



# Forms

- ❑ Traditionally, the term 'form' has referred to a printed document that contains spaces for you to fill in information.
- ❑ HTML **borrow**s the concept of a form to refer to different elements that **allow you to collect information from visitors to your site.**



# The star of all forms



Google Search

I'm Feeling Lucky

## □ Common forms

- registering as a member
- Shopping online
- signing up for newsletters or mailing lists
- ...and many many more

# Form Controls

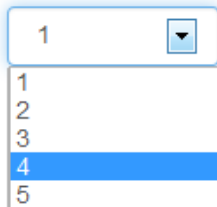
# Form Controls

## □ Adding Text

- Text input (single-line)
- Password input
- Text area (multi-line)

## □ Making Choices

- Radio buttons
- Check boxes
- Drop-down boxes



Email

elgi@ait.gr|

Password

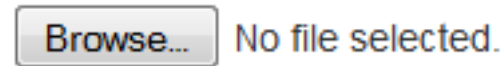
••••••••

My comments...|

- ☒ Option one is this and that—be sure to include why it's great
- ☐ Option two can be something else and selecting it will deselect option one
- ☒ Option one is this and that—be sure to include why it's great

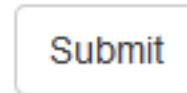
# Form Controls (cont..)

## □ Uploading files



## □ Submitting Forms

- Buttons
- Image Buttons



# **How forms Work**



# Game of thrones

Who would you like to see on the Iron Throne?

Answers

- ☐ Tyrion Lannister
- ☐ Daenerys Targaryen
- ☐ Jon Snow
- ☐ Arya Stark
- ☒ Khal Drogo X No he is dead...

Vote

Show Results

2

The name of each form control is sent to the server along with the value the user enters or selects.

3

The server processes the information using a programming language such as PHP, C#, VB.net, or Java. It may also store the information in a database.

4

The server creates a new page to send back to the browser based on the information received.

5

Thank you, your vote is registered but you will have to wait for the writer to end the books :)



# Important!

- ❑ A form may have several form controls, each gathering different information.
- ❑ The server needs to know which piece of inputted data corresponds with which form element.

Form field name



vote=John Snow

Value



# Form structure- <form>

- ❑ Form controls live inside a <form> element.
- ❑ **Important:** This element **should always carry** the **action** attribute and will usually have a **method** and **id** attribute.

```
<form action="http://www.example.com/subscribe.php"  
      method="get">  
  <p>This is where the form controls will appear.  
  </p>  
</form>
```

# Form attributes-action

---

- ❑ Every **<form>** element requires an **action** attribute.
- ❑ Its **value** is the **URL for the page on the server** that will **receive** the **information** in the form when it is submitted.

# Form attributes-method

- ❑ Forms can be sent using one of two methods: **get** or **post**.
- ❑ **get** method: the values from the form are added to the end of the URL specified in the action attribute
  - short forms (such as search boxes)
  - when you are just retrieving data from the web server (not sending information that should be added to or deleted from a database)
- ❑ **post** method: the values are sent in what are known as HTTP headers. You should use this type if your form:
  - allows users to upload a file
  - is very long
  - contains sensitive data (e.g. passwords)
  - adds information to, or deletes information from, a database

**Note:** If the method attribute is not used, the form data will be sent using the **get** method.

```
<form action="http://www.example.com/subscribe.php" method="get">
```

.....

```
</form>
```

# Form attributes-id

---

- **Id:** the value is used to identify the form distinctly from other elements on the page
- often used by scripts — such as those that check you have entered information into fields that require values)

# Form Input Types

# <input>

## <input>

- The <input> element is used to create several different form controls. The value of the type attribute determines what kind of input they will be creating.

÷ Text

÷ Password

÷ Radio

÷ Checkbox

÷ File

÷ Image

÷ .....



# <input type="text">

- ❑ When the type attribute has a value of text, it creates a singleline text input.
- ❑ **name**
  - When users enter information into a form, the server needs to know which form control each piece of data was entered into.
    - ÷ For example, in a login form, the server needs to know what has been entered as the username and what has been given as the password.
    - ÷ Thus, each **form control requires a name attribute**.
    - ÷ The value of this attribute identifies the form control and is sent along with the information they enter to the server.
- ❑ **maxlength**
  - Used to limit the number of characters a user may enter into the text field.
  - Its value is the number of characters they may enter.
- ❑ **size**
  - The size attribute specifies the visible width, in characters, of an <input> element.

```
<form action="http://www.example.com/login.php">  
  <p>Username:  
    <input type="text" name="username" size="15"  
      maxlength="30" />  
  </p>  
</form>
```

Username:

# <input type="password">

---

□ it creates a text box that acts just like a single-line text input, except the characters are blocked out.

