ES6 Interview Questions

Read Top ES6 Interview Questions, ECMAScript 2015 Interview Questions

Q1. What is ES6?

Es6 or ECMASCRIPT 2015 is sixth major release of ECMAScript language which comes with a lot of new features and syntax for writing web applications in Javascript. As currently, not all browsers support ES6, they support pre-versions of ES6. So to write web applications in ES6 that will support all Browsers we needed tools like Babel and Webpack.

Q2. List some new features of ES6

New Features in ES6.

1. Support for constants (also known as “immutable variables”)
2. Block-Scope support for both variables, constants, functions
3. Arrow Functions
4. Extended Parameter Handling
5. Template Literals
6. Extended Literals
7. Enhanced Regular Expression
8. Enhanced Object Properties
9. Destructuring Assignment
10. Modules, Classes, Iterators, Generators
11. Support for Map/Set & WeakMap/WeakSet
12. Promises, Meta-Programming ,Internationalization & Localization

Read More from http://es6-features.org/

Q3. What is Babel?

Babel is one of the most popular JavaScript transpilers and becomes the industry standard. It allows us to write ES6 code and convert it back in pre-Es6 JavaScript that browser supports.

For example look the below code snippet.
In ES6 (ECMASCRIPT 2015)

```
const PI = 3.141593;
```

```
PI > 3.0 ;
export{PI};
```

In ES5 after conversion

```
"use strict";
Object.defineProperty(exports, "__esModule", {
  value: true
});
var PI = 3.141593;
PI > 3.0;
exports.PI = PI;
```

Q4. List steps to install Babel?

Installation: In order to install Babel, you require node.js and NPM. Make sure Node.js is installed on your server.

To check node installed or not run below commands on your terminal.

```
node -v
npm -v
```

Installing Babel : We can install Babel CLI locally by running below command on terminal.

```
npm install --save-dev babel-cli
```

Q5. What is Webpack?

Webpack allows you to run an environment that hosts babel. Webpack is opensource javascript module bundler which takes modules with dependencies and generates static assets representing those modules.

Q6. List benefits of using Webpack?

Benefits of using Webpack.

1. It bundles your multiple modules and packs it into a single .js file.
2. It comes with integrated dev server. A small express app for local development. You simply include one Javascript tag pointed to the server, like localhost:8080/assets/bundle.js, and get live code updating and asset management for free.

Q7. Explain Constants in Es6?

Constants also are known as immutable variables are a special type of variables whose content is not changed. In Es6 a constant is defined using const keyword. Constants in Es6 enable protection to overwrite a variable value, improve performance and helps programmers to write readable and cleaner code.

Example

In Es6

```
const WEBSITE_URL = "http://www.abc.com";
WEBSITE_URL="new url"; // generate an error;
console.log(WEBSITE_URL);
```

In prior version of Es6

```
//  and only in global context and not in a block scope
Object.defineProperty(typeof global === "object" ? global : window, "WEBSITE_URL"
value: "http://www.abc.com", enumerable: true,
writable:     false,
configurable: false
});
console.log(WEBSITE_URL);
```

Also, Read Best Node JS Interview Questions

Q8. What are template literals in Es6?

Template literals are the string with embedded code and variables inside. Template literal allows concatenation and interpolation in much more comprehensive and clear in comparison with prior versions of ECMAScript.

Let see an example of concatenating a string in JavaScript.

```
var a="Hello";
var b="John";
var c = a+ " " + b;
Console.log(c); //outputs Hello John;
```

In ES6 concatenation and interpolation is done by backtick “ in a single line. To interpolate a variable simply put in to {} braces forwarded by $ sign.>/p>

// In ES6

```
let a="Hello";
let b="John";
let c=`${a} ${b}`;
console.log(c); //outputs Hello John;
```

Q9. What is Spread Operator in ES6?

Spread Operator provides a new way to manipulate array and objects in Es6.A Spread operator is represented by … followed by the variable name.

Example :

```
let a =[7,8,9];
let b=[1,2,3,...a,10];
console.log(b); // [1,2,3,7,8,9,10]
```

So spread operator spreads the contents of variable a and concatenates it in b.

Another Example

```
function print(...z){
 console.log(z);
}
print(1,2,3,4);//[1,2,3,4]
```

Q10. Explain Destructuring Assignment in ES6?

Destructing assignment in another improvement in Es6. It allows us to extract data from array and objects into separate variables.

Example

```
let full_name =['John','Deo'];
let [first_name,last_name]=full_name;
console.log(first_name,last_name);
// outputs John Deo
```

Another example

```
let c=[100,200,330,400];
let [a,...b]=c;
console.log(a,b);
// outputs 100 [200, 330, 400]
```

Q11. How to create a Javascript class in ES6?

In Es6 you can create a class using the Class keyword.Below is sample javascript class in ES6.

```
class User{
    constructor(name,age) {
        this.name  = name;
        this.age = age;
    }
    getData() {
        console.log(this.name + " is " + this.age + " years old !");
    }
}
var user = new User("foo", 7);
```

```
s1.getData();
```

Q12. What is Set in ES6?

Set is a collection of unique values. The values could be also primitives or object references.

Creating a Set in Javascript

```
let set = new Set();
set.add(1);
set.add('1');
set.add({ key: 'value' });
console.log(set); // Set {1, '1', Object {key: 'value'}}
```

Q13. Explain Generator function in ES6?

Generators are functions that can be exited and later re-entered. Their context (variable bindings) will be saved across re-entrances. A function keyword followed by an asterisk defines a generator function, which returns a Generator object.

Generator Function Example.

```
function* generator(i) {
  yield i;
  yield i + 10;
}
var gen = generator(10);
console.log(gen.next().value);
// expected output: 10
console.log(gen.next().value);
```

Further reading: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/function*

Q14. Explain WeakMap in ES6?

WeakMaps provide a way to extend objects from the outside without interfering with garbage collection. Whenever you want to extend an object but can't because it is sealed - or from an external source - a WeakMap can be applied. WeakMap was also introduced by ES6 in 2015

Further reading : https://stackoverflow.com/questions/29413222/what-are-the-actual-uses-of-es6-weakmap