

Sliders are thin tracks with one or more handles on them that the user can drag along the track.

The goal of a slider is to provide an alternative input method for defining a numerical value; the slider represents a range, and sliding a handle along the track defines a value within this range.

Sliders can be in either horizontal or vertical orientation. When horizontal, the left end of the track usually represents the lowest value, while in a vertical orientation, the bottom of the slide is usually the lowest value.

To use script.aculo.us's slider capabilities, you'll need to load the slider.js module along with the prototype.js module. So your minimum loading for script.aculo.us will look like this –

```
<script type="text/javascript" src="/javascript/prototype.js"></script>
<script type="text/javascript" src="/javascript/scriptaculous.js?load=slider"></script>
```

Creating a Slider Control

Creating a slider is, as usual, a matter of constructing a custom object over a few existing elements in your page's DOM. You'll need two elements here, one for the *handle* and one for the *track* as follows –

```
new Control.Slider(handle, track [ , options ] );
```

The track element is usually a `<div>`, and the handle element is a `<div>` or `` within the track element. Both can be passed either by their `id=` or by direct DOM references, as usual.

Sliders Options

You can use one or more of the following options while creating your Slider object.

Option	Description
Axis	Defines the orientation of the control as <i>horizontal</i> or <i>vertical</i> . The default orientation is <i>horizontal</i> .
Range	Defines the range of the slider values as an instance of a Prototype ObjectRange instance. Defaults to 0 through 1.
Values	Defines the discrete set of values that the slider can acquire. If omitted, all values within the range can be set.
sliderValue	Sets the initial value of the slider. If omitted, the value represented by the leftmost <i>or top-most</i> edge of the slider is the initial value.
Disabled	If true, it creates a slide that is initially disabled. Obviously defaults to false.
setValue	Will update the slider's value and thus move the slider handle to the appropriate position.
setDisabled	Will set the slider to the disabled state <i>disabled = true</i> .
setEnabled	Will set the slider to the enabled state <i>disabled = false</i> .

You can provide the following callbacks in the options parameter –

Option	Description
--------	-------------

onSlide Called whenever the Slider is moved by dragging. The called function gets the slider value as its parameter.

onChange Called whenever the Slider has finished moving or has had its value changed via the setSlider Value function. The called function gets the slider value as its parameter.

Sliders Example

```

<html>
  <head>
    <title>Sliders Example</title>

    <script type="text/javascript" src="/javascript/prototype.js"></script>
    <script type="text/javascript" src="/javascript/scriptaculous.js?load=slider"
></script>

    <script type="text/javascript">

      window.onload = function() {
        new Control.Slider('handle1' , 'track1',{
          range: $R(1,100), values: [1,25,50,75,100], sliderValue: 1, onChange:
function(v){
          $('changed').innerHTML = 'Changed Value : '+v;
        },

        onSlide: function(v) {
          $('sliding').innerHTML = 'Sliding Value: '+v;
        }
      });
      new Control.Slider('handle2' , 'track2', { range: $R(1,100),
axis:'vertical', sliderValue: 1, onChange: function(v){
          $('changed').innerHTML = 'Changed Value : '+v;
        }

        onSlide: function(v) {
          $('sliding').innerHTML = 'Sliding Value: '+v;
        }
      });
    </script>

    <style type="text/css">

      h1{ font-size: 1.5em; }

      .track {
        background-color: #aaa;
        position: relative;
        height: 0.5em; width: 10em;
        cursor: pointer; z-index: 0;
      }

      .handle {
        background-color: red;
        position: absolute;
        height: 1em; width: 0.25em; top: -0.25em;
        cursor: move; z-index: 2;
      }

      .track.vertical {
        width: 0.5em; height: 10em;
      }

      .track.vertical .handle {
        width: 1em; height: 0.25em; top: 0; left: -0.25em;
      }
    </style>
  
```

```

</style>
</head>
<body>

<h1>Simple sliders</h1>

<div>
  <div></div>
</div>

<p></p>
<p></p>

<div>
  <div></div>
</div>

</body>
</html>

```

Points to note:

- You can change the slider image of any slider using CSS. Use CSS properties *background-image: urltrack.gif* and *background-repeat: no-repeat* to set the slider image.
- The range value can be specified using *R(minValue, maxValue)*. For example, R1, 100.
- The range value can be specified in terms of specific values. For example values: [1,25,50,75,100]. In this case, the slider would only achieve the discrete values listed as the handle was moved.
- At any time, the value of the slider can be set under program control by calling the *setValue* method of the slider instance, as in: *sliderInstance.setValue50;*

This will produce following result –