## □ **name**

- The name attribute indicates the name of the password input, which is sent to the server with the password the user enters.

## □ **size, maxlength**

- It can also carry the size and maxlength attributes like the single-line text input.

```
<form action="http://www.example.com/login.php">
  <p>Username:
    <input type="text" name="username" size="15"
      maxlength="30" />
  </p>
  <p>Password:
    <input type="password" name="password" size="15"
      maxlength="30" />
  </p>
</form>
```

Username:

Password:

# <textarea>

---

## □ <textarea>

- is used to create a mutli-line text input.
- Unlike other input elements **it is not an empty** element.
  - ÷ It should have an opening and a closing tag.
  - ÷ Any text that appears between the opening <textarea> and closing </textarea> tags will appear in the text box when the page loads.

```
<!DOCTYPE html>
<html>
<body>

<textarea rows="4" cols="50">
  Please enter your comments...
</textarea>

</body>
</html>
```

Please enter your comments...

# Radio button-`<input type="radio"`

☐ Radio buttons allow users to pick just one of a number of options.

## ☐ **name**

- The name attribute is sent to the server with the value of the option the user selects.
- the value of the **name attribute** should be **the same for all of the radio buttons**

## ☐ **value**

- The value attribute indicates the value that is sent to the server for the selected option.
- The **value of each of the buttons** in a group **should be different**

## ☐ **checked**

- The checked attribute can be used to indicate which value (if any) should be selected when the page loads.

```
<!DOCTYPE html>
<html>
<body>
  <input type="radio" name="optionsRadios" value="option1" checked>
  Option one
  <input type="radio" name="optionsRadios" value="option2">
  Option two
</body>
</html>
```

☒ Option one ☐ Option two



# Checkbox-`<input type="checkbox">`

☐ Checkboxes allow users to select (and unselect) one or more options in answer to a question.

## ☐ **name**

- The **name** attribute is sent to the server with the value of the option(s) the user selects.
- When a question provides users with options for answers in the form of checkboxes, the value of the **name** attribute should be the same for all of the buttons that answer that question.

## ☐ **value**

- The **value** attribute indicates the value sent to the server if this checkbox is checked.

## ☐ **checked**

- The **checked** attribute indicates that this box should be checked when the page loads.

```
<!DOCTYPE html>
<html>
<body>

<form action="">
<input type="checkbox" name="options" value="one" checked>One<br>
<input type="checkbox" name="options" value="two">Two <br>
<input type="checkbox" name="options" value="three">Three<br>
<input type="checkbox" name="options" value="four">Four<br>
<input type="checkbox" name="options" value="five">Five<br>
</form>

</body>
</html>
```

- ☒ One
- ☐ Two
- ☐ Three
- ☐ Four
- ☐ Five

# Drop-down list box

---

## □ **<select>**

- A drop down list box allows users to select one option from a drop down list.
- It contains two or more `<option>` elements.

## □ **name**

- indicates the name of the form control being sent to the server, along with the value the user selected.

## □ **<option>**

- is used to specify the options that the user can select from.
- The words between the opening `<option>` and closing `</option>` tags will be shown to the user in the drop down box.

## □ **value**

- The **<option> element** uses the **value attribute** to indicate the value that is sent to the server along with the name of the control if this option is selected.

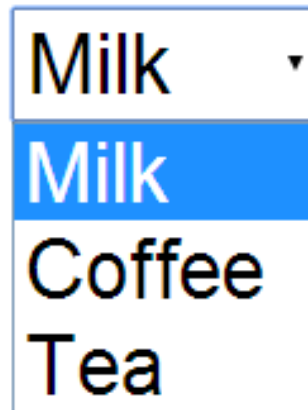
# Drop-down list box (cont..)

## ☐ selected

- can be used to indicate the option that should be selected when the page loads.
  - If this attribute is not used, the first option will be shown when the page loads. If the user does not select an option, then the first item will be sent to the server as the value for this control.
- ☐ Similar to radio buttons (in that only one option can be selected).

```
<!DOCTYPE html>
<html>
<body>
<select>
  <option>Milk</option>
  <option>Coffee</option>
  <option>Tea</option>
</select>

</body>
</html>
```



Milk ▾

Milk

Coffee

Tea

# Radio buttons or drop-downs?

---

- ❑ If users need to see **all options at a glance**
  - **radio buttons** are better suited
- ❑ If there is a **very long list** of options (such as a list of countries),
  - **drop down list** boxes work better

# Multiple Select Box-<select>

---

## ☐ size

- You can turn a drop down select box into a box that shows more than one option by adding the size attribute.
- Its value should be the number of options you want to show at once.
- Unfortunately, the way that browsers have implemented this attribute is not perfect, and it should be tested thoroughly if used (in particular in Firefox and Safari on a Mac).

