**HTML5 Forms**

**Introduction:**

Html Forms are used to collect group the input elements and also used for various purposes like form validation (User Input validation ).

The HTML <form> element defines a form that is used to collect user input:The HTML <form> element defines a form that is used to collect user input:

Example:

<form>

.

*form elements*

.

</form>

**Form Elements**

Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more**.**

## **The <input> Element**

The <input> element is the most important form element.

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

**Various Input Elements**

|  |  |
| --- | --- |
| **Type** | **Description** |
| <input type="text"> | Defines a one-line text input field |
| <input type="radio"> | Defines a radio button (for selecting one of many choices) |
| <input type="submit"> | Defines a submit button (for submitting the form) |

**1.Modernizr**

**What is Modernizr**

It’s a collection of superfast tests – or “detects” as we like to call them – which run as your web page loads, then you can use the results to tailor the experience to the user.

**Why Modernizr**

All web developers come up against differences between browsers and devices. That’s largely due to different feature sets: the latest versions of the popular browsers can do some awesome things which older browsers can’t – but we still have to support the older ones.

Modernizr makes it easy to deliver tiered experiences: make use of the latest and greatest features in browsers which support them, without leaving less fortunate users high and dry.

## **Syntax**

Before you start using Modernizr, you would have to include its javascript library in your HTML page header as follows −

<script src="modernizr.min.js" type="text/javascript"></script>

As mentioned above, you can create CSS rules based on the feature availability and these rules would apply automatically on the webpage if browser does not support a new featur.

**1.1 Modernizr Using CSS**

**Example:**

/\* In your CSS: \*/  
.no-audio #music {  
 display: none; /\* Don't show Audio options \*/  
}  
.audio #music button {  
 /\* Style the Play and Pause buttons nicely \*/  
}  
  
<!-- In your HTML: -->  
<div id="music">  
   
 <audio>  
 <source src="audio.ogg" />  
 <source src="audio.mp3" />  
 </audio>  
   
 <button id="play">Play</button>  
 <button id="pause">Pause</button>  
</div>

**Note:** *Here it is notable that you need to prefix "no-" to every feature/property you want to handle for the browser which does not support them.*

**1.2 Modernizr Using JavaScript**

**if (Modernizr.audio) {  
 /\* properties for browsers that  
 support audio \*/  
}  
  
else{  
 /\* properties for browsers that  
 does not support audio \*/  
}**

## 

## **1.3 Features detected by Modernizr**

|  |  |  |
| --- | --- | --- |
| **Feature** | **CSS Property** | **JavaScript Check** |
| @font-face | .fontface | Modernizr.fontface |
| Canvas | .canvas | Modernizr.canvas |
| Canvas Text | .canvastext | Modernizr.canvastext |
| HTML5 Audio | .audio | Modernizr.audio |
| HTML5 Audio formats | NA | Modernizr.audio[format] |
| HTML5 Video | .video | Modernizr.video |
| HTML5 Video Formats | NA | Modernizr.video[format] |
| rgba() | .rgba | Modernizr.rgba |
| hsla() | .hsla | Modernizr.hsla |
| border-image | .borderimage | Modernizr.borderimage |
| border-radius | .borderradius | Modernizr.borderradius |
| box-shadow | .boxshadow | Modernizr.boxshadow |
| Multiple backgrounds | .multiplebgs | Modernizr.multiplebgs |
| opacity | .opacity | Modernizr.opacity |
| CSS Animations | .cssanimations | Modernizr.cssanimations |
| CSS Columns | .csscolumns | Modernizr.csscolumns |
| CSS Gradients | .cssgradients | Modernizr.cssgradients |
| CSS Reflections | .cssreflections | Modernizr.cssreflections |
| CSS 2D Transforms | .csstransforms | Modernizr.csstransforms |
| CSS 3D Transforms | .csstransforms3d | Modernizr.csstransforms3d |
| CSS Transitions | .csstransitions | Modernizr.csstransitions |
| Geolocation API | .geolocation | Modernizr.geolocation |
| Input Types | NA | Modernizr.inputtypes[type] |
| Input Attributes | NA | Modernizr.input[attribute] |
| localStorage | .localstorage | Modernizr.localstorage |
| sessionStorage | .sessionstorage | Modernizr.sessionstorage |
| Web Workers | .webworkers | Modernizr.webworkers |
| applicationCache | .applicationcache | Modernizr.applicationcache |
| SVG | .svg | Modernizr.svg |
| SVG Clip Paths | .svgclippaths | Modernizr.svgclippaths |
| SMIL | .smil | Modernizr.smil |
| Web SQL Database | .websqldatabase | Modernizr.websqldatabase |
| IndexedDB | .indexeddb | Modernizr.indexeddb |
| Web Sockets | .websockets | Modernizr.websockets |
| Hashchange Event | .hashchange | Modernizr.hashchange |
| History Management | .historymanagement | Modernizr.historymanagement |
| Drag and Drop | .draganddrop | Modernizr.draganddrop |
| Cross-window Messaging | .crosswindowmessaging | Modernizr.crosswindowmessaging |
| addTest() Plugin API | NA | Modernizr.addTest(str,fn) |

