JAVASCRIPT | DAY 4 | CLASSWORK

Exercise 1

Remember the other day’s exercise where you had to create a function about providing information about the weather. Now if you have finished it successfully it is time to upgrade it to the version 2.0.

You should create a nice looking responsive front-end for your program (mobile friendly) and introduce more detailed weather descriptions. (e.g. If the temperature is above 32°C, the program could say, "The weather is hot." etc). Moreover you also need to display a proper image for the current weather condition.

Exercise 2

Create a circle and center it on the page.

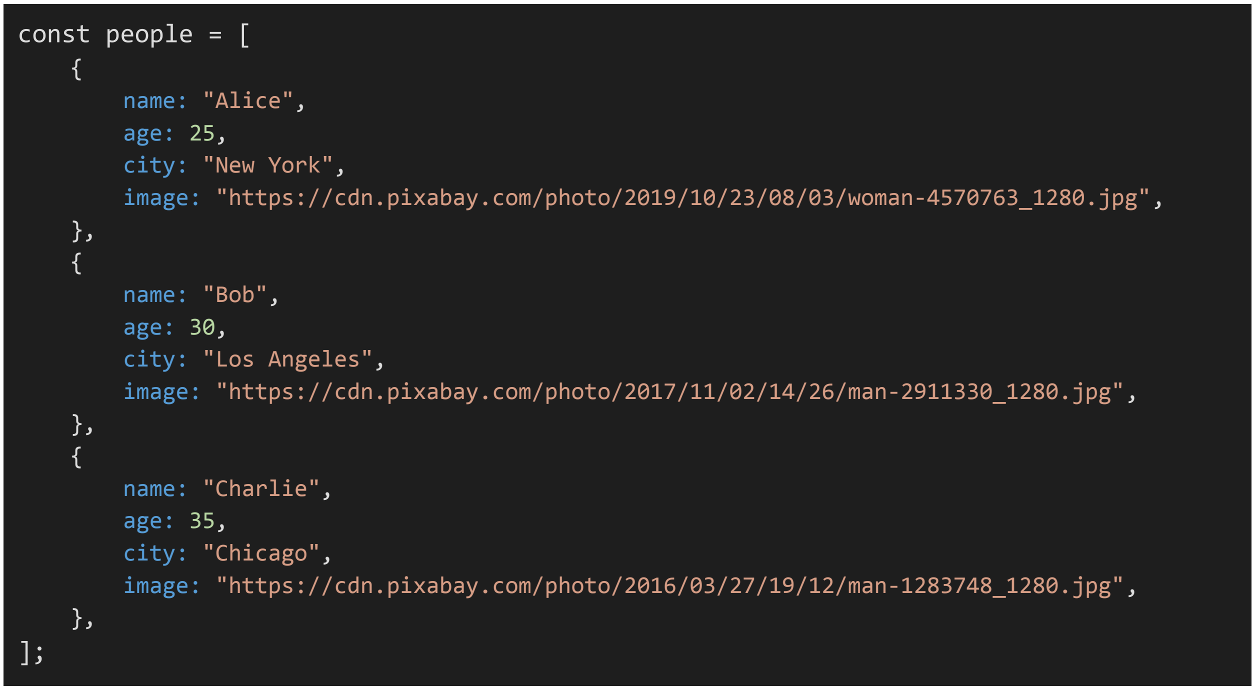
This circle will respond to different mouse events by displaying messages and changing the background color.

- **Hover:** Display a message below the circle that reads "You are in the circle" when the mouse is over it.
- **Mouse Leave:** Show a message that says "You are outside of the circle" when the mouse leaves the circle

- **Click:** Change the circle’s background color to gray upon a single mouse click on the circle.
- **Double-Click:** Change the circle’s background color to blue when you double-click on the circle.

Exercise 3

Create a program that defines an array of objects, where each object represents a person with properties such as name, age, and city.



```
const people = [
    {
        name: "Alice",
        age: 25,
        city: "New York",
        image: "https://cdn.pixabay.com/photo/2019/10/23/08/03/woman-4570763_1280.jpg",
    },
    {
        name: "Bob",
        age: 30,
        city: "Los Angeles",
        image: "https://cdn.pixabay.com/photo/2017/11/02/14/26/man-2911330_1280.jpg",
    },
    {
        name: "Charlie",
        age: 35,
        city: "Chicago",
        image: "https://cdn.pixabay.com/photo/2016/03/27/19/12/man-1283748_1280.jpg",
    },
];
```

Use a for loop to iterate through the array, and for each person, create a Bootstrap card dynamically using JavaScript to display their details (Name, Age, City). Insert these cards into a specific HTML element, such as a <div> with an ID like #result

If the person's age is above 30, display an additional note inside the card, such as "This person is above 30 years old."

Exercise 4

Create a JavaScript function to create random background colors each time we click a button that we had on the website.

Hint:

- Consider the RGB format for color values
- Consider the Global DOM built-in properties and methods

Exercise 5

To solve this task, you will need two arrays: students and grades. In students, store the names of students (e.g., ["John", "Jane"]). Correspondingly, in gades, store their grades (ranging from 0 to 100) at the same indexes as their names. For example:

Students = [“John”, “Jane”]

MathGrades = [70, 85]

Next, use a variable to store a student's name. This name should be compared against the names in the Students array. When a match is found, display the student's name and their math grade. The output should be formatted nicely, like so:
"John has reached 70 points in Math this season."

Furthermore the grades should also be checked and change the color based on the following conditions

- Less than 60: Red
- Between 60 and 70: Yellow
- Over 70: Green
- Exactly 100: Blue

Hint: Use the prompt() function to create an alert window where the user can enter a name. This name can then be stored in a variable. For example:

```
let result = prompt("type your name");
```

Challenge

Exercise 6

Many people love games of chance and especially the thrill of "roll of the dice". For this exercise, create a page with a two-sided layout for two players based on the idea of "roll of the dice".

When the page loads, a prompt method should appear to receive a string (the name for each player), then display their names in the header of the page, for example, "Thomas vs. Martin".

In each half of the layout, place a button that calls a function. This function should create one random number. Next, display the number separately and visually appealing in each player's half, so that the player can see what they "rolled". Once both players have rolled, announce the winner based on the highest score! Do your best to create a rich game-design using CSS or Bootstrap.

Consider the following programming principles while solving the Challenge:

- Use of best practices in semantic elements, code indentation and comments
- Give your page a responsive layout with at least two breakpoints
- Correct use of attributes with functions
- Declaring variables as global or local, when appropriate
- Best practices in naming variables
- Manipulating the DOM with methods like .getElementById() and properties like .innerHTML

CodeFactory
Vienna
- We create developers! -