

# HTML5 Forms Notes Part B

## HTML5 input types

HTML5 introduced a number of new input types. These new input types give hints to the browser about what type of keyboard layout to display for on-screen keyboards. Users are more easily able to enter the required information without having to change their keyboard and only see the appropriate keys for that input type.

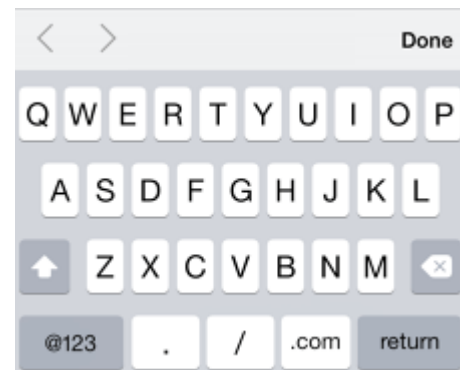
### Input type

`url`

For entering a URL. It must start with a valid URI scheme, for

example `http://`, `ftp://` or `mailto:.`

### Typical Keyboard



`tel`

For entering phone numbers. It does **not** enforce a particular syntax for validation, so if you want to ensure a particular format, you can use pattern.

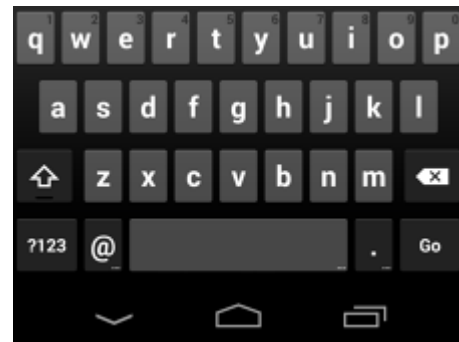


## Input type

email

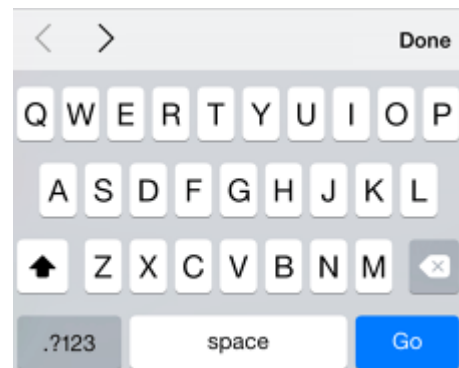
For entering email addresses, and hints that the @ should be shown on the keyboard by default. You can add the multiple attribute if more than one email address will be provided.

## Typical Keyboard



search

A text input field styled in a way that is consistent with the platform's search field.



number

For numeric input, can be any rational integer or float value.



range

For number input, but unlike the number input type, the value is less important. It is displayed to the user as a slider control.

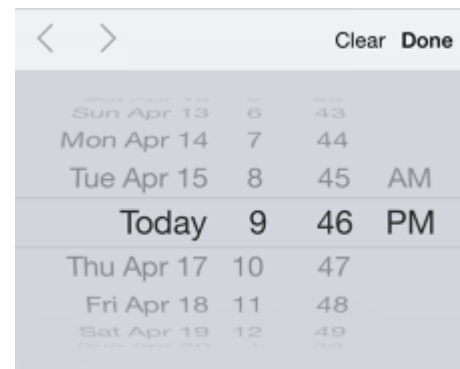


## Input type

`datetime-local`

For entering a date and time value where the time zone provided is the local time zone.

## Typical Keyboard

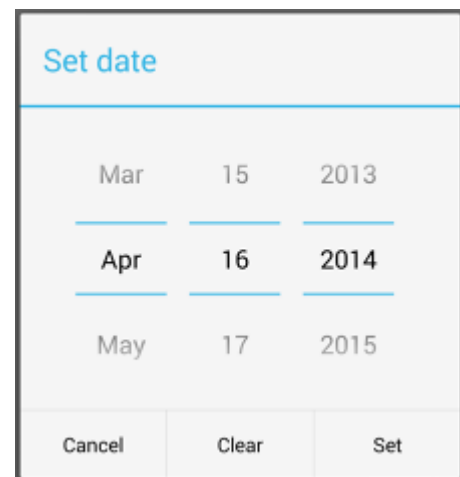


A screenshot of a mobile application showing a date and time picker for the `datetime-local` input type. The interface includes navigation arrows, 'Clear' and 'Done' buttons, and a list of dates and times. The current selection is 'Today' at 9:46 PM.

Date	Hour	Minute	Period
Sun Apr 13	6	43	
Mon Apr 14	7	44	
Tue Apr 15	8	45	AM
<b>Today</b>	<b>9</b>	<b>46</b>	<b>PM</b>
Thu Apr 17	10	47	
Fri Apr 18	11	48	
Sat Apr 19	12	49	

`date`

For entering a date (only) with no time zone provided.