## ☐ multiple

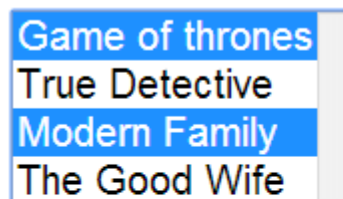
- You can allow users to select multiple options from this list by adding the multiple attribute with a value of multiple.
- It is a good idea to tell users if they can select more than one option at a time. It is also helpful to indicate that on a PC they should hold down the control key while selecting multiple options and on a Mac they should use the command key while selecting options.

```
<!DOCTYPE html>
<html>
<body>

<form action="form_action.php">
<select name="series" multiple>
  <option value="game">Game of thrones</option>
  <option value="true">True Detective</option>
  <option value="modern">Modern Family</option>
  <option value="good">The Good Wife</option>
</select>
<input type="submit" value="submit">
</form>
```

<p>Hold down the Ctrl (windows) / Command (Mac) button to select multiple options.</p>

```
</body>
</html>
```



submit



Hold down the Ctrl (windows) / Command (Mac) button to select multiple options.



# File input - `<input type="file">`

---

- ❑ Allows users to upload a file
- ❑ It creates a box that looks like a text input followed by a **browse** button.

When you are allowing users to upload files, the **method** attribute on the **<form>** element must have a value of **post**. (You cannot send files using the HTTP **get** method.)

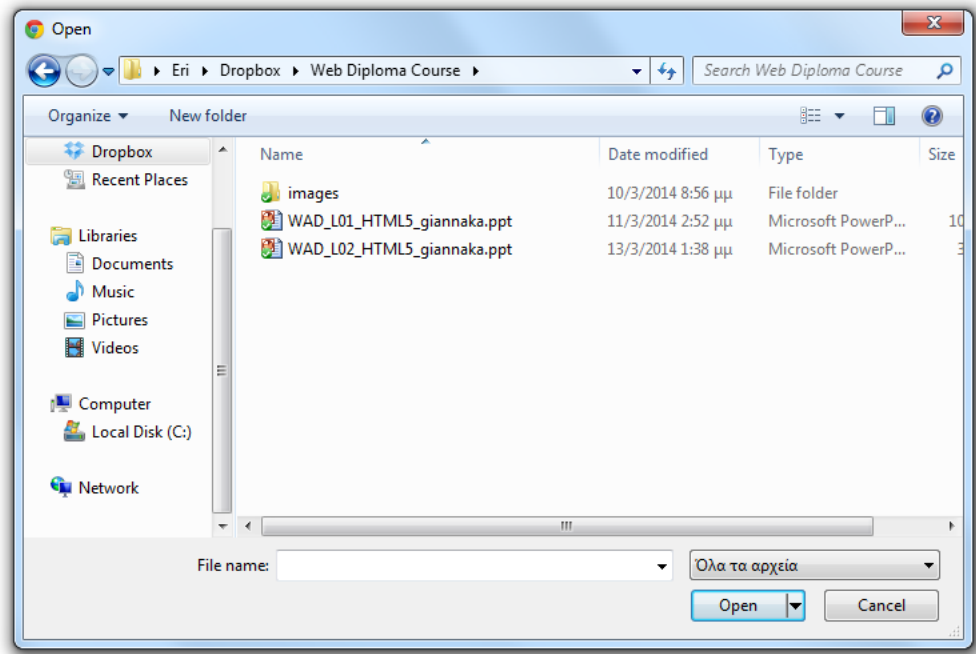
```
<!DOCTYPE html>
<html>
<body>
<form action="parse_file.php" method="post">
  <p>Browse for a file to upload: </p>
  <input type="file" name="uploaded_file">
  <br /><br />
  <input type="submit" value="Upload It Now">
</form>

</body>
</html>
```

Browse for a file to upload:

Επιλογή αρχείου Δεν έχει επιλεγεί κανένα αρχείο

Upload It Now



# Submit Button-`<input type="submit">`

---

❑ It is used to send a form to the server.

## ❑ **name**

❑ It can use a **name** attribute but it does not need to have one.

