Hyper Text Markup Language 5

(HTML 5)

National Telecommunication Institute - Egypt

Why HTML 5?

- The newest version of HTML.
- Designed to deliver rich content without the need for additional plugins.
- Delivers everything from animation, graphics, music, and movies.
- Builds complicated web sites.
- Cross-platform; Designed to work with PCs, Tablets, Smartphones, or a Smart TVs.

Main Rules for HTML5

- New features based on HTML, CSS, DOM, and JavaScript.
- No need for external plugins.
- Error handling is easier.
- HTML5 is device-independent.
- HTML5 is supported by all major browsers like: Chrome, Firefox, Internet Explorer, Safari, and Opera.

HTML5 Codes

The HTML5 DOCTYPE declaration

- To declare the DOCTYPE in HTML 5 use:

```
<!DOCTYPE html>
```

HTML5 Codes

The HTML5 character encoding

- Very simple (The default character encoding in HTML5 is UTF-8)

```
<meta charset="UTF-8">
```

HTML5 Codes

The HTML5 Document

Example:

```
<!DOCTYPE html>
<html><head>
    <meta charset="UTF-8">
    <title>Document Title</title>
</head>
<body>
    HTML tags
</body></html>
```

HTML5Shiv

- IE8 and earlier, does not allow styling of the new HTML5 semantic elements.
- The HTML5Shiv is a JavaScript solution for this problem.
- Download and link the “html5shiv.js” file to your HTML file.

```
<head>
  <!--[if lt IE 9]>
    <script src="/js/html5shiv.js"></script>
  <![endif]-->
</head>
```

HTML 5 New Semantic Elements

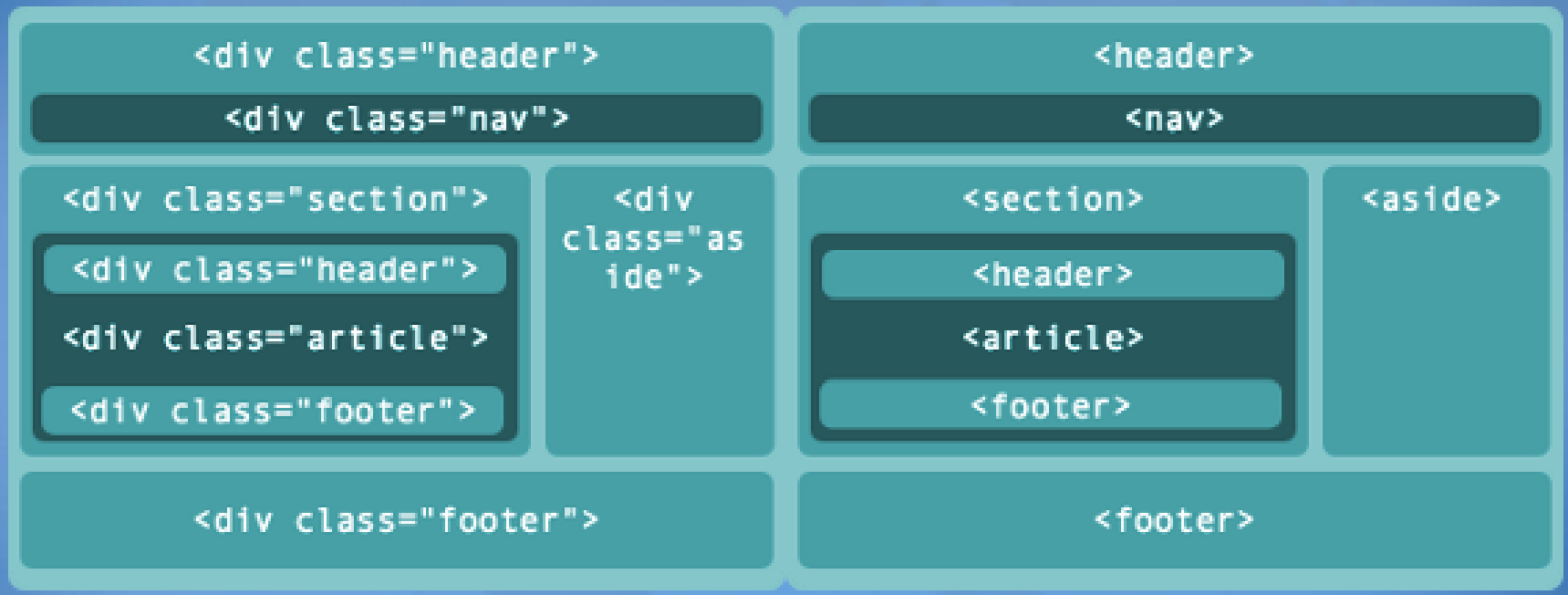
- Many web sites contain HTML code like: to indicate navigation, header, and footer.
- HTML5 offers new semantic elements to define different parts of a web page like:

