

The Quick Refresher

JAVASCRIPT INTERVIEW QUESTION + ANSWER

Var, Let, Const
Closure
Unshift() & pop()
0, Null, Undefined
Anonymous Function
Unescape() & escape()
Event Bubbling
New Operator

Learn all in one place



Q1 What is difference between VAR, LET & CONST?

*“ The difference between **var** and **let** is that **var** is function scoped while **let** is block-scoped ”*

keyword	const	let	var
global scope	NO	NO	YES
function scope	YES	YES	YES
block scope	YES	YES	NO
can be reassigned	NO	YES	YES



var



const



let

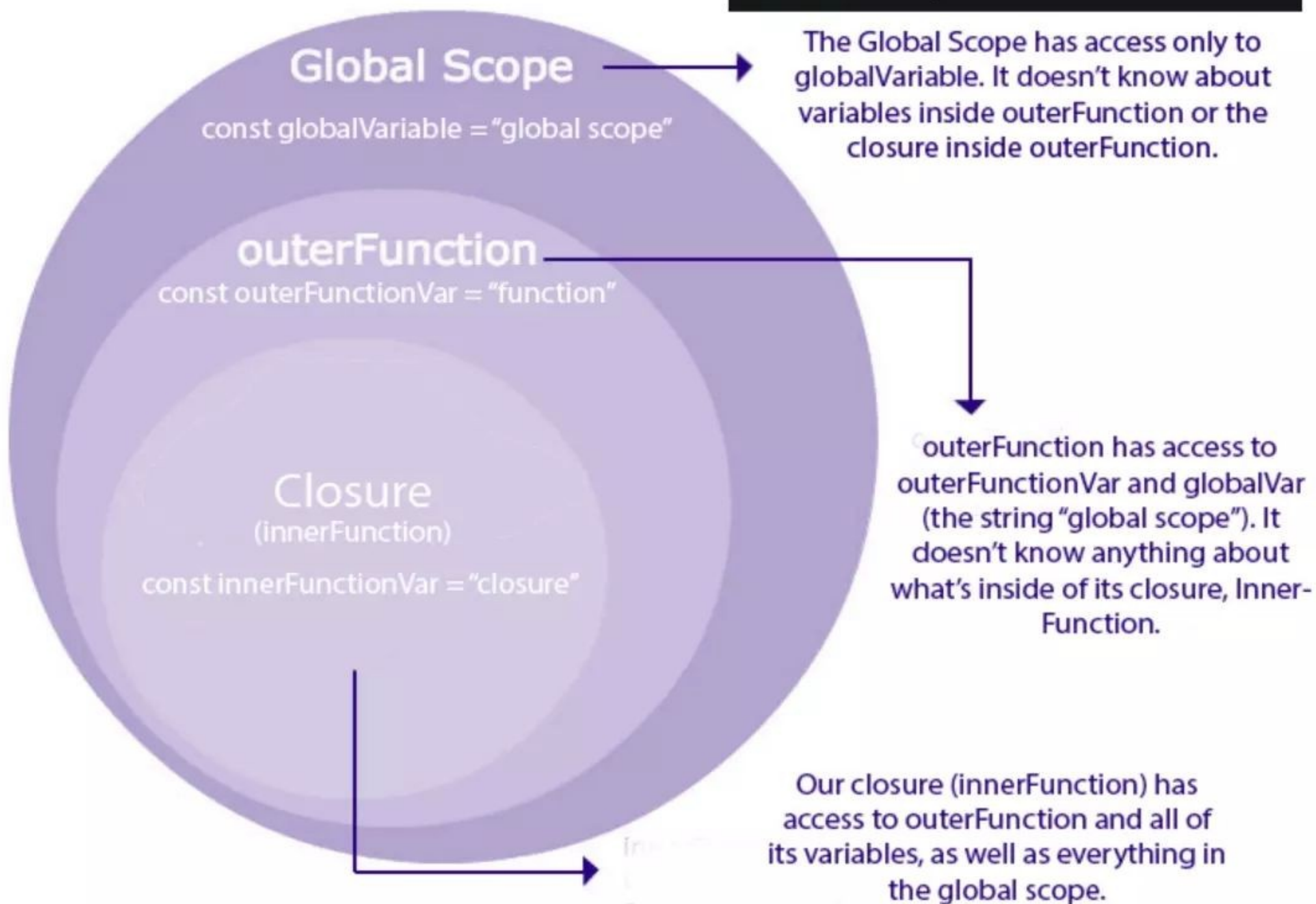
Q2 What is **Closure** in JavaScript?

Closure in JavaScript means an inner function have access to the variables that are defined in the outer function. Whenever we are declaring a function inside another function we are actually creating closures.

