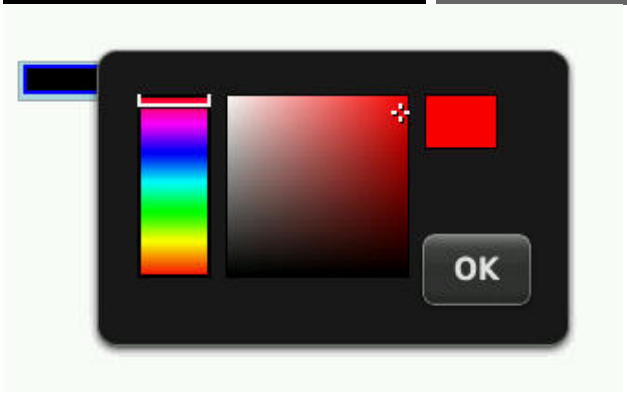
<input type=color>

INTRODUCTION

For years we used hundreds of lines of JavaScript for selecting colors, and now it's bundled in the browser.

Here is how it looks on a mobile device: the first screenshot is from Opera mobile, the second one with Chrome mobile, and the third one on Blackberry. Note that FireFox mobile has also support, even if we do not show a screenshot here.



TYPICAL USE

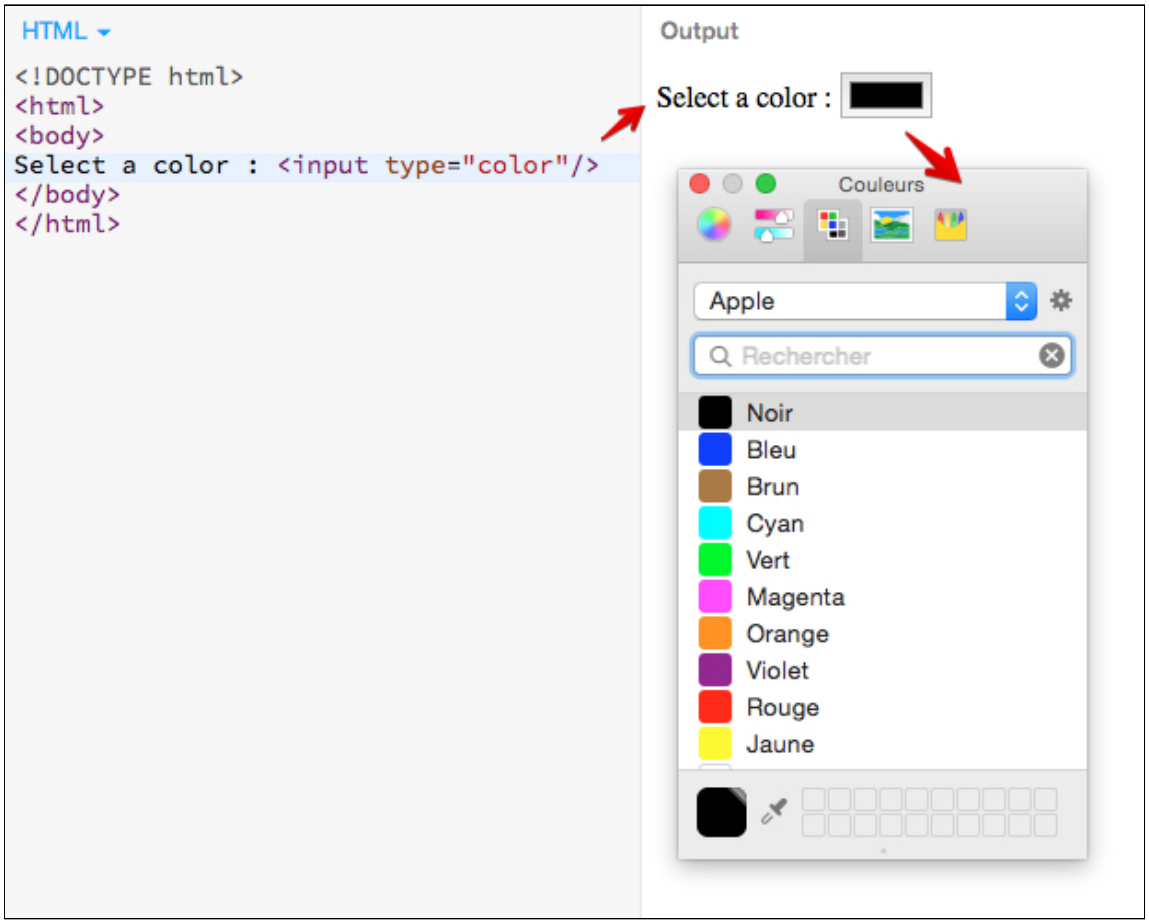
Inserting a color chooser is as simple as that:

```
<!DOCTYPE html>
<html>
<body>
   Choose a color : <input type="color" value="#FF00FF"/>
</body>
</html>
```