## ❑ **value**

❑ The **value** attribute is used to control the text that appears on a button.

- It is a good idea to specify the words you want to appear on a button because the default value of buttons on some browsers is 'Submit query' and this might not be appropriate for all kinds of form.

# Image button-`<input type="image">`

---

- ❑ If you want to use an image for the submit button, you can give the type attribute a value of image.
- ❑ The src, width, height, and alt attributes work just like they do when used with the `<img>` element

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>
    <form name="myform" action="http://www.mydomain.com/myformhandler.php" method="POST">

      <br>
      <input type="text" size="25" value="name">
      <br>
      <input type="image" src="media/click.png" name="image" width="60" height="60"><br>

    </form>
  </body>
</html>
```

name



# Button `<button>` & Hidden Controls

## □ `<button>`

- The `<button>` element was introduced to allow users more control over how their buttons appear, and to allow other elements to appear inside the button.

□ This means that you can combine text and images between the opening `<button>` tag and closing `</button>` tag.

## □ `<input type="hidden">`

- These form controls are not shown on the page (although you can see them if you use the **View Source** option in the browser). They allow web page authors to add values to forms that users cannot see.

```
<!DOCTYPE html>  
<html>  
<body>  
  
<button type="button">I am a button. Click Me!</button>  
  
</body>  
</html>
```

I am a button. Click Me!

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>
    <form name="myform" action="http://www.mydomain.com/myformhandler.php" method="POST">
      <div align="center">
        <input type="text" size="25" value="Enter your name here!">
        <input type="hidden" name="Language" value="English">
        <br><br>
      </div>
    </form>
  </body>
</html>
```

Enter your name here!



# Form Labels - `<label>`

- ❑ Indicate the purpose of each form field next to it.
  - each form control should have its own **`<label>`** element as this makes the form accessible to vision-impaired users.
- ❑ The **`<label>`** element can be used in two ways.
  - Wrap around both the text description and the form input (as shown on the first line of the example to your right).
  - Be kept separate from the form control and use the **`for`** attribute to indicate which form control it is a label for (as shown with the radio buttons).
- ❑ **`for`**
  - The **`for`** attribute states which form control the label belongs to.
    - ÷ Note how the radio buttons use the **`id`** attribute. The value of the **`id`** attribute uniquely identifies an element from all other elements on a page. The value of the **`for`** attribute matches that of the **`id`** attribute on the form control it is labelling.

```
<!DOCTYPE html>
<html>
<body>

<p>Click on one of the text labels to toggle the related control:</p>

<form action="demo_form.asp">
  <label for="male">Male</label>
  <input type="radio" name="sex" id="male" value="male"><br>
  <label for="female">Female</label>
  <input type="radio" name="sex" id="female" value="female"><br><br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

Click on one of the text labels to toggle the related control:

Male ☐

Female ☐

Submit

# Grouping Form Elements-**<fieldset>**

---

- You can group related form controls together inside the **<fieldset>** element. This is particularly helpful for longer forms.
- **<legend>**
  - The **<legend>** element can come directly after the opening **<fieldset>** tag and contains a caption which helps identify the purpose of that group of form controls.

```
<!DOCTYPE html>
<html>
<body>

<form>
  <fieldset>
    <legend>Personal Info:</legend>
    Name: <input type="text" name="name"><br>
    Surname: <input type="text" name="surname"><br>
    Hometown: <input type="text" name="home">
  </fieldset>
</form>

</body>
</html>
```

Name:

Surname:

Hometown:

Personal Info:

Name:

Surname:

Hometown:

**WHAT HAS HTML5  
BROUGHT TO  
FORMS?**



# Validation

- ❑ **ONE OF THE** problems with HTML 4 forms is that they're just dumb fields.
- ❑ Validation is required on the server, of course, but you have to duplicate it in the user's browser with JavaScript to give them the seamless experience they deserve.
- ❑ **Wouldn't it be great if browsers had built-in validation for some of the most common data types that we collect?**

**HTML5 forms provide exactly that.**

# Validation (cont..)

- ❑ The new HTML 5 `input` types are *self validating* on the client side
  - eliminates the need to add complicated JavaScript
  - reduces the amount of invalid data submitted
    - ÷ Reduces Internet traffic between the server and the client to correct invalid input.
- ❑ ***The server should still validate all user input.***
- ❑ When a user enters data into a form then submits the form the browser immediately checks the self-validating elements to ensure that the data is correct

# NEW HTML5 FORM ELEMENTS





# New form field types

---

- ❑ Many forms need to gather information such as dates, email addresses, and URLs.
  - Traditionally been done using **text inputs**.
- ❑ HTML5 introduces **new form controls** to standardize the way that some information is gathered.
  - **Note:** Older browsers that do not recognize these inputs will just treat them as a single line text box.

# Date input-`<input type="date">`

- This will create a date input in browsers that support the new HTML5 input types.