## 

## 

## **2. New Input Types**

HTML5 added several new input types:

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

**3. Type = Search**

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

**Example:**

<form>

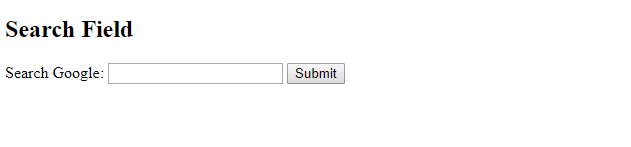
Search Google:

<input type="search" name="googlesearch">

<input type="submit" name="searchSubmit">

</form>

**Output:**

****

**4. Type = tel**

The <input type="tel"> is used for input fields that should contain a telephone number.

***Note:*** *The tel type is currently only supported in Safari 8*.

**Example:**

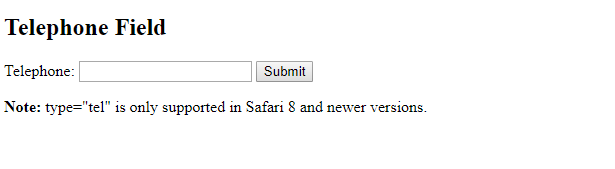
**<form>**

**Telephone:**

**<input type="tel" name="usrtel">**

**</form>**

**Output:**

****

**5. Type = url and email**

**5.1 Type 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.

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.

**Example:**

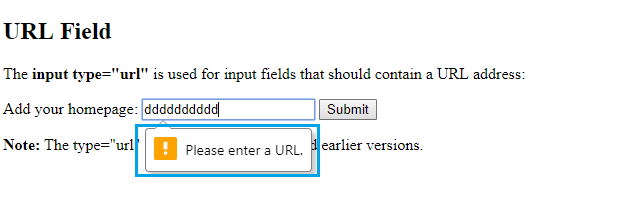
**<form>**

**Add your homepage:**

**<input type="url" name="homepage">**

**</form>**

**Output:**

****

*It validates the text whether it is URL or not*

**5.2 Type 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.

**Example**

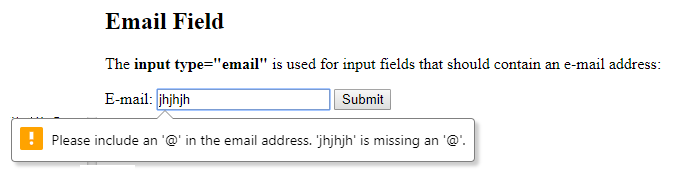
**<form>**

**E-mail:**

**<input type="email" name="email">**

**</form>**

**Output**

****

*Similar to URL email field validate a text whether a email or not*

**6. Type = Date/ Time**

**6.1 Type = 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.

**Example:**

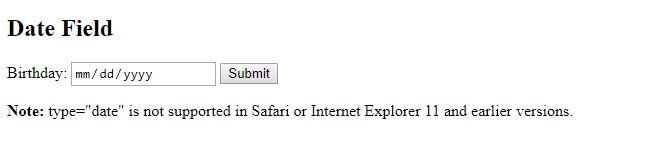
**<form>**

**Birthday:**

**<input type="date" name="bday">**

**</form>**

**Output:**

****

**6.2 Type = Time**

The <input type="time"> allows the user to select a time (no time zone).

