

LING 408/508: Computational Techniques for Linguists

Lecture 20

Administrivia

- More new stuff to cover today
- Any other questions on the homework?
 - (Now due next Monday midnight)




```
49 function random_td() {  
50     var row = Math.floor(Math.random() * 4);  
51     var col = Math.floor(Math.random() * 4);  
52     return document.getElementById("puzzle").rows[row].cells[col];  
53 }
```

Javascript Forms

Register for Miss Bakers Space Cadets

Personal details

First Name

Last Name
 

Date of Birth

Email

URL

Telephone

Shoesize

Flying Skill level (1 low - 100 high)

Example:

- <http://html5doctor.com/demos/forms/forms-example.html>
- HTML Forms:
 - allow the user to input information
 - multiple named input fields for text, numbers, radio buttons, check boxes etc. can be defined within a form
 - values can be sent to a Web server (using GET or POST) by clicking on a button
 - web server implementation: *later in this course*
 - we'll use forms and call javascript functions (*browser-side functionality only*)

BMI: Javascript

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Javascript BMI</title>
5 <script>
6 function computeBMI(e) {
7     var weight = e.form.weight.value;
8     var height = e.form.height.value;
9     var output = document.getElementById("output");
10    var scale = e.form.units[0].checked ? 10000 : 703;
11    var bmi = weight * scale / (height * height);
12    var range;
13    if (bmi < 18.5) {
14        range = "underweight"
15    } else if (bmi < 25) {
16        range = "normal"
17    } else if (bmi < 30) {
18        range = "overweight"
19    } else {
20        range = "obese"
21    }
22    output.innerHTML =
23        "BMI: " + bmi.toFixed(2) +
24        "<br>Range: " + range;
25 }
26 </script>
```

```
27 </head>
28 <body>
29 <h1>Javascript BMI</h1>
30 <div>
31 <form action="" method="GET">
32 Weight (kg/lbs): <input type="text" name="weight" size=5>
33 <br>
34 Height (cm/ins): <input type="text" name="height" size=5>
35 <br>
36 <input type="radio" name="units" value="kg" checked>kg-cm
37 <input type="radio" name="units" value="lbs">lbs-ins
38 <br>
39 <input type="button" name="button" Value="Click" onClick="computeBMI(this)">
40 </form>
41 </div>
42 <div id="output"></div>
43 </body>
44 </html>
```

Javascript BMI

Weight (kg/lbs):

Height (cm/ins):

kg-cm lbs-ins

Click

File: bmi-js.html

BMI: Javascript

- https://www.w3schools.com/html/html_forms.asp

Radio Button Input

`<input type="radio">` defines a **radio button**.

Radio buttons let a user select ONE of a limited number of choices:

Example

```
<form>
  <input type="radio" name="gender" value="male" checked>
  Male<br>
  <input type="radio" name="gender" value="female">
  Female<br>
  <input type="radio" name="gender" value="other"> Other
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

- Male
- Female
- Other

Text Input

`<input type="text">` defines a one-line input field for **text input**:

Example

```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

Try it Yourself »

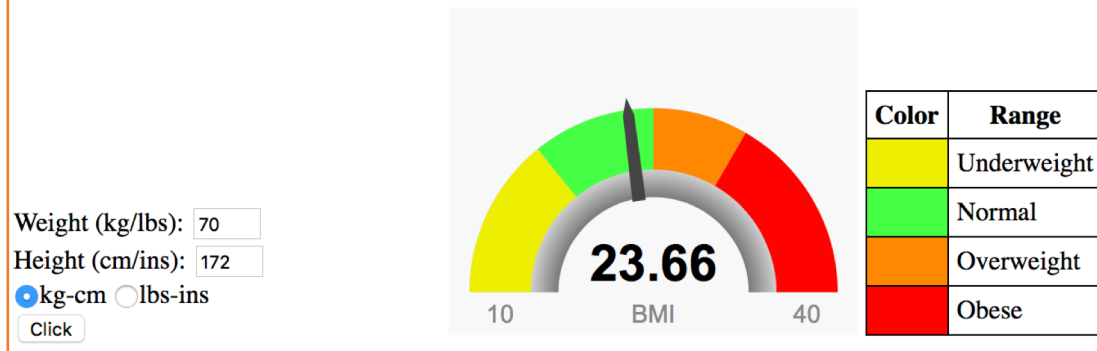
This is how it will look like in a browser:

First name:

Last name:

BMI: SVG

Javascript SVG/BMI



A Universal Gauge for Your Web Dashboard



Steffen Ploetz, 16 Dec 2014



4.95 (19 votes)

Rate this:

JavaScript plugin gaugeSVG to generate widely configurable SVG gauge for a web dashboard

 [Download source and sample - 6.6 KB](#)

 [Download source and sample - v2.0 - 8.3 KB](#)

 [Download source and sample - v2.1 - 8.7 KB](#)

<http://www.codeproject.com/Articles/604502/A-universal-gauge-for-your-web-dashboard>

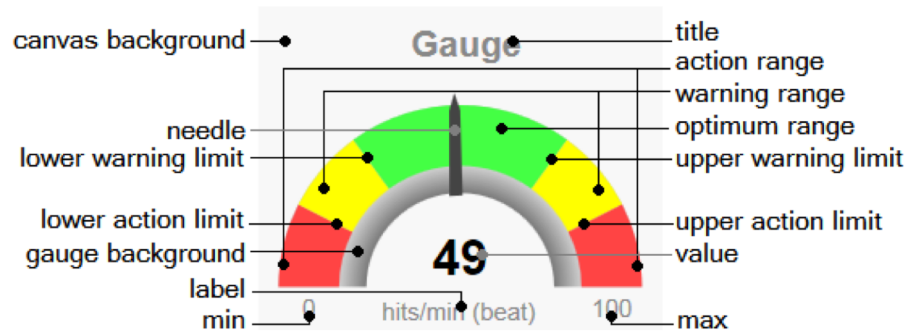
bmi-gauge.html

Download gaugeSVG.js

from the course webpage (I've modified the original code a bit)

gaugeSVG.js

- **Note:** I've modified his code slightly to allow for different colors for lower and upper warning ranges



- **title:** [string] The title text that is displayed above the gauge. It can be an empty string to be suppressed. Default is an empty string.
- **titleColor:** [#rrggb] The title color. Default is "#888888".
- **value:** [float] The value to display. Default is $(\text{max} - \text{min}) / 2.0$. Values below **min** are shown as **min**. Values above **max** are shown as **max**.
- **valueColor:** [#rrggb] The value text color. Default is "#000000". The accurate value is shown in the center of the gauge as text.
- **label:** [string] The label displayed below the value text. It can be an empty string to be suppressed. Default is empty string. Typically used to display the value's measuring unit.
- **labelColor:** [#rrggb] The label text color. Default is "#888888".
- **min:** [float] The minimum of the gauge display range. Will be displayed as text at the gauge start point, if **showMinMax** is true.
- **max:** [float] The maximum of the gauge display range. Will be displayed as text at the gauge end point, if **showMinMax** is true.
- **showMinMax:** [bool] Hide or display the **min** and **max** gauge display range values as text. Default is true.
- **minmaxColor:** [#rrggb] The **min** and **max** value's text color. Default is "#888888".
- **canvasBackColor:** [#rrggb] The background color of the gauge canvas. Default is "#f8f8f8".
- **gaugeWidthScale:** [float] The width of the gauge arc. Default is 1.0. Meaningful values range from 0.15 to 1.5. Lower values show a smaller arc, higher values show a thicker arc.
- **gaugeBorderColor:** [float] The gauge arc border color. Default is "#cccccc".
- **gaugeBorderWidth:** [#rrggb] The gauge arc border width. Default is 0.

- **gaugeBackColor:** [#rrggb] The gauge arc background color. Default is "#cccccc".
- **showGaugeShadow:** [bool] Hide or display a gauge arc shadow. Default is true. The gauge shadow is made of a SVG radial gradient. The gradient start color is the **gaugeShadowColor**. The gradient stop color is the **gaugeBackColor**.
- **gaugeShadowColor:** [#rrggb] The gauge arc shadow color. Default is "#000000".
- **gaugeShadowScale:** [float] The width of the gauge arc's shadow. Default is 1.0. Meaningful values range from 0.8 to 1.5. Lower values show a smaller shadow, higher values show a thicker shadow.
- **lowerActionLimit:** [float] The lower action limit or a negative value, if not desired. Default is $(\text{max} - \text{min}) * 0.15 + \text{min}$.
- **lowerWarningLimit:** [float] The lower warning limit or a negative value, if not desired. Default is $(\text{max} - \text{min}) * 0.30 + \text{min}$.
- **upperWarningLimit:** [float] The upper warning limit or a negative value, if not desired. Default is $(\text{max} - \text{min}) * 0.70 + \text{min}$.
- **upperActionLimit:** [float] The upper action limit or a negative value, if not desired. Default is $(\text{max} - \text{min}) * 0.85 + \text{min}$.
- **needleColor:** [#rrggb] The gauge needle color. Default is "#444444".
- **optimumRangeColor:** [#rrggb] The optimum range color. Default is "#44ff44".
- **warningRangeColor:** [#rrggb] The warning range color. Default is "#ffff00".
- **actionRangeColor:** [#rrggb] The action range color. Default is "#ff4444".

bmi-gauge.html

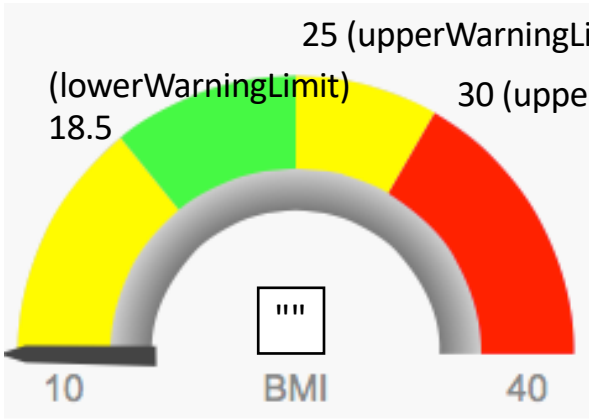
```
55 <body>
56 <h1>Javascript SVG/BMI</h1>
57 <div>
58 <form action="" method="GET">
59 Weight (kg/lbs): <input type="text" name="weight" size=5>
60 <br>
61 Height (cm/ins): <input type="text" name="height" size=5>
62 <br>
63 <input type="radio" name="units" value="kg" checked>kg-cm
64 <input type="radio" name="units" value="lbs">lbs-ins
65 <br>
66 <input type="button" name="button" Value="Click" onClick="computeBMI(this)">
67 </form>
68 </div>
69 <div style="width:100px;color:red" id="output"></div>
70 <div id="gauge-div" style="width: 250px; height: 200px"></div>
71 <div>
72 <table>
73 <tr><th>Color</th><th>Range</th></tr>
74 <tr>
75 <td class="color" style="background-color: #eeee00"></td><td>Underweight</td>
76 </tr>
77 <tr>
78 <td class="color" style="background-color: #44ff44"></td><td>Normal</td>
79 </tr>
80 <tr>
81 <td class="color" style="background-color: #ff8800"></td><td>Overweight</td>
82 </tr>
83 <tr>
84 <td class="color" style="background-color: #ff0000"></td><td>Obese</td>
85 </tr>
86 </div>
87 </body>
```


bmi-gauge.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Javascript/SVG BMI</title>
5 <style>
6 div { display: inline-block }
7 table { border: 1px solid black; border-collapse: collapse }
8 td, th { border: 1px solid black; padding: 5px }
9 .color { width: 30px; height: 30px }
10 </style>
11 <script src="gaugeSVG.js"></script>
12 <script>
13 var gauge;
14 window.onload = function(){
15     gauge = new GaugeSVG({id: "gauge-div", value: 10,
16         min: 10, max: 40,
17         label: "BMI",
18         lowerWarningLimit: 18.5, upperWarningLimit: 25,
19         warningLowerRangeColor: "#eeee00",
20         warningUpperRangeColor: "#ff8800",
21         actionRangeColor: "#ff0000",
22         upperActionLimit: 30, lowerActionLimit: -1});
23     gauge.gaugeVAL.childNodes[0].textContent = ""; // don't display initially
24 };
```

