Introduction in HTML5

www.instigate-training-center.am

Topics

- HTML5 Introduction
- HTML5 New Elements
- HTML5 Semantic Elements
- HTML5 Input Types
- HTML5 Form Elements
- HTML5 Form Attributes

HTML5 Introduction

- HTML5 is the latest standard for HTML.
- HTML5 was designed to replace both HTML 4, XHTML, and the HTML DOM Level 2.
- It was specially designed to deliver rich content without the need for additional plugins. The current version delivers everything from animation to graphics, music to movies, and can also be used to build complicated web applications.
- HTML5 is also cross-platform. It is designed to work whether you are using a PC, or a Tablet, a Smartphone, or a Smart TV.
- HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

HTML5 Introduction (cont.)

- Some rules for HTML5 were established:
  - New features should be based on HTML, CSS, DOM, and JavaScript
  - The need for external plugins (like Flash) should be reduced
  - Error handling should be easier than in previous versions
  - Scripting has to be replaced by more markup
  - HTML5 should be device-independent
  - The development process should be visible to the public
- Some of the most interesting new features in HTML5 are:
  - The <canvas> element for 2D drawing
  - The <video> and <audio> elements for media playback
  - Support for local storage
  - New content-specific elements, like <article>, <footer>, <header>, <nav>, <section>
  - New form controls, like calendar, date, time, email, url, search

HTML5 New Elements

- Today, several elements in HTML 4.01 are obsolete, never used, or not used the way they were intended. All those elements are removed or re-written in HTML5.
- To better handle today's internet needs, HTML5 has also included new elements for drawing graphics, displaying media content, for better page structure and better form handling, and several new APIs, such as drag and drop, get the geographical position of a user, store local data, and more.
- The New <canvas> Element
  - <canvas> Defines graphic drawing using JavaScript
- New Form Elements
  - <datalist> Defines pre-defined options for input controls
  - <keygen> Defines a key-pair generator field (for forms)
  - <output> Defines the result of a calculation

HTML5 New Elements (cont.)

- New Media Elements
  - <audio> Defines sound or music content
  - <embed> Defines containers for external applications (like plug-ins)
  - <source> Defines sources for <video> and <audio>
  - <track> Defines tracks for <video> and <audio>
  - <video> Defines video or movie content
- New Semantic/Structural Elements
  - <article> Defines an article in the document
  - <dialog> Defines a dialog box or window
  - <details> Defines additional details that the user can view or hide
  - <figure> Defines self-contained content, like illustrations, diagrams, photos, code listings, etc.
  - <footer> Defines a footer for the document or a section
  - <header> Defines a header for the document or a section

HTML5 New Elements (cont.)

- Removed Elements
  - The following HTML 4.01 elements has been removed from HTML5:
    - <acronym>
    - <applet>
    - <basefont>
    - <big>
    - <center>
    - <dir>
    - <font>
    - <frame>
    - <frameset>
    - <noframes>
    - <strike>
    - <tt>

HTML5 Semantic Elements

- Semantic elements = Elements with meaning.
- A semantic element clearly describes its meaning to both the browser and the developer.
- Examples of non-semantic elements: <div> and <span> - Tells nothing about its content.
- Examples of semantic elements: <form>, <table>, and <img> - Clearly defines its content.
- New Semantic Elements in HTML5
  - HTML5 offers new semantic elements to clearly define different parts of a web page:
  - <header>,<nav>,<section>,<article>,<aside>,<figure>, <figcaption>, <footer>, <details>, <summary>, <mark>,<time>

HTML5 Semantic Elements (cont.)

- HTML5 <section> Element
  - The <section> element defines a section in a document.
  - According to W3C's HTML5 documentation: "A section is a thematic grouping of content, typically with a heading."
- HTML5 <article> Element
  - The <article> element specifies independent, self-contained content.
  - An article should make sense on its own and it should be possible to distribute it independently from the rest of the web site.
  - Examples of where an <article> element can be used:
    - Forum post, Blog post, News story, Comment
- HTML5 <nav> Element
  - The <nav> element defines a set of navigation links.
  - The <nav> element is intended for large blocks of navigation links. However, not all links should be inside a <nav> element!