Note: In this chapter we are simplifying the examples, as we usually embed input elements in a `<form>...</form>`.

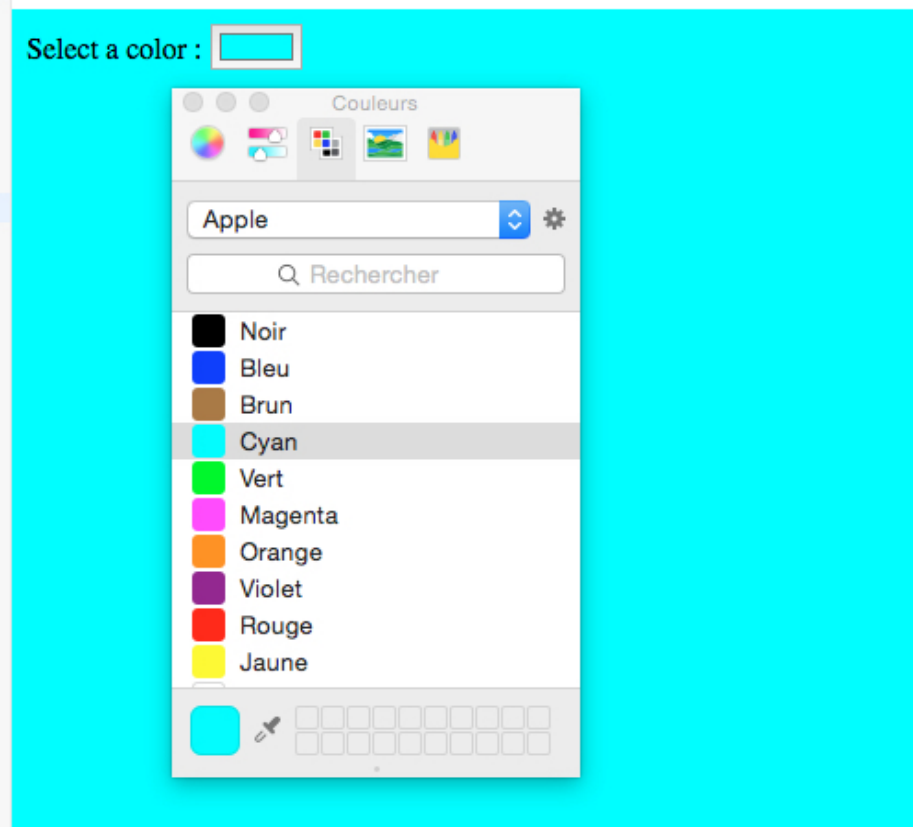
Try `<input type="color">` online on this JS Bin example. Or do it here in your browser: just click the purple square (and if you don't see a light purple square, it means that you are using either Safari or Internet Explorer - see "current support" below): <input type="color" value="#FF00FF" />

Here is the result on Google Chrome:



EXAMPLE: CHANGING THE BACKGROUND COLOR OF THE PAGE

The `<input type="color">` can fire `change` or `input` events. Here is an example that changes the background color of the page when a color is chosen. Try it online at JS Bin.