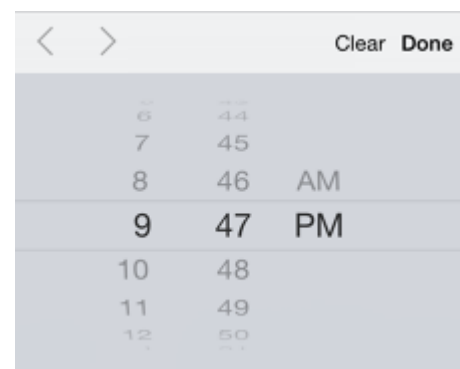


A screenshot of a mobile application showing a date picker for the `date` input type. The interface includes a title 'Set date', a list of dates, and 'Cancel', 'Clear', and 'Set' buttons. The current selection is 'Apr 16 2014'.

Month	Day	Year
Mar	15	2013
<b>Apr</b>	<b>16</b>	<b>2014</b>
May	17	2015

`time`

For entering a time (only) with no time zone provided.



A screenshot of a mobile application showing a time picker for the `time` input type. The interface includes navigation arrows, 'Clear' and 'Done' buttons, and a list of times. The current selection is 9:47 PM.

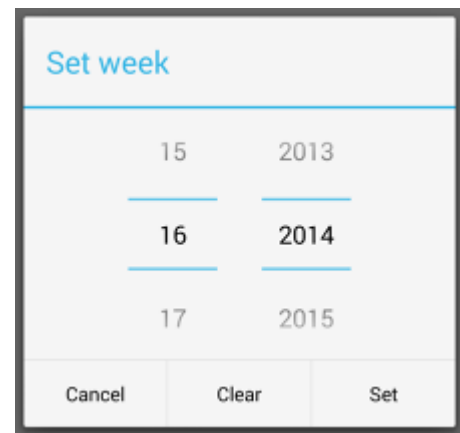
Hour	Minute	Period
6	44	
7	45	
8	46	AM
<b>9</b>	<b>47</b>	<b>PM</b>
10	48	
11	49	
12	50	

## Input type

`week`

For entering a week (only) with no time zone provided.

## Typical Keyboard

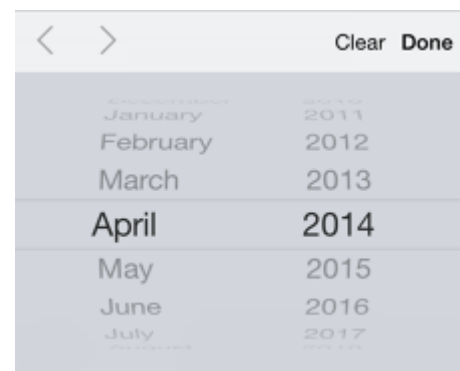


A dialog box titled "Set week" with a light blue header. It contains two columns of input fields. The first column has three fields with values 15, 16, and 17. The second column has three fields with values 2013, 2014, and 2015. Each field is a simple text box with a blue underline. At the bottom, there are three buttons: "Cancel", "Clear", and "Set".

Set week		
15	2013	
16	2014	
17	2015	
Cancel	Clear	Set

`month`

For entering a month (only) with no time zone provided.

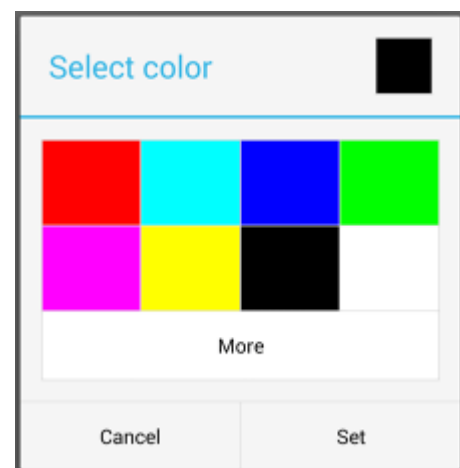


A dialog box for selecting a month and year. It has a light gray header with navigation arrows on the left and "Clear" and "Done" buttons on the right. The main area is a table with months in the first column and years in the second column. The year 2014 is highlighted in bold. The months are listed from January to July.

		Clear	Done
January	2011		
February	2012		
March	2013		
<b>April</b>	<b>2014</b>		
May	2015		
June	2016		
July	2017		

`color`

For picking a color.



A dialog box titled "Select color" with a light blue header. It features a 2x4 grid of color swatches: red, cyan, blue, green in the first row, and magenta, yellow, black, white in the second row. Below the grid is a "More" button. At the bottom, there are two buttons: "Cancel" and "Set".

Select color			
Red	Cyan	Blue	Green
Magenta	Yellow	Black	White
More			
Cancel	Set		

# Offer suggestions during input with datalist

The `datalist` element isn't an input type, but a list of suggested input values to associated with a form field. It lets the browser suggest autocomplete options as the user types. Unlike select elements where users must scan long lists to find the value they're looking for, and limiting them only to those lists, `datalist` element provides hints as the user types.

```
<label for="frmFavChocolate">Favorite Type of Chocolate</label>
<input type="text" name="fav-choc" id="frmFavChocolate"
list="chocType">
<datalist id="chocType">
  <option value="white">
  <option value="milk">
  <option value="dark">
</datalist>
```

