



Technical University of Sofia
Faculty of Computer Systems and Control

Web Programming

Lecture 2 – HTML Tags

Sen. Assist. Prof. PhD A. Tasheva

HTML tags again

- html
- head
- body
- title
- p
- h1, h2, ..., h6
- img

- br
- i
- b
- ul
- ol
- li

Comments

- `<!-- This is a comment -->`

- `<!--`

This paragraph,

is also a

comment...

`-->`

Special HTML

- < → <
- > → >
- & → &
- → space

Anchor Tag (Links)

Absolute HREFs specify fully qualified URLs.

```
<a href="http://www.tu-sofia.bg/">TU</a>
```

```
<a href="reldoc.html">In this directory!</a>
```

```
<a href="a/doc.html">In sub-directory a!</a>
```

Relative HREFs are relative to the directory containing the current HTML file.

Hypertext Links

Links can contain images and other text-level elements

```
<a href="www.tu-sofia.bg">  
    
</a>
```

Hypertext Links

Link to Section.

```
<a href="#par1">
```

Go to paragraph 1.

```
</a>
```

Name a Section.

```
<a name="par1">
```

Paragraph 1.

```
</a>
```

Tables

- `<table>...</table>`
- `<th>...</th>` for header row
- `<tr>...</tr>` for each row
- `<td>...</td>` for each element in a row

Table Example

```
<table border="1">
```

```
<tr>
```

```
  <td>row 1, cell 1</td>
```

```
  <td>row 1, cell 2</td>
```

```
</tr>
```

```
<tr>
```

```
  <td>row 2, cell 1</td>
```

```
  <td>row 2, cell 2</td>
```

```
</tr>
```

```
</table>
```

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

Table properties

- **Align**
 - The Align attribute gives the horizontal alignment of the table as a whole
 - Legal values are LEFT, RIGHT, and CENTER. Left is the default.
- **Border**
 - This specifies the width in pixels of the border around the table
 - This is in addition to the border around each cell
 - The default is zero.

Table properties

- **CellSpacing**

- This gives the space in pixels between adjacent cells. Drawn as a 3D line if Border is nonzero, otherwise empty space in the background color is used.
- The default is usually about 3.

- **CellPadding**

- Determines the empty space, in pixels, between the cell's border and the element
- The default is 1.

Table properties

- **Width and Height**

- Specifies the width of the table, either in pixels (<table width="250"...>) or as a percentage of the current browser window width (width="75%").
- The same with Height.

- **BgColor**

- Specify the background color of the table
- Works also with TR, TD and TH

```
class MainActivity extends AppCompatActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}

public void onClick(View view) {
    Intent intent = new Intent(this, Call
```

FORMS AND USER INPUT

The Form tag

- A form is an area that can contain form elements.

```
<form></form>
```

- HTML forms are used to pass data to a server.
- HTML forms gather information from user input.

Common Form Elements

- Commonly used form elements includes:
 - Text fields
 - Radio buttons
 - Checkboxes
 - Submit buttons
- And also:
 - select lists, textarea, fieldset, legend, and label elements.

Text Input Fields

- Used when you want the user to type letters, number, etc.

```
<form>
```

```
  First name: <input type="text"  
  name="firstname" />
```

```
<br/>
```

```
  Last name: <input type="text"  
  name="lastname" />
```

```
</form>
```

Password Field

- Used when the user needs to enter password.
- The characters in a password field are masked (shown as asterisks or circles).

```
<form>
```

```
    Password: <input type="password"  
              name="pwd" />
```

```
</form>
```

Submission Button

- When user clicks on the “Submit” button, the content of the form is sent to another file.

```
<form name="input"
  action="html_form_action.asp"
  method="get"> Username:
  <input type="text" name="user"/>
  <br/>
  <input type="submit" value="Submit"/>
</form>
```

Check-Boxes

- Used when you want the user to select one or more options of a limited number of choices.

```
<form>
```

```
<input type="checkbox" name="bike" value="bike"/> I have a bike
```

```
<br/>
```

```
<input type="checkbox" name="car" value="car"/> I have a car
```

```
</form>
```

Radio Buttons

- Used when you want the user to select one of a limited number of choices.

```
<form>
```

```
<input type="radio" name="sex"  
value="male"/> Male
```

```
<br/>
```

```
<input type="radio" name="sex"  
value="female"/> Female
```

```
</form>
```

Drop-down Menu

- Used when you want user to respond with one specific answer with choices you have given.
- The `<select>` tag is used to create a select list (drop-down list).
- The `<option>` tags inside the select element define the available options in the list.

