

**OBJECT<sup>TM</sup>**

TECHNOLOGIES

HTML forms



# What will be covered

- Creating form
- Form attributes
- Use of get and post
- Form elements
- Different input types
- Attributes of input
- Select control
- Textarea, File upload and Hidden form field
- New in HTML 5
- New semantic elements
- HTML 5 input types
- HTML 5 form attributes
- HTML 5 input restrictions
- HTML multimedia and canvas

# Creating form

OBJECT  
TECHNIQUE

- HTML Forms are required when you want to collect some data from the site visitor. For example during user registration you would like to collect information such as name, email address, credit card, etc.
- A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc.
- The back-end application will perform required processing or the passed data based on defined business logic inside the application.
- There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.
- The HTML <form> tag is used to create an HTML form

# Form Attributes

OBJECT

```
<form action="/my-handling-form-page" method="post">  
</form>
```

**action :** This attribute is used to specify backend script ready to process your passed data.

**method :** This attribute is used to specify method to be used to upload data. The most frequently used are GET and POST methods.

**enctype :** This is used to specify how the browser encodes the data before it sends it to the server. Possible values are:

- **application/x-www-form-urlencoded** - This is the standard method most forms use in simple scenarios.
- **multipart/form-data** - This is used when you want to upload binary data in the form of files like image, word file etc.

# Use of get and post

OBJECT  
Technologies

- Before the browser sends the information, it encodes it using a scheme called URL encoding. In this scheme, name/value pairs are joined with equal signs and different pairs are separated by the ampersand.

name1=value1&name2=value2&name3=value3

- The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the ? character.

<http://www.test.com/index.htm?name1=value1&name2=value2>

- The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called QUERY\_STRING.

# Use of get and post

OBJECT

Sr No	get method	post method
1	The GET method is restricted to send upto 1024 characters only.	The POST method does not have any restriction on data size to be sent.
2	GET can't be used to send binary data, like images or word documents, to the server.	The POST method can be used to send ASCII as well as binary data.
3	Never use GET method if you have password or other sensitive information to be sent to the server.	It is safe to send sensitive information like password or account numbers using post method
4	Get methods can not be bookmarked	Post methods can be bookmarked
5	Get is the default method	Post is not the default method

### The <input> Element

- The most important form element is the <input> element.
- The <input> element can be displayed in several ways, depending on the **type** attribute.

### The <select> Element

- The <select> element defines a drop-down list.

### The <textarea> Element

- The <textarea> element defines a multi-line input field (a **text area**)

### The <button> Element

- The <button> element defines a clickable button

# Different input types

OBJECT  
HIGHLIGHTS

- Input Type Text

<input type="text"> defines a one-line text input field.

```
<form>  
First name: <br>  
<input type="text" name="firstname"><br>  
Last name: <br>  
<input type="text" name="lastname">  
</form>
```

- Input Type Password

<input type="password"> defines a password field:

```
<form>  
User name: <br>  
<input type="text" name="username"><br>  
User password: <br>  
<input type="password" name="psw">  
</form>
```

# Different input types

OBJECT

## Input Type Submit

- **<input type="submit">** defines a button for submitting form data to a form-handler.
- The form-handler is typically a server page with a script for processing input data.
- The form-handler is specified in the form's action attribute

```
<form action="action_page.php">  
First name: <br>  
<input type="text" name="firstname" value="Mickey"><br>  
Last name: <br>  
<input type="text" name="lastname" value="Mouse"><br><br>  
<input type="submit" value="Submit">  
</form>
```

# Different input types

## OBJECT

### Input Type Reset

- <input type="reset"> defines a **reset button** that will reset all form values to their default values:

```
<form action="action_page.php">  
First name:<br>  
<input type="text" name="firstname" value="Mickey"><br>  
Last name:<br>  
<input type="text" name="lastname" value="Mouse"><br><br>  
<input type="submit" value="Submit">  
<input type="reset">  
</form>
```

### Input Type Radio

- <input type="radio"> defines a **radio button**.
- Radio buttons let a user select ONLY ONE of a limited number of choices

# Different input types

OBJECT

```
<form>
  <input type="radio" name="gender" value="male" checked>
  Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
  <input type="radio" name="gender" value="other"> Other
</form>
```

