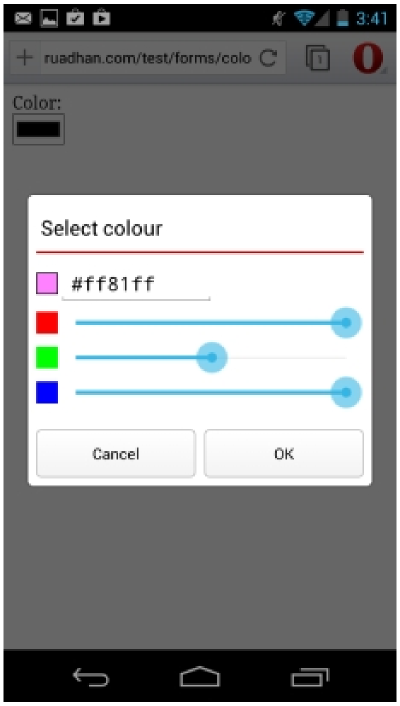
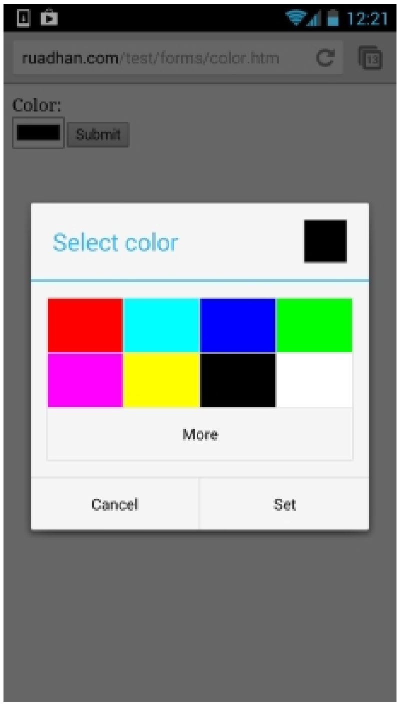
<input type=color>

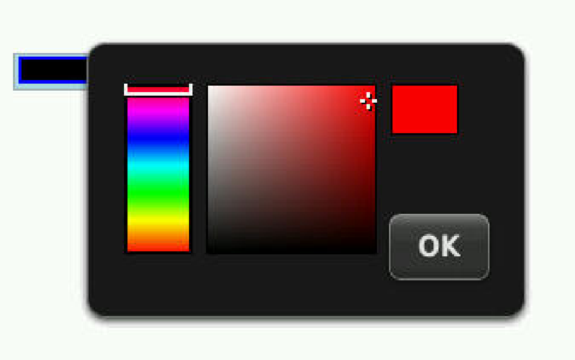
INTRODUCTION

For years we used hundreds of lines of JavaScript for selecting colors, but 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 with Chrome mobile, and the third on Blackberry. Note that, although no screenshot is shown here, FireFox mobile also has support.