Source code:

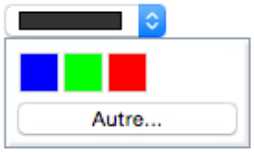
```
<!DOCTYPE html>
<html>
<body>
 Select a color : <input type="color"id="colorChooser"/>
  <script>
    var colorInputField =document.querySelector("#colorChooser");
    colorInputField.addEventListener('input',function(evt) {
        document.body.style.backgroundColor =this.value;
    }, false);
 </script>
</body>
</html>
```

PROPOSE A LIMITED CHOICE OF COLORS

By default, the color selector offers many options that may either frighten some users or just not be appropriate to the purpose of the application.

Good news: it is possible to restrict the proposed choices, and also simplify the user interface, by using a `<datalist>` with some `<option>` elements inside. As of today (April 2015), this only works in Google Chrome and Opera.

Example: if you are using Opera or Chrome, click the black rectangle on the right: . The following should be displayed:



Online example at JS Bin

Source code extract:

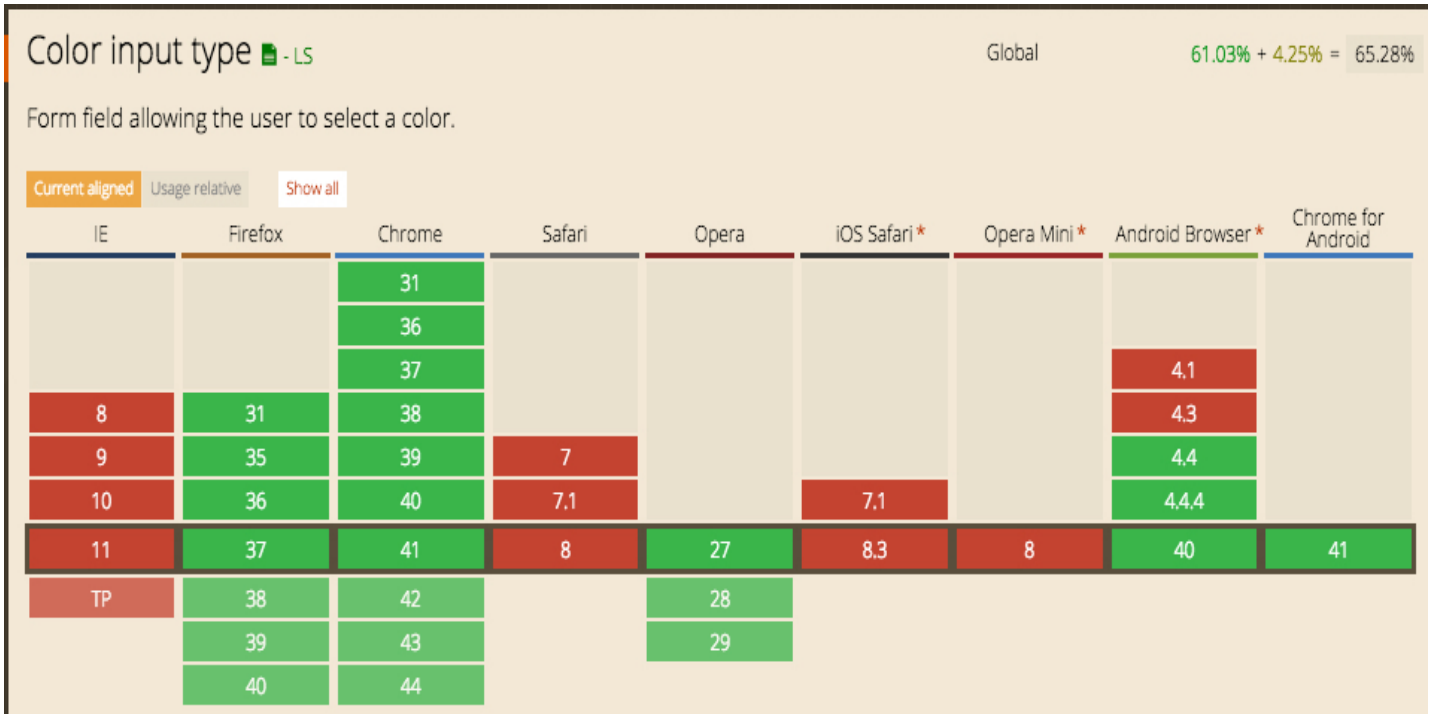
```
<input type="color" value="#333333" list="colors">
<datalist id="colors">
   <option>#0000FF</option>
   <option>#00FF00</option>
   <option>#FF0000</option>
</datalist>
```

