

JS FOLLOW TO LEARN MORE JS

Interview Questions on Functions

JS

< SWIPE ≡



What is a **First Class Functions**?

- In JavaScript, functions are first-class objects.
- **First class functions** means when functions in that language are treated like any other variable
- **For example:** a function can be passed as an argument to other function, can be returned by another function and can be assigned as a value to a variable.



```
const run = () => {  
  // logic  
}
```

```
run(); // invoke it using a variable
```

What is Higher Order Function?

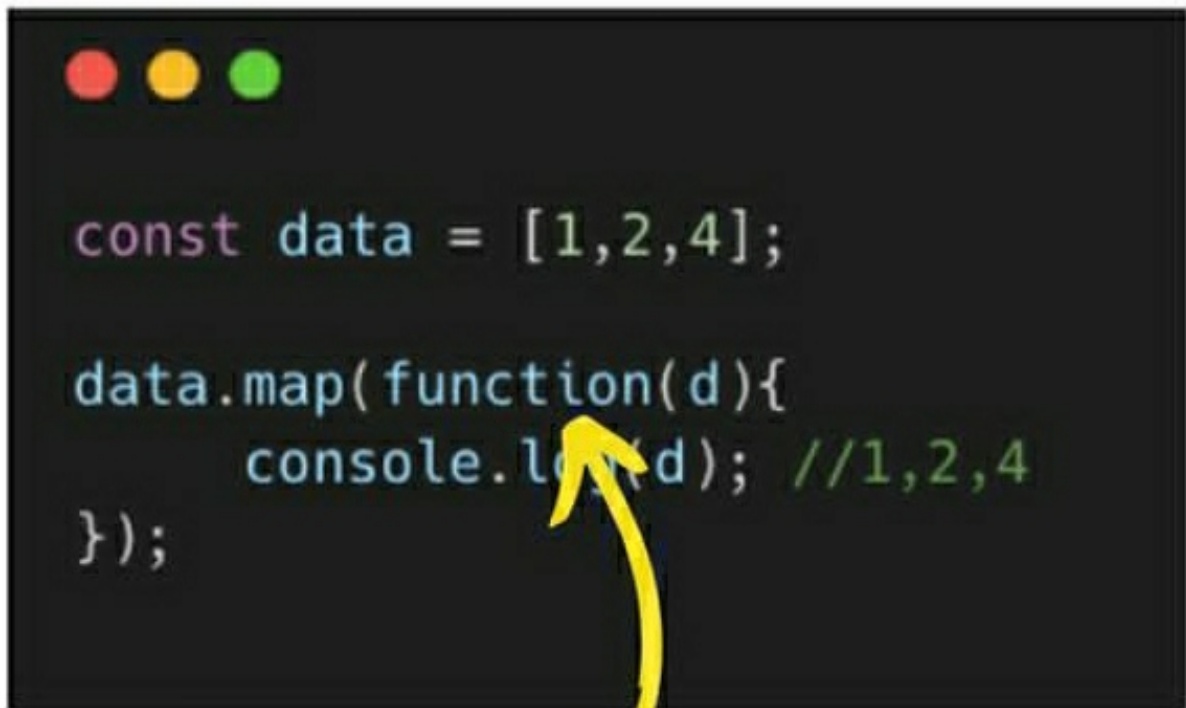
A function that returns another function as an argument or returns a function as a return value or both is called **Higher Order Function**

```
const data = [1,2,4];  
  
data.map(function(d){  
    console.log(d); //1,2,4  
});
```

*here, **map** is a Higher Order function
because its accepting other func as
argument*

What is Callback Function?

- **A callback function** is a function that is passed as an argument into another function and then invoked inside the outer function to complete some kind of routine or action.



```
const data = [1,2,4];  
  
data.map(function(d){  
    console.log(d); //1,2,4  
});
```

*here, **inner function** is a **callback func** because its passed as argument to other function*