## Input Type Checkbox

- **<input type="checkbox">** defines a **checkbox**.
- Checkboxes let a user select ZERO or MORE options of a **limited** number of choices

```
<form>
  <input type="checkbox" name="vehicle1" value="Bike"> I have a
  bike<br>
  <input type="checkbox" name="vehicle2" value="Car"> I have a
  car
</form>
```

# Attributes for input element

Attribute	Description
type	Indicates the type of input control and for text input control it will be set to <b>text</b> .
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	This can be used to provide an initial value inside the control.
size	Allows to specify the width of the text-input control in terms of characters.
maxlength	Allows to specify the maximum number of characters a user can enter into the text box.
checked	Set to <i>checked</i> if you want to select it by default. Used for radio button and check boxes



## Select box control

OBJECT

- A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

```
'  
<form>  
<select name="dropdown">  
<option value="Maths" selected>Maths</option>  
<option value="Physics">Physics</option>  
</select>  
</form>
```

- This will produce following result:

# Select box control

OBJECT

- Following is the list of important attributes of <select> tag:

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
size	This can be used to present a scrolling list box.
multiple	If set to "multiple" then allows a user to select multiple items from the menu.

- Following is the list of important attributes of <option> tag:

Attribute	Description
value	The value that will be used if an option in the select box box is selected.
selected	Specifies that this option should be the initially selected value when the page loads.
label	An alternative way of labeling options

# The <textarea> Element

## OBJECT

- This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

```
<form>
Description : <br />
<textarea rows="5" cols="50" name="description">
Enter description here...
</textarea>
</form>
```

- It produces following result :

Description :

Enter description here...

# The <textarea> Element

Following is the list of attributes for <textarea> tag.

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
rows	Indicates the number of rows of text area box.
cols	Indicates the number of columns of text area box

# File Upload Box

## OBJECT

- If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the `<input>` element but type attribute is set to `file`.

- Example

```
<form>
<input type="file" name="fileupload" accept="image/*" />
</form>
```

- This will produce following result

**Choose File** No file chosen

# Hidden Form Controls

## OBJECT

- Hidden form controls are used to hide data inside the page which later on can be pushed to the server.

- This control hides inside the code and does not appear on the actual page.

- Example :

```
<form>
<p>This is page 10</p>
<input type="hidden" name="pagename" value="10" />
<input type="submit" name="submit" value="Submit" />
<input type="reset" name="reset" value="Reset" />
</form>
```

This is page 10

# New in HTML5

## OBJECT

HTML5 introduces a number of new elements and attributes that helps in building a modern website.

- **New Semantic Elements** – These are like <header>, <footer>, and <section>. This helps in improving the readability of web page.
- **Forms 2.0** – Improvements to HTML web forms where new attributes have been introduced for <input> tag which helps in form validation without the effort of scripting.
- **Canvas** – This supports a two-dimensional drawing surface that you can program with JavaScript.
- **Audio & Video** – You can embed audio or video on your web pages without resorting to third-party plugins.
- **Geolocation** – Now visitors can choose to share their physical location with your web application.

# New Semantic Elements

OBJECT

- 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 <article> - Clearly defines its content.
- Many web sites contain HTML code like: <div id="nav"> <div class="header"> <div id="footer"> to indicate navigation, header, and footer.
- HTML5 offers new semantic elements to define different parts of a web page. E.g
  - <article>
  - <figure>
  - <footer>
  - <header>
  - <nav>
  - <section>
  - <summary>
  - <time>

# New Semantic Elements

## OBJECT

- These elements do not affect the look and feel but just used to increase the readability of the web page.
- Example: use of article and header

```
<article>
<header>
  <h1>What Does WWF Do?</h1>
  <p>WWF's mission:</p>
</header>
<p>WWF's mission is to stop the degradation of our planet's
natural environment,
and build a future in which humans live in harmony with
nature.</p>
</article>
```

# HTML5 input types

OBJECT

HTML5 added several new input types:

- color
- date
- datetime
- email
- month
- number
- range
- search
- tel
- time
- url
- week



# HTML 5 form attributes

OBJECT

- **placeholder** - It allows us to set placeholder text used for short descriptions. This text is deleted as soon as input gets focus.