TYPICAL USE

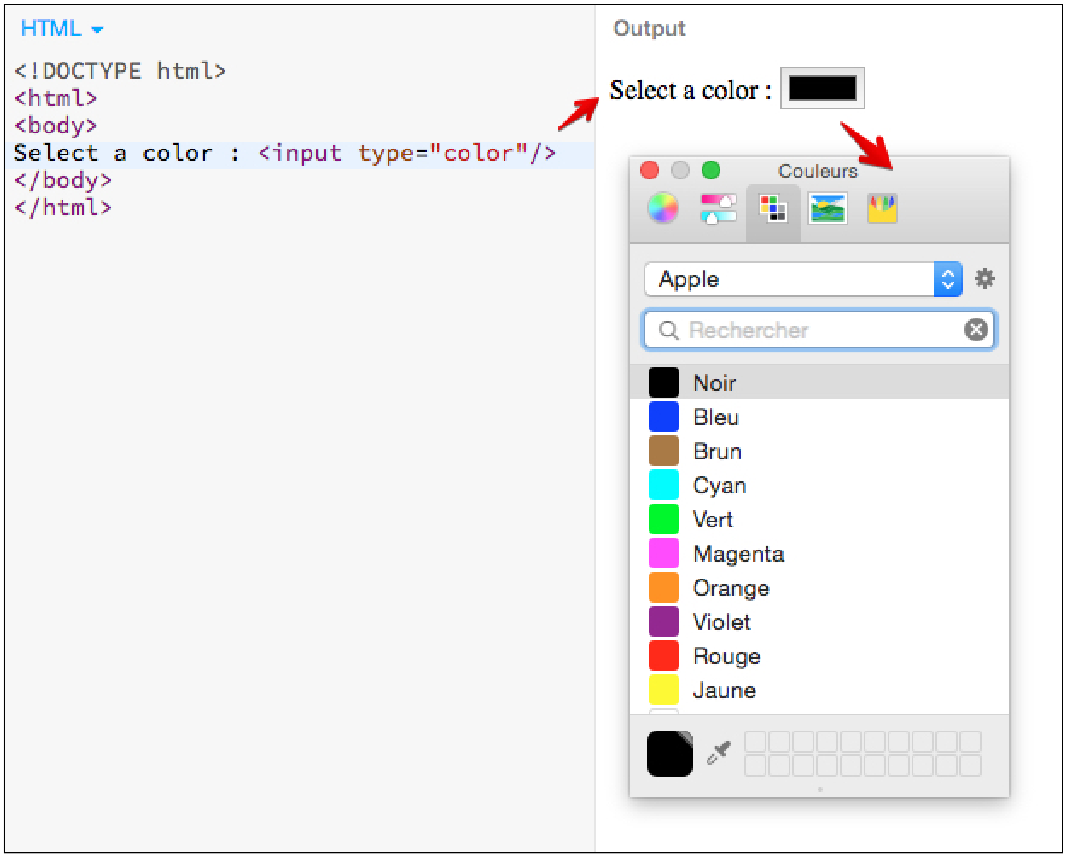
Inserting a color chooser is as simple as:

```
<!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):🟪

Here is the result on Google Chrome:

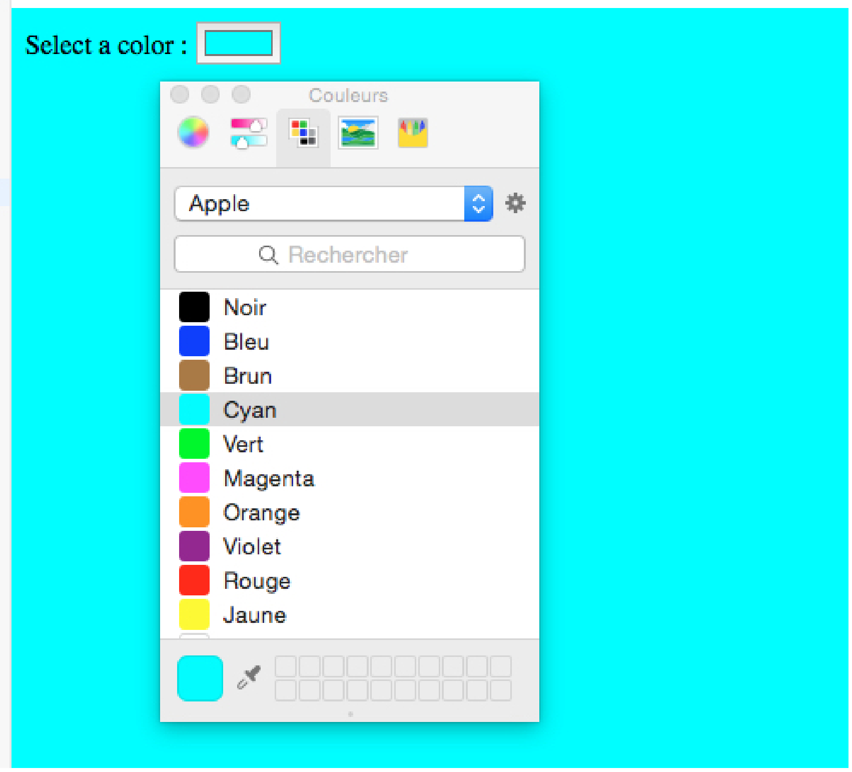


HTML