Closure Example ->

```
function developer(){  
  var name = 'aditya';  
  function displayName(){  
    alert(name);  
  }  
  return displayName;  
}  
var devName = developer();  
devName();
```

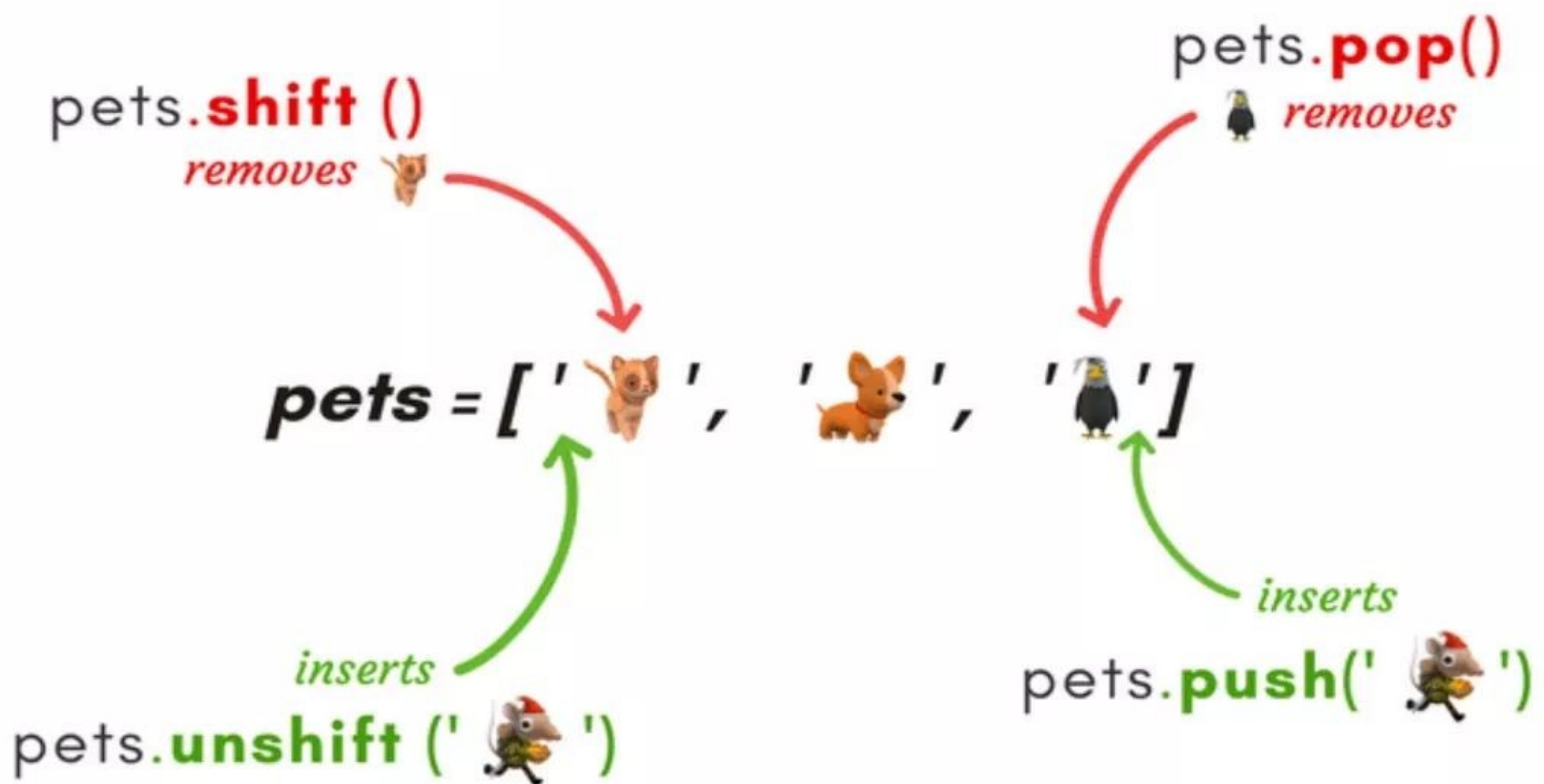
Complete overview -





Q3 Explain `unshift()` & `pop()` methods

push/pop/shift/unshift in Javascript



Well **pop** and **push** are easy enough to remember. As for **shift** and **unshift**

29 Easy to remember if you mentally drop the "f" in shift / unshift: **shift** removes elements and **unshift** adds them :) – Mar 28 '16 at 8:22



Q4 What is difference between 0, NULL & UNDEFINED?

0 - It's just the number zero.

null - Null means that something was defined but is not set.

undefined - Undefined most typically means a variable has been declared, but not defined.

```
let a = null;  
console.log(a);  
// null
```

```
let b;  
console.log(b);  
// undefined
```

Non-zero value



null



0



undefined





Q5 What does the following statement declares?

```
var myArray = [[[]]];
```

It declares a **three dimensional array**

Q6 Define event bubbling?