Drop-down Menu

- Example:

```
<p>Select a fruit:</p>
```

```
<select name="Fruits">
```

```
<option selected> Apples </option>
```

```
<option> Bananas </option>
```

```
<option> Oranges </option>
```

```
</select>
```

TextArea

- The `<textarea>` tag defines a multi-line text input control.
- A text area can hold an unlimited number of characters, and the text renders in a fixed-width font (usually Courier).
- The size of a textarea can be specified by the `cols` and `rows` attributes, or even better; through CSS' `height` and `width` properties.

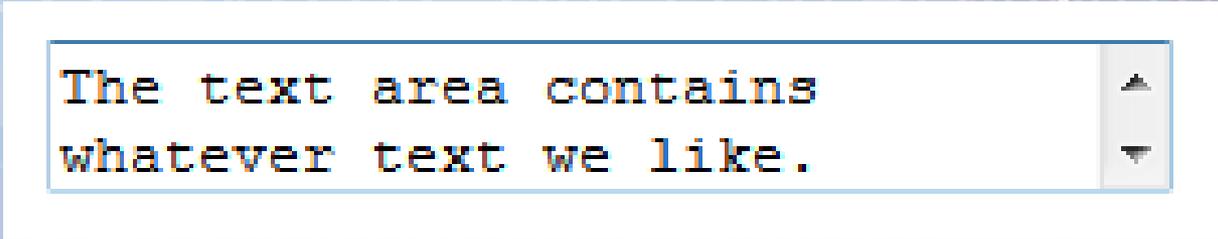
TextArea

```
<textarea rows="2" cols="20">
```

The text area contains whatever text we like.

And it has the same formatting! << ^_^ >>

```
</textarea>
```



```
The text area contains  
whatever text we like.
```

Fieldset

- Defines a border around elements in a form.
- The <fieldset> tag is used to logically group together elements in a form.
- The <fieldset> tag draws a box around the related form elements.
- The <legend> tag defines a caption for the fieldset element.

Fieldset

```
<fieldset>
```

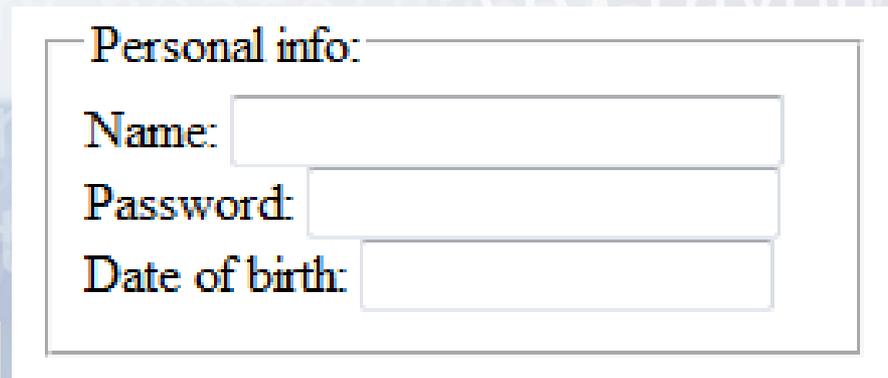
```
<legend>Personal info:</legend>
```

```
Name: <input type="text" size="23" /><br />
```

```
Password: <input type="password" size="19" /><br />
```

```
Date of birth: <input type="text" size="16" />
```

```
</fieldset>
```



Personal info:

Name:

Password:

Date of birth:

```
class MainActivity extends AppCompatActivity {  
    @Override  
    public void onCreate(Bundle savedInstanceState)  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
}
```

HTML5 BASICS

```
public void onClick(View view)  
    Intent intent = new Intent(new Call
```

Background

- HTML5 is new standard for HTML. It's an output of joint efforts from W3C (World Wide Web Consortium) and WHATWG (Web Hypertext Application Technology Working Group).
- The previous version of HTML, is HTML 4.01, came in 1999.
- HTML5 is still a work in progress. However, the major browsers support many of the new HTML5 elements and APIs.

Why we need HTML5?

Early days



Today



PC/Mac



Android Tablet



iPhone



Android



iPad



Windows Phone

Why we need HTML5?

- ❑ It introduces more efficient markup to eliminate the use of external plugging like Adobe Flash Player.
- ❑ HTML5 also reduces the use of scripting languages and it's more SEO friendly.
- ❑ It is device independent and supported by latest versions of all major web browsers.

Basic/Minimal Structure of HTML5 Document

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="utf-8"/>  
<link href="css/some-stylesheet.css" rel="stylesheet"/>  
<script src="scripts/some-script.js">  
</script>  
</head>  
< body>  
...  
< /body>  
< /html>
```

New Features of HTML5

- New Semantic\Structural Elements of HTML5
- New Attributes
- New Input Elements of HTML5
- Canvas Element
- Media Elements(Video\Audio)
- Drag and Drop Feature
- Geolocation Support
- SVG (Scalable Vector Graphics)
- Web Storage
- Offline Support

The Figure Element

- HTML Older version Code

```

```

```
<p>Image of Mars. </p>.
```

- HTML5

```
<figure>
```

```

```

```
<figcaption>
```

```
<p>This is an image of something interesting.</p>
```

```
</figcaption>
```

```
</figure>
```

No More Types for Scripts and Links

- HTML Older Version Code.