<article>	<aside>	<details>	<figcaption>
<figure>	<footer>	<header>	<main>
<mark>	<nav>	<section>	<summary>

Why Semantic Elements?

- In HTML4, developers uses their own id/class names to style elements like: header, footer, navigation and container, This made it impossible for search engines to identify the correct web page content.
- With the new HTML5 elements (<header> <footer> <nav> <section> <article>), this will become easier.

Why Semantic Elements?



Why Semantic Elements?

HTML <section>

- The <section> element defines a section in a document.
- It is a group of contents, with a heading.
- A webpage could be split into many sections (Ex: introduction, content, and contact information).

```
<section>
  <h1>Welcome</h1>
  <p>This is our website for....</p>
</section>
```

HTML <article>

- The <article> element specifies independent, self-contained content.
- Examples of where an <article> element can be used:
  - Forum post, Blog post, Newspaper article.

```
<article>
  <h1>What We Offer</h1>
  <p>We are a nonprofit organization aiming to……</p>
</article>
```

HTML <header>

- The <header> element specifies a header for a document or a section.
- The <header> element should be used as a container for introductory content.
- You can have several <header> elements in one document.

HTML <header>

```
<article>
  <header>
    <h1>……</h1>
    <p>……</p>
  </header>
  <p>……</p>
</article>
```

HTML <footer>

- The <footer> element specifies a footer for a document or section.
- A <footer> element should contain information about its containing element.
- A footer contains copyright information, links to terms of use, contact information.
- You may have several <footer> elements in one document.

HTML <footer>

```
<footer>
 <p>Contact information: </p>
  <p><a  href="mailto:someone@example.com">
 someone@example.com</a></p>
 </p>
</footer>
```

HTML <nav>

- The <nav> element defines a set of navigation links.
- Notice 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.

```
<nav>
  <a href="/html/">HTML</a>
  <a href="/css/">CSS</a>
  <a href="/js/">JavaScript</a>
</nav>
```

HTML <aside>

- The <aside> element defines some content aside from the content it is placed in (like a sidebar).

```
<aside>
  <h4>NEWS</h4>
  <p>……..</p>
</aside>
```

HTML <figure> and <figcaption>

- The purpose of a figure caption is to add an explanation to an image.

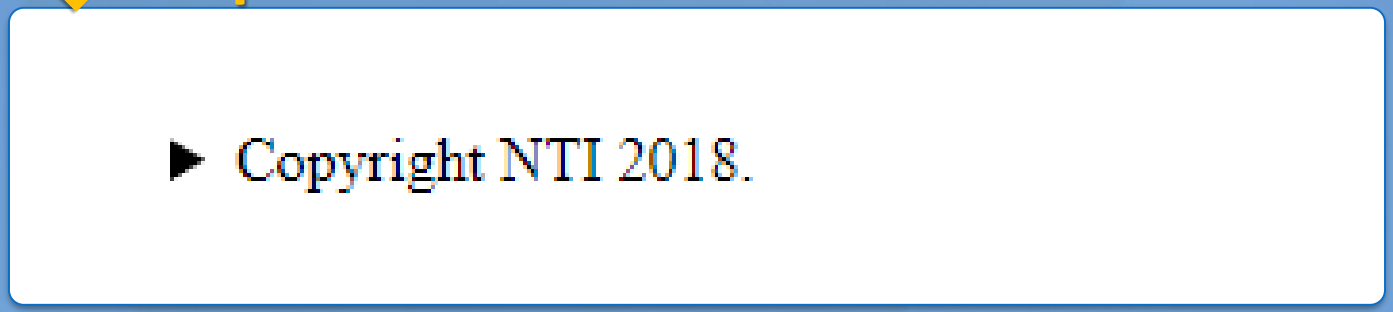
```
<figure>
  <img src="pic1.jpg">
  <figcaption>Fig1. – Cat Image.</figcaption>
</figure>
```