Note that the `id` of the `<datalist>` element should be the same as the value of the `list` attribute of the input field.

Warning: color values must use the CSS hexadecimal notation, using 'blue', 'green' and 'red' does not work in current implementations.

CURRENT SUPPORT FOR <INPUT TYPE="COLOR">

As of April 2015, Safari and Internet Explorer still do not support this input type, as shown in the table below:

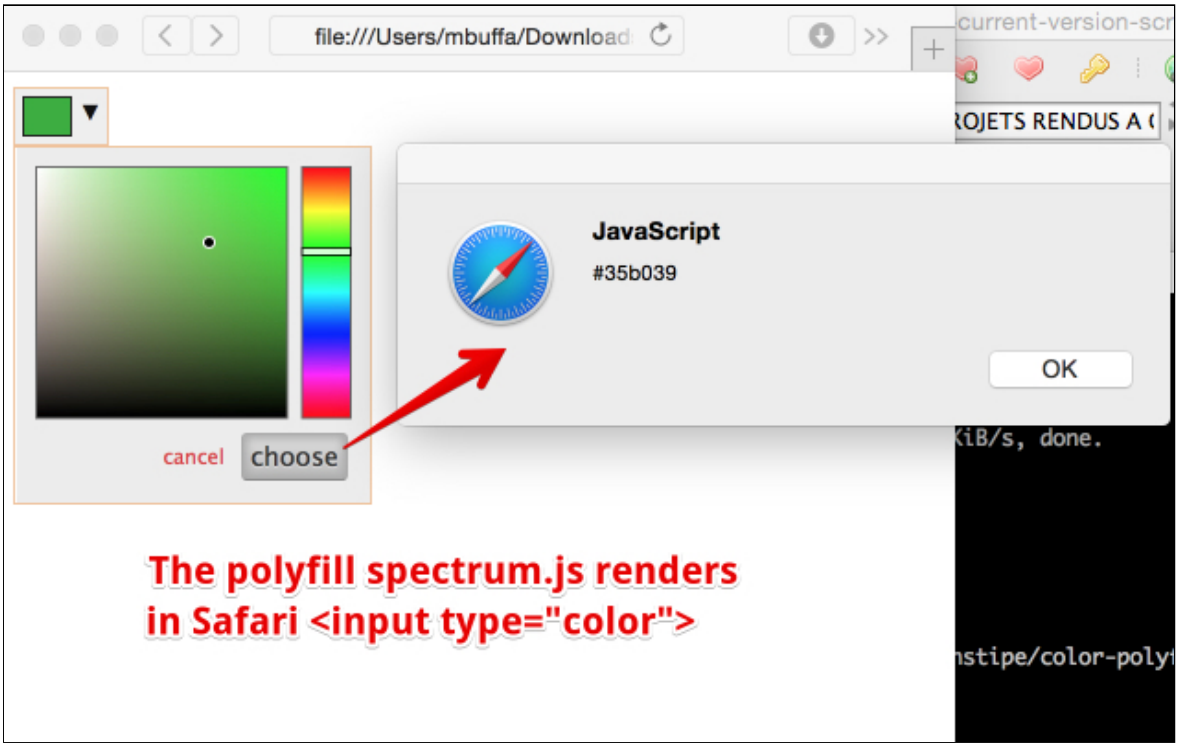


Check for mobile support.

Several polyfills are available

If you click on the link to the caniuse.com support table, you will find links to polyfills. There are a few available on the Web, and some are included in Modernizr.com (a JavaScript library that detects HTML5 and CSS3 features in the user's browser).