```
<!DOCTYPE html>
<html>
<body>

<form action="demo_form.asp">
  Birthday: <input type="date" name="bday">
  <input type="submit">
</form>

</body>
</html>
```

Birthday:

Μάρτιος 2014

| Δευ | Τρι | Τετ | Πεμ | Παρ | Σαβ | Κυρ |
|-----|-----|-----|-----|-----|-----|-----|
| 24  | 25  | 26  | 27  | 28  | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |
| 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| 24  | 25  | 26  | 27  | 28  | 29  | 30  |
| 31  | 1   | 2   | 3   | 4   | 5   | 6   |

# Date input-`<input type="datetime"`

- ❑ The `datetime` input type enables the user to enter a date (year, month, day), time (hour, minute, second, fraction of a second) and the time zone set to UTC (Coordinated Universal Time or Universal Time, Coordinated).
- ❑ Currently, most of the browsers render `datetime` as a text field;
  - Chrome renders an up-down control
  - Opera renders a date and time control.



```
<!DOCTYPE html>
<html>
<body>

<form action="demo_form.asp">
  Birthday (date and time): <input type="datetime" name="bdaytime">
  <input type="submit">
</form>

</body>
</html>
```

Birthday (date and time):

# <input type="time"

- ❑ The **time input type** enables the user to enter an hour, minute, seconds and fraction of second.
- ❑ The HTML5 specification indicates that a time must have two digits representing the hour, followed by a colon (:) and two digits representing the minute.
  - Optionally, you can also include a colon followed by two digits representing the seconds and a period followed by one or more digits representing a fraction of a second



```
<!DOCTYPE html>
<html>
<body>

<form action="demo_form.asp">
  Select a time: <input type="time" name="usr_time">
  <input type="submit">
</form>

</body>
</html>
```

Select a time:

# <input type="month"

- ❑ The **month input type** enables the user to enter a year and month in the format **yyyy-mm**, such as **2012-01**.
- ❑ If the user enters the data in an improper format (e.g., **January 2012**) and submits the form, a callout stating that an invalid value was entered appears.





```
<!DOCTYPE html>
<html>
<body>

<form action="demo_form.php">
  Birthday (month and year): <input type="month" name="bdaymonth">
  <input type="submit">
</form>

</body>
</html>
```

Birthday (month and year):

-----

----

Μάρτιος 2014 ▾

| Δευ | Τρι | Τετ | Πεμ | Παρ | Σαβ | Κυρ |
|-----|-----|-----|-----|-----|-----|-----|
| 24  | 25  | 26  | 27  | 28  | 1   | 2   |
| 3   | 4   | 5   | 6   | 7   | 8   | 9   |
| 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| 24  | 25  | 26  | 27  | 28  | 29  | 30  |
| 31  | 1   | 2   | 3   | 4   | 5   | 6   |

Υποβολή

# <input type="week"

- ❑ The **week input type** enables the user to select a year and week number in the format `yyyy-wnn`, where `nn` is 01–53—for example, `2012-w01` represents the first week of 2012. Internet Explorer, Firefox and Safari render a text field.
  - Chrome renders an up-down control.
  - Opera renders *week control* with a down arrow that, when clicked, brings up a calendar for the current month with the corresponding week numbers listed down the left side.



```

<!DOCTYPE html>
<html>
<body>

<form action="demo_form.asp">
  Select a week: <input type="week" name="year_week">
  <input type="submit">
</form>

</body>
</html>

```

Select a week: Εβδομάδα 12, 2014

Μάρτιος 2014

| Εβδομάδα | Δευ | Τρι | Τετ | Πεμ | Παρ | Σαβ | Κυρ |
|----------|-----|-----|-----|-----|-----|-----|-----|
| 9        | 24  | 25  | 26  | 27  | 28  | 1   | 2   |
| 10       | 3   | 4   | 5   | 6   | 7   | 8   | 9   |
| 11       | 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 12       | 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| 13       | 24  | 25  | 26  | 27  | 28  | 29  | 30  |
| 14       | 31  | 1   | 2   | 3   | 4   | 5   | 6   |

# <input type="email">

- Browsers that support HTML5 validation will check that the user has provided information in the correct format of an email address.
- Some smart phones also optimize their keyboard to display the keys you are most likely to need when entering an email address (such as the @ symbol).



