4) ES6/ES7 Features.

ES é a abreviação para ECMAScript que por sua vez significa “European Computer Manufacturers Association”.

Objetivos:

* Ser uma linguagem melhor para construir aplicações complexas;
* Resolver problemas antigos do JavaScript;
* Facilidade no desenvolvimento de *libraries;*

1-Features ES6:

* 1. Declaração de variáveis:

1.1.1 var x let

A diferença principal entre o var e o let é que enquanto o primeiro tem escopo de função, o segundo possui escopo de bloco. Em resumo, let conserta o antigo problema causado pelo hoisting fazendo com que a declaração de variáveis funcione da forma esperada pela maioria dos desenvolvedores.

1.1.2 let x const

const funciona de forma semelhante. A única diferença é que as variáveis criadas não podem ser reatribuídas. É comum achar que const deixa a variável imutável, assim como algumas libs como ImmutableJS. Isso não é verdade. As propriedades de um objeto, por exemplo, podem ser alteradas:

1.2 Parâmetro de Funções

1.2.1“default parameters”

Os parâmetros de funções têm undefined como valor default. Porém, em alguns casos, pode ser necessário utilizar um outro valor. O ES6 introduziu uma nova forma, bem mais simples, de se fazer isso. Basta adicionar o valor default na definição do parâmetro desejado.

1.2.2 “rest parameters”

Na versão atual do JavaScript podemos utilizar o objeto arguments para pegar todos os parâmetros de uma função. O arguments porém, apresenta alguns problemas:

-O objeto parece com um array, mas não é exatamente um

-Todos os parâmetros da função são automaticamente atribuídos ao arguments. Não temos uma forma clara de diferenciar os parâmetros.

Com esses problemas em mente, os Rest Parameters foram adicionados no ES6.

* 1. Programação Funcional

1.3.1 “arrow functions”

Os arrow functions são um excelente syntax sugar na criação de funções. Com funções de apenas uma linha, podemos simplesmente omitir o return e as chaves.

Mas o verdadeiro benefício das arrows functions não está na expressividade, ele se encontra na resolução de um antigo problema da linguagem: o this.

1.3.2 “destructuring”

Uma nova forma de declarar variáveis extraindo valores de objetos e arrays é através do destructuring.

1.4 Orientação a Objetos

Vamos falar agora sobre a feature mais polêmica do ES6: classes.

Classes nos dão uma sintaxe amigável que definem o estado e o comportamento de objetos que representam as abstrações que usamos diariamente.

1.4.1 constructor

O método constructor tem como tarefa fazer a inicialização da instância. Ele é chamado automaticamente na criação da mesma e garante que ela esteja em um estado válido.

1.4.2 getters/setters

Como getters e setters são muito comuns na utilização de classes, o ES6 veio com um syntax sugar para lidar com os mesmos. A principal função dos getters/setters é proteger os dados internos das instâncias de um objeto.

1.4.3 extends

Podemos utilizar o conceito de herança com o extends.

1.5Módulos

Um dos grandes problemas do JavaScript era não ter um sistema built-in de módulos assim como a maioria das linguagens possui. A maioria dos desenvolvedores que já trabalhou em projetos grandes sabe a importância dos módulos: organizar o sistema, aumentar o reuso e diminuir a complexidade de cada pequeno trecho do seu code base.

O ES6 possui várias outra features que vieram para concertar erros das versões anteriores e facilitar a programação com essa ferramenta.

5) Instanciação de Variáveis.

Hoisting é uma maneira pela qual as variáveis são declaradas em JS, ou seja, quando declaradas elas são erguidas, colocadas no topo do escopo, mas sua inicialização não.

Uma das boas práticas em JavaScript é sempre declarar as variáveis com a palavra reservada ‘var’ para conter o valor em seu escopo local e manter o escopo global limpo.

-Variável Global

Variáveis globais são todas aquelas definidas fora de alguma função. Isso porque cada função gera seu próprio escopo. A variável global pode ser acessada por qualquer função. Todas as variáveis que não forem declaradas com a palavra reservada ‘var’ serão consideradas parte do escopo global.

-Variável por parâmetro

Quando declaramos uma função temos a opção de indicar alguns parâmetros para elas. Tais parâmetros são considerados como variáveis que recebem um valor na hora da execução da função. Esses valores são utilizados dentro da função. Caso esse parâmetro seja uma variável global o valor dela não se altera.

-Instanciação usando uma IIFE

IIFE é a abreviação para Imediately Invoked Function Expression, que pode ser traduzida para Expressão de Função Invocada Imediatamente. Esse tipo de função é executado no mesmo momento que está sendo interpretada. Os parênteses que envolvem a função fazem dela uma expressão e os parênteses no final da declaração executa a função. Esse tipo de função também pode ser armazenado em uma variável. Como toda função, a IIFE também pode receber parâmetros.