# Frontend Interview Questions

### Reverse a string in Javascript:

Ans: let str = "hello";

let reversed = str.split("").reverse().join("");

### Output of the code

var a={},

b={key:'b'},

c={key:'c'};

a[b]=123;

a[c]=456;

console.log(a[b]);

Ans: 456, When setting an object property, JavaScript will implicitly **stringify** the parameter value. In this case, since b and c are both objects, they will *both* be converted to "[object Object]". As a result, a[b] anda[c] are both equivalent to a["[object Object]"] and can be used interchangeably. Therefore, setting or referencing a[c] is precisely the same as setting or referencing a[b].

### Console output of the below code:

(function(x) {

return (function(y) {

console.log(x);

})(2)

})(1);

Ans: 1, since x is not defined in the inner function, the scope of the outer function is searched for a defined variable x, which is found to have a value of 1

### How do you clone an object in Javascript?

var obj = {a: 1 ,b: 2}

var objclone = Object.assign({},obj);

### What is closure in Javascript

Ans: A closure is an inner function that has access to the variables in the outer (enclosing) function’s scope chain. The closure has access to variables in three scopes; specifically: (1) variable in its own scope, (2) variables in the enclosing function’s scope, and (3) global variables.

# Frontend Interview Questions

### How does one share data between components in Angular?

* Ans: **Parent to child using @Input decorator**
* **Child to parent using @ViewChild decorator**
* **@Output** property and services

### What is AOT compilation? What are the advantages of AOT?

Ans: Angular provides two types of compilation:

* JIT(Just-in-Time) compilation
* AOT(Ahead-of-Time) compilation

In JIT compilation, the application compiles inside the browser during runtime. Whereas in the AOT compilation, the application compiles during the build time.

The advantages of using AOT compilation are:

* Since the application compiles before running inside the browser, the browser loads the executable code and renders the application immediately, which leads to **faster rendering**.
* In AOT compilation, the compiler sends the external HTML and CSS files along with the application, eliminating separate AJAX requests for those source files, which leads to **fewer ajax requests**.
* Developers can detect and handle errors during the building phase, which helps in **minimizing errors**.
* The AOT compiler adds HTML and templates into the JS files before they run inside the browser. Due to this, there are no extra HTML files to be read, which provide **better security** to the application.

By default, angular builds and serves the application using JIT compiler:

ng build

ng serve

For using AOT compiler following changes should be made:

ng build --aot

ng serve --aot

### What are HTTP interceptors in Angular?

Ans: Using the HttpClient, interceptors allow us to intercept incoming and outgoing HTTP requests. They are capable of handling both HttpRequest and HttpResponse. We can edit or update the value of the request by intercepting the HTTP request, and we can perform some specified actions on a specific status code or message by intercepting the answer.

### How are observables different from promises?

Ans:

| **Promise** | **Observable** |
| --- | --- |
| Emits a single value | Emits multiple values over a period of time |
| Not Lazy | Lazy. An observable is not called until we subscribe to the observable |
| Cannot be cancelled | Can be cancelled by using the unsubscribe() method |
|  | Observable provides operators like map, forEach, filter, reduce, retry, retryWhen etc. |

### **What is view encapsulation in Angular?**

View encapsulation specifies if the component's template and styles can impact the entire program or vice versa.

**Angular offers three encapsulation methods:**

* **Native:**The component does not inherit styles from the main HTML. Styles defined in this component's @Component decorator are only applicable to this component.
* **Emulated (Default):**The component inherits styles from the main HTML. Styles set in the @Component decorator are only applicable to this component.
* **None:**The component's styles are propagated back to the main HTML and therefore accessible to all components on the page. Be wary of programs that have None and Native components. Styles will be repeated in all components with Native encapsulation if they have No encapsulation.