HTML <details>

- The <details> tag specifies additional details that the user can view or hide on demand.
- The <details> tag can be used to create an interactive widget that the user can open and close. Any sort of content can be put inside the <details> tag.
- The content of a <details> element should not be visible unless the open attribute is set.
- Not Supported in IE.

HTML <details>

Output:



HTML <mark>

- The <mark> tag defines a part of your text to be highlighted.

```
<p>Do not forget to <mark>Register</mark> now.</p>
```

HTML <main>

- The <main> tag specifies the main content of a webpage.
- The content inside the <main> element should be unique to the document. It should not contain any content that is repeated across documents.

HTML <main>

```
<main>
        <h1>……</h1>
        <p>……</p>
<article>
          <h1>……</h1>
          <p>……p>
 </article>
 <article>
          <h1>……</h1>
          <p>……p>
 </article>
</main>
```

HTML5 Videos

- The HTML5 <video> element specifies a standard way to embed a video in a web page.

```
<video width="320" height="240" controls autoplay loop
                muted  poster="img1.jpg">
    <source src="movie.mp4" type="video/mp4">
    <source src="movie.ogg" type="video/ogg">
    Your browser does not support the video tag.
</video>
```

HTML5 Videos

- Controls: attribute adds video controls, like play, pause, and volume.
- Autoplay: attribute used to start the video automatically.
- Loop: attribute used to start the video over again, every time it is finished.
- Muted: attribute used to start the video muted (no sound played).

HTML5 Videos

- Poster: attribute specifies an image to be shown while the video is downloading, or until the user hits the play button.
- Width and Height: Defines the width and the height. <source>: element specifies alternative video files the browser may choose from.
- Text: between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

HTML5 Audio

- The HTML5 <audio> element used to play an audio file a web page.

```
<audio controls>
    <source src="sound.ogg" type="audio/ogg">
    <source src="sound.mp3" type="audio/mpeg">
    Your browser does not support the audio element.
</audio>
```

HTML5 Audio

- Controls: attribute adds audio controls, like play, pause, and volume.
- Loop: attribute used to start the audio over again, every time it is finished.
- Muted: attribute specifies that the audio output should be muted.

HTML5 Audio

- <source>: element allows to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.
- Text: between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

HTML5 new from inputs

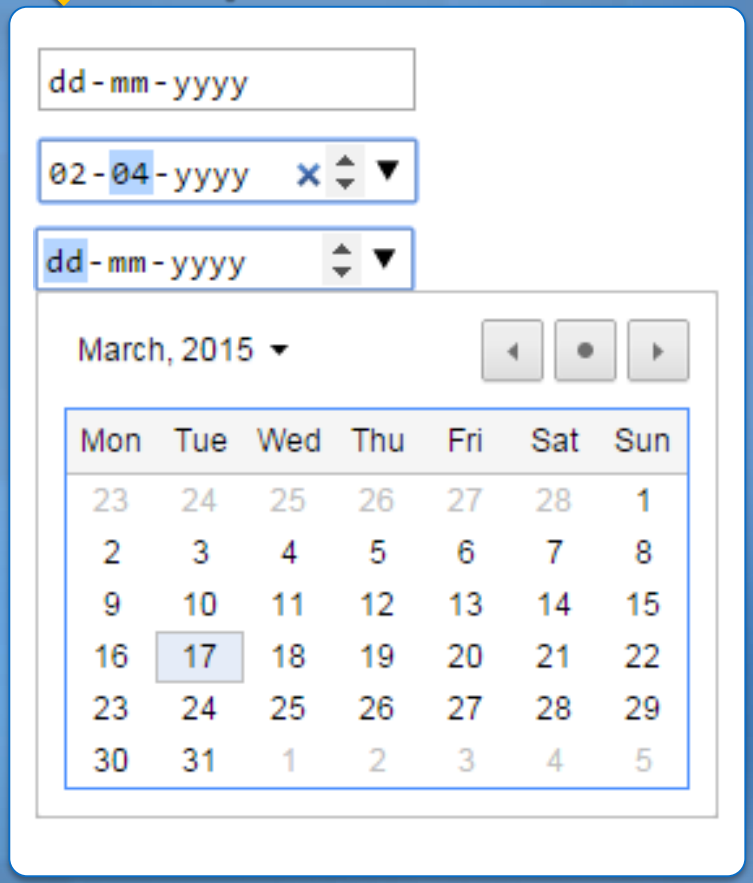
- <input type="color">
  - Used for input fields that should contain a color.