```
<!DOCTYPE html>
<html>
<body>
Select a color : <input type="color"/>
</body>
</html>
```

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:

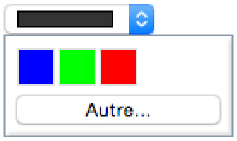
```
<!DOCTYPE html>
<html>
<body>
  Select a color : <input type="color"id="colorChooser"/>
  <script>
     var colorInputField =document.querySelector("#colorChooser");
     colorInputField.addEventListener('input',function(evt) {
          document.body.style.backgroundColor =this.value;
      }, false);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

Good news: it is possible to restrict the choices, and also simplify the user interface, by using a `<datalist>` with some`<option>` elements inside. This feature is not yet (as at April 2015) supported by all browsers, check this compatibility table.

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:

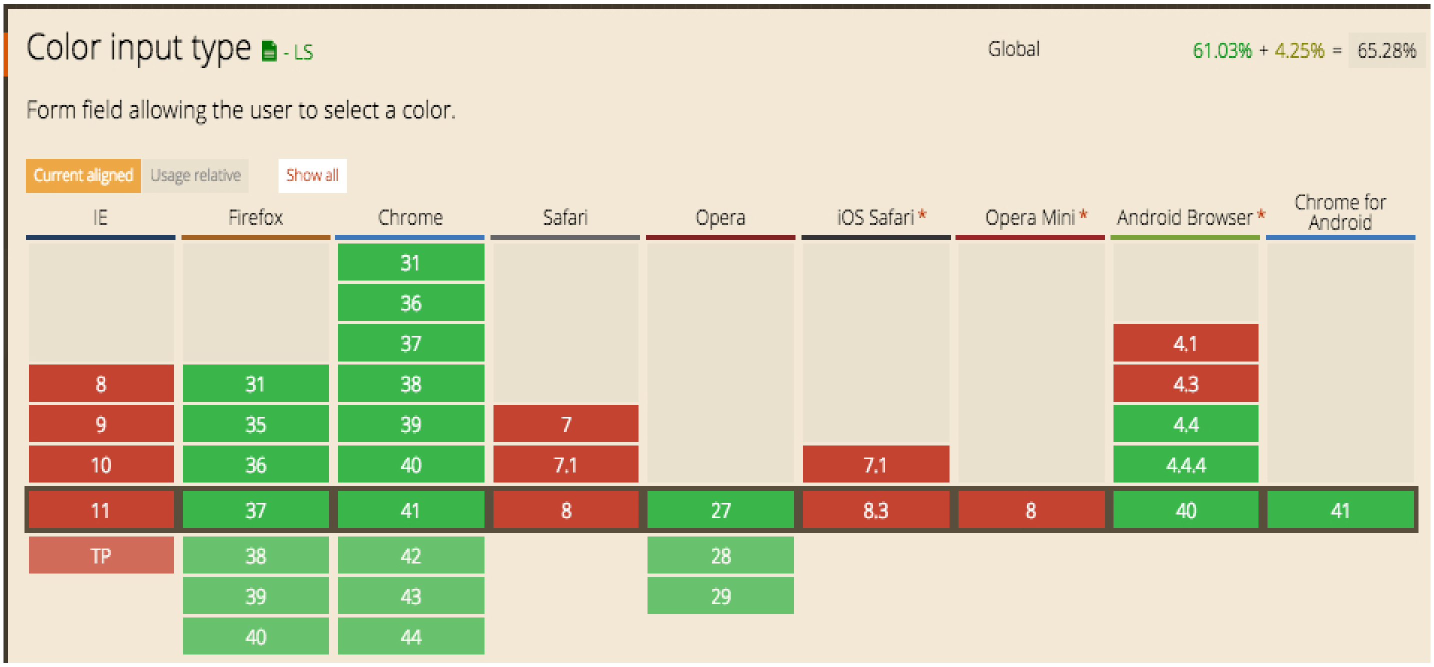
```
<input type="color" value="#333333"list="colors">
<datalist id="colors">
    <option>#0000FF</option>
    <option>#00FF00</option>
    <option>#FF0000</option>
</datalist>
```

