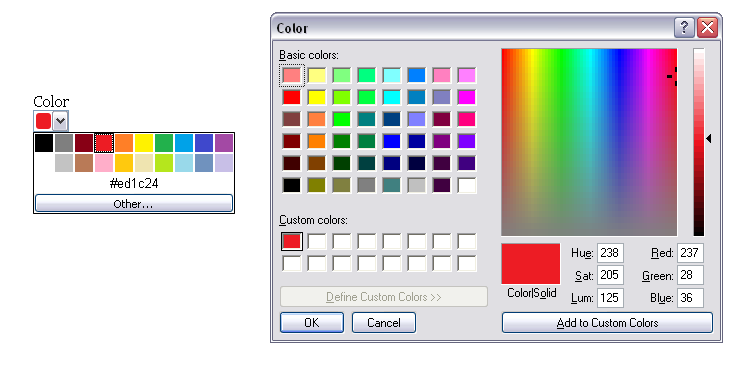
**Input Types :**

**Color**

The color input type is pretty self-explanatory: it allows the user to select a color and returns the hex value for that color. It is anticipated that users will either be able to type the value or select from a color picker, which will either be native to the operating system or a browser’s own implementation.

Example:

<input id="color"name="color"type="color">



**Date and Time**

If you’ve ever booked tickets online, you will have come across date pickers to help you quickly and easily choose the date you require. Perhaps you’ve even implemented a date picker on your own website. Generally, this is done using a JavaScript library such as jQuery, Dojo, or YUI. It can be a pain when you need to load a whole library and associated plug-ins just to implement a simple date picker. Well, with HTML5 we get that functionality baked into the browser. Not only that, but we don’t have to stop at just electing a single date; we can select a week, month, time, date and time, and even date and time with a time zone using the different input types. The markup is pretty straightforward.

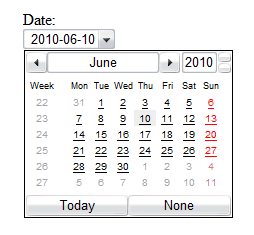
Example 1:

<input id="dob"name="dob"type="date">

We can go a step further by using the min and max attributes to ensure the user can only choose from a specified date range.

Example 2:

<input id="startdate"name="startdate"min="2012-01-01"max="2013-01-01"type="date">

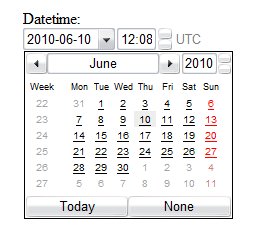


**Datetime**

We can combine the date and time by using type="datetime" for specifying a precise time on a given day

Example:

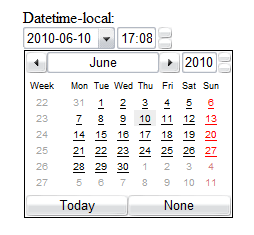
<input id="entry-day-time"name="entry-day-time"type="datetime">



**Datetime-local**

we can achieve slightly more granular control by selecting a precise time on a given day with a local time zone variation using type="datetime-local".

Example:<input id="arrival-time"name="arrival-time "type="datetime-local">

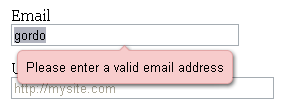


**Email**

In rendering terms, the email input type is no different than a standard text input type and allows for one or more e-mail addresses to be entered. Combined with the required attribute, the browser is then able to look for patterns to ensure a valid e-mail address has been entered. Naturally, this checking is rudimentary, perhaps looking for an @ character or a period (.) and not allowing spaces.

Example (not working):

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

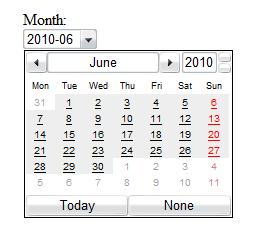


**Month**

Shows type="month", which might, for example, be used for a credit card expiry date.

Example:

<input id="expiry"name="expiry"type="month"required>

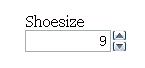


**Number**

number, as you might expect, is used for specifying a numerical value.

Example:

<input type="number"min="5"max="18"step="0.5"value="9"name="shoe-size">

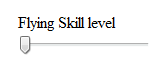


**Range**

The range input type is similar to number but more specific. It represents a numerical value within a given range. Why the difference, I hear you cry? Because when you’re using range, the exact value isn’t important. It also allows browsers to offer a simpler control than fornumber.

Example:

<input id="skill"type="range"min="1"max="100"value="0">



**Search**

 It’s probably the most common action performed on the Web every day

Example:

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



**Tel**

tel differs from email and url in that no particular syntax is enforced. Phone numbers differ around the world, making it difficult to guarantee any type of specific notation except for allowing only numbers and perhaps a + symbol to be entered. It’s possible that you can validate specific phone numbers (if you can guarantee the format) using client-side validation. type="tel" is marked up.

Example :

<input type="tel"name="tel"id="tel"required>

**url**

The url input type, as you might expect, is for web addresses. You can use the multiple attribute to enter more than one URL. Liketype="email", a browser will carry out simple validation on these fields and present an error message on form submission. This is likely to include looking for forward

slashes, periods, and spaces, and possibly detecting a valid top-level domain (such as .com or .co.uk).

Example:

<input type="url"name="url"required>

**Week**

Example:

<input id="vacation"name="vacation"type="week">