bmi-gauge.html

```
<script src="gaugeSVG.js"></script>
<script>
var gauge;
window.onload = function(){
  gauge = new GaugeSVG({fid: "gauge-div", value: 10,
    min: 10, max: 40,
    label: "BMI",
    lowerWarningLimit: 18.5, upperWarningLimit: 25,
    warningLowerRangeColor: "#eeee00",
    warningUpperRangeColor: "#ff8800",
    actionRangeColor: "#ff0000",
    upperActionLimit: 30, lowerActionLimit: -1});
  gauge.gaugeVAL.childNodes[0].textContent = ""; // don't display initially
};
```



To set the value:
gauge.refresh(bmi, true);

animation
true|false

bmi-gauge.html

```
25 function validNumber(x) {
26     return /^[0-9\.]+$/.test(x);
27 //     return ((x | 0) > 0 && x % 1 == 0); // convert to 32-bit integer and no fractional part
28 }
29 function computeBMI(e) {
30     var weight = e.form.weight.value;
31     var height = e.form.height.value;
32     var o = document.getElementById("output");
33     o.innerHTML = "";
34
35     if (validNumber(weight) && validNumber(height)) {
36         var scale = e.form.units[0].checked ? 10000 : 703; // kg-cm
37         var bmi = weight * scale / (height * height);
38         var range;
39         if (bmi < 18.5) {
40             range = "underweight"
41         } else if (bmi < 25) {
42             range = "normal"
43         } else if (bmi < 30) {
44             range = "overweight"
45         } else {
46             range = "obese"
47         }
48         gauge.refresh(bmi.toFixed(2), true)
49     } else {
50         o.innerHTML = "Error: height and weight must be positive numbers"
51     }
52 }
```