```
<link rel="stylesheet" href="path/to/stylesheet.css" type="text/css" />
```

```
<script type="text/javascript" src="path/to/script.js"></script>
```

- HTML5

```
<link rel="stylesheet" href="path/to/stylesheet.css" />
```

```
<script src="path/to/script.js"></script>
```

Make your Content Editable

- HTML Older Version

We need to add some client side script to make the things happened.

- HTML5

```
<ul contenteditable="true">
```

```
<li> Break mechanical cab driver. </li>
```

```
<li> Drive to abandoned factory
```

```
<li> Watch video of self </li>
```

```
</ul>
```

New Semantic/Structural Elements of HTML5

- <article>
- <aside>
- <header>
- <footer>
- <section>
- <nav>
- <summary>
- <figure>
- <figcaption>
- Many more...

New Input Elements of HTML5

- `<input type = "color" />`
- `<input type = "date" />`
- `<input type = "datetime" />`
- `<input type = "datetime-local" />`
- `<input type = "email" />`
- `<input type = "month" />`
- `<input type = "number" />`
- `<input type = "range" />`
- `<input type = "search" />`
- `<input type = "tel" />`
- `<input type = "time" />`
- `<input type = "url" />`
- `<input type = "week" />`

Email Input Element

- HTML Older Version Code.

```
<form action="" method="get">  
  <label for="email">Email:</label>  
  <input id="email" name="email" type="text" />  
  <button type="submit"> Submit Form </button>  
</form>
```

- HTML5

```
<form action="" method="get">  
  <label for="email">Email:</label>  
  <input id="email" name="email" type="email" />  
  <button type="submit"> Submit Form </button>  
</form>
```

Required Attribute

- HTML Older Version

We need to required some client side script to validate the things.

- HTML5

```
<form method="post" action="">
```

```
  <label for="someInput"> Your Name: </label>
```

```
  <input type="text" id="someInput"  
  name="someInput" placeholder= "John Doe"  
  required>
```

```
  <button type="submit">Go</button>
```

```
</form>
```

Placeholder

- HTML Older Version

We need to add some client side script to make the things happened.

- HTML5

```
<form method="post" action="">
```

```
  <label for="someInput"> Your Name: </label>
```

```
  <input type="text" id="someInput"
name="someInput" placeholder="Enter your name"
required>
```

```
  <button type="submit">Go</button>
```

```
</form>
```

Media Elements

<Audio> Tag

- The <Audio> element is a standard way to embed an audio file on a web page.
- HTML5 solves the dependence on external third party plug ins (like Flash)
- Supported Format's: .Ogg, .MP3 and .WAV

```
<audio src="newsong.ogg" controls="controls">
```

```
<p><Your browser does not support HTML5</p>
```

```
</audio>
```

Media Elements

<Video> Tag

- The < Video > element is a standard way to embed an video file on a web page.
- HTML5 solves the dependence on external third party plug ins (like Flash)
- Supported Format's: .Ogg, . MP4 and . WebM

```
<video src="video.AVI" type="video/ AVI" controls />
```

```
<p><Your browser does not support HTML5</p>
```

```
</video>
```

Drag N Drop

- HTML5 supports drag\drop with the help of scripting.
- No Need for any third party plugin(Jquery\DOJO).
- We have to just add **draggable=true** attribute on to the element you want to make moveable.
- There are a number of different events to attach to for monitoring the entire drag and drop process:
 - **dragstart**
 - **drag**
 - **dragenter**
 - **dragleave**
 - **dragover**
 - **drop**
 - **dragend**

Canvas Element

- `<canvas>` Element helps browser to draw shapes and images without any plugin.
- `<canvas>` element is used to draw graphics, on the fly, via scripting.
- `<canvas>` has several methods for drawing paths, boxes, circles, characters, and adding images.

```
<canvas id="mycanvas" width="400" height="300">
```

Your browser does not support HTML5

```
</canvas>
```

Offline Support

With HTML5 it is easy to make an offline version of a web application, by creating a cache manifest file. It offers following benefits

- Offline Browsing
- Speed
- Reduced server load

```
<html manifest="example.appcache">
```

```
...
```

```
</html>
```

The cache manifest file is a simple text file that lists the resources the browser should cache for offline access.

To test your browser for HTML5

- <http://html5test.com/>

**Thank you for
your attention!**



References

- http://w3schools.com/html/html5_intro.asp
- <http://www.html-5-tutorial.com>
- <http://www.html5tutorial4u.com>