Username

- **autofocus** - Adding it to an input automatically focuses that field when the page is rendered
- **autocomplete** - The autocomplete attribute helps users complete forms based on earlier input.
- **required** - the browser requires the user to enter data into that field before submitting the form. This replaces the basic form validation implemented with JavaScript

# HTML5 form attributes

## OBJECT

First Name

Last name  This is a required field

- list and the datalist element - The list attribute enables the user

to associate a list of options with a particular field. The value of the list attribute must be the same as the ID of a datalist element. datalist element represents a predefined list of options for form controls

<label>Your favorite fruit:

```
<datalist id="fruits">
<option value="Blackberry">Blackberry</option>
<option value="Blackcurrant">Blackcurrant</option>
<option value="Blueberry">Blueberry</option>
```



<!---->

</datalist>

If other, please specify:

```
<input type="text" name="fruit" list="fruits">
</label>
```

# HTML 5 input restrictions

## OBJECT

### The min and max Attributes

- The **min** and **max** attributes specify the minimum and maximum values for an **<input>** element.
- The min and max attributes work with the following input types: number, range, date, datetime-local, month, time and week.

Enter a date before 1980-01-01:

```
<input type="date" name="bday" max="1979-12-31">
```

Enter a date after 2000-01-01:

```
<input type="date" name="bday" min="2000-01-02">
```

Quantity (between 1 and 5):

```
<input type="number" name="quantity" min="1" max="5">
```

# HTML 5 input restrictions

## OBJECT

### The pattern Attribute

- The **pattern** attribute specifies a regular expression that the `<input>` element's value is checked against.
- The pattern attribute works with the following input types: text, search, url, tel, email, and password.

```
Country code: <input type="text" name="country_code" pattern="[A-Za-z]{3}" title="Three letter country code">
```

### The step Attribute

- The **step** attribute specifies the legal number intervals for an `<input>` element.
- Example: if `step="3"`, legal numbers could be -3, 0, 3, 6, etc.
- The step attribute can be used together with the max and min attributes to create a range of legal values.

# HTML multimedia

## OBJECT

- Multimedia comes in many different formats. It can be almost anything you can hear or see.
- Examples: Images, music, sound, videos, records, films, animations, and more.
- Web pages often contain multimedia elements of different types and formats.
- Multimedia elements (like audio or video) are stored in media files.
- Multimedia files have formats and different extensions like: .swf, .wav, .mp3, .mp4, .mpg, .wmv, and .avi.
- MP4 is supported by HTML5. MP3 is the newest format for compressed recorded music.
- Only MP3, WAV, and Ogg audio are supported by the HTML5 standard.
- Before audio and video could only be played in a browser with a plug-in (like flash)

# HTML multimedia

## OBJECT

```
<audio controls>
```

```
  <source src="horse.ogg" type="audio/ogg">
```

```
  <source src="horse.mp3" type="audio/mpeg">
```

Your browser does not support the audio element.

```
</audio>
```

- controls attribute adds audio/video controls, like play, pause, and volume.

- The <source> element allows you to specify alternative audio/video files which the browser may choose from

```
<video width="320" height="240" controls>
```

```
  <source src="movie.mp4" type="video/mp4">
```

```
  <source src="movie.ogg" type="video/ogg">
```

Your browser does not support the video tag.

```
</video>
```

- HTML5 element <canvas> gives you an easy and powerful way to draw graphics using JavaScript. It can be used to draw graphs, make photo compositions or do simple (and not so simple) animations.

- Here is a simple <canvas> element which has only two specific attributes **width** and **height** plus all the core HTML5 attributes like id, name and class etc

```
<canvas id="mycanvas" width="100" height="100"></canvas>
```



- The <canvas> is initially blank, and to display something, a script first needs to access the rendering context and draw on it.
- The canvas element has a DOM method called **getContext**, used to obtain the rendering context and its drawing functions. This function takes one parameter, the type of context **2d**.

# HTML5 Canvas

OBJECT

```
var canvas = document.getElementById("mycanvas");
```

```
if (canvas.getContext){
```

```
    var ctx = canvas.getContext('2d');
```

```
    // drawing code here
```

```
}
```

```
else {
```

```
    // canvas-unsupported code here
```



## Html Forms

Forms are one of the most important elements in the HTML tags which is used for creating user interface and accepting input from the user using different types of input controls like text filed, password field, check boxes, radio buttons, drop down boxes etc. We will throw a light on all these features in this chapter.

### Introduction to HTML Forms

The `<form>` HTML element represents a document section containing interactive controls for submitting information. Web forms are one of the main points of interaction between a user and a website or application. Forms allow users to enter data, which is generally sent to a web server for processing and storage.