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



Color input type - LS

Global 61.03% + 4.25% = 65.28%

Form field allowing the user to select a color.

Current aligned | Usage relative | Show all

IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		31						
		36						
		37					4.1	
8	31	38					4.3	
9	35	39	7				4.4	
10	36	40	7.1		7.1		4.4.4	
11	37	41	8	27	8.3	8	40	41
TP	38	42		28				
	39	43		29				
	40	44						

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">):

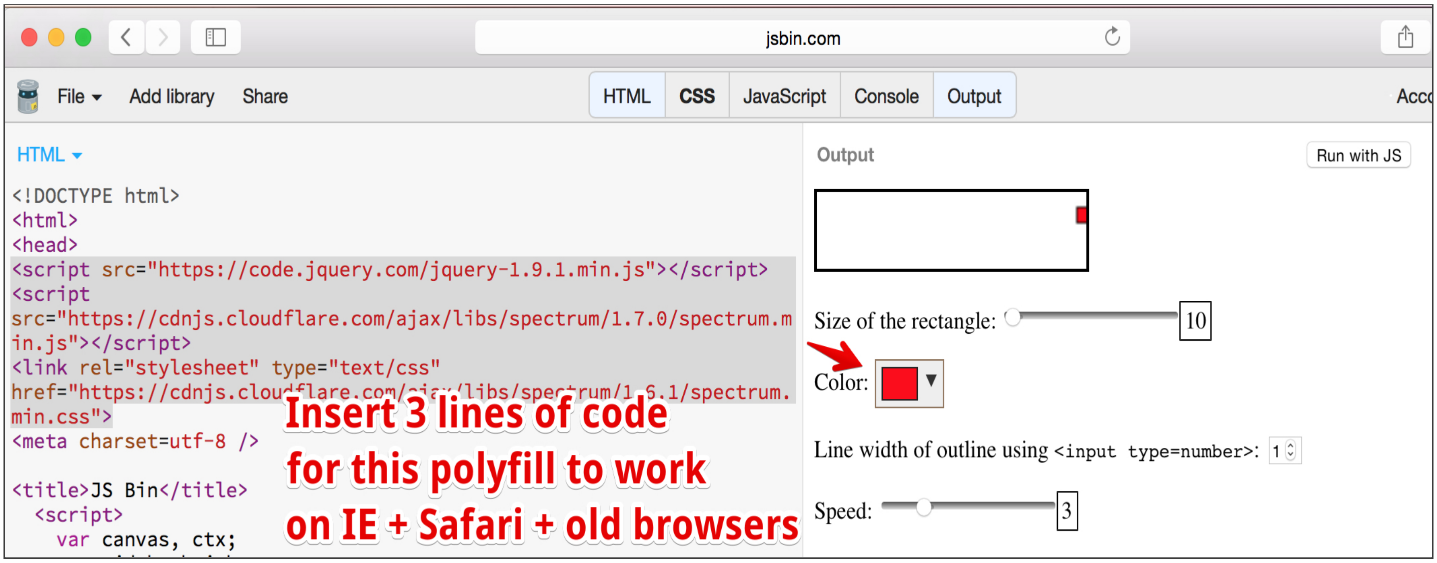
For those of you who would prefer using a CDN, here is another example that uses this polyfill, directly on JS Bin. You just have to include two lines in the HTML and your `<input type=range>` will work on Safari and IE too :-) Notice that this polyfill does not fire the input event, only the change event, when a color is picked.