HTML5 Semantic Elements (cont.)

- HTML5 <aside> Element
  - The <aside> element defines some content aside from the content it is placed in (like a sidebar).
  - The aside content should be related to the surrounding content.
- HTML5 <header> Element
  - The <header> element specifies a header for a document or section.
  - The <header> element should be used as a container for introductory content.
- HTML5 <footer> Element
  - The <footer> element specifies a footer for a document or section.
  - A footer typically contains the author of the document, copyright information, links to terms of use, contact information, etc.
  - You can have several <footer> elements in one document.

HTML5 Semantic Elements (cont.)

- HTML5 <figure> and <figcaption> Elements
  - The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
  - 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 should not affect the flow of the document.
  - The <figcaption> tag defines a caption for a <figure> element.
- The elements explained above are all block elements (except <figcaption>).
- To get these elements to work properly in older browsers, set the display property to block in your style sheet (this causes older browsers to render these elements correctly):
  - header, section, footer, aside, nav, main, article, figure

    {display: block; }

HTML5 Input Types

- HTML5 has several new input types for forms. These new features allow better input control and validation.
  - color type is used for input fields that should contain a color.
  - date type allows the user to select a date.
  - datetime type allows to select a date and time (with time zone).
  - datetime-local type allows to select a date and time (no time zone).
  - email type is used for input fields that should contain an e-mail address.
  - month type allows the user to select a month and year.
  - number type is used for input fields that should contain a numeric value.
  - range type is used for input fields that should contain a value from a range of numbers.
  - search type is used for search fields
  - tel type is used for input fields that should contain a telephone number.
  - url type is used for input fields that should contain a URL address.

HTML5 Form Elements

- HTML5 has the following new form elements:<datalist>, <keygen>, <output>
- HTML5 <datalist> Element
  - <datalist> element specifies a list of pre-defined options for an <input> element.
  - <datalist> element is used to provide an "autocomplete" feature on <input> elements. Users will see a drop-down list of pre-defined options.
- HTML5 <keygen> Element
  - The purpose of the <keygen> element is to provide a secure way to authenticate users.
  - <keygen> tag specifies a key-pair generator field in a form.
  - When the form is submitted, two keys are generated, one private and one public.
  - The private key is stored locally, and the public key is sent to the server. The public key could be used to generate a client certificate to authenticate the user in the future.

HTML5 Form Elements (cont.)

- HTML5 <output> Element
  - <output> element represents the result of a calculation (like one performed by a script).

  -
    ```
    <form oninput="x.value=parseInt(a.value)+parseInt(b.value)">0
    <input type="range" id="a" value="50">100 +
    <input type="number" id="b" value="50">=
    <output name="x" for="a b"></output>
    </form>
    ```

HTML5 Form Attributes

- HTML5 has several new attributes for <form> and <input>.
  - autocomplete attribute specifies whether a form or input field should have autocomplete on or off.
  - <form> novalidate attribute specifies that the form-data (input) should not be validated when submitted.
  - autofocus specifies that an <input> element should automatically get focus when the page loads.
  - <input> form attribute specifies one or more forms an <input> element belongs to.
  - formaction attribute specifies the URL of a file that will process the input control when the form is submitted. formaction attribute overrides the action attribute of the <form> element.
  - <input> formenctype attribute specifies how the form-data should be encoded when submitting it to the server (only for forms with method="post")
  - formmethod attribute defines the HTTP method for sending form-data to the action URL.

HTML5 Form Attributes (cont.)

- formnovalidate specifies that the <input> element should not be validated when submitted, overrides the novalidate attribute of the <form> element.
- formtarget attribute specifies a name or a keyword that indicates where to display the response that is received after submitting the form.
- height and width attributes specify the height and width of an <input>.
- list attribute refers to a <datalist> element that contains pre-defined options for an <input> element.
- min and max attributes specify the minimum and maximum value for an <input> element.
- multiple specifies that the user is allowed to enter more than one value in the <input> element.
- pattern attribute specifies a regular expression that the <input> element's value is checked against.
- step attribute specifies the legal number intervals for an <input> element.

Thank You

Ինստիգեյթ
Ուսումնական Կենտրոն