A web form's HTML is made up of one or more form controls, plus some additional elements like labels or span elements to show some error messages. The controls can be single or multi-line text fields, dropdown boxes, buttons, checkboxes, or radio buttons.

Form controls can also be programmed to enforce specific formats or values to be entered (form validation).

All forms start with a `<form>` element, like this:



```
<form action="/my-handling-form-page" method="post">...</form>
```

This element formally defines a form. It's a container element like a `<section>` or `<footer>` element, but specifically for containing forms. There are many attributes for form element but many of its attributes are optional, but it's standard practice to always set at least the action and method attributes.

- The action attribute defines the location (URL) where the form's collected data should be sent when it is submitted.
- The method attribute defines which HTTP method to send the data with (usually get or post).

### Use of get and post

Before the browser sends the information, it encodes it using a scheme called URL encoding. In this scheme, name/value pairs are joined with equal signs and different pairs are separated by the ampersand.

```
name1=value1&name2=value2&name3=value3
```

The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the '?' character.

```
http://www.test.com/index.htm?name1=value1&name2=value2
```

The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called `QUERY_STRING`.

get method

post method

## get method

The GET method is restricted to send upto 1024 characters only

GET can not be used to send binary data like images or word documents to the server

Never use GET method if the sensitive information like password needs to be send to the server.

GET methods can be bookmarked

GET is the default method

## post method

The POST method does not have any restriction on data size to be sent.

The POST method can be used to send ASCII as well as binary data.

It is safe to send sensitive information like password or account numbers using post method

Post methods can not be bookmarked

Post is not the default method

## Different Form elements



### The <input> Element

The most important form element is the <input> element. The <input> element can be displayed in several ways, depending on the type attribute.

#### Input Type Text

<input type="text"> defines a one-line text input field.

```
<form>
```

```
First name:<br>
<input type="text" name="firstname"><br>
Last name:<br>
<input type="text" name="lastname">
</form>
```

#### Input Type Password

<input type="password"> defines a password field:

```
<form>
```

```
User name:<br>
<input type="text" name="username"><br>
User password:<br>
<input type="password" name="psw">
```

## Different Form elements

### The <input> Element

The most important form element is the <input> element. The <input> element can be displayed in several ways, depending on the type attribute.

#### Input Type Text

<input type="text"> defines a one-line text input field.

#### <form>

```
First name:<br>
<input type="text" name="firstname"><br>
Last name:<br>
<input type="text" name="lastname">
</form>
```

#### Input Type Password

<input type="password"> defines a password field:

↳

```
<form>
User name:<br>
<input type="text" name="username"><br>
User password:<br>
<input type="password" name="psw">
</form>
```

#### Input Type Radio

<input type="radio"> defines a radio button. Radio buttons let a user select ONLY ONE of a limited number of choices

```
<form>
<input type="radio" name="gender" value="male" checked>
Male<br>
<input type="radio" name="gender" value="female"> Female<br>
<input type="radio" name="gender" value="other"> Other
</form>
```

Input Type Checkbox <input type="checkbox"> defines a checkbox. Checkboxes let a user select ZERO or MORE options of a limited number of choices

```
<form>
```

```
<input type="checkbox" name="vehicle1" value="Bike"> I have a
```

## Input Type Radio

The `<input type="radio">` defines a radio button. Radio buttons let a user select ONLY ONE of a limited number of choices

```
<form>
  <input type="radio" name="gender" value="male" checked>
  Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
  <input type="radio" name="gender" value="other"> Other
</form>
```

Input Type Checkbox `<input type="checkbox">` defines a checkbox. Checkboxes let a user select ZERO or MORE options of a limited number of choices

```
<form>
  <input type="checkbox" name="vehicle1" value="Bike"> I have a
  bike<br>
  <input type="checkbox" name="vehicle2" value="Car"> I have a
  car
</form>
```

## The `<select>` Element

The `<select>` element defines a drop-down list. A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

```
<form>
  <select name="dropdown">
    <option value="Maths" selected>Maths</option>
    <option value="Physics">Physics</option>
  </select>
</form>
```

This will produce following result:

```
[Maths ▾]
```

## The `<textarea>` Element

The `<textarea>` element defines a multi-line input field (a text area). This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML `<textarea>` tag.

