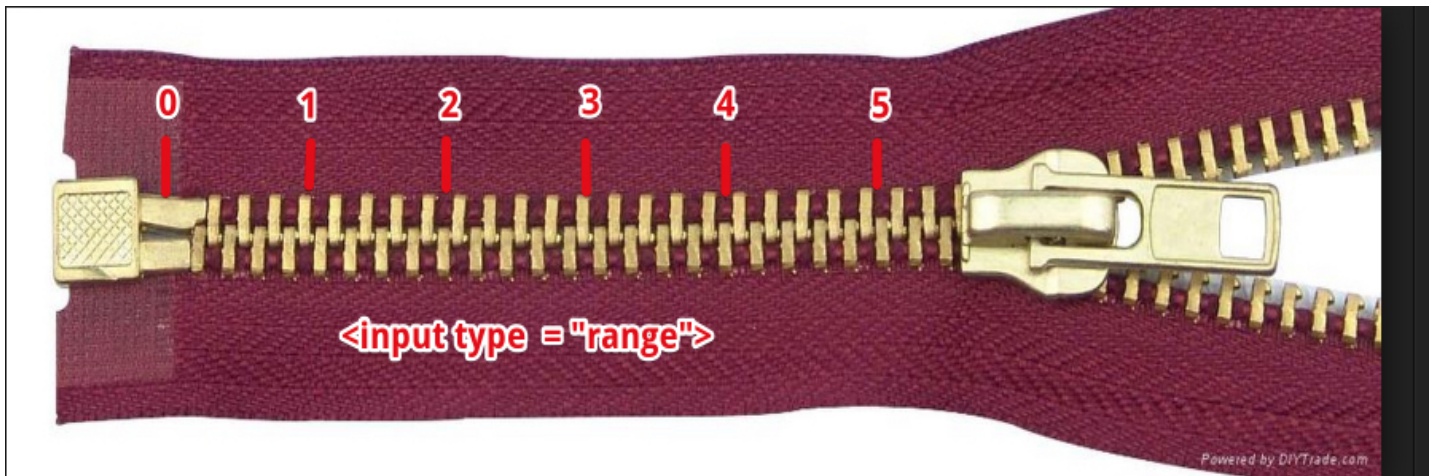
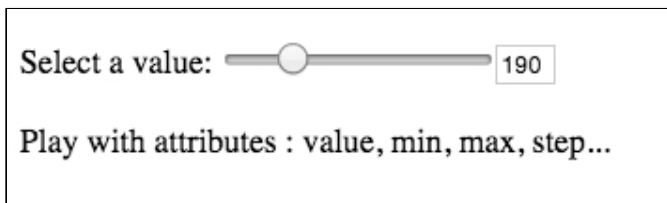


```
<input type="range">
```

