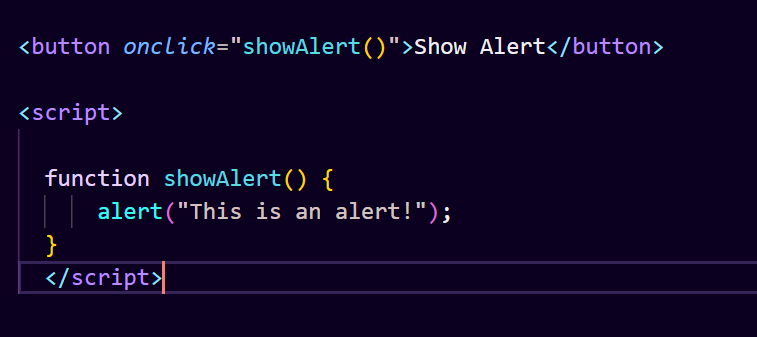
# Advance JavaScript

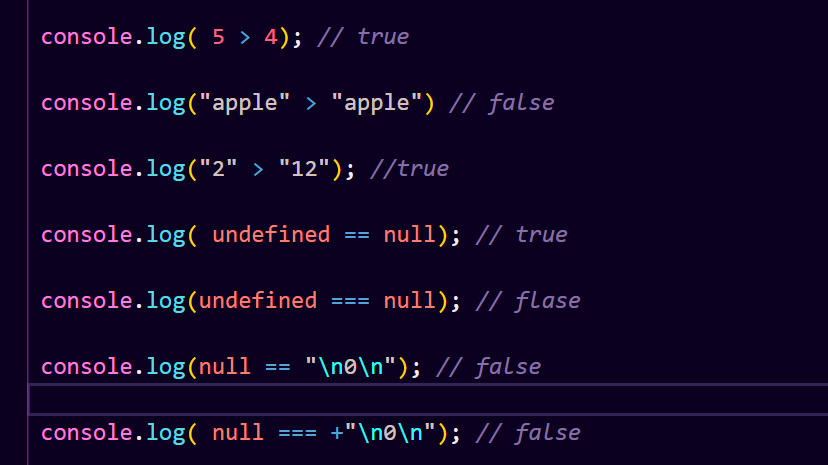
* **Write a program to Show an alert**

**Ans : **

In this HTML file, we have a button that calls the showAlert() function when clicked. The showAlert() function simply displays an alert dialog with the message "This is an alert!".

* **What will be the result for these expressions?**

**Ans :**

****

● **Will alert be shown?**

**Ans : if ("0") { alert( 'Hello'); }**

Yes, the alert will be shown.

* **What is the code below going to output? alert( null || 2 || undefined );**

**Ans :** The code alert( null || 2 || undefined ); will output 2.

This is because the logical OR (||) operator returns the first truthy value it encounters when evaluating from left to right. In this case:

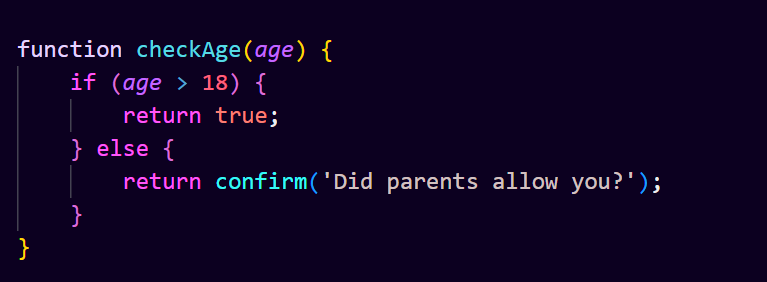
null is falsy.

2 is truthy, so it's returned.

undefined is not reached because 2 was already encountered and returned.