## Label and name inputs properly

Forms are hard to fill out on mobile. The best forms are the ones with the fewest inputs. Good forms provide semantic input types. Keys should change to match the user's input type; users pick a date in a calendar. Keep your user informed. Validation tools should tell the user what they need to do before submitting the form

---

## The importance of labels

The `label` element provides direction to the user, telling them what information is needed in a form element. Each `label` is associated with an input element by placing it inside the `label` element, or by using the "for" attribute. Applying labels to form elements also helps to improve the touch target size: the user can touch either the label or the input in order to place focus on the input element.

```
<label for="frmAddressS">Address</label>
<input type="text" name="ship-address" required id="frmAddressS"
  placeholder="123 Any Street" autocomplete="shipping street-
address">
```

# Label sizing and placement

Labels and inputs should be large enough to be easy to press. In portrait viewports, field labels should be above input elements, and beside them in landscape. Ensure field labels and the corresponding input boxes are visible at the same time. Be careful with custom scroll handlers that may scroll input elements to the top of the page hiding the label, or labels placed below input elements may be covered by the virtual keyboard.

## Use placeholders

The placeholder attribute provides a hint to the user about what's expected in the input, typically by displaying the value as light text until the user starts typing in the element.

```
<input type="text" placeholder="MM-YYYY" ...>
```

Remember

- Placeholders disappear as soon as the user starts typing in an element, thus they are not a replacement for labels. They should be used as an aid to help guide users on the required format and content.

## Use metadata to enable auto-complete

Users appreciate when websites save them time by automatically filling common fields like names, email addresses and other frequently used fields, plus it helps to reduce potential input errors – especially on virtual keyboards and small devices.

Browsers use many heuristics to determine which fields they can [auto-populate based on previously specified data by the user](#), and you can give hints to the browser by providing both the name attribute and the autocomplete attribute on each input element.

For example, to hint to the browser that it should auto-complete the form with the user's name, email address and phone number, you should use:

```
<label for="frmNameA">Name</label>
<input type="text" name="name" id="frmNameA"
  placeholder="Full name" required autocomplete="name">
```

```

<label for="frmEmailA">Email</label>
<input type="email" name="email" id="frmEmailA"
  placeholder="name@example.com" required autocomplete="email">

<label for="frmEmailC">Confirm Email</label>
<input type="email" name="emailC" id="frmEmailC"
  placeholder="name@example.com" required autocomplete="email">

<label for="frmPhoneNumA">Phone</label>
<input type="tel" name="phone" id="frmPhoneNumA"
  placeholder="+1-555-555-1212" required autocomplete="tel">

```

## Recommended, input name and autocomplete attribute values

autocomplete attribute values are part of the current [WHATWG HTML Standard](#). The most commonly used autocomplete attributes are shown below.

The autocomplete attributes can be accompanied with a section name, such as **shipping** given-name or **billing** street-address. The browser will auto-complete different sections separately, and not as a continuous form.

Content type	name attribute	autocomplete attribute
Name	name fname mname lname	<ul style="list-style-type: none"> <li>name (full name)</li> <li>given-name (first name)</li> <li>additional-name (middle name)</li> <li>family-name (last name)</li> </ul>
Email	email	email

Content type	name attribute	autocomplete attribute
Address	address city region province statezip zip2 postal country	<p>For one address input:</p> <ul style="list-style-type: none"> <li>o street-address</li> </ul> <p>For two address inputs:</p> <ul style="list-style-type: none"> <li>o address-line1</li> <li>o address-line2</li> <li>• address-level1 (state or province)</li> <li>• address-level2 (city)</li> <li>• postal-code (zip code)</li> <li>• country</li> </ul>
Phone	phone mobile country-code area-codeexchange suffix ext	tel
Credit Card	ccname cardnumber cvc ccmmonth ccyearexp-date card-type	<ul style="list-style-type: none"> <li>• cc-name</li> <li>• cc-number</li> <li>• cc-csc</li> <li>• cc-exp-month</li> <li>• cc-exp-year</li> <li>• cc-exp</li> <li>• cc-type</li> </ul>
Username	username	<ul style="list-style-type: none"> <li>• username</li> </ul>



Content type	name attribute	autocomplete attribute
Passwords	password	<ul style="list-style-type: none"> <li>current-password (for sign-in forms)</li> <li>new-password (for sign-up and password-change forms)</li> </ul>

# The autofocus attribute

On some forms, for example the Google home page where the only thing you want the user to do is fill out a particular field, you can add the `autofocus` attribute. When set, desktop browsers immediately move the focus to the input field, making it easy for users to quickly begin using the form. Mobile browsers ignore the `autofocus` attribute, to prevent the keyboard from randomly appearing.

Be careful using the autofocus attribute because it will steal keyboard focus and potentially preventing the backspace character from being used for navigation.

```
<input type="text" autofocus ...>
```

## Reference

LePage, Pete. "Label and Name Inputs Properly | Web Fundamentals - Google Developers." *Label and Name Inputs Properly | Web Fundamentals - Google Developers*. Google Developers - Web Fundamentals, 02 June 2011. Web. 15 Mar. 2016.

## Authors

**Pete LePage:** Pete is a Developer Advocate

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Terms of Service](#).