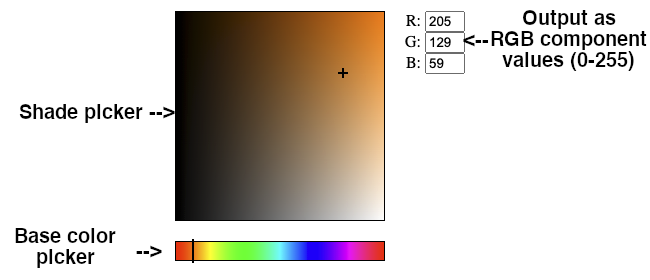
# Coding assignment - Color Picker

| Time needed | approximately 2 hours |
| --- | --- |
| Technology to use | TS or ES5+ |



## App description

Prepare a simple color picker that looks like the one in the picture above. There are two main components that the user can interact with - the bottom bar and the squared shade picker.

The bottom bar selects so called base color. This is the base color that in the shade picker is represented by the top right corner. The colors on the bar go left to right in this order:

red (#f00), yellow (#ff0), green (#0f0), aqua (#0ff), blue (#00f), magenta (#f0f) and back to red (#f00). The currently selected base color is represented by the vertical bar..

The shade picker has two axes:

* The vertical axis represents the transition to white with linear interpolation - base color at the top, white color at the bottom.
* The horizontal axis represents the transition to black with linear interpolation - base color on the right, black color on the left.

The currently selected shade is represented by a cross pointer.

Represent the final color by its components (red, green and blue) and their corresponding value between 0 and 255. Display them to the right of the picker as is shown on the picture.

Changing the base color does not reset the position of the pointer in the shade picker.

## Implementation description

You can use either Typescript or any Ecmascript version as you like. We also leave it up to you whether you use any bundling technology (e.g. webpack, parcel, etc.) or just a plain unbundled app.

We also allow the use of frameworks and libraries, should you choose to utilize them. You are not allowed to use any libraries for color picking.

The app should work with the current version of Chrome.

To send us the assignment either share your repository with the list of people provided to you by HR. Or if you choose not to utilize a repository, send us the source code compressed as a zip file.

## What will be evaluated

The evaluation will focus on the functionality of the app and the formal code quality. By this we mean code separation, used design pattern, consistency in style etc. The performance of the code (How fast it is evaluated and executed by the browser.) and the time it took you to implement it are not important.

## Bonus tasks

If you finish the main task and you are willing to spend some more time with the app, you can complete one or both of these bonus tasks:

* Implement a copy to clipboard button that stores styling rgb string in the user’s clipboard. (e.g. “rgb(234, 12, 68)”)
* Implement a reverse functionality where changing the values in the input fields sets the corresponding base color and pointer location in the shade picker.