* **The following function returns true if the parameter age is greater than 18. Otherwise it asks for a confirmation and returns its result:**

**Ans :**

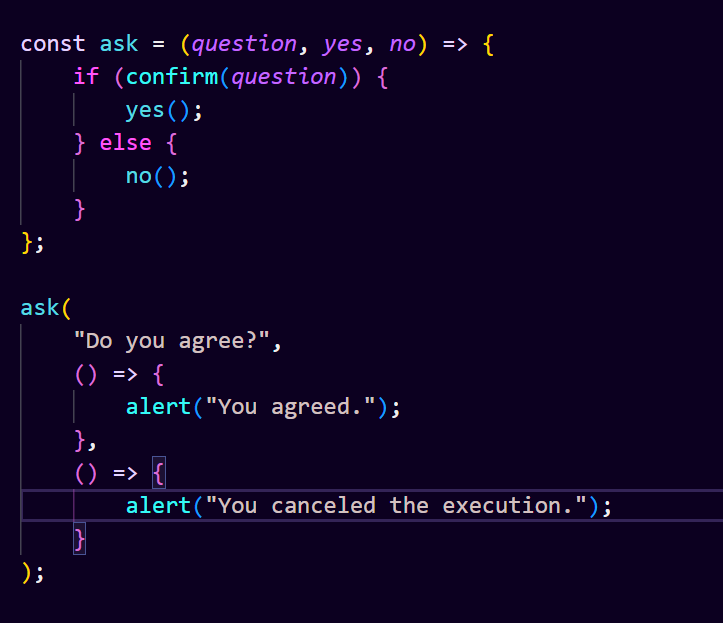
****

If the age parameter is greater than 18, the function immediately returns true.

If the age parameter is 18 or less, the function prompts the user with a confirmation dialog asking if their parents allowed them. The function then returns the result of this confirmation dialog.

* **Replace Function Expressions with arrow functions in the code below: Function**

**Ans :**

****

The ask function is defined using an arrow function syntax.

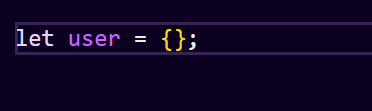
The yes and no callback functions are also defined using arrow function syntax.

# Data Types and Objects

* **Write the code, one line for each action**

1. **Create an empty object user**

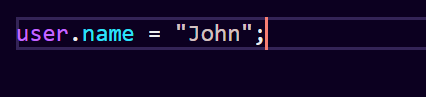
**Ans :**

****

This code declares a variable user and assigns an empty object {} to it. Now, user is an empty object ready to hold properties and methods.

1. **Add the property name with the value John.**

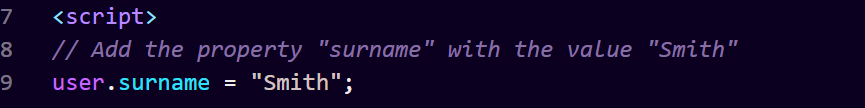
**Ans :**

****

You can add the property "name" with the value "John" to the user object like this:

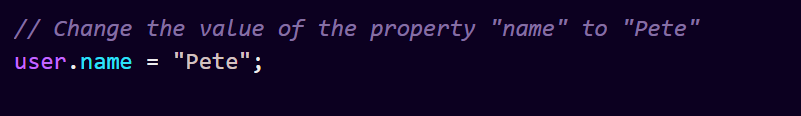
After executing this line, the user object will have a property named "name" with the value "John".

1. **Add the property surname with the value Smith.**

**Ans :** ****

user will have the property "surname" with the value "Smith".

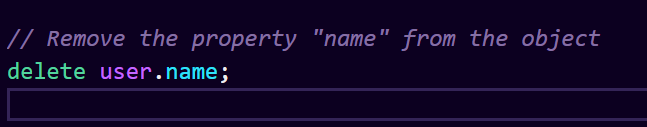
1. **Change the value of the name to Pete.**

**Ans :** ****

The value of the property "name" will be changed to "Pete".