```
<!DOCTYPE html>
<html>
<body>

<form action="demo_form.asp">
  E-mail: <input type="email" name="email">
  <input type="submit">
</form>

</body>
</html>
```

E-mail:

Υποβολή

**Note:** type="email"

! Συμπεριλάβετε το σύμβολο "@" στη διεύθυνση ηλεκτρονικού ταχυδρομείου. Από τη διεύθυνση "elgiaitgr" λείπει το σύμβολο "@".

Internet Explorer 9 and earlier versions.

# URL input-`<input type="url"`

- A URL input can be used when you are asking a user for a web page address.
- Browsers that support HTML5 validation will check that the user has provided information in the format of a URL.
  - Some smart phones also optimize their keyboard to display the keys you are most likely to need when entering a URL.



```
<!DOCTYPE html>
<html>
<body>

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

<p><b>Note:</b> The type="url" is not supported in Internet Explorer 9 and
earlier versions.</p>

</body>
</html>
```

Add your homepage:

Υποβολή

! Εισαγάγετε μια διεύθυνση URL.

**Note:** The type="url" is not supported in Internet Explorer 9 and earlier versions.

# Search input-`<input type="search">`

□ If you want to create a single line text box for search queries, HTML5 provides a special search input.

## □ **placeholder**

- On any text input, you can also use an attribute called **placeholder** whose value is text that will be shown in the text box until the user clicks in that area. Older browsers simply ignore this attribute.





```
<!DOCTYPE html>
<html>
<body>
<form>
  Search Wad: <input type="search" name="wadsearch"><br>
  Search Wad with Placeholder:
  <input type="search" name="wadsearch" placeholder="Search Wad"><br>
  <input type="submit">
</form>
</body>
</html>
```

Search Wad:

Search Wad with Placeholder:

Υποβολή

# <input type="color">

- ❑ The color input type enables the user to enter a color.
- ❑ At the time of this writing, most browsers render the color input type as a text field in which the user can enter a hexadecimal code or a color name.
- ❑ In the future, when you click a color input, browsers will likely display a color picker similar to the Microsoft Windows color dialog shown

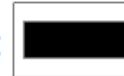


```
<!DOCTYPE html>
<html>
<body>

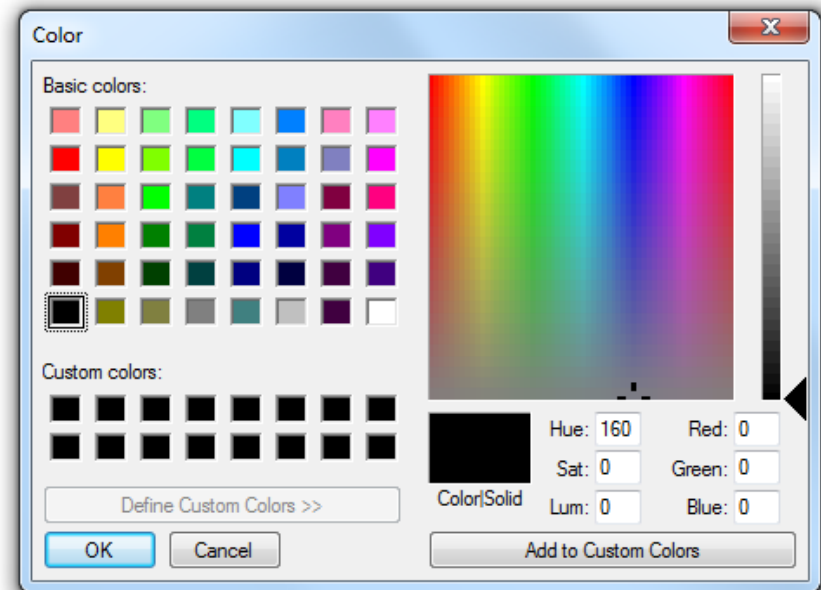
<form>
  What color are you waering today?:
  <input type="color" name="favcolor"><br>
  <input type="submit">
</form>

</body>
</html>
```

What color are you waering today?:



Υποβολή



# <input type="number">

- The **number input type** enables the user to enter a numerical value—mobile browsers typically display a numeric keypad for this **input type**.
- Internet Explorer, Firefox and Safari display a text field in which the user can enter a number.
  - Chrome and Opera render a spinner control for adjusting the number.



# <input type="number"

- ❑ The **min** attribute sets the minimum valid number.
- ❑ The **max** attribute sets the maximum valid number.
- ❑ The **step** attribute determines the increment in which the numbers increase.
- ❑ The **value** attribute sets the initial value displayed in the form
- ❑ If the user attempts to enter an invalid value by typing in the text field, a callout pointing to the **number input** element will instruct the user to enter a valid value.