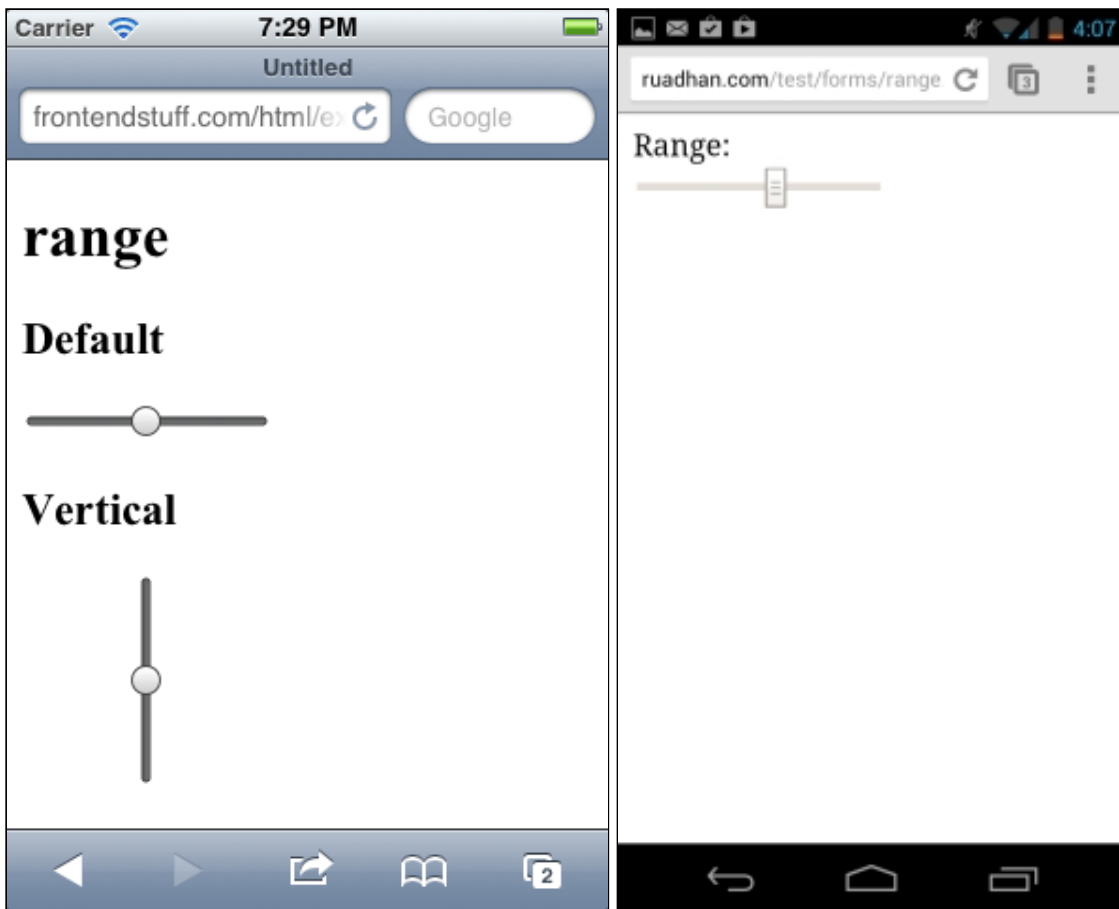


This input type renders as a slider. It accepts the same attributes as the `<input type="number"/>`: `min`, `max`, `step` and `value`.

Example of rendering on a desktop:



And on mobile devices: (left IOS, right Android)



TYPICAL USE

The basic use is to specify at least the `value`, `min` and `max` attributes, and eventually the `step` attribute, too - like this:

```
<input id="slider6" type="range" min="0"max="10" step="2" value="5">
```

But most of the time, you will need a visual feedback that shows the current value selected by the slider.

[This online example at JS Bin](#) shows how to add a visual feedback using a very short JavaScript function and an `<output>` element. The result is shown in the first screenshot of this page, and can be tried here live in your browser, just click and drag the small cursor of the slider (or use up and down arrow keys when the field has the focus):

Select a value:

Source code:

```
20. <!DOCTYPE html>
    <html>
      <head>
        <style>
          #rangeValue1 {
            border:1px solid black;
            padding:2px;
          }
        </style>
        <script>
          window.onload = function() {
            // Called when the page is loaded, for displaying initial
            value in the output
            printValue('slider1','rangeValue1');
          }
          function printValue(sliderId,outputId) {
            var x =document.getElementById(outputId);
            var y =document.getElementById(sliderId);
            x.value = y.value;
          }
        </script>
      </head>
      <body>
        <form >
          <label for="slider1">Select a value:</label>
          <input id="slider1" type="range"
            min="100" max="500" step="10"value="150"
            oninput="printValue('slider1','rangeValue1')"/>
          <output id="rangeValue1"></output>
        </form>
        <br/>
        Play with attributes: value, min, max, step...
      </body>
    </html>
```

SNAPPING BEHAVIOR AND THE `STEP` ATTRIBUTE

When you click and drag the slider, it "jumps" to some snap points corresponding to the integer values of the range defined by the `min` and `max` attributes, and the "size of the jumps" depends on

the value of the `step` attribute.

Try these examples and look at their behavior
([complete online version is also available at JS Bin](#)):



`value=5 min=0, max=10 step=1:`



`value=12 min=10, max=50 step=4:`



Note that in the previous example, the default value displayed is 14, not 12 (the value just above `min` plus an integer `step` value). 12 is not possible so it's been "snapped" to 14.

`value=5 min=0, max=10 step="0.5":`



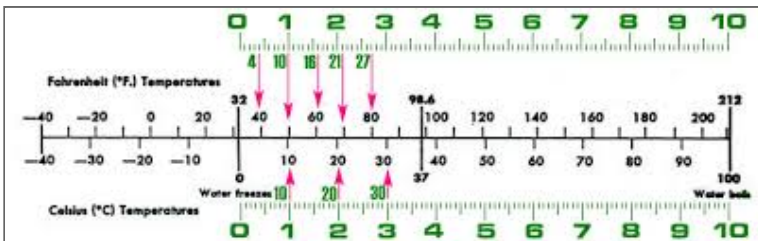
In the previous example, it's necessary to add quotes for setting `step="0.5"` (while HTML5 authorizes not using quotes for setting integer values to attributes).

`value=5 min=0, max=10 step="any":`



WARNING: Using a `step` attribute with an integer value will make the slider jump corresponding to the `step` value. By default, omitting the `step` attribute is equivalent to `step="1"`, so for accepting float values, it is necessary to use `step="any"`, or step equal to a floating point value, such as `step="0.5"`.

ADDING "TICKS" TO THE RANGE SLIDER USING A `<datalist>` ELEMENT



Using the `<datalist>` element, it's possible to display "ticks" above the range slider, at given positions.


This feature is not yet (in 2015) supported by all browsers, [check this compatibility table](#).


```


<label for="slider2">value=5 min=0, max=10 step=1, ticks at 2, 4, 6,
8 and 10:</label>
<input id="slider2" type="range"
      list="ticks2"
      min="0" max="10" step="1" value="5"/>
<datalist id="ticks2">
  <option>0</option>
  <option>2</option>
  <option>4</option>
  <option>6</option>
  <option>8</option>
  <option>10</option>
12. </datalist>


```

Here is a complete online example at [JS Bin](#) that shows multiple possibilities. You can try this using the sliders below:

value=5 min=0, max=10 step=1, ticks at 2, 4, 6, 8 and 10: 

value=20 min=10, max=50 step=5, ticks at 0, 10, 20, 30, 40 and 50: 

value=5 min=0, max=10 step="0.5", ticks at 0, 0.5, 1, 2, 4, 8: 

value=5 min=0, max=10 step="any", ticks at 0, 5 and 10: 

EXTERNAL RESOURCES

- Visit [Dudley Storey's blog](#), and in particular [his post on HTML5 sliders](#).
- A script with the code from the above resource, that automatically generates ticks, depending on the min, max and step attributes, codepen demonstration here: see the Pen [Auto-Generated HTML5 range input Ticks](#) by Dudley Storey (@dudleystorey) on [CodePen](#).

STYLING THE SLIDERS

You can use CSS for "standard" styling (size, color, background color, etc), and special pseudo classes that depend on browser vendors for styling the cursor. Here are some examples of what can be done:

[Example at JS Bin](#)

Or try it here:

value=0.5 min=0, max=1 step="0.1":



HTML:

```
<p>
<label for="sliderVolume">value=0.5 min=0, max=11 step="0.1":
</label><p>
<input id="sliderVolume" type="range"min="0" max="1" step="0.1" value="0.5"
oninput="printValue('sliderVolume','volumeValue')"class="custom"/>
<output id="volumeValue"/>
</p>
```

CSS:

```
.custom { // horizontal rectangle of the slider
  width: 60%;
  height:15px;
  border-radius: 8px;
  box-shadow: inset 0 0 5px #333;
  background-image:linear-gradient(to
right, lightGreen, blue 100%, orange);
  transition: background 450ms;
}

10. /* Change the Slider Button Color Webkit (Opera, Chrome, Safari) */
.custom::-webkit-slider-thumb {
  -webkit-appearance: none !important;
  background-color: #AAA;
  background-image:
  linear-gradient(to bottom, #EEE, #AAA);
  border: 2px solid #999;
  height:30px;
  width:30px;
  border-radius: 15px;

20. }
```

```

/* Change the Slider Button Color FireFox */
.custom::-moz-range-thumb {
    background-color: #AAA;
    background-image:
        linear-gradient(to bottom, #EEE, #AAA);
    border: 1px solid #999;
    height:30px;
    width:30px;
    border-radius: 15px;
31. }

```

JavaScript (just for changing the linear gradient of the CSS background color, the CSS transition (line 7 of the above code) makes the color change animated/smooth:

```

function setVolume(sliderID, textbox) {
    var x =document.getElementById(textbox);
    var y =document.getElementById(sliderID);
    x.value = y.value;
    volumeListener(sliderID);
}

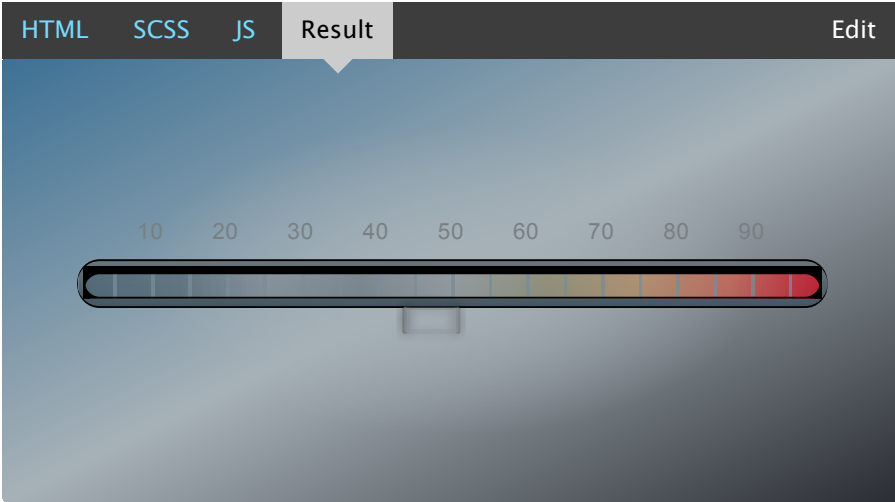
function volumeListener(id) {
10.     var slider=document.getElementById(id);
        var val = (slider.value -slider.min) / (slider.max - slider.min);
        val *= 100; // val is between 0 and 1, values of the min and max
attributes
                // We scale this to appropriate values for a gradient
        var cssProp = 'linear-gradient(to right , lightGreen, blue
'+val +'%, red)';
        slider.style.backgroundImage =cssProp;
}

```

EXTERNAL RESOURCES WITH VERY IMPRESSIVE SLIDER STYLINGS

- A must see: [Ana Tudor's custom range sliders](#)

Some of her CodePens:



[HTML](#)[SCSS](#)[JS](#)[Result](#)[Edit](#)

KNOWLEDGE CHECK 5.4.6 (NOT GRADED)

How do you add "ticks" along a slider created with the `<input type=range>` element?

- ☐ Use only CSS
 - ☐ Use the `datalist` element
 - ☐ Use the `step` attribute
-