1. **Remove the property name from the object.**

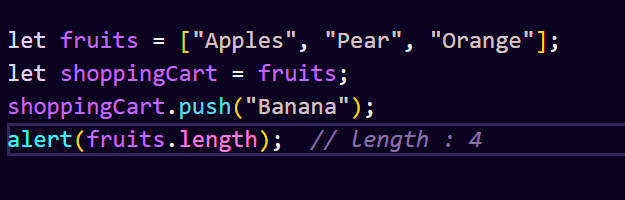
**Ans :**

****

The property "name" will be removed from the user object.

* **Is array copied?**

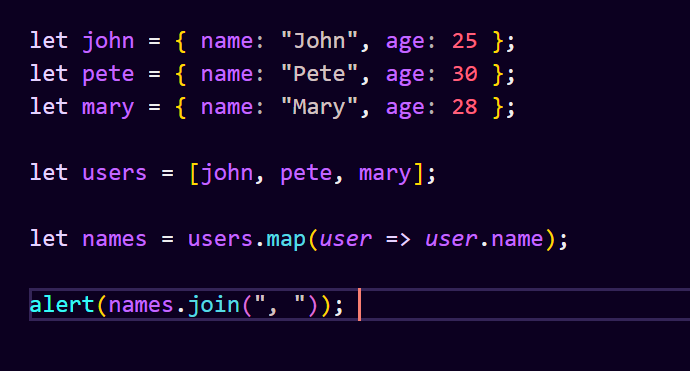
**Ans :**

****

After executing this code, fruits and shoppingCart both reference the same array, which now contains "Apples", "Pear", "Orange", and "Banana". So, the length of fruits array will be 4, as it includes the new item "Banana".