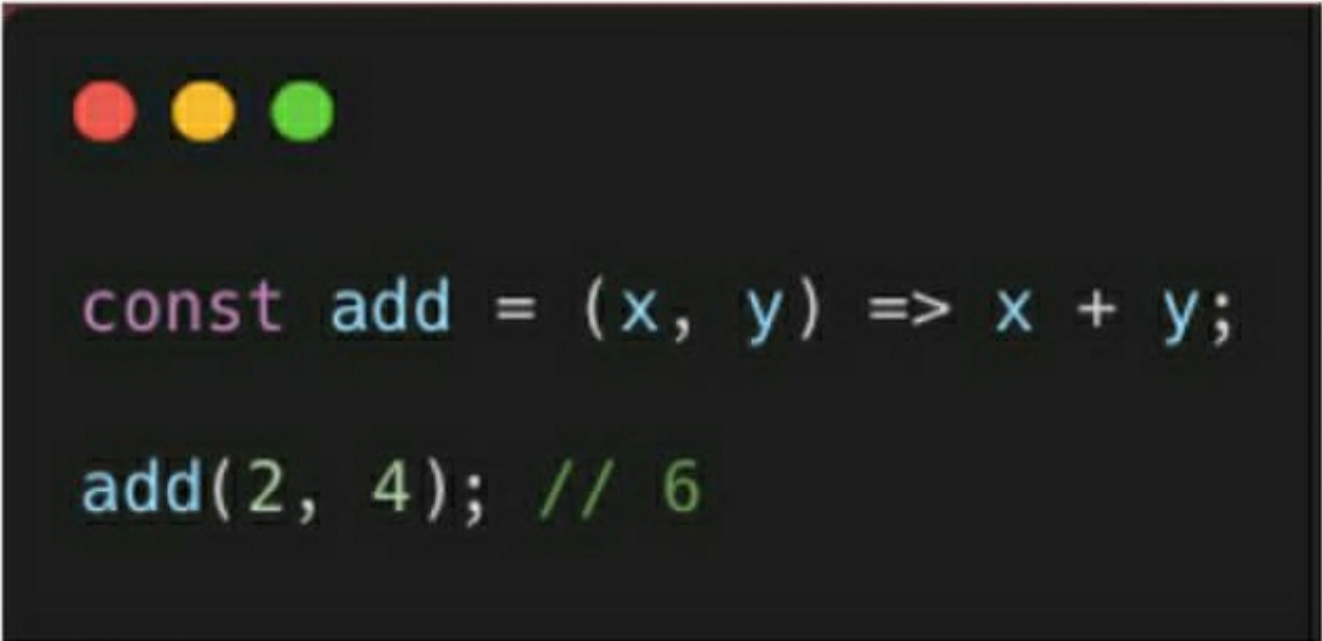
What is the Currying Function?

- It is a process of taking a function with multiple arguments and turning it into a sequence of functions each with only a single argument.
- Currying is named after Mathematician **Haskell Curry**
- Curry functions are great to improve code **reusability** and **functional composition**

```
function calculateVolume(length){  
  return function(breadth){  
    return function(height){  
      return length*breadth*height;  
    }  
  }  
}  
  
console.log(calculateVolume(2)(4)(2)); //16
```

What is a Pure Function?


- A Pure function is a function where the return value is only determined by its arguments without any side effects.
- If you call a function with same arguments it will always return the same result.



```
const add = (x, y) => x + y;  
add(2, 4); // 6
```

What is a **Arrow Function**?

- **Arrow function(=>)** is one of the features introduced in the ES6 version of JavaScript.
- It allows you to create functions in a cleaner way compared to regular functions
- Arrow functions don't have their own bindings to **this** , **arguments** , or **super**



```
const add = (x, y) => x + y;  
add(2, 4); // 6
```

What is a **Anonymous Function**?

- **Anonymous** is a function that does not have any name associated with it.
- Normally we use the function keyword before the function name to define a function in JavaScript, however, in anonymous functions in JavaScript, we use only the function keyword without the function name



```
var greet = function () {  
    console.log("Join @nishasingla05");  
};  
  
greet();
```


What is a **IIFE Function**?

- **An IIFE (Immediately Invoked Function Expression)**
- **IIFE** is a JavaScript function that runs as soon as it is defined.
- You can use **IIFEs** to prevent global variables' definition issues, alias variables, protect private data, and avoid conflicts of using many libraries that export the same object name

```
(function(){  
    console.log('This is a IIFE function');  
})();
```



Jayesh Deshmukh

Follow



Like



Comment



Share



Save