Below is an example with the polyfill spectrum.js. See how it renders in Safari (the same code renders natively on browsers that support <input type="color">):



And here is the HTML code (note, you need to first download the lib from the Web site):

```
<!doctype html>
<html>
<head>
 <title>Spectrum polyfill example, for input type=color</title>
 <link rel="stylesheet" type="text/css"href="../spectrum.css">
 <script type="text/javascript"src="../docs/jquery-1.9.1.js"></script>
 <script type="text/javascript"src="../spectrum.js"></script>
</head>
<body>
    <input type="color" onchange="alert(this.value);"></p>
</body>
</html>
```

MAIN PROBLEMS OF THIS ELEMENT

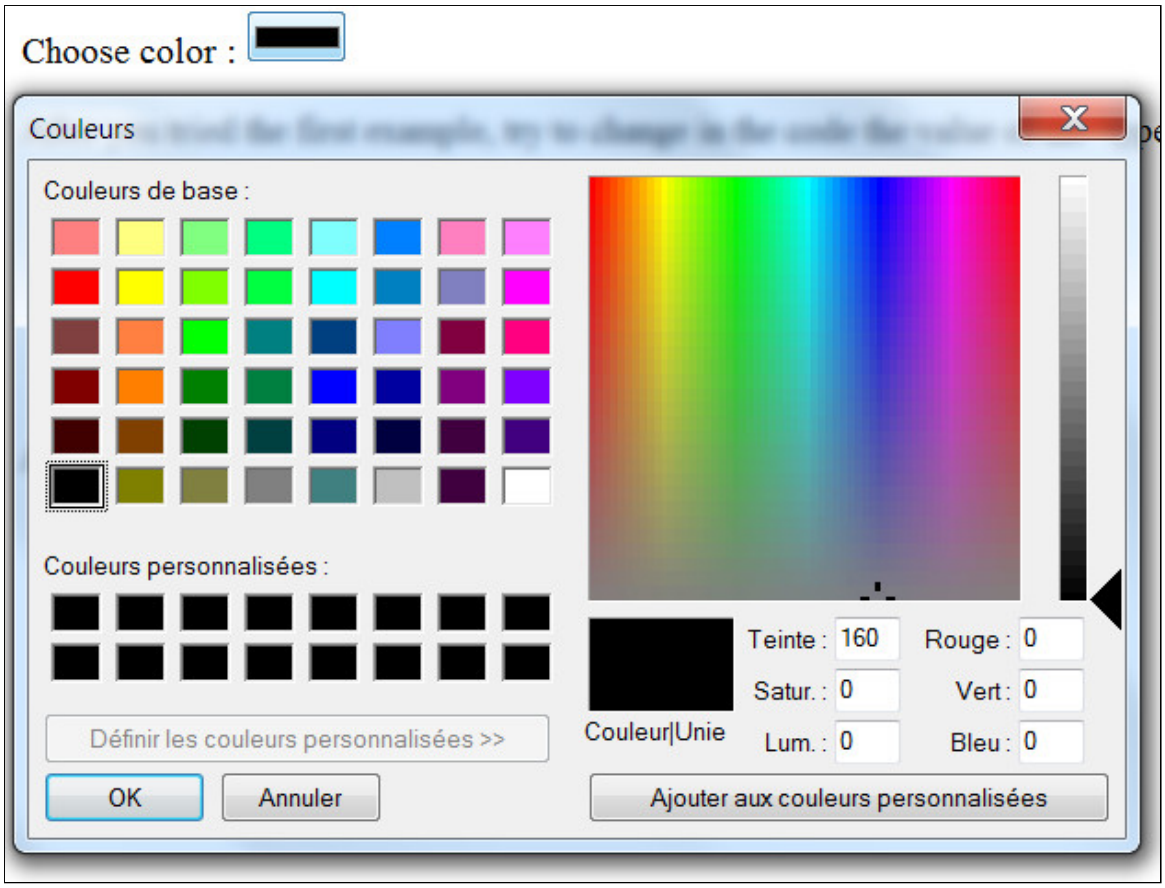
The main critic Web designers make about this element is related to its default appearance being strongly dependent on the browser and its underlying operating system. Changing the*look and feel* is not possible, except with the use of the options we saw already in the previous sections of this page. This problem is also true for other input elements that renders as complex widgets, like `<input type="date">` and its variants.