Output:

HTML5 new from inputs

- <input type="date">
  - Used for input fields that should contain a date.

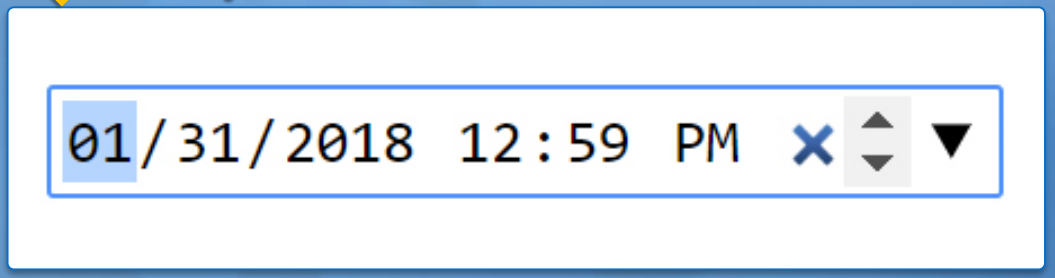
Output:



HTML5 new from inputs

- <input type="datetime-local">
  - Adds a date and time input field, with no time zone.

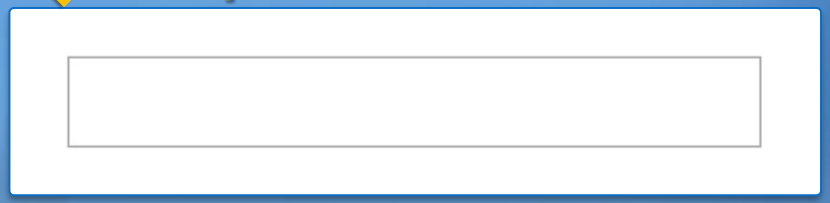
Output:



HTML5 new from inputs

- <input type="email">
  - Adds an input fields that should contain an
  - e-mail address. Can be automatically validated when submitted.

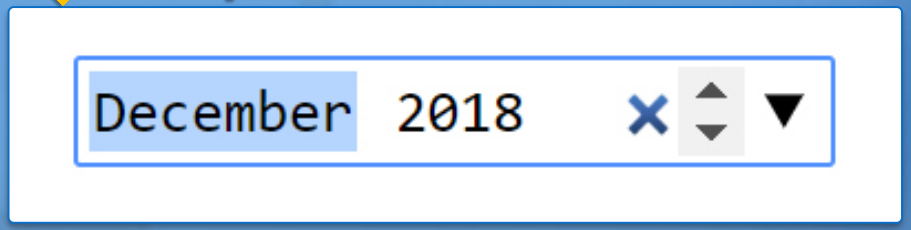
Output:



HTML5 new from inputs

- <input type="month">
  - Adds an input fields that allows the user to select a month and year.