## The <textarea> Element

The <textarea> element defines a multi-line input field (a text area). This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

```
<form>
  Description : <br />
  <textarea rows="5" cols="50" name="description">
    Enter description here...
  </textarea>
</form>
```

It produces following result :

Description :

Enter description here...

## Button Element

This element defines a clickable button.

<input type="submit"> defines a button for submitting form data to a form-handler. The form-handler is typically a server page with a script for processing input data. The form-handler is specified in the form's action attribute

<input type="reset"> defines a reset button that will reset all form values to their default values

<input type="button"> defines the button which is used for calling some java script code for performing some client side operations.

## File upload element

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to file. Example

```
<form>
  <input type="file" name="fileupload" accept="image/*" />
</form>
```

This will produce following result

Choose File No file chosen

## File upload element

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the `<input>` element but type attribute is set to file. Example

```
<form>
<input type="file" name="fileupload" accept="image/*" />
</form>
```

This will produce following result

Choose File No file chosen

## Hidden form element

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. Example:

```
<form>
<p>This is page 10</p>
<input type="hidden" name="pagename" value="10" />
<input type="submit" name="submit" value="Submit" />
<input type="reset" name="reset" value="Reset" />
</form>
```

This is page 10

Submit Reset

## Advantages of HTML5

As the name suggests, HTML5 is the 5th version of HTML (Hypertext Markup Language). HTML5 is the latest specification of the HTML language, and represented a major break with previous markup practices. HTML5 comes with a variety of new and advanced features that make the life of developers more flexible.

The purpose of the profound changes to the language was to standardize the many new ways in which developers were using it, as well as to encourage a single set of best practices with regards to web development.

## Encouraging Semantic Markup

Semantic markup means markup which has meaning, rather than markup which simply looks a certain way. For example, the `<h1>` tag implies that the content of the element is the title or headline of the entire document. That semantic meaning would

## Advantages of HTML 5

As the name suggests, HTML5 is the 5th version of HTML (Hypertext Markup Language). HTML5 is the latest specification of the HTML language, and represented a major break with previous markup practices. HTML5 comes with a variety of new and advanced features that make the life of developers more flexible.

The purpose of the profound changes to the language was to standardize the many new ways in which developers were using it, as well as to encourage a single set of best practices with regards to web development.

### Encouraging Semantic Markup

Semantic markup means markup which has meaning, rather than markup which simply looks a certain way. For example, the `<h1>` tag implies that the content of the element is the title or headline of the entire document. That semantic meaning would be lost if we just made the text bold and large without using the appropriate tag.

HTML has always had a little bit of semantic markup available to it: heading tags, the link rel attribute, and document metadata. But it wasn't enough. In previous versions of the language, common structural elements like page headers, navigation menus, and main content sections were all indicated with the same HTML element, the `<div>` tag. In HTML, there are a host of new semantic elements intended to indicate the basic structure of a page:

```
<header>
<nav>
<main>
<article>
<aside>
<section>
<footer>

example :

<nav class="crumbs">
<ol>
    <li class="crumb"><a href="#">Bikes</a></li>
    <li class="crumb"><a href="#">BMX</a></li>
    <li class="crumb">Jump Bike 3000</li>
</ol>
</nav>
```

This improves web page readability and understandability for the programmer which helps in easy modification and maintenance.

### Audio & Video Tags

With Audio and Video tags, developers can embed videos or audio into their websites. For styling the video tag, developers can use CSS and CSS3.

```
<!DOCTYPE html>
<html>
<body>
<video width="400px" height="300px" controls>
<source src=" movie.mp4 " type="video/mp4">
</video>
</body>
</html>
```

- width and height are for setting the dimensions for the video element
  - controls attribute creates buttons for playback (Play, Pause, etc.)
  - src (source) tag provides the URL or media source
  - type represents the video type
- HTML was originally created for (hyper-)text documents, with perhaps a few images, not rich media pages with audio and video. They required users to add special plugins like flash to their browsers. With audio and video tags, there is no dependency on external plugins.

### Canvas Tag

The canvas tag in HTML5 helps users draw graphics or images on the fly using JavaScript. We can use it for drawing paths, boxes, circles, adding images, etc. The canvas tag has two attributes: width and height for setting the width and height of the canvas.

```
<!DOCTYPE html>
<html>
<body>
<canvas id="Canvas1" width="400" height="100" style="border:2px solid;">
</canvas>
```

- width and height are for setting the dimensions for the video element

- controls attribute creates buttons for playback (Play, Pause, etc.)
- src (source) tag provides the URL or media source
- type represents the video type

HTML was originally created for (hyper-)text documents, with perhaps a few images, not rich media pages with audio and video. They required users to add special plugins like flash to their browsers. With audio and video tags, there is no dependency on external plugins.