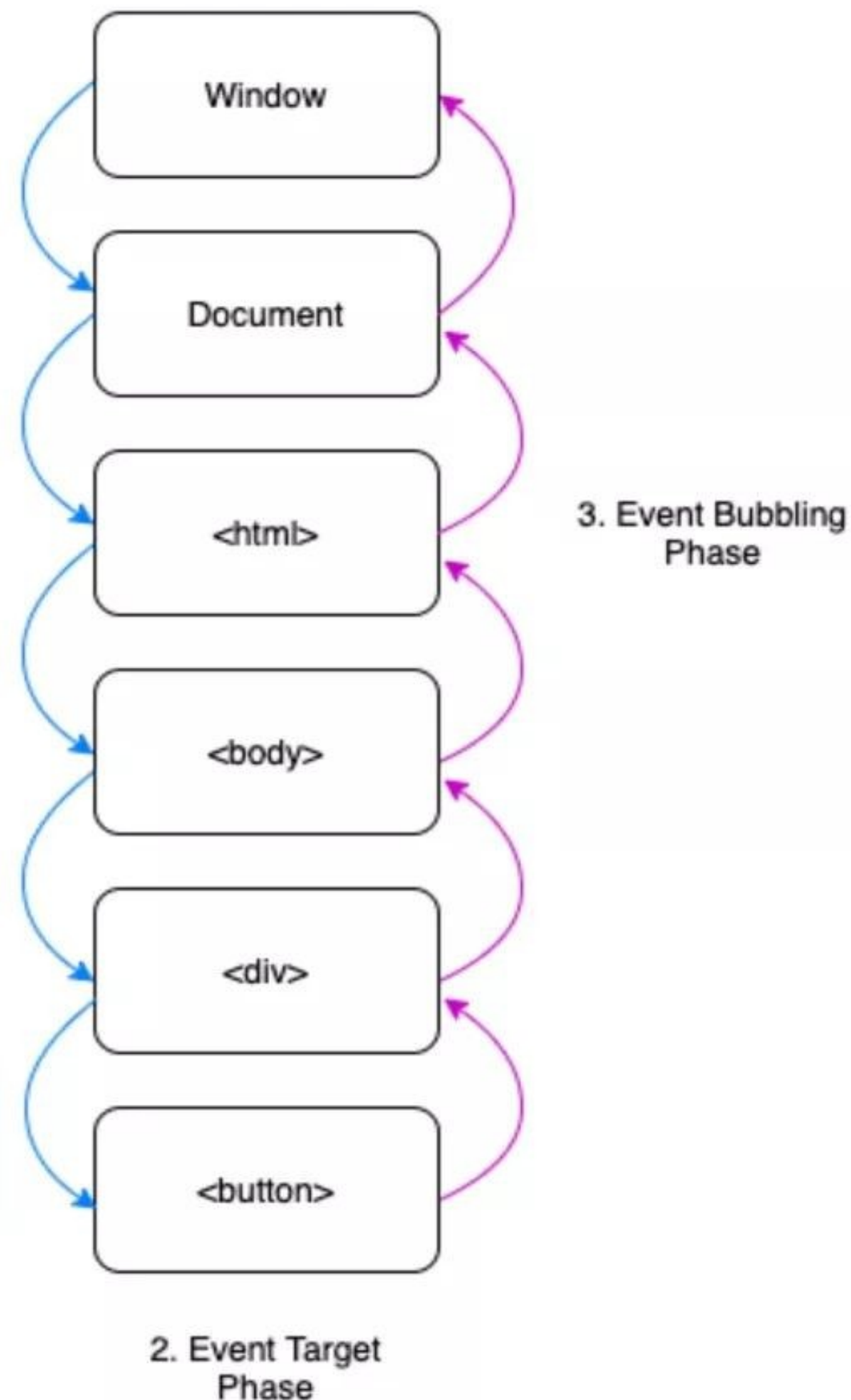
JavaScript allows DOM elements to be nested inside each other. In such a case, **if the handler of the child is clicked**, the handler of parent will also work as if it were clicked too.

1. Event Capturing Phase

3. Event Bubbling Phase

Q7 How can a value be appended to an array?

```
arr[arr.length] = value;
```





Q8 What is **new operator** in JavaScript?

The **new operator** is used to **construct a user-defined object from a function**. The **this** inside the function is set to the new object. So any assignments we are not with the this will be added as a property to this object.

If we **explicitly return** something inside the function, then the this binding will be ignored and an empty object will be assigned to the **caller variable**.

```
function developer(salary, role){
  this.salary = salary; //this refers to newly created object here
  this.role = role;
}
var employee = new developer('$450,000', 'SeniorWebDev');
console.log(employee.salary); // $450,000
```

```
var obj1 = new MyFunc();
    ↓ 1. Creates an empty object
{ }
    ↓ 2. Assigns MyFunc.prototype
{ __proto__ = MyFunc.prototype }
    ↓ 3. Assign properties and functions declared with this keyword
{ __proto__ = MyFunc.prototype, x = 100 }
    ↓ 4. Returns newly created object
var obj1 = { __proto__ = MyFunc.prototype, x = 100 }
```

© TutorialsTeacher.com



Q9 Define **unescape()** and **escape()** functions?

The **escape()** function is responsible for coding a string so as to make the transfer of the information from one computer to the other, across a network.

```
document.write(escape("Hello! I am Aditya"));  
//output : Hello%21%20I%20am%20Aditya
```

The **unescape()** function is very important as it decodes the coded string. It works in the following way. For example:

```
document.write(unescape("Hello%21%20I%20am%20Aditya"));  
//Output :: Hello! I am Aditya
```

