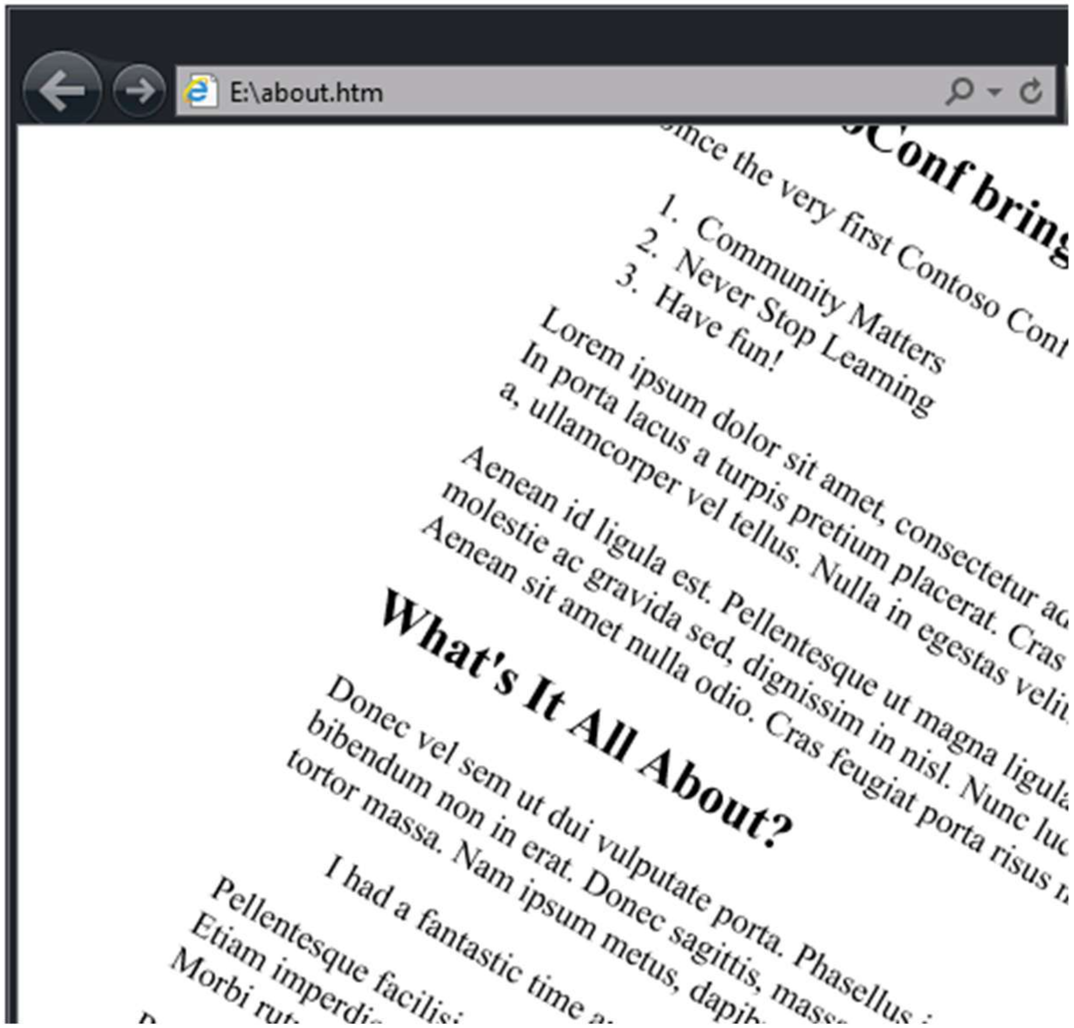
Using CSS3, you can:

- Transform, rotate, and skew elements

```
article {
      transform: rotate(30deg);
}
```

- Generate shapes

```
#circle {
            width: 200px;
            height: 200px;
            background: blue;
            border-radius: 50%;
}
```



Conf bring

nce the very first Contoso Conf

1. Community Matters
2. Never Stop Learning
3. Have fun!

Lorem ipsum dolor sit amet, consectetur ad
In porta lacus a turpis pretium placerat. Cras
a, ullamcorper vel tellus. Nulla in egestas veli

Aenean id ligula est. Pellentesque ut magna ligula
molestie ac gravida sed, dignissim in nisl. Nunc luc
Aenean sit amet nulla odio. Cras feugiat porta risus n

What's It All About?

Donec vel sem ut dui vulputate porta. Phasellus
bibendum non in erat. Donec sagittis, mass
tortor massa. Nam ipsum metus, dapib

I had a fantastic time a

Pellentesque facilisi
Etiam imperdi
Morbi rut