### Canvas Tag

The canvas tag in HTML5 helps users draw graphics or images on the fly using JavaScript. We can use it for drawing paths, boxes, circles, adding images, etc. The canvas tag has two attributes: width and height for setting the width and height of the canvas.

```
<!DOCTYPE html>
<html>
  <body>
    <canvas id="Canvas1" width="400" height="100" style="border:2px solid;">
    </canvas>
  </body>
</html>
```

### New types for input tags

input is an attribute which is an old attribute but in HTML, it is reintroduced with new values like email, month, number, range, search, tel, color, week, url, time, date, datetime-local etc. These are the new values which can be contained by the input tag.

### Validation attributes

HTML5 added new attributes that declaratively set validation rules for a given input field. These new attributes include:

- required: Specifies that the field can't be blank
- min/max: Specifies the range of allowed values for numeric inputs
- minlength/maxlength: Specifies the range of allowed length for text inputs
- step: For a step of n, a numeric input value is only valid if it is a multiple of n
- pattern: Specifies a regular expression that a text input must match

## Validation attributes

- required: Specifies that the field can't be blank
  - min/max: Specifies the range of allowed values for numeric inputs
  - minlength/maxlength: Specifies the range of allowed length for text inputs
  - step: For a step of n, a numeric input value is only valid if it is a multiple of n
  - pattern: Specifies a regular expression that a text input must match
- These attributes help in reducing java script coding for validation of data entered by user.

## CSS selectors

HTML5 also introduced some new CSS selectors, including two new pseudo-classes — :valid and :invalid. These match any input whose value has passed or failed validation, respectively.

For example, we can automatically mark invalid fields with a red border:

```
input:invalid {  
    border: 2px solid red;  
}  
  
input:valid {  
    border: 2px solid green;  
}
```

## New Form Elements and Attributes

HTML5 added several new input types: color, date, datetime, email, month, number, range, search, tel, time, url, week. These new elements will look like :

Type color:

Type time:

Type number:

Type range:

## New Form Elements and Attributes

HTML5 added several new input types: `color`, `date`, `datetime`, `email`, `month`, `number`, `range`, `search`, `tel`, `time`, `url`, `week`. The new elements will look like :

Type color:

Type time:

Type number:

Type range:

Type search:

Type tel:

## HTML 5 form attributes

**placeholder** - It allows us to set placeholder text used for short descriptions. This text is deleted as soon as input gets focus.

Username

Username

- autofocus** - Adding it to an input automatically focuses that field when the page is rendered
- autocomplete** - The autocomplete attribute helps users complete forms based on earlier input.

## HTML 5 input restrictions

**required** - the browser requires the user to enter data into that field before submitting the form. This replaces the basic form validation implemented with JavaScript

First Name

1. This is a required field

## HTML 5 form attributes

**placeholder** - It allows us to set placeholder text used for short descriptions. This text is deleted as soon as input gets focus.

**Username**

**Username**

**autofocus** - Adding it to an input automatically focuses that field when the page is rendered

**autocomplete** - The autocomplete attribute helps users complete forms based on earlier input.

## HTML 5 input restrictions

**required** - the browser requires the user to enter data into that field before submitting the form. This replaces the basic form validation implemented with JavaScript

First Name

Last Name

This is a required field

## The min and max Attributes

The min and max attributes specify the minimum and maximum values for an <input> element. The min and max attributes work with the following input types: number, range, date, datetime-local, month, time and week.

Enter a date before 1980-01-01:

```
<input type="date" name="bday" max="1979-12-31">
```

Enter a date after 2000-01-01:

```
<input type="date" name="bday" min="2000-01-02">
```

Quantity (between 1 and 5):

```
<input type="number" name="quantity" min="1" max="5">
```

## The pattern Attribute

The pattern attribute specifies a regular expression that the <input> element's value is checked against. The pattern attribute

### The pattern Attribute

The pattern attribute specifies a regular expression that the <input> element's value is checked against. The pattern attribute works with the following input types: text, search, url, tel, email, and password.

```
Country code: <input type="text" name="country_code" pattern="[A-Za-z]{3}" title="Three letter country code">
```

### The step Attribute

The step attribute specifies the legal number intervals for an <input> element. Example: if step="3", legal numbers could be -3, 0, 3, 6, etc. The step attribute can be used together with the max and min attributes to create a range of legal values.