Example on JS Bin

We just added these two lines inside the `<head>..</head>`element:

```
<scriptsrc="https://code.jquery.com/jquery-1.9.1.min.js"></script>
<scriptsrc="https://cdnjs.cloudflare.com/ajax/libs/spectrum/1.7.0/spectrum.min.js">
</script>
<link rel="stylesheet" type="text/css"

href="https://cdnjs.cloudflare.com/ajax/libs/spectrum/1.6.1/spectrum.min.css">
```



```
<!DOCTYPE html>
<html>
<head>
<script src="https://code.jquery.com/jquery-1.9.1.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/spectrum/1.7.0/spectrum.m
in.js"></script>
<link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/spectrum/1.6.1/spectrum.
min.css">
<meta charset=utf-8 />

<title>JS Bin</title>
  <script>
    var canvas, ctx;
```

Output

Run with JS

Size of the rectangle: 10

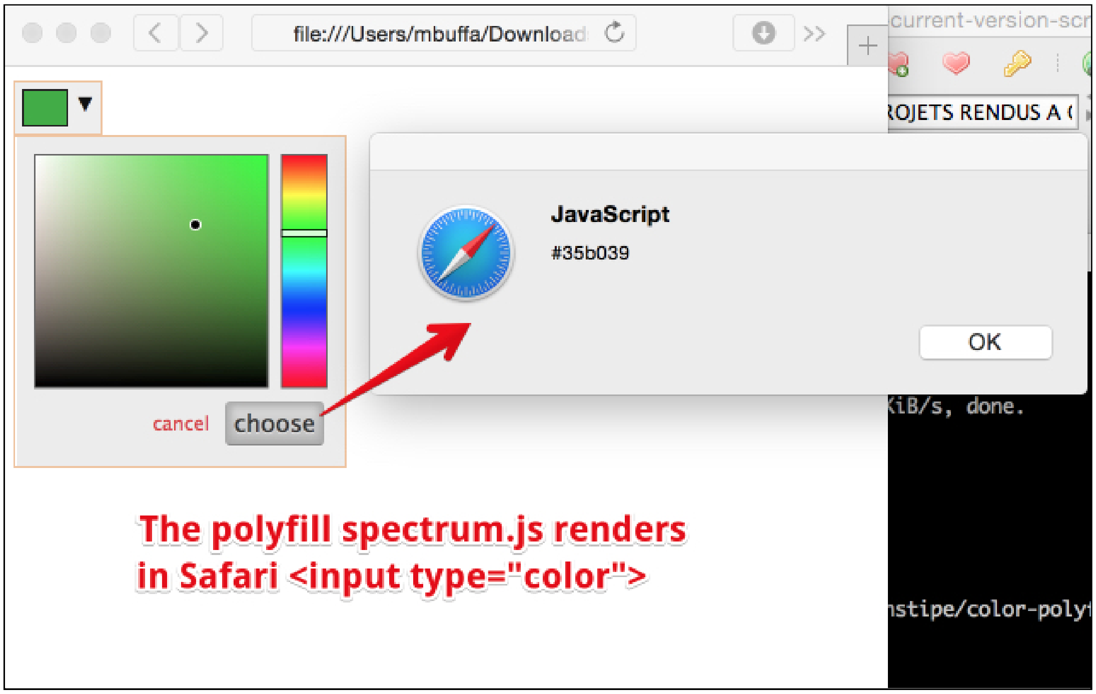
Color:

Line width of outline using `<input type=number>`: 1

Speed: 3

Insert 3 lines of code for this polyfill to work on IE + Safari + old browsers

Another example of this polyfill. This time we downloaded it and made our own HTML page:



JavaScript

#35b039

The polyfill spectrum.js renders in Safari <input type="color">

And here is the HTML code (note, you first need to 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>
```

THE MAIN PROBLEMS WITH THIS ELEMENT

The main criticism that 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 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.

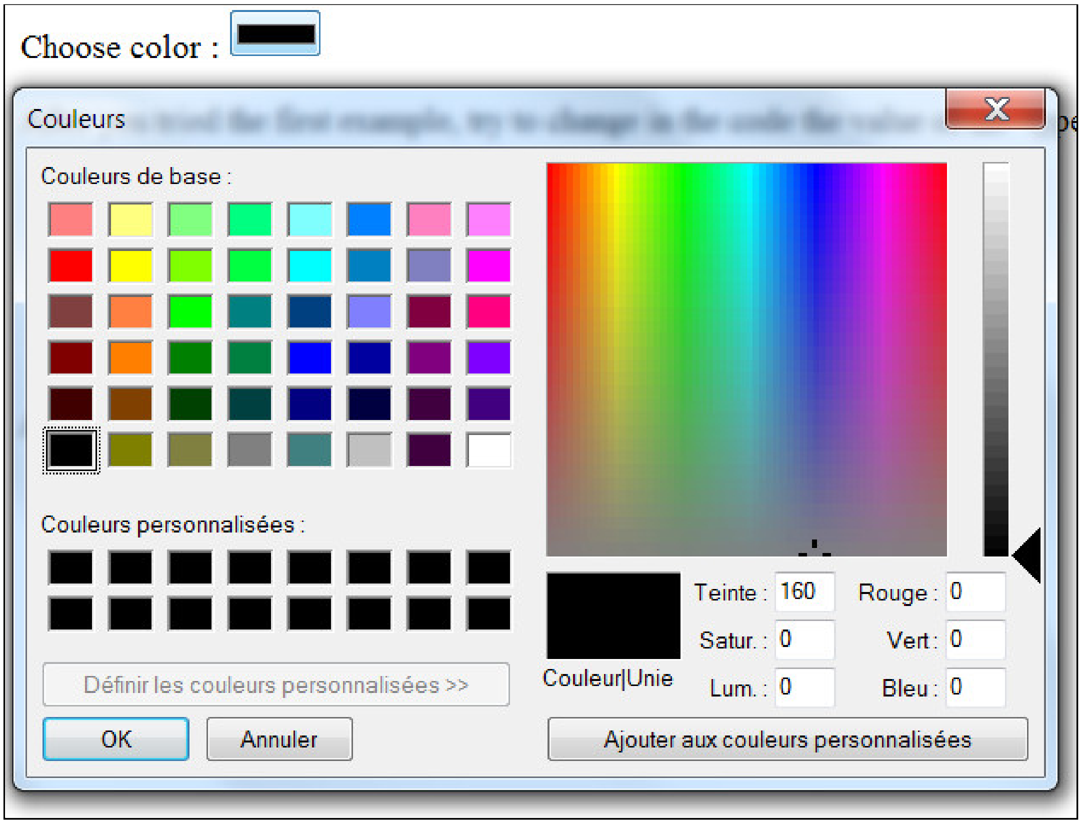
Another problem is that there is no way to control where the dialog that contains the color chooser will appear - no positioning via CSS or JavaScript is possible. The specification does not say anything about how to position it over the page, thus the result is vendor specific.