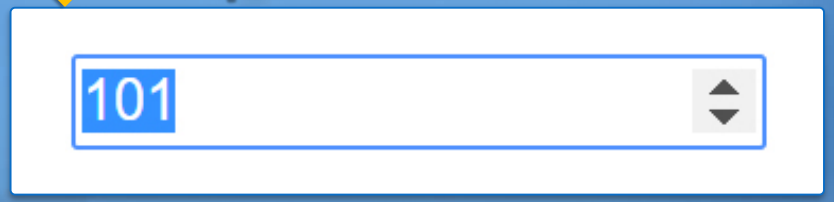
Output:



HTML5 new from inputs

- <input type="number">
  - Defines a numeric input field. Max and Min and Step attributes values can be added.

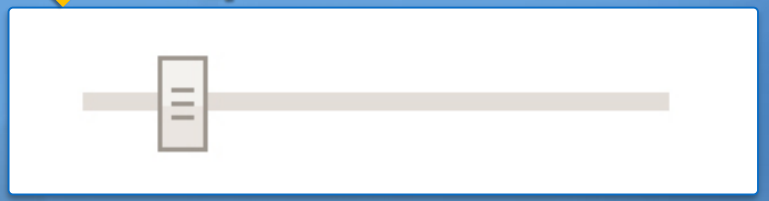
Output:



HTML5 new from inputs

- <input type="range">
  - Adds a slider control for entering a number whose exact value is not important.

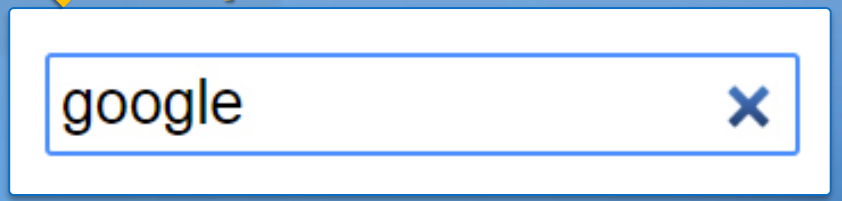
Output:



HTML5 new from inputs

- <input type="search">
  - Used to add search fields.

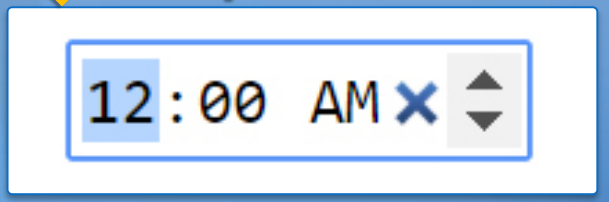
Output:



HTML5 new from inputs

- <input type="time">
  - Allows the user to select a time (no time zone).

Output:



HTML5 new from inputs

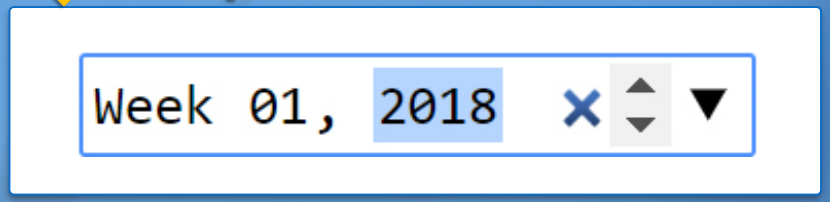
- <input type="url">
  - Used for input fields that should contain a URL address. The url field is automatically validated when submitted.

Output:

HTML5 new from inputs

- <input type="week">
  - Allows the user to select a week and year, a date picker shows up in the input field.

Output:



Data lists

- The <datalist> tag is used to provide an "autocomplete" feature on <input> element.

```
<input name="country" list="country_name">
<datalist id="country_name">
   <option value="Albania">
   <option value="Algeria">
   <option value="Austria">
</datalist>
```

Placeholder attribute

- Specifies a hint displayed in the input field before the user enters a value.
- Works only with these input types: text, search, url, tel, email, and password.

```
<input type="text" placeholder="Please enter your name" >
```

Autofocus attribute

- it specifies that an <input> element should automatically get focus when the page loads.

```
<input type="text" autofocus >
```

Tabindex attribute

- The tabindex attribute specifies the tab order of an element.

```
<ul>
   <li><a href="index.html" tabindex="1">Home</a></li>
   <li><a href="about.html" tabindex="3">About Us</a></li>
   <li><a href="contact.html" tabindex="2">Contact Us</a></li>
</ul>
```

Accesskey attribute

- Specifies a shortcut key to activate/focus an element. (using “alt+accesskey”).

```
<ul>
   <li>
     <a href="http://www.google.com" accesskey="g">
     Google</a>
   </li>
</ul>
```

Table Columns <colgroup>

- The <colgroup> tag specifies a group of one or more columns in a table for styling, instead of repeating the styles for each cell, in every row.
- The <colgroup> tag must be the second child of a <table> element, after the <caption> element.
- The <col> tag is used inside the <colgroup> tag to call each column separately.

