```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="Data Visualisation">
    <meta name="keywords" content="HTML, CSS, D3">
    <meta name="author" content="Dhanveer Ramnauth">
    <meta name="description" content="Data Visualisation">

    <title>Task 2.2 D3 Data Binding</title>
    <script src="https://cdn.jsdelivr.net/npm/d3@7"></script>
    <link href="css/style.css" rel="stylesheet">
</head>

<body>
    <h1>The D3 Journey starts here...</h1>

    <script>
        //dataset
        let dataset = [14,  5, 26, 23,  9,
                        1,  2,  3,  4,  5,
                        6,  7,  8,  9, 10,
                       11, 12, 13, 14, 15,
                       16, 17, 18, 19, 20,
                       21, 22, 23, 24, 25,
                       26, 27, 28, 29, 30];
        //canvas size
        let w = 500;
        let h = 200;

        //ratio between width of canvas and length of the dataset
        //used to compute element spacing
        let ratio = (w / dataset.length);

        //define svg canvas
        let svg = d3.select("body")      // in body element
                    .append("svg")       //add the svg
                    .attr("width", w)    //set width attribute
                    .attr("height", h); //set height attribute

        //spacing between the bars (by modifying width)
        let padding = 1; // size between bar
        let height_multiplier = 4; //make bar big

        svg.selectAll("rect")
            .data(dataset) //bind data
            .enter() //creates a new placeholder for each bit of data
            .append("rect") //add svg rect
            .attr("x", function (d, i) { //X coord
                return i * ratio; // w / dataset.length - norm
            })
            .attr("y", function (d) { //Y coord
                //we need to set the y value to the top of the bar so its not upside down
                return h - d * height_multiplier; //d*height_multiplier is bar height
            })
            .attr("width", ratio - padding) //ratio is the maximum element size and padding is gap
            .attr("height", function (d) {
                return d * height_multiplier; //multiply the current data with a multiplier
            })
            .attr("fill", function(d, i)
            {
                    //https://www.learnui.design/tools/data-color-picker.html
                    let color_palette = [
                        [  0,  63,  92],
                        [ 47,  75, 124],

                        [102,  81, 145],
                        [160,  81, 149],
                        [212,  80, 135],
                        [249,  93, 106],
                        [255, 124,  67],
                        [255, 166,   0]
                    ];


                    //----------------INDEXING-------------------------------

                    //Max value is 30
                    //color pallete length is 8
```

```
                    /*
                        color_index_ratio calculation:

                        30 / 8 =  3.75
                        3.75 * 2 = 7.5;
                        floor(7.5) = 7;

                        7 is max index value so we are safe
                    */

                    let color_index_ratio = Math.floor( ( d / color_palette.length ) * 2 ); // BAD WAY TO DO THIS
                    let element = color_palette[color_index_ratio]; // slightly cleaner than inlining this.

                    //-----------------------------------------------------


                    return `rgb(${element[0]}, ${element[1]}, ${element[2]})`;
            });

    </script>

    <br>

    <footer>
        Data Visualisation <br>Dhanveer Ramnauth
    </footer>

</body>

</html>
```