The solution proposed by the W3C and its contributors is called *Web Components*, a new approach for designing

HTML5 widgets, that will be covered 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):

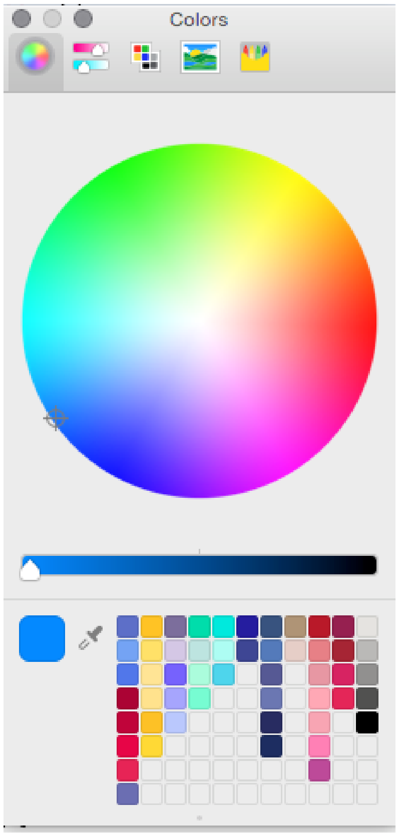

Choose color :
Couleurs
Couleurs de base :
Couleurs personnalisées :
Définir les couleurs personnalisées >>
OK
Annuler
Couleur|Unie
Ajouter aux couleurs personnalisées

Teinte : 160 Rouge : 0

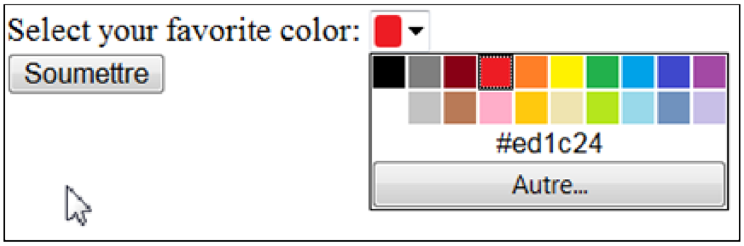
Satur. : 0 Vert : 0

Lum. : 0 Bleu : 0

June 2015 / Chrome / Mac OS X Yosemite:

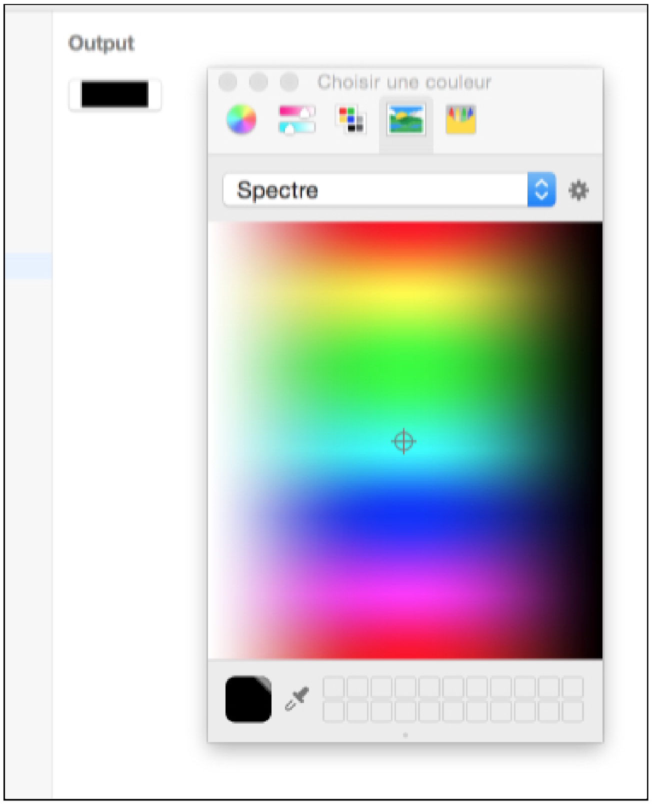


In Opera:


Select your favorite color:

#ed1c24

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 thoroughly customize the look'n' feel of this input type using only CSS and HTML attributes?

○ Yes

○ No