UD5. Activity 2

MULTIMEDIA CONTENT IMPLEMENTATION

Toni Barceló García

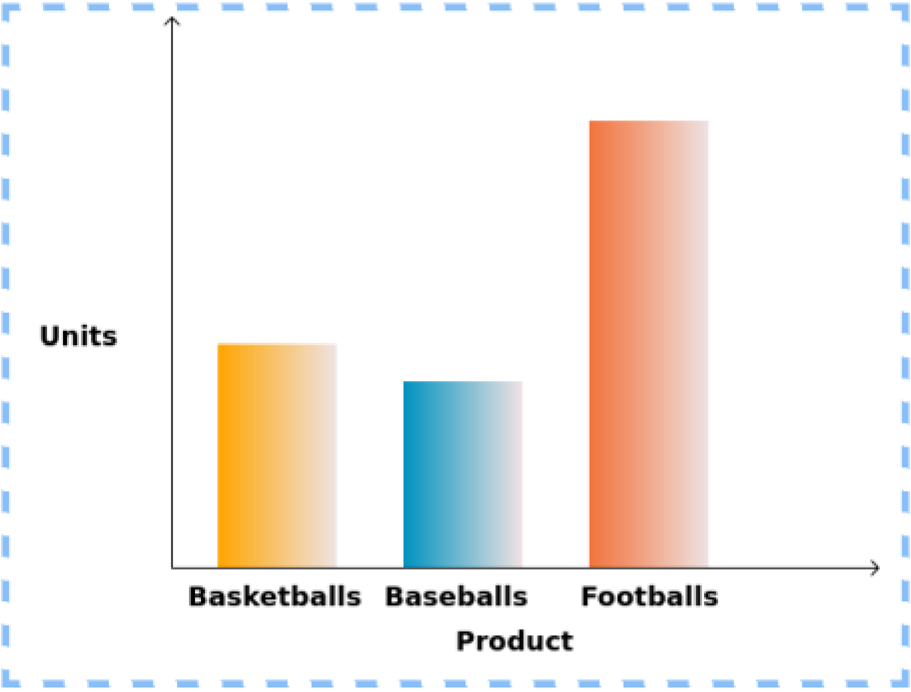
IES JOAN RAMIS I RAMIS | DAW

Content Table

UD5. Activity 2

Use an HTML5 Canvas in order to:

- Paint a graphic chart similar to this one:



- The background colour of each bar has to be a done with a gradient.
- Your data must follow this structure in Javascript (JSON), in order to achieve the result:

```
var sales = [{
        product: "Basketballs",
        units: 150
},      {
        product: "Baseballs",
        units: 125
},      {
        product: "Footballs",
        units: 300
```

Source Github project

If you clone the project, you will have all the activities I have done so far sorted by folders in “activities” root directory.

Source project:

https://github.com/abarcelogarcia/abarcelogarcia.github.io

Root folder for this activity:

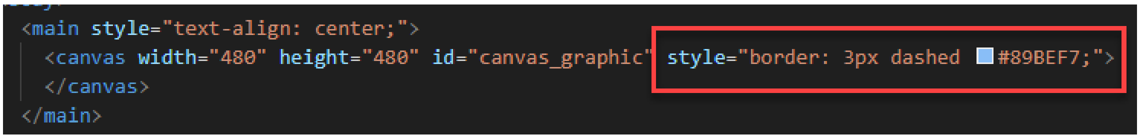
activities/ UD5A_2_Canvas

The website structure

Directory	Files	Concept
root	index.html	main website file.
	index.css	css stylesheet file
	index.js	JS file

CANVAS

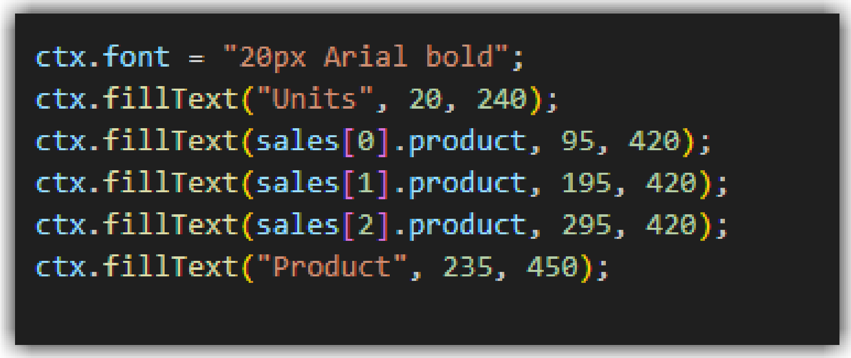
I have created the canvas and added the style of its outline



```
<main style="text-align: center;">
  <canvas width="480" height="480" id="canvas_graphic" style="border: 3px dashed #89BEF7;">
  </canvas>
</main>
```