### Assignments

1. Create a login form which accepts login id and password from the user. Form should have login(submit) and clear(reset) button. For submitting the data observe query string for get and post method

UID :	<input type="text"/>
PWD :	<input type="password"/>
<input type="button" value="LOGIN"/>	<input type="button" value="CLEAR"/>

2. Create a form like one below :

Visitor Information	
Email Address:	<input type="text"/>
Do you use web for:	<input checked="" type="checkbox"/> Research <input type="checkbox"/> Purchase
Occupation:	(Select One) ▾
Want us to send you junk email?	<input checked="" type="checkbox"/> Absolutely <input type="checkbox"/> No way, man
<input type="button" value="SUBMIT"/> <input type="button" value="RESET"/>	

3. Create a web page for the registration of the user :



### Assignments

1. Create a login form which accepts login id and password from the user. Form should have login(submit) and clear(reset) button. For submitting the data observe query string for get and post method

UID :	<input type="text"/>
PWD :	<input type="password"/>
<b>LOGIN</b>	<b>CLEAR</b>

2. Create a form like one below :

Visitor Information	
Email Address:	<input type="text"/>
Do you use web for:	<input type="checkbox"/> Research <input type="checkbox"/> Purchase (Select One) ▾
Occupation:	
Want us to send you junk email?	<input type="radio"/> Absolutely <input type="radio"/> No way, man
SUBMIT	RESET

3. Create a web page for the registration of the user :

First Northern Bank 

### New User Registration

In order to use FNB's All Day Banking Service, please complete the following details and then the submit button to complete the registration cycle.

Last name   
First name   
Your preferred user id   
Email Address   
Account number   
Credit card number   
Online password   
Online password (repeated)

**3. Create a web page for the registration of the user :**



## New User Registration

In order to use FNB's All Day Banking Service, please complete the following details and then the submit button to complete the registration cycle.

last name	<input type="text"/>
first name	<input type="text"/>
your preferred user id	<input type="text"/>
Email Address	<input type="text"/>
Account number	<input type="text"/>
Credit card number	<input type="text"/>
online password	<input type="text"/>
online password (repeated)	<input type="text"/>

4. Create a form having following layout :

### **Express Registration**

please note that all fields that have an asterisk (\*) are required to process your registration.

Personal Details	Address
Title: <input type="text"/> (Mr/Mrs/Miss)	Address: <input type="text"/>
First Name: <input type="text"/>	Town: <input type="text"/>
Last Name: <input type="text"/>	County/State: <input type="text"/>
Email Address: <input type="text"/>	Country: <input type="text"/>
Telephone: <input type="text"/>	Postcode: <input type="text"/>
Cell Phone: <input type="text"/>	
Comment:	<input type="text"/>
<input type="checkbox"/> friend <input type="checkbox"/> site owner <input type="checkbox"/> search engine	

4. Create a form having following layout :

### Express Registration

please note that all fields that have an asterisk (\*) are required to process your registration.

Personal Details		Address	
Title:	(Mr/Mrs/Miss)	Address:	-
First Name:	*	Town:	-
Last Name:	*	Country/State:	-
Email Address:	*	Country:	United States
Telephone:	*	Postcode:	-
Cell Phone:	*	Comment	<input type="text"/>
Feedback	<input type="text"/>	Hear us from?	<input type="checkbox"/> Friend <input type="checkbox"/> Site owner <input type="checkbox"/> Search Engine
<b>Security Details</b>			
Choose Password:	<input type="password"/>	Confirm Password:	<input type="password"/>

5. Create a form for accepting birthdate, birthtime, email and name from user
1. For email give validation as required and max length as 15
  2. For name supply datalist for the user to select
  3. Make all the field compulsory
  4. Display hints in email and name text field for the user instead of creating labels
  6. Use audio and video tags for displaying in audio, video in web page

## Important Questions for: Html Forms

### Interview Questions

- 1.Inline and Block key difference ?
- 2.Writing a element ?,eg : Input type..
- 3.Date Format do you know about Pipe?
- 4.Write for each loop ?