

---

---

# Forms

FSDI 103

---

---

# Forms

The FORM element is used to create a data input form.

A region using forms is enclosed within the <FORM> </FORM> tags.

The FORM element has four attributes:

| Attribute | Description  |
|-----------|--|
| action    | Backend script ready to process your passed data.  |
| method    | Method to be used to upload data. The most frequently used are GET and POST methods.   |
| target    | Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.  |
| enctype   | <p>You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are:</p> <ul style="list-style-type: none"><li>• <b>application/x-www-form-urlencoded</b> - This is the standard method most forms use in simple scenarios.</li><li>• <b>multipart/form-data</b> - This is used when you want to upload binary data in the form of files like image, word file etc.</li></ul> |

# Example

```
<Form Action="getemp.asp" Method="post">
```

(all form elements inside)

```
</Form>
```

# Form INPUT

Only used within a FORM element and is denoted by <INPUT>.

Attributes:

**NAME:** The name of the particular element.

**MAXLENGTH:** The maximum number of characters that can be entered by users in a text field.

**SIZE:** Specifies the size of the field and depends on its type.

**SRC:** Denote URL for an image.

**VALUE:** Contain the initial value displayed to users.

**TYPE:** Defines the type of data used in the field.

**CHECKED:** Indicates that a checkbox or radio button is selected.

**DISABLED:** Prevents the field from receiving focus.

**ALIGN:** Alignment if image is used.

**READONLY:** Prevents modification of the contents of the field.

# Checkbox

An object where several values can be selected at the same time.  
The checkbox is submitted as separate name/value pair for each selected value.

Checkbox that are grouped together should have

- a different name.
- a unique value.

# Example

<FORM>

What pets do you own?

<P><INPUT TYPE=CHECKBOX name = petdog value="dog"> DOG

<BR><INPUT TYPE=CHECKBOX name = petcat value="cat"> CAT

<BR><INPUT TYPE=CHECKBOX name = petbird value="bird"> BIRD

<BR><INPUT TYPE=CHECKBOX name = petfish value="fish"> FISH

</FORM>

# Radio Boxes

An object that defines an item where only one value can be selected from a set of possibilities.

A set is defined as the group of radio boxes with the same NAME attribute. Never set more than one radio box at a time with the CHECKED attribute.

Radio buttons:<br>

```
<input type="radio" name="radiobutton" value="myValue1"> male<br>
```

```
<input type="radio" name="radiobutton" value="myValue2" checked> female
```

Radio buttons:

☐ male

☒ female

# Text Type

Specifies a single line text entry field.

Can be used with the MAXLENGTH and SIZE attributes (MAXLENGTH >= SIZE)

```
<P><B> First Name:</B> <INPUT NAME="fname" TYPE = text MAXLENGTH=30 SIZE  
=30></P>
```

```
<P><B> Last Name:</B> <INPUT NAME="lname" TYPE = text MAXLENGTH=30 SIZE  
=30></P>
```



# Password

Same as text except the text entered by the user is obscured.  
Use the MAXLENGTH and SIZE attributes.

```
<P><B> Enter Your Password:</B>
```

```
<INPUT NAME="password" TYPE = password MAXLENGTH=30 SIZE =30></P>
```

# Hidden

`<input type="hidden" ...>`

- Creates a control similar to a text control
  - User does not see control
  - User can not easily change the value
- Useful for keeping track of data as the user traverses a collection of pages

`<input type="hidden" name="hiddendata" value="Hidden Data in Here">`

# SUBMIT and RESET

SUBMIT: Used to submit the form's content, as specified by the ACTION attribute.

RESET: Set all fields in the form to their initial values.

```
<P>INPUT TYPE=SUBMIT>
```

```
<INPUT TYPE=RESET>
```

```
<P><INPUT TYPE=SUBMIT VALUE = "Place Your Order">
```

```
<INPUT TYPE=RESET VALUE = "Start over">
```

# Button

Creates a button whose use can be defined through scripting and onClick event.

Use to create a back button.


Only useful to browsers that support scripting.


<FORM>


<P><INPUT TYPE="button" VALUE="Back to Last Document" onClick="history.back( )">

</P>

</FORM>

A submit button: 

A reset button: 

A plain button: 

# Image button

`<input type="image" ...>`

- Displays an image that behaves like a submit button
- The SRC attribute specifies the location of an image file
- The ALT attribute specifies some text to render if the image is not displayable

`<input type="image" src="button.png" alt="Submit">`

# Textarea

Let users enter more than one line of text.

Uses attributes ROWS and COLS to size.

WRAP Attribute:

OFF: No wrapping

VIRTUAL: Display wraps but long lines are sent as one line.

