

# Assignment (Advance JavaScript)

---

## MODULE: 1 (Introduction and Code Quality)

➤ Write a program to Show an alert

**Ans:**

```
<html>

<head>

    <script type = "text/JavaScript">

        function demo(){

            alert("This is an alert dialog box"); }

    </script>

</head>

</html>
```

➤ What will be the result for these expressions?

**Ans:**

1)  $5 > 4$

**Ans. True**

2) "apple" > "pineapple"

**Ans. False** (Dictionary comparison, hence false. "a" is smaller than "p".)

3) "2" > "12"

**Ans. True** (Again, dictionary comparison, first char "2" is greater than the first char "1".)

4) `undefined == null`

**Ans. True** (Values null and undefined equal each other only.)

5) `undefined === null`

**Ans. False**

6) `null == "\n0\n"`

**Ans. False** (null only equals undefined.)

7) `null === +"\n0\n"`

**Ans. False**

➤ Will alert be shown?

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

**Yes it will.**

Any string except an empty one (and "0" is not empty) becomes true in the logical context.

➤ What is the code below going to output? `Alert(null || 2 || undefined);`

**Ans:** The answer is 2. Because null and undefined refers to empty string.

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

**Ans:**

```

Function checkAge(age){
    if(age>18)
        {return true;}
    else
        { //...return confirm('did parents allow you?') }
}

```

➤ Replace Function Expressions with arrow function in the code

below:

```

function ask(question, yes, no) {
    if (confirm(question)) yes();
    else no();
}

```

```

ask(
    "Do you agree?",
    function () { alert( "You agreed." ); },
    function () { alert( "You canceled the execution." ); }
);

```

Ans:

```

function ask(question, yes, no) {
    if (confirm(question)) yes();
    else no();
}

```

```

ask(
    "Do you agree?",
    () => alert( "You agreed." ),
    () => alert( "You canceled the execution." ) );

```

## **MODULE: 2 (Data Types and Objects)**

➤ Write the code, one line for each action:

**Ans:**

a) Create an empty object user.

**Ans:** `let user = {};`

b) Add the property name with the value John.

**Ans:** `user.name = "John";`

c) Add the property surname with the value Smith.

**Ans:** `user.surname = "Smith";`

d) Change the value of the name to Pete.

**Ans:** `user.name = "pete";`

e) Remove the property name from the object.

**Ans:** `delete user.name;`

➤ Is array copied?

```
let fruits = ["Apples", "Pear", "Orange"]; // push a new value  
into the "copy" let shoppingCart = fruits;  
shoppingCart.push("Banana"); // what's in fruits?  
alert( fruits.length ); // ?
```

**Ans:** Yes array is copied.

```
Let fruits = [“Apple”, “Pear”, “Orange”];
```

```
Let shoppingCart = fruits;
```

```
shopingCart.push(“Banana”);
```

```
console.log(shoppingCart);
```

```
alert(fruits.length); // ans:- fruits length is 4.
```

➤ Map to names

```
let john = { name: "John", age: 25 };
```

```
let pete = { name: "Pete", age: 30 };
```

```
let mary = { name: "Mary", age: 28 };
```

```
let users = [ john, pete, mary ];
```

```
let names = /* ... your code */
```

```
alert( names ); // John, Pete, Mary
```

**Ans:**

```
Let john = { name: “john”, age: 25 };
```

```
Let pete = { name: “pete”, age: 22 };
```

```
Let mary = { name: “mary”, age: 18 };
```

```
Let users = [john, pete, mary];
```

```
Let names = users.map(val => val.name);
```

```
alert(names);
```

### ➤ Map to objects

```
let john = { name: "John", surname: "Smith", id: 1 };  
let pete = { name: "Pete", surname: "Hunt", id: 2 };  
let mary = { name: "Mary", surname: "Key", id: 3 };  
let users = [ john, pete, mary ];  
let usersMapped = /* ... your code ... */  
  
/* usersMapped = [  
  { fullName: "John Smith", id: 1 },  
  { fullName: "Pete Hunt", id: 2 },  
  { fullName: "Mary Key", id: 3 }  
] */  
alert( usersMapped[0].id ) // 1  
  
alert( usersMapped[0].fullName ) // John Smith
```