Another problem is that there is no way to control where the dialog that contains the color chooser will appear, no positioning via CSS or JavaScript is possible.The specification does not say anything how to position it over the page, thus the result is vendor specific.

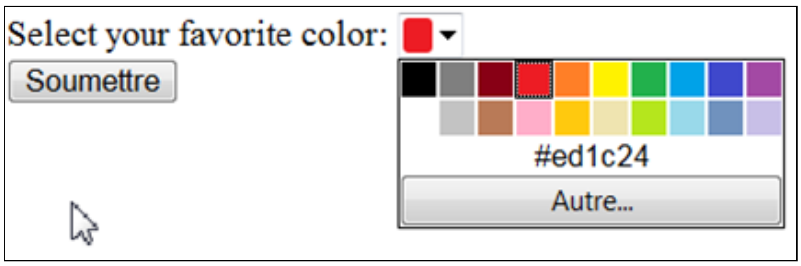
The solution proposed by the W3C and its contributors is called *Web Components*, a new approach for designing HTML5 widgets, that will be studied in the HTML5 Part-2 course.

Below are the different look'n'feels on different versions of different browsers.

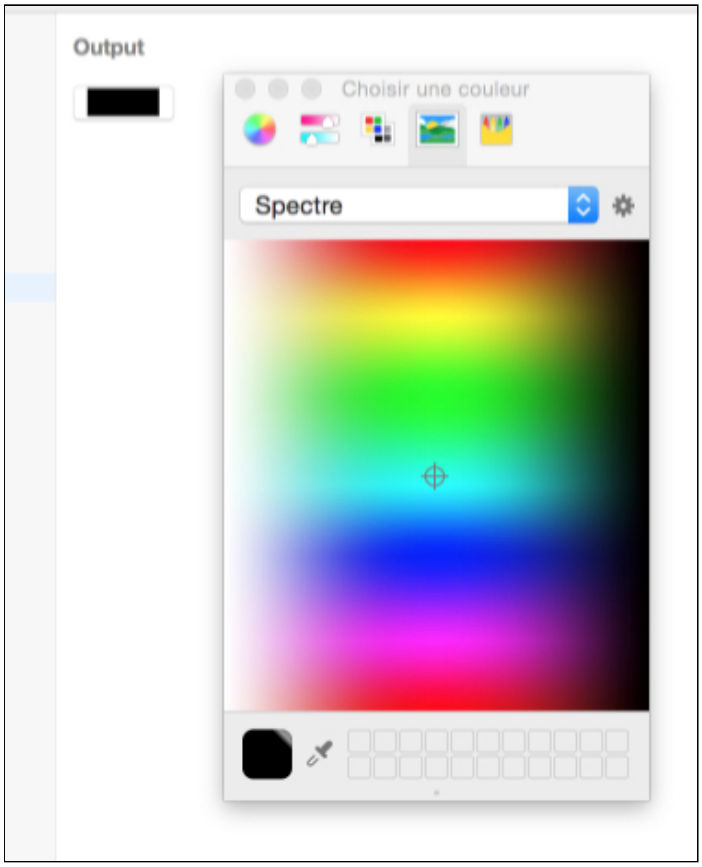
In Google Chrome (screenshot from 2014):



In Opera:



In FireFox:



KNOWLEDGE CHECK 5.4.2 (NOT GRADED)

On mobile devices, `<input type=color>` pops up a dialog that is adapted to each operating system (IOS, Android, etc.). On desktops, the native implementations differ in their look'n' feel. Is it possible to customize deeply the look'n' feel of this input type using only CSS and HTML attributes?

- Yes
- No