Titles

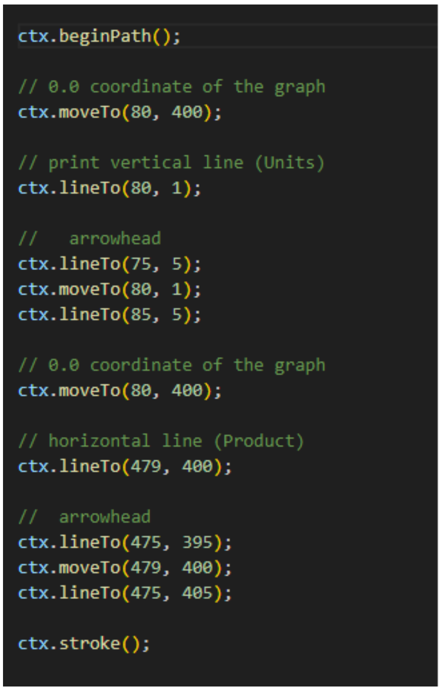
Next, I have created the titles of the coordinates and the values



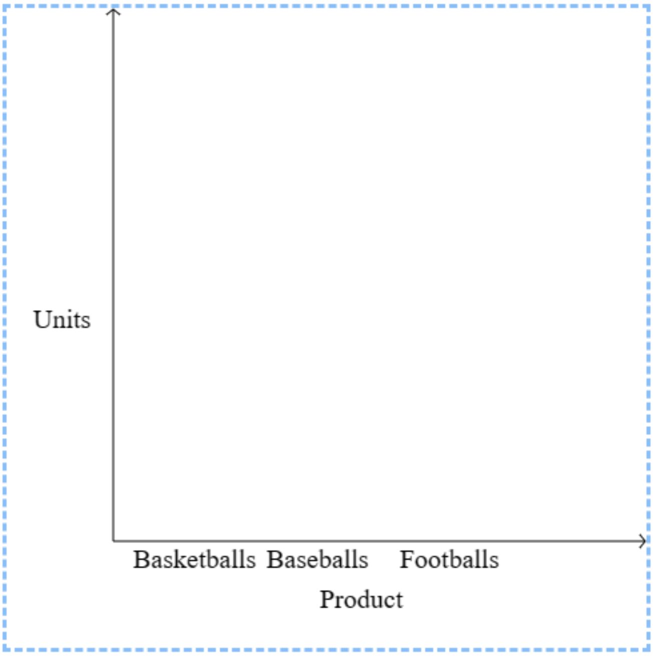
```
ctx.font = "20px Arial bold";
ctx.fillText("Units", 20, 240);
ctx.fillText(sales[0].product, 95, 420);
ctx.fillText(sales[1].product, 195, 420);
ctx.fillText(sales[2].product, 295, 420);
ctx.fillText("Product", 235, 450);
```

Axis

The next step is to create the coordinate axis, starting with the vertical.

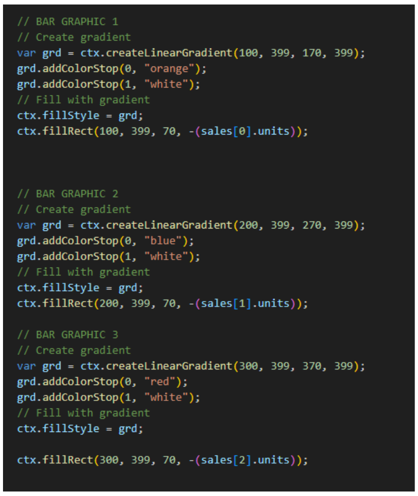


```
ctx.beginPath();

// 0.0 coordinate of the graph
ctx.moveTo(80, 400);

// print vertical line (Units)
ctx.lineTo(80, 1);

//   arrowhead
ctx.lineTo(75, 5);
ctx.moveTo(80, 1);
ctx.lineTo(85, 5);

// 0.0 coordinate of the graph
ctx.moveTo(80, 400);

// horizontal line (Product)
ctx.lineTo(479, 400);

//  arrowhead
ctx.lineTo(475, 395);
ctx.moveTo(479, 400);
ctx.lineTo(475, 405);

ctx.stroke();
```