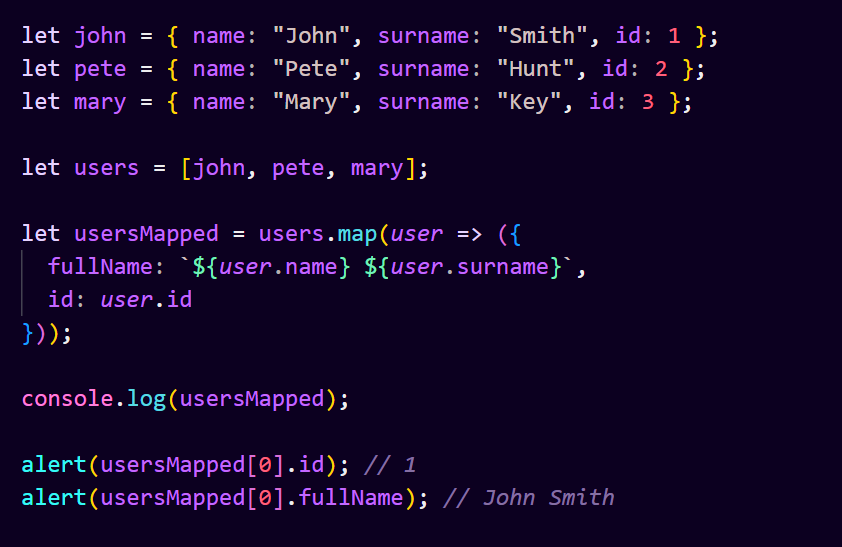
* **Map to names**

**Ans :**

****

* **Map to objects**

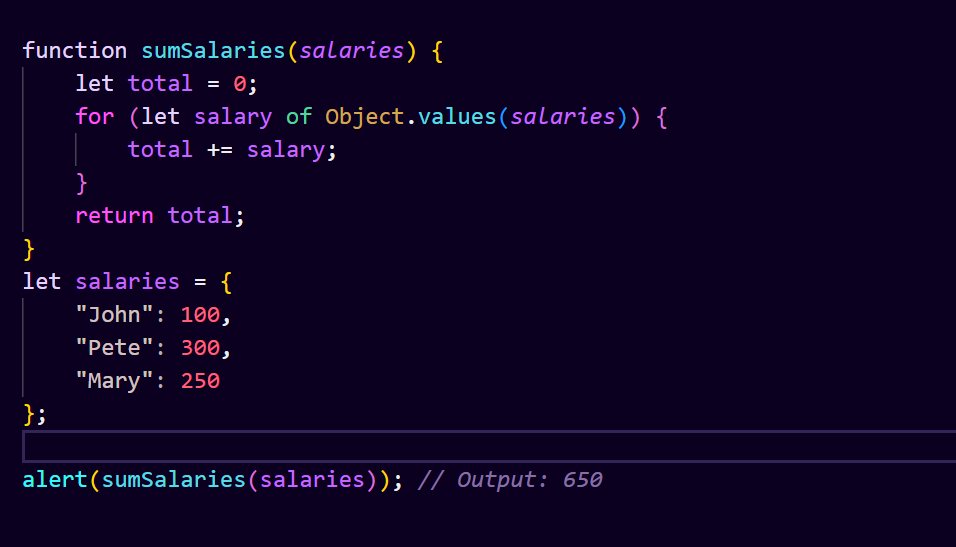
**Ans :**

****

* **Sum the properties There is a salaries object with arbitrary number of salaries. Write the function sumSalaries(salaries) that returns the sum**

**of all salaries using Object.values and the for..of loop.If salaries is empty, then the result must be 0.**

**Ans :**

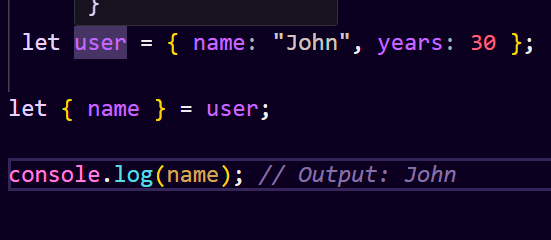
****

* **Destructuring assignment We have an object: Write the Destructuring assignment that reads:**

**Ans :**

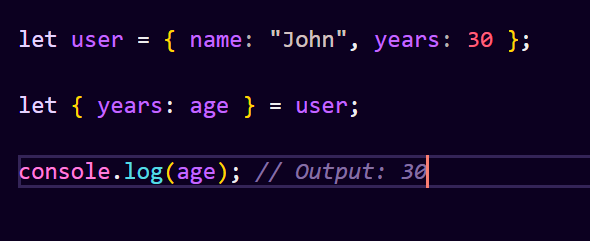
1. **Name property into the variable name.**

**Ans :**

****

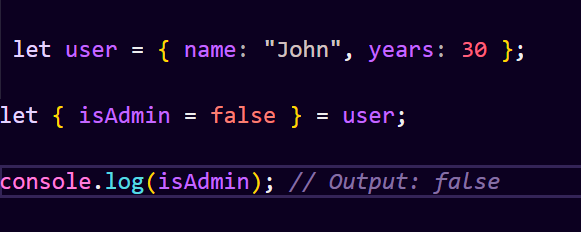
1. **Year’s property into the variable age.**

**Ans :**

****

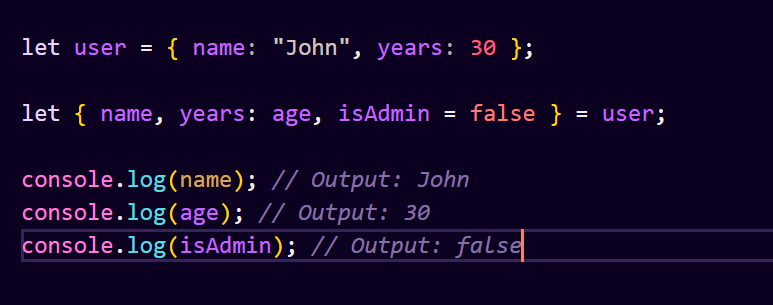
1. **isAdmin property into the variable isAdmin (false, if no such property)**

**Ans :**

****

1. **let user = { name: "John", years: 30};**

**Ans :**

****