```
<!DOCTYPE html>
<html>
<body>

<form>
  Quantity (between 1 and 5):
  <input type="number" name="quantity" min="1" max="5" step="2">
  <input type="submit">
</form>

<p><b>Note:</b> type="number" is not supported in Internet Explorer 9 and
earlier versions.</p>

</body>
</html>
```

Quantity (between 1 and 5):

**Note:** type="number" is not supported in Internet Explorer 9 and earlier versions.

! Η τιμή πρέπει να είναι μικρότερη ή ίση του 5.

# <input type="range">

- ❑ The **range input type** appears as a *slider* control in
- ❑ You can set the minimum and maximum and specify a value.
- ❑ The **range input type** is *inherently self-validating* when it is rendered by the browser as a slider control, because *the user is unable to move the slider outside the bounds of the minimum or maximum value.*



```
<!DOCTYPE html>
<html>
<body>

<form action="demo_form.asp" method="get">
Points: 0<input type="range" name="points" min="1" max="10">10
<input type="submit">
</form>

<p><b>Note:</b> type="range" is not supported in Internet Explorer 9 and
earlier versions.</p>

</body>
</html>
```

Points: 0  10

**Note:** type="range" is not supported in Internet Explorer 9 and earlier versions.



# <input type="tel"

- ❑ The tel input type enables the user to enter a telephone number—mobile browsers typically display a keypad specific to entering phone numbers for this input type.
- ❑ At this time, the tel input type is rendered as a text field in all of the browsers.
- ❑ **HTML5 does not self validate the tel input type.**



# datalist Element

---

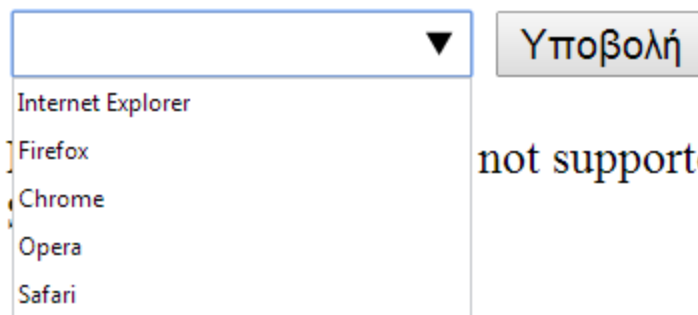
- ❑ Provides input options for a `text` input element.
- ❑ One of the most used JavaScript widgets over has been the text box autocomplete widget.
- ❑ In HTML5 it is replaced via the new `DATALIST` element.

```
<!DOCTYPE html>
<html>
<body>

<form action="demo_form.php" method="get">
<input list="browsers" name="browser">

<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
<input type="submit">
</form>

</body>
</html>
```



Internet Explorer  
Firefox  
Chrome  
Opera  
Safari

Υποβολή

not supported in Internet Explorer 9 and earlier versions, or in

# NEW FORM ATTRIBUTES

# autocomplete Attribute

- ❑ The **autocomplete attribute** can be used on **input** types to automatically fill in the user's information based on previous input—such as name, address or e-mail.
- ❑ You can enable **autocomplete** for an entire form or just for specific elements.
- ❑ For example, an online order form might set **autocomplete = "on"** for the name and address **inputs** and set **autocomplete = "off"** for the credit card and password **inputs** for security purposes.



```
<!DOCTYPE html>
<html>
<body>

<form action="demo_form.php" autocomplete="on">
  First name:<input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  E-mail: <input type="email" name="email" autocomplete="off"><br>
  <input type="submit">
</form>

</body>
</html>
```

# Validation

- ❑ If you want to bypass validation, you can add the **novalidate** attribute

```
<!DOCTYPE html>
<html>
<body>

<form action="demo_form.asp" novalidate>
E-mail: <input type="email" name="user_email">
<input type="submit">
</form>

</body>
</html>
```



# NEW ATTRIBUTES FOR <INPUT>



# *autofocus* Attribute

- The **autofocus attribute**—an optional attribute that can be used in only one `input` element on a form—automatically gives the focus to the `input` element, allowing the user to begin typing in that element immediately.