### Ans:

```
Let john = { name: "john", age: 25 };  
Let pete = { name: "pete", age: 22 };  
Let mary = { name: "mary", age: 18 };  
Let users = [john, pete, mary];  
  
Let usersMapped = users.map(user => ({  
  fullName: `${user.name} ${user.surname}`,  
  id: user.id  
}));  
  
alert(usersMapped[0].id );  
alert(usersMapped[0].fullname );
```

➤ Sum the properties There is a salaries object with arbitrary number of salaries. Write the function sumSalaries 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.

```
Let salaries = {  
  "john": 100,  
  "pete": 300,  
  "mary": 250  
};  
  
Alert( sumSalaries(salaries) ); //650
```

**Ans:**

```
Let salaries = {  
  "john": 100,  
  "pete": 300,  
  "mary": 250 };  
  
Function sumSalaries(salaries) {  
  Let sum = 0;  
  for (let salary of Object.values (salaries)) {  
    sum += salary; }  
  return sum;  
}  
  
alert( sumSalaries(salaries) );
```

➤ Destructuring assignment we have an object: write the Destructuring assignment that reads:

- a) Name property into the variable name.
- b) Year's property into the variable age.

c) isAdmin property into the variable isAdmin (false, if no such property)

d) let user = {name: "John", years:30};

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

let {name, years: age, isAdmin = false} = user;

alert ( name ); //John

alert (age); //30

alert (isAdmin); //false

➤ Turn the object into JSON and back Turn the user into JSON and then read it back into another variable.

user = {name: "John Smith", age: 35 };

**Ans:** let user = {  
    name: "John Smith",  
    age: 35  
};

Let usernew = JSON.parse(JSON.stringify(user));



## **MODULE: 3 (Document, Event and Controls)**

➤ Create a program to hide/show the password

**Ans:**

[https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/Hide\\_Show\\_Password.html](https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/Hide_Show_Password.html)

➤ Create a program that will select all the classes and loop over and whenever I click the button the alert should show

**Ans:**

[https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/Click\\_btn\\_alertbox.html](https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/Click_btn_alertbox.html)

➤ Create a responsive header using proper JavaScript

**Ans:**

[https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/Responsive\\_Header\\_withJS.html](https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/Responsive_Header_withJS.html)

➤ Create a form and validate using JavaScript

**Ans:**

[https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/form\\_validate\\_func.html](https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/form_validate_func.html)

- Create a modal box using css and Js with three buttons

**Ans:**

[https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/modal\\_box.html](https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/modal_box.html)

- Use external JS library to show slider

**Ans:**

<https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/slider.html>

- Prevent the browser when I click the form submit button

**Ans:**

[https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/form\\_submit\\_btn.html](https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/form_submit_btn.html)

## **MODULE: 4 (New Request)**

### **➤ What is JSON**

**Ans:** JSON stands for JavaScript Object Notation

JSON is a lightweight format for storing and transporting data

JSON is often used when data is sent from a server to a web page

JSON is “self-describing” and easy to understand

### **➤ What is promises**

**Ans:** "Producing code" is code that can take some time

"Consuming code" is code that must wait for the result

A Promise is a JavaScript object that links producing code and consuming code

A JavaScript Promise object can be:

- Pending
- Fulfilled
- Rejected

```
myPromise.then{  
  
function(value){/*code if successful*/},  
  
function(error){/*code if failed*/}  
  
};
```

➤ Write a program of promises and handle that promises also

**Ans:**

[https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/promise\\_task.html](https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/promise_task.html)

➤ Use fetch method for calling an API <https://fakestoreapi.com/products>

**Ans:**

[https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/fetch\\_API.html](https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/fetch_API.html)

➤ Display all the product from the API in your HTML page

**Ans:**

[https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/fetch\\_API.html](https://github.com/KunalPatel30/Assignment/blob/dce96d63f9130dd8acccc53d2068bca9ffaa786a/Advance%20JS%20Practicle/fetch_API.html)