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

**Example:**

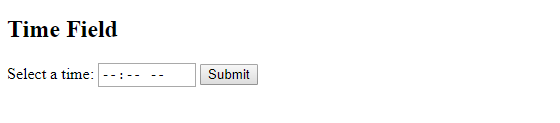
**<form>**

**Select a time:**

**<input type="time" name="usr\_time">**

**</form>**

**Output:**

****

***Note:*** *type="time" is not supported in Safari or Internet Explorer 12 and earlier versions.*

**7. Type = Number**

The <input type="number"> defines a **numeric** input field.

You can also set restrictions on what numbers are accepted.

The following example displays a numeric input field, where you can enter a value from 1 to 5:

**Example:**

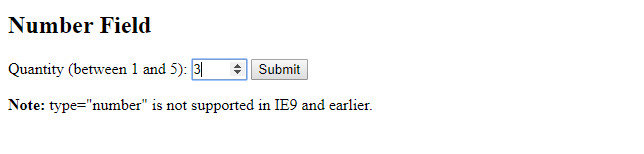
**<form>**

**Quantity (between 1 and 5):**

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

**</form>**

**Output:**

****

***Note:*** *type="number" is not supported in IE9 and earlier.*

**8. Type = Range**

The <input type="range"> defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100.

However, you can set restrictions on what numbers are accepted with the min, max, and step attributes:

*Min max and step attributes used for, to the set the range between which the the value should be entered. It validate the content in the box dynamically at real time.*

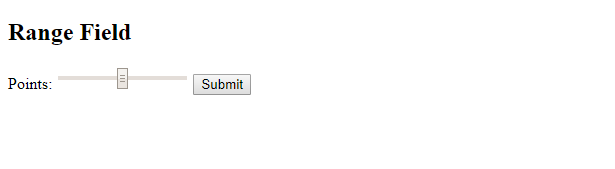
Example:

<form>

<input type="range" name="points" min="0" max="10">

</form>

**Output:**

****

***Note 1:*** Depending on browser support:<br>The input type "range" can be displayed as a slider control.

***Note 2*:** type="range" is not supported in Internet Explorer 9 and earlier versions.

**9. Type = Color**

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

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

**Example:**

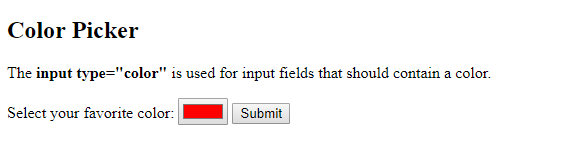
**<form>**

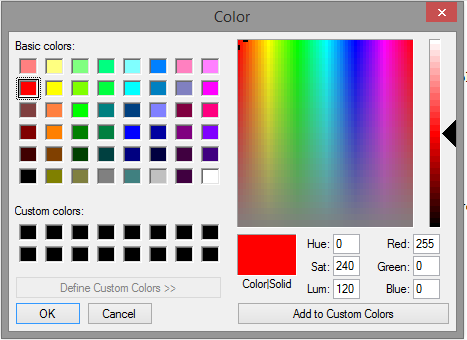
**Select your favorite color:**

**<input type="color" name="favcolor">**

**</form>**

**Output:**

****



**Depending on browser support a color picker can pop-up when you enter the input field. Like This**

**Reference**

1. **https://www.w3schools.com/html/html\_forms.asp**
2. **https://modernizr.com/https://modernizr.com/**
3. **https://www.tutorialspoint.com/html5/html5\_modernizr.htm**
4. **https://www.w3schools.com/html/html\_form\_input\_types.asp**