```
<!DOCTYPE html>
<html>
<body>

<form action="demo_form.asp">
  First name: <input type="text" name="fname" autofocus><br>
  Last name: <input type="text" name="lname"><br>
  <input type="submit">
</form>

</body>
</html>
```

First name:

Last name:

Υποβολή

# autocomplete Attribute

- ❑ The **autocomplete attribute** can be used on **input** types to automatically fill in the user's information based on previous input—such as name, address or e-mail.
- ❑ You can enable **autocomplete** for an entire form or just for specific elements.
- ❑ For example, an online order form might set **autocomplete = "on"** for the name and address **inputs** and set **autocomplete = "off"** for the credit card and password **inputs** for security purposes.



```
<!DOCTYPE html>
<html>
<body>

<form action="demo_form.asp" autocomplete="on">
  First name:<input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  E-mail: <input type="email" name="email" autocomplete="off"><br>
  <input type="submit">
</form>

</body>
</html>
```

First name:

Last name:

E-mail:

# The required attribute

- ❑ The new **required** attribute can be used on `<textarea>` and most input fields (except when the type attribute is hidden, image, or some button types such as submit).
- ❑ The browser will not allow the user to submit the form if required fields are empty.



```
<!DOCTYPE html>
<html>
<body>

<form action="demo_form.asp">
  Username: <input type="text" name="usrname" required>
  Name: <input type="text" name="name">
  <input type="submit">
</form>

</body>
</html>
```

Username:

Name:

Υποβολή



Συμπληρώστε αυτό το πεδίο.

# <input> form Attribute

- The form attribute specifies one or more forms an <input> element belongs to.

```
<!DOCTYPE html>
<html>
<body>
```

```
<form action="demo_form.asp" id="form1">
First name: <input type="text" name="fname"><br>
<input type="submit" value="Submit">
</form>
```

```
<p>The "Last name" field below is outside the form element, but still part
of the form.</p>
```

```
Last name: <input type="text" name="lname" form="form1">
```

```
</body>
</html>
```

First name:

The "Last name" field below is outside the form element, but still part of the form.

Last name:



# <input> formaction Attribute

- ❑ The formaction attribute specifies the URL of a file that will process the input control when the form is submitted.
- ❑ The **formaction** attribute overrides the **action** attribute of the <form> element.





```
<!DOCTYPE html>
<html>
<body>

<form action="demo_form.php">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="submit" value="Submit"><br>
  <input type="submit" formaction="demo_admin.asp" value="Submit as admin">
</form>

</body>
</html>
```

First name:

Last name:

# <input> formmethod Attribute

---

- ❑ The **formmethod** attribute defines the HTTP method for sending form-data to the action URL.
- ❑ The **formmethod** attribute overrides the **method** attribute of the <form> element.



```
<!DOCTYPE html>
<html>
<body>

<form action="demo_form.asp" method="get">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="submit" value="Submit">
  <input type="submit" formmethod="post" formaction="demo_post.asp"
value="Submit using POST">
</form>

</body>
</html>
```

First name:

Last name:

Submit

Submit using POST

# <input> formnovalidate attribute

---

- If you want to bypass validation, you can add the **form novalidate attribute** to input type submit :

```
<input type = "submit" value = "Submit" novalidate />
```



# <input> formtarget Attribute

---

- ❑ The **formtarget** attribute specifies a name or a keyword that indicates where to display the response that is received after submitting the form.

```
<!DOCTYPE html>
<html>
<body>

<form action="demo_form.php">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="submit" value="Submit as normal">
  <input type="submit" formtarget="_blank" value="Submit to a new
window/tab">
</form>

</body>
</html>
```

First name:

Last name:

Submit as normal

Submit to a new window/tab

# Other attributes

---

- ❑ `<input>` required Attribute
- ❑ `<input>` placeholder Attribute
- ❑ `<input>` multiple Attribute
- ❑ `<input>` min and max Attributes

# Summing it up

---

- ❑ Whenever you want to collect information from visitors you will need a form, which lives inside a **<form>** element.
- ❑ Information from a form is sent in name/value pairs.
- ❑ Each form control is given a name, and the text the user types in or the values of the options they select are sent to the server.
- ❑ HTML5 introduces new form elements which make it easier for visitors to fill in forms.



Questions?



# Sources

## Introducing HTML5

Bruce Lawson, Remy Sharp

New Riders Publishing, 2010



## HTML and CSS: Design and Build Websites

Jon Duckett

ISBN: 978-1-118-00818-8



## □ Additional Sources

- W3Schools: <http://www.w3schools.com/>
- Mozilla Developer Network: <https://developer.mozilla.org>

**Lets get our hands dirty**

