**Questions**

**HTML CSS:**

1. Give an explanation on these tags and what they do?
   1. <div>

The <div> tag defines a division or a section in an HTML document.

The <div> element is often used as a container for other HTML elements to style them with CSS or to perform certain tasks with JavaScript.

* 1. <ul>

The <ul> tag defines an unordered (bulleted) list.

Use the <ul> tag together with the <li> tag to create unordered lists.

* 1. <td>

The <td> tag defines a standard cell in an HTML table.

* 1. <thead>

The <thead> tag is used to group header content in an HTML table.

The <thead> tag must be used in the following context: As a child of a <table> element, after any <caption>, and <colgroup> elements, and before any <tbody>, <tfoot>, and <tr> elements.

The thead, tbody, and tfoot elements will not affect the layout of the table by default. However, you can use CSS to style these elements.

* 1. <section>

The <section> tag defines sections in a document, such as chapters, headers, footers, or any other sections of the document.

* 1. <img>

The <img> tag defines an image in an HTML page.

* 1. <form>

The <form> tag is used to create an HTML form for user input.

* 1. <label>

The <label> tag defines a label for an <input> element.

* 1. <input>

The <input> tag specifies an input field where the user can enter data.

* 1. <option>

The <option> tag defines an option in a select list.

<option> elements go inside a <select> or <datalist> element.

1. Describe the following common CSS units of length:
   1. Cm

absolute units, 1cm should be exactly 1 centimeter

* 1. Em

relative units, depend on the font and may be different for each element in the document. equal to the computed value of the font-size property

* 1. In

absolute units, inch, not recommend

* 1. Mm

absolute units, mm, not recommend

* 1. Px

absolute units, pixel units – 1px is equal to 0.75pt.

* 1. Vh

relative units, 1/100th of the window's height.

* 1. Vw

relative units, 1/100th of the window's width.

* 1. Rem

relative units, The rem (for “root em”) is the font size of the root element of the document. Unlike the em, which may be different for each element, the rem is constant throughout the document.

* 1. %

relative units, percentage of their parent.

**JavaScript, AJAX, JQuery**

1. What is the use of the ‘this’ keyword?

The JavaScript this keyword refers to the object it belongs to.

1. What is a function?

A JavaScript function is a block of code designed to perform a particular task.

A JavaScript function is executed when "something" invokes it (calls it).

1. Create a function that takes height in cm and weight in kg as parameters, it should return the BMI using the two inputs present.

(weight in kg)

(height in meters) \* (height in meters)

<https://en.wikipedia.org/wiki/Body_mass_index>

function BMI(cm,kg) {

var result = kg/(cm/100\*cm/100)

return result

}

console.log(BMI(171,90));

**6)**

Accessing objects in JS

1. **var** car **=** {
2. brands: {
3. BMW: {
4. 6-series: {
5. sold: 1805
6. }
7. },
8. Tesla: {
9. Model-S: {
10. sold: 200
11. },
12. "Model-A": {
13. sold: 14
14. },
15. }
16. }
17. }

Access the number of 6-series cars sold

car.brands.BMW["6-series"]

Access the number of Model-A cars sold

car.brands.Tesla["Model-A"]

**7)**

**var** restaurants **=** [

{name: "Ollies", group: "Castelo Concepts"},

{name: "Wagyu", group: "Castelo Concepts"},

{name: "Zaks", group: "Castelo Concepts"},

{name: "Black Salt", group: "Black Sheep"},

{name: "Salt and Barrel", group: "Black Sheep"}

]

Return an array of just the restaurants in the group Castelo Concepts

Create an array all the names of the restaurants

var restaurants = [

{name: "Ollies", group: "Castelo Concepts"},

{name: "Wagyu", group: "Castelo Concepts"},

{name: "Zaks", group: "Castelo Concepts"},

{name: "Black Salt", group: "Black Sheep"},

{name: "Salt and Barrel", group: "Black Sheep"}

]

var result = restaurants.filter(function(restaurant) {

return restaurant.group === "Castelo Concepts";

});

var nameonly = result.map(function(n) {

return n.name;

});

console.log(nameonly);

**8)**

Using Getters and Setters create a class called Pet it should take a name and an age

Three methods should be associated with the class, Age() = will return the age of the pet

Age(age) 🡪 sets the age of the animal (resets to the value given)

Info() returns the most updated age and name of the pet in a string.

Initialise a new pet ensure these are the outputs.

Outputs: pet.Age // “I’m 4 year’s old”

pet.Age = 14;

pet.Age // “I’m 14 year’s old”

pet.Info // “My name is {given name}, I am 14 years old”

class Pet {

constructor({age,name}) {

this.age = age;

this.name = name;

}

get Age() {

return `I’m ${this.age} year’s old.`

}

set Age(n) {

this.age = n;

}

info () {

return `My name is ${this.name} and I'm ${this.age} year's old.`;

}

}

const pet = new Pet({age: 44, name: 'Tom'});

console.log(pet.Age);

pet.Age = 14;

console.log(pet.Age);

console.log(pet.info());

**9)**

Construct a button in an html file, when pressed, the button calls this api: <https://randomuser.me/api/>, every time a user presses the button you should add the random user’s name to the html page using JQuery.

**10)**

Construct a button in an html file, when pressed, the button calls this api: <https://randomuser.me/api/>, get information for two users when you have this information fire off a second request to <https://api.sunrise-sunset.org/> api, you will need to pass the users latitude and longitude into the api call, once you have the information using JQuery add the users timezone to the html page as well as the Sunrise and Sunset time.