

## 5. HTML5 – Web Forms 2.0

Web Forms 2.0 is an extension to the forms features found in HTML4. Form elements and attributes in HTML5 provide a greater degree of semantic mark-up than HTML4 and free us from a great deal of tedious scripting and styling that was required in HTML4.

### The `<input>` element in HTML4

HTML4 input elements use the **type** attribute to specify the data type. HTML4 provides following types –

Type	Description
text	A free-form text field, nominally free of line breaks.
password	A free-form text field for sensitive information, nominally free of line breaks.
checkbox	A set of zero or more values from a predefined list.
radio	An enumerated value.
submit	A free form of button initiates form submission.
file	An arbitrary file with a MIME type and optionally a file name.
image	A coordinate, relative to a particular image's size, with the extra semantic that it must be the last value selected and initiates form submission.
hidden	An arbitrary string that is not normally displayed to the user.
select	An enumerated value, much like the radio type.
textarea	A free-form text field, nominally with no line break restrictions.
button	A free form of button which can initiates any event related to button.

Following is the simple example of using labels, radio buttons, and submit buttons –

```
...  
<form action="http://example.com/cgi-script.pl" method="post">  
  
  <p>  
  
    <label for="firstname">first name: </label>  
    <input type="text" id="firstname"><br />  
  
    <label for="lastname">last name: </label>
```

```

<input type="text" id="lastname"><br />

<label for="email">email: </label>
<input type="text" id="email"><br>

<input type="radio" name="sex" value="male"> Male<br>
<input type="radio" name="sex" value="female"> Female<br>
<input type="submit" value="send"> <input type="reset">

</p>
</form>

...

```

## The <input> element in HTML5

Apart from the above-mentioned attributes, HTML5 input elements introduced several new values for the **type** attribute. These are listed below.

**NOTE:** Try all the following example using latest version of **Opera** browser.

Type	Description
<b>datetime</b>	A date and time (year, month, day, hour, minute, second, fractions of a second) encoded according to ISO 8601 with the time zone set to UTC.
<b>datetime-local</b>	A date and time (year, month, day, hour, minute, second, fractions of a second) encoded according to ISO 8601, with no time zone information.
<b>date</b>	A date (year, month, day) encoded according to ISO 8601.
<b>month</b>	A date consisting of a year and a month encoded according to ISO 8601.
<b>week</b>	A date consisting of a year and a week number encoded according to ISO 8601.
<b>time</b>	A time (hour, minute, seconds, fractional seconds) encoded according to ISO 8601.
<b>number</b>	It accepts only numerical value. The step attribute specifies the precision, defaulting to 1.
<b>range</b>	The range type is used for input fields that should contain a value from a range of numbers.
<b>email</b>	It accepts only email value. This type is used for input fields that should contain an e-mail address. If you try to submit a simple text, it forces to enter only email address in email@example.com format.
<b>url</b>	It accepts only URL value. This type is used for input fields that should contain a URL address. If you try to submit a simple text, it forces to enter

	only URL address either in http://www.example.com format or in http://example.com format.
--	---

## HTML5 - datetime

A date and time (year, month, day, hour, minute, second, fractions of a second) encoded according to ISO 8601 with the time zone set to UTC.

### Example

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

    <form action = "/cgi-bin/html5.cgi" method = "get">
      Date and Time : <input type = "datetime" name = "newinput" />
      <input type = "submit" value = "submit" />
    </form>

  </body>
</html>
```

### Output

Date and Time :

## HTML5 - datetime local

---

A date and time (year, month, day, hour, minute, second, fractions of a second) encoded according to ISO 8601, with no time zone information.

### Example

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

    <form action = "/cgi-bin/html5.cgi" method = "get">
      Local Date and Time : <input type = "datetime-local" name = "newinput" />
      <input type = "submit" value = "submit" />
    </form>

  </body>
</html>
```

### Output

Local Date and Time :

## HTML5 – date

---

A date (year, month, day) encoded according to ISO 8601.