PHYSICAL: Word wraps and text is sent with wrap points.

```
<textarea rows="30" cols="50" name="bigtext">
```

The preformatted initial text is sandwiched within the tag.

```
</textarea>
```

# Menus

Drop-down menus are created using the SELECT tag

- Attribute SIZE determines how many rows to display at once
- Each option is enclosed in an OPTION tag

```
<select name="country" size="5">
```

```
<option value="AB">Abkhazia</option>
```

```
...
```

```
<option value="ZB">Zimbabwe</option>
```

```
</select>
```

# Labels

The LABEL tag specifies that the enclosed item is a label for the named form element. For example, clicking the label will shift the focus or change the state of the associated form element.

```
<input type="checkbox" name="doglover" id="dogs"checked>  
<label for="dogs">I like dogs</label>  
<br>  
<input type="checkbox" name="catlover" id="cats">  
<label for="cats">I like cats</label>  
<br>  
<input type="checkbox" name="piglover" id="pigs">  
<label for="pigs">I like pigs</label>
```



# Fieldset

The FIELDSET tag is used to group together a set of related form elements. The LEGEND tag assigns a caption to a field set

```
<fieldset>
<legend>Personal Information</legend>
First Name: <input type="text" name="fn" size="20">
<br>
Last Name: <input type="text" name="ln" size="20">
<br>
Date of Birth: <input type="text" name="dob"
size="10">
</fieldset>
```

# Date

The `<input type="date">` is used for input fields that should contain a date.

Depending on browser support, a date picker can show up in the input field.

```
<form action="/action_page.php">  
  Birthday:  
  <input type="date" name="bday">  
  <input type="submit">  
</form>
```

Birthday:

# Email

The `<input type="email">` is used for input fields that should contain an e-mail address.

Depending on browser support, the e-mail address can be automatically validated when submitted.

Some smartphones recognize the email type, and adds ".com" to the keyboard to match email input.

```
<form action="/action_page.php">  
  E-mail:  
  <input type="email" name="email">  
  <input type="submit">  
</form>
```

E-mail:

# Search

The `<input type="search">` is used for search fields (a search field behaves like a regular text field).

```
<form action="/action_page.php">  
  Search Google:  
  <input type="search" name="googlesearch">  
  <input type="submit">  
</form>
```

Search Google:

# URL

The `<input type="url">` is used for input fields that should contain a URL address.

Depending on browser support, the url field can be automatically validated when submitted.

Some smartphones recognize the url type, and adds ".com" to the keyboard to match url input.

```
<form action="/action_page.php">  
  Add your homepage:  
  <input type="url" name="homepage">  
  <input type="submit">  
</form>
```

Add your homepage:

# Number and range

The number and range input types also accept parameters for min, max and step. In most cases you can leave out step as it defaults to 1.

Here you see an example including both a number input, typically displayed as a 'roller' and a range input displayed as a 'slider':

Age:   
Satisfaction:



# HTML Validations

## 1. Required

The simplest change you can make to your forms is to mark a text input field as 'required':

Your Name: `<input type="text" name="name" required>`

Your Name:

Your Name:

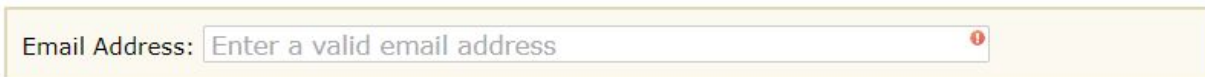
# HTML validations

## 2. input type

This is where HTML5 really gets interesting and more useful. Along with the text input type, there are now a host of other options, including email, url, number, tel, date and many others.

By changing the input type to email while also using the required attribute, the browser can be used to validate (in a limited fashion) email addresses:

Email Address: `<input type="email" name="email" required placeholder="Enter a valid email address">`

A screenshot of a web form. It features a label 'Email Address:' followed by a text input field. The input field has a light blue placeholder text that reads 'Enter a valid email address'. To the right of the input field, there is a small red circular icon with a white exclamation mark inside, indicating a validation error. The entire form is enclosed in a thin yellow border.

Note that for this example we've made use of another HTML5 attribute placeholder which lets us display a prompt or instructions inside the field.



# HTML Validation

## 3. Pattern

we can improve on this by making use of the `pattern` attribute which accepts a JavaScript regular expression. So the code above becomes:

Website: `<input type="url" name="website" required pattern="https?://.+">`

Now our input box will only accept text starting with `http://` or `https://` and at least one additional character:

Website:



starting with http

# Styling forms in CSS

Use the width property to determine the width of the input field.

Use the padding property to add space inside the text field.

When you have many inputs after each other, you might also want to add some margin, to add more space outside of them.

```
input[type=text] {  
  width: 100%;  
  padding: 12px 20px;  
  margin: 8px 0;  
  box-sizing: border-box;  
}
```

First Name

Last Name

# Styling forms in CSS

Use the border property to change the border size and color, and use the border-radius property to add rounded corners.

```
input[type=text] {  
  border: 2px solid red;  
  border-radius: 4px;  
}
```

First Name

If you only want a bottom border, use the border-bottom property.

First Name

```
input[type=text] {  
  border: none;  
  border-bottom: 2px solid red;  
}
```

# Styling forms in CSS

Use the background-color property to add a background color to the input, and the color property to change the text color.

```
input[type=text] {  
  background-color: #3CBC8D;  
  color: white;  
}
```



John

If you want an icon inside the input, use the background-image property and position it with the background-position property. Also notice that we add a large left padding to reserve the space of the icon.

```
input[type=text] {  
  background-color: white;  
  background-image: url('searchicon.png');  
  background-position: 10px 10px;  
  background-repeat: no-repeat;  
  padding-left: 40px;  
}
```



Search..

# Styling forms in CSS

Styling input buttons.

```
input[type=button], input[type=submit], input[type=reset] {  
  background-color: #4CAF50;  
  border: none;  
  color: white;  
  padding: 16px 32px;  
  text-decoration: none;  
  margin: 4px 2px;  
  cursor: pointer;  
}
```