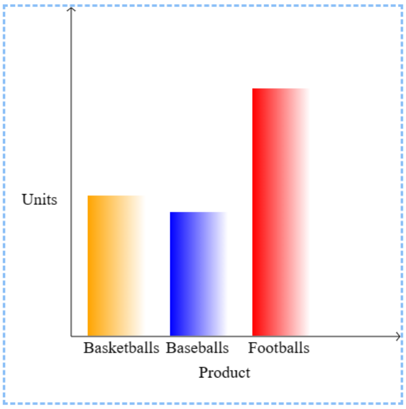


Bars

Finally, I add the data bars.

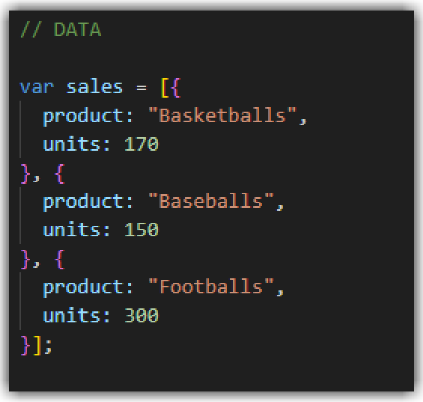


```
// BAR GRAPHIC 1
// Create gradient
var grd = ctx.createLinearGradient(100, 399, 170, 399);
grd.addColorStop(0, "orange");
grd.addColorStop(1, "white");
// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(100, 399, 70, -(sales[0].units));


// BAR GRAPHIC 2
// Create gradient
var grd = ctx.createLinearGradient(200, 399, 270, 399);
grd.addColorStop(0, "blue");
grd.addColorStop(1, "white");
// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(200, 399, 70, -(sales[1].units));

// BAR GRAPHIC 3
// Create gradient
var grd = ctx.createLinearGradient(300, 399, 370, 399);
grd.addColorStop(0, "red");
grd.addColorStop(1, "white");
// Fill with gradient
ctx.fillStyle = grd;

ctx.fillRect(300, 399, 70, -(sales[2].units));
```

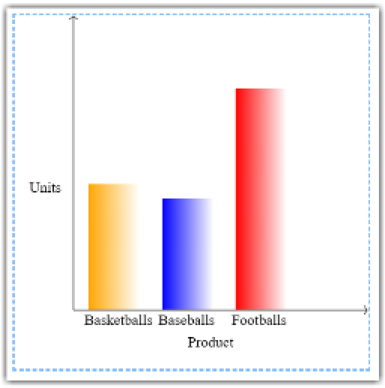


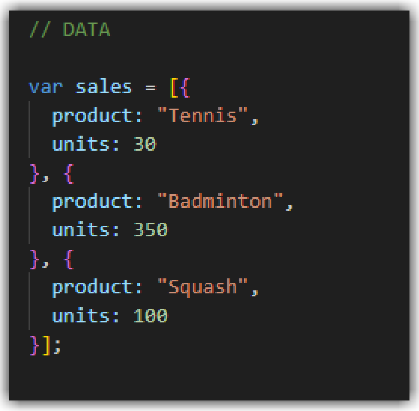
Json

For the chart data I have used the json structure provided by variables that allows to display the chart dynamically by modifying the values of the structure.

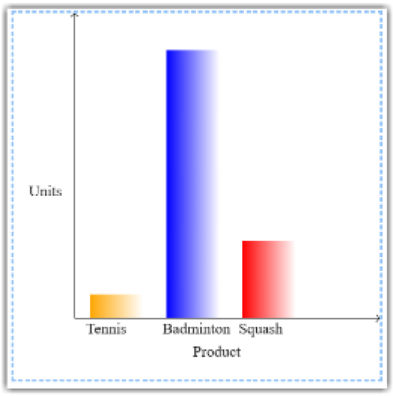


```
// DATA

var sales = [{
  product: "Basketballs",
  units: 170
}, {
  product: "Baseballs",
  units: 150
}, {
  product: "Footballs",
  units: 300
}];
```





```
// DATA

var sales = [{
  product: "Tennis",
  units: 30
}, {
  product: "Badminton",
  units: 350
}, {
  product: "Squash",
  units: 100
}];
```


Units
Tennis
Badminton
Squash
Product