**Questions**

**HTML CSS:**

1. Give an explanation on these tags and what they do?
   1. <div> **division**
   2. <ul> **unordered list**
   3. <td> **table cell**
   4. <thead> **table head**
   5. <section> **section**
   6. <img> **image**
   7. <form> **form**
   8. <label> **label for input**
   9. <input> **input box**
   10. <option> **option list**
2. Describe the following common CSS units of length:
   1. Cm **10mm**
   2. Em **Relative to the font-size of the element**
   3. in **inch**
   4. mm **0.1cm**
   5. px **pixels**
   6. vh **Relative to 1% of the height of the viewport\***
   7. vw **Relative to 1% of the width of the viewport\***
   8. rem **Relative to font-size of the root element**
   9. % **Relative to the parent element**

**JavaScript, AJAX, JQuery**

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

**this keyword refers to an object, that object which is executing the current bit of js code.**

1. What is a function?

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

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>

**var bmiCalulate = (height, weight) => {**

**return height / (weight/100) /(weight/100)}**

**bmiCalulate(52,155) //52kg and 155cm people**

**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

Access the number of Model-A cars sold

**“6-series” & “Model – S” have problem. But I guess the answer is**

console.log(car.brands.BMW['6-series'].sold)

console.log(car.brands.Tesla['Model-A].sold)

**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

**var arr = restaurants.filter(function(restaurants){**

**return restaurants.group == "Castelo Concepts"**

**})**

Create an array all the names of the restaurantsI

**for (i=0; i < restaurants.length; i++){**

**arr2.push(restaurants[i].name);**

**}**

**console.log(arr2)**

**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”

**var pet = {**

**name: 'Tom' ,**

**age: 4,**

**get Info() {**

**return 'My name is ' + this.name + '.I am '+ this.age +' years old'**

**},**

**set Age(age) {**

**this.age = age;**

**},**

**get Age(){**

**return 'I’m ' + this.age + ' year’s old'**

**}**

**}**

**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.

See q9.html

**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.