Table Columns <colgroup>

```
<table>
   <colgroup>
      <col>
      <col class="red">
      <col>
   </colgroup>
   <tr>
      <td>cell1</td><td>cell2</td><td>cell3</td>
   </tr>
   <tr>
      <td>cell4</td><td>cell5</td><td>cell6</td>
   </tr>
</table>
```

Table Header and Footer and Body

- The <thead>, <tbody> and <tfoot> elements are used together to specify each part of a table (header, body, footer).
- The <thead> tag must be used as a child of a <table> element, after the <caption>, and the <colgroup> elements.

Table Header and Footer and Body

```
<table>
   <thead><tr>
         <td>Header 1</td><td>Header 2</td><td>Header 3</td>
      </tr></thead>
   <tfoot><tr>
         <td>Footer 1</td><td>Footer 2</td><td>Footer 3</td>
      </tr></tfoot>
   <tbody><tr>
         <td>Cell 1</td><td>Cell 2</td><td>Cell 3</td>
      </tr></tbody>
</table>
```

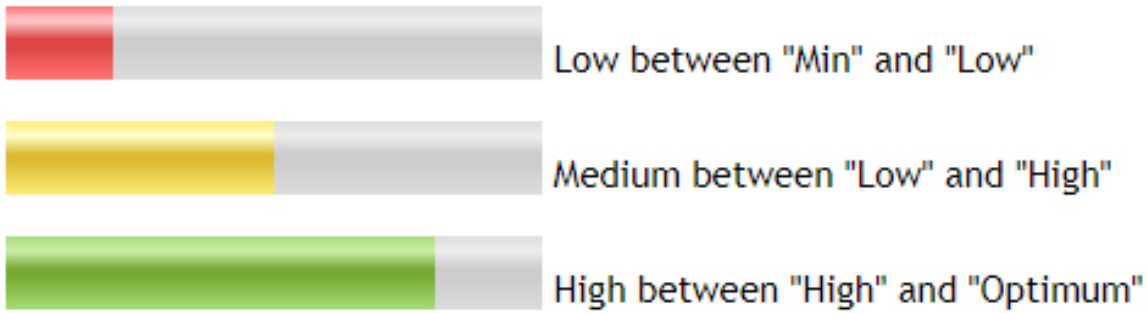
Meter Tag <meter>

- Used to display a measurement on a known scale.
- Measurements fall in any of the three categories: low, medium or high.
- The low category will be shown in red color, medium in yellow color and high in green color.
- Use the “min”, “max”, “low”, “high” and “optimum” attributes the values on the scale, and the “Value” attribute to set the desired value.
- Low: between "Min" and "Low" attributes.
- Medium: between "Low" and "High" attributes.
- High: between "High" and "Optimum" attributes.

Meter Tag <meter>

```
<p><meter value="20" min="0" max="100" low="25" high="75"
          optimum="100"></meter></p>
<p><meter value="50" min="0" max="100" low="25" high="75"
          optimum="100"></meter></p>
<p><meter value="80" min="0" max="100" low="25" high="75"
          optimum="100"></meter></p>
```

Output:



Progress Bar <progress>

- Used to create progress bars.
- You can specify the value of “max” attribute to set how much work the task requires in total and the “value” attribute to set how much of this work has been completed.
- Progress bar that has a value attribute is known as “determinate” and progress bar that doesn’t have a value attribute is known as being “indeterminate”.

Progress Bar <progress>

```
<p>
    <progress max="100"></progress>
</p>
<p>
    <progress value="60" max="100"></progress>
</p>
```

Output: