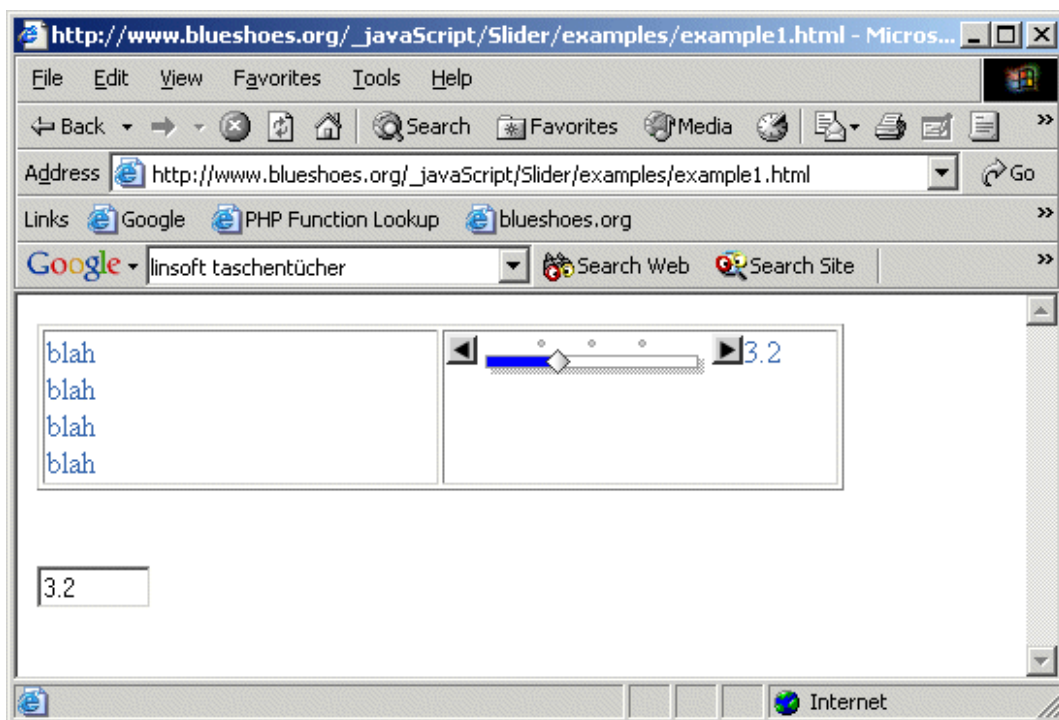# Bs_Slider Howto

Version   1.2, updated 2002/12/09
Author   Andrej Arn

The Bs_Slider is a javascript/dhtml component.

We are going to create this slider:

You can find similar final code in the directory slider/examples/example1.html

The js libraries needed are:
>   /_bsJavascript/lib/LibCrossBrowser.js
>   /_bsJavascript/lib/EventHandler.js
>   /_bsJavascript/core/form/Bs_FormUtil.lib.js
>   /_bsJavascript/components/slider/Bs_Slider.class.js

Note: If you have not installed the full BlueShoes framework then you need to put these files somewhere yourself, and your paths to the files will look different. For example you could put all these files into a subdir of the webroot called /bsSlider/.

Then we need some onLoad code. That is:

```
initCrossBrowserLib();
```

and the code to create your slider(s), like this:

```
1.  mySlider = new Bs_Slider();
2.  mySlider.objectName = 'mySlider';
3.  mySlider.width         = 121;
4.  mySlider.height        = 26;
5.  mySlider.minVal        = 0;
6.  mySlider.maxVal        = 10;
7.  mySlider.valueInterval = 0.1;
8.  mySlider.arrowAmount   = 1;
9.  mySlider.defaultVal    = 0;
10.    mySlider.imgBasePath   = '/_bsJavascript/components/slider/img/';
11.    mySlider.setBackgroundImage('bob/background.gif', 'no-repeat');
12.    mySlider.setSliderIcon('bob/slider.gif', 13, 21);
13.    mySlider.setArrowIconLeft('img/arrowLeft.gif', 16, 16);
14.    mySlider.setArrowIconRight('img/arrowRight.gif', 16, 16);
15.    mySlider.useInputField = 3;
16.    mySlider.colorbar = new Array();
17.    mySlider.colorbar['color']           = 'blue';
18.    mySlider.colorbar['height']          = 2;
19.    mySlider.colorbar['widthDifference'] = -10;
20.    mySlider.colorbar['offsetLeft']      = 5;
21.    mySlider.colorbar['offsetTop']       = 12;
22.    mySlider.draw('sliderDiv1');
```

On line 22 we draw the slider and have it render into an element on the page. In our case it's a div with the ID "sliderDiv1". So add:

```
<div id="sliderDiv1"></div>
```

To see how we can grab the actual value out of the slider element, we attach an onChange event to it. Add the code:

```
mySlider.attachOnChange(bsSliderChange);
```

bsSliderChange is a global function. We could also attach some js code that would be evaluated. Let's define the function like this:

```
function bsSliderChange(obj){
  document.f.t.value = obj.getValue();
}
```

The param obj is always a reference to the slider object itself. In all attached events you can expect to get that. So we use the getValue() method of the slider object to get the value, and assign it to a form field. the form is called "f", the field is called "t". Add the form to your page like this:

```
<form name="f"><input type="text" name="t" value="" size=6></form>
```

Now that's it. if it does not work for you, get the example source code and run it (and check what's missing/wrong in your code).

Three more things to say:

The attached onchange event to update a form field was just an example. If you want to use the slider in a form you don't need to do this. The slider uses a built in hidden form field called "vf" + div-id ("vfsliderDiv1" in our example).

The slider api offers tons of options to set. You can customize the look and behavior a lot.

Basically a slider component is a nice and easy way for the user to input data. If you want to be old-browser-compatible, that can be done but requires some additional code. Even browsers without javascript (or js turned off) can input some data. For those we offer an input field. You could also offer radio buttons, for example to choose between 1-5-10 (or low-medium-high or so).

```
<div id="sliderDiv1">
  <noscript>
    <input type="text" name="myInputField" size="3" value="0">
  </noscript>
</div>
```