### Example

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

    <form action = "/cgi-bin/html5.cgi" method = "get">
```

```

        Date : <input type = "date" name = "newinput" />
        <input type = "submit" value = "submit" />
    </form>
</body>
</html>

```

## Output

Date :

## HTML5 – month

A date consisting of a year and a month encoded according to ISO 8601.

### Example

```

<!DOCTYPE HTML>
<html>
    <body>

        <form action = "/cgi-bin/html5.cgi" method = "get">
            Month : <input type = "month" name = "newinput" />
            <input type = "submit" value = "submit" />
        </form>

    </body>
</html>

```

## Output

Month :

## HTML5 - week

A date consisting of a year and a week number encoded according to ISO 8601.

## Example

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

    <form action = "/cgi-bin/html5.cgi" method = "get">
      Week : <input type = "week" name = "newinput" />
      <input type = "submit" value = "submit" />
    </form>

  </body>
</html>
```

## Output

Week :

## HTML5 – time

---

A time (hour, minute, seconds, fractional seconds) encoded according to ISO 8601.

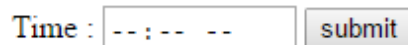
### Example

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

    <form action = "/cgi-bin/html5.cgi" method = "get">
      Time : <input type = "time" name = "newinput" />
      <input type = "submit" value = "submit" />
    </form>

  </body>
</html>
```

### Output



## HTML5 – number

---

It accepts only numerical value. The step attribute specifies the precision, defaulting to 1.

### Example

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

    <form action = "/cgi-bin/html5.cgi" method = "get">
```

```

        Select Number : <input type = "number" min = "0" max = "10" step "1"
                        value = "5" name = "newinput" />
        <input type = "submit" value = "submit" />
    </form>

</body>
</html>

```

## Output

Select Number :

## HTML5 – range

The range type is used for input fields that should contain a value from a range of numbers.

### Example

```

<!DOCTYPE HTML>
<html>
    <body>

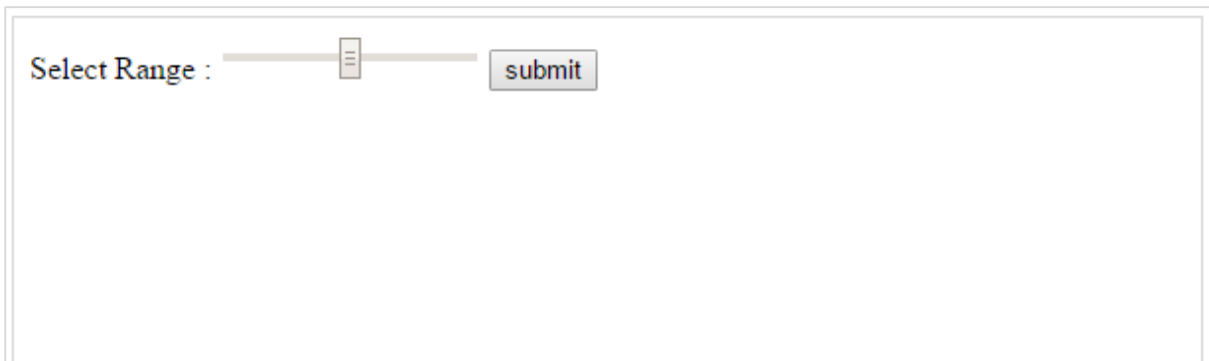
        <form action = "/cgi-bin/html5.cgi" method = "get">
            Select Range : <input type = "range" min = "0" max = "10" step "1"
                          value = "5" name = "newinput" />
            <input type = "submit" value = "submit" />
        </form>

    </body>
</html>

```



## Output



## HTML5 - email

It accepts only email value. This type is used for input fields that should contain an e-mail address. If you try to submit a simple text, it forces to enter only email address in email@example.com format.

