

## BMI Calculator

default.js

```

1. // bmiRanges array stores the data of the BMI classes
2. var bmiRanges = [
3.   { min: 0, max: 18.4, label: 'Underweight', color: Color.blue },
4.   { min: 18.5, max: 24.9, label: 'Healthy Weight', color: Color.green },
5.   { min: 25, max: 29.9, label: 'Overweight', color: Color.red },
6.   { min: 30, max: Infinity, label: 'Obesity', color: Color.red }
7. ];
8.
9. /*Function start() calls userInfo() which collects information
10. aswell as calls table() which makes the design of the chart*/
11. function start() {
12.   userInfo();
13.   table();
14. }
15.
16. /*Function userInfo() gathers the information needed for the calculations and
17. runs it by storing the calculations into variables. It calls on array bmiRanges
18. to figure out the group the BMI belongs to. Ie Healthy Weight / Underweight*/
19. function userInfo() {
20.   var height = readFloat("What is your height in cm? ");
21.   var weight = readFloat("What is your weight in kg? ");
22.   var m2 = (height / 100) ** 2;
23.
24.   var BMI = Math.trunc(weight / m2);
25.
26.   insertText("BMI = " + BMI, "20pt Arial", Color.black, (170 + 140) / 2, 375);
27.
28.   pointer(BMI);
29.
30.   for (var i = 0; i < bmiRanges.length; i++) {
31.     if (BMI >= bmiRanges[i].min && BMI <= bmiRanges[i].max) {
32.       var group = bmiRanges[i].label;
33.       var color = bmiRanges[i].color;
34.       insertText(group, "20pt Arial", color, 130, 75);
35.       break;
36.     }
37.   }
38. }
39.
40. /*Function table() makes the designs and inputs what is needed.*/
41. function table() {
42.   drawRect(140, 75, Color.blue, 0, 200);
43.   drawRect(140, 75, Color.green, 140, 200);
44.   drawRect(140, 75, Color.red, 270, 200);
45.
46.   insertText("Underweight", "10pt Arial", Color.black, 30, 240);
47.   insertText("Healthy Weight", "10pt Arial", Color.black, 160, 240);
48.   insertText("Overweight/Obesity", "10pt Arial", Color.black, 275, 240);
49. }
50. }
51.
52. /*Function pointer() uses parameter BMI so that the variable could be exported
53. to this function. It then makes variables for coordinates to find out where the
54. pointer needs to be.*/
55. function pointer(BMI) {
56.   var startX = 0;
57.   var endX = 410;
58.   var y = 190;
59.
60.   //Calculate the percentage of BMI progress across the ranges
61.   var progressPercentage;
62.   if (BMI < 18.5) {
63.     progressPercentage = BMI / 18.5;
64.   } else if (BMI >= 18.5 && BMI <= 24.9) {
65.     progressPercentage = (BMI - 18.5) / (24.9 - 18.5);
66.   } else {
67.     progressPercentage = (BMI - 25) / (29.9 - 25);
68.   }
69.
70.   //Calculate the x-coordinate of the pointer based on the progress percentage
71.   var x;
72.   if (BMI < 18.5) {
73.     x = startX + progressPercentage * (140 - startX);
74.   } else if (BMI >= 18.5 && BMI <= 24.9) {
75.     x = 140 + progressPercentage * (270 - 140);
76.   } else {
77.     x = 270 + progressPercentage * (endX - 270);
78.   }

```

```
79.
80.     var arrow = new Polygon();
81.     arrow.addPoint(x - 10, y - 20);
82.     arrow.addPoint(x, y);
83.     arrow.addPoint(x + 10, y - 20);
84.     arrow.setColor(Color.black);
85.     add(arrow);
86. }
87.
88. /*Function insertText() prints out the text to show it on the screen; it uses
89. parameters to simplify the program and make it more efficient and understandable */
90. function insertText(label, font, color, x, y) {
91.     var txt = new Text(label, font);
92.     txt.setColor(color);
93.     txt.setPosition(x, y);
94.     add(txt);
95. }
96.
97. /*Function drawRect() uses parameters aswell so that when the function is called
98. the desired measurments and inputed easily.*/
99. function drawRect(width, height, color, x, y) {
100.     var rect = new Rectangle(width, height);
101.     rect.setColor(color);
102.     rect.setPosition(x, y);
103.     add(rect);
104. }
```