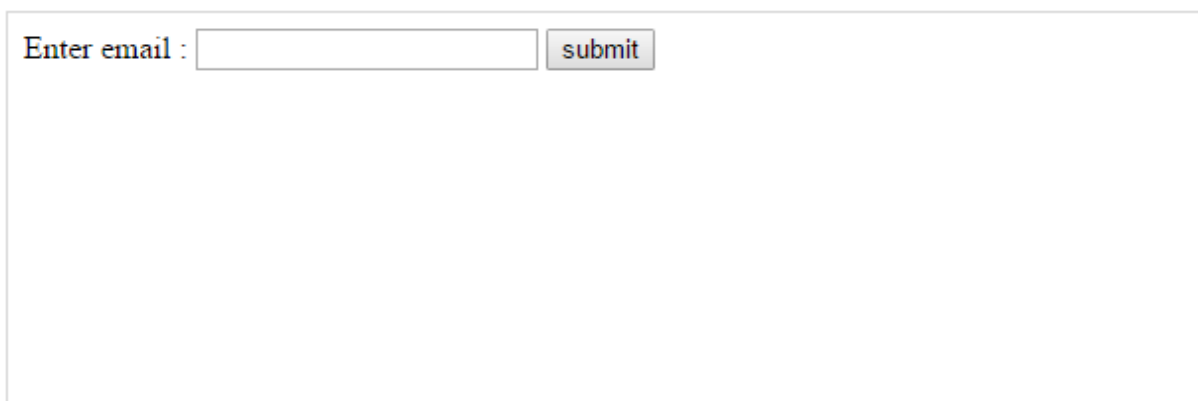
## Example

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

    <form action = "/cgi-bin/html5.cgi" method = "get">
      Enter email : <input type = "email" name = "newinput" />
      <input type = "submit" value = "submit" />
    </form>

  </body>
</html>
```

## Output



## HTML5 – URL

It accepts only URL value. This type is used for input fields that should contain a URL address. If you try to submit a simple text, it forces to enter only URL address either in **http://www.example.com** format or in **http://example.com** format.

### Example

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

    <form action = "/cgi-bin/html5.cgi" method = "get">
      Enter URL : <input type = "url" name = "newinput" />
      <input type = "submit" value = "submit" />
    </form>

  </body>
</html>
```

### Output

Enter URL :

## The <output> element

HTML5 introduced a new element <output> which is used to represent the result of different types of output, such as output written by a script.

You can use the **for** attribute to specify a relationship between the output element and other elements in the document that affected the calculation (for example, as inputs or parameters). The value of the for attribute is a space-separated list of IDs of other elements.

```
<!DOCTYPE HTML>
```

```

<html>

  <script type="text/javascript">
    function showResult()
    {
      x = document.forms["myform"]["newinput"].value;
      document.forms["myform"]["result"].value=x;
    }
  </script>

  <body>

    <form action="/cgi-bin/html5.cgi" method="get" name="myform">
      Enter a value : <input type="text" name="newinput" />
      <input type="button" value="Result" onclick="showResult();" />
      <output name="result"/>
    </form>

  </body>

</html>

```

It will produce the following result –

Enter a value :

## The placeholder attribute

HTML5 introduced a new attribute called **placeholder**. This attribute on `<input>` and `<textarea>` elements provide a hint to the user of what can be entered in the field. The placeholder text must not contain carriage returns or line-feeds.

Here is the simple syntax for placeholder attribute –

```
<input type="text" name="search" placeholder="search the web"/>
```

This attribute is supported by latest versions of Mozilla, Safari and Chrome browsers only.

```
<!DOCTYPE HTML>
<html>

  <body>

    <form action="/cgi-bin/html5.cgi" method="get">
      Enter email : <input type="email" name="newinput"
placeholder="email@example.com"/>
      <input type="submit" value="submit" />
    </form>

  </body>

</html>
```

This will produce the following result –



## The autofocus attribute

This is a simple one-step pattern, easily programmed in JavaScript at the time of document load, automatically focus one particular form field.

HTML5 introduced a new attribute called **autofocus** which would be used as follows –

```
<input type="text" name="search" autofocus/>
```

This attribute is supported by latest versions of Mozilla, Safari and Chrome browsers only.

```

<!DOCTYPE HTML>

<html>

  <body>

    <form action="/cgi-bin/html5.cgi" method="get">
      Enter email : <input type="text" name="newinput" autofocus/>
      <p>Try to submit using Submit button</p>
      <input type="submit" value="submit" />
    </form>

  </body>

</html>

```

## The required attribute

Now you do not need to have JavaScript for client-side validations like empty text box would never be submitted because HTML5 introduced a new attribute called **required** which would be used as follows and would insist to have a value –

```
<input type="text" name="search" required/>
```

This attribute is supported by latest versions of Mozilla, Safari and Chrome browsers only.

```

<!DOCTYPE HTML>
<html>
  <body>
    <form action="/cgi-bin/html5.cgi" method="get">
      Enter email : <input type="text" name="newinput" required/>
      <p>Try to submit using Submit button</p>
      <input type="submit" value="submit" />
    </form>
  </body>
</html>

```

It will produce the following result –

Enter email :

Try to submit using Submit button

submit