Curso: Introdução a JavaScript ES6 Essencial

Instituição: Digital Innovation One

Link: https://web.digitalinnovation.one/course/javascript-es6-essencial/learning/183aad79-0e6d-4acb-880f-b0e179824a81/

AULA 01:

HISTÓRIA E CONCEITOS:

Histórico:

* Lançado em 1995
* Criado por Brendan Eich;
* ECMAScript é uma especificação da linguagem de programação;
* O ECMAScript é controlado pelo TC39 que é um reunião onde os membros fazem propostas para adicionar novas features às linguagens e onde elas são votadas baseadas em maioria de votos absolutos. O fluxo da proposta funciona da seguinte forma:
  + Stage 0: Strawman – o conteúdo é revisado;
  + Stage 1: Proposal – surge a proposta formal dessa nova feature;
  + Stage 2: Draft – Primeira versão da proposta, com descrição da sintaxe, semântica, etc...;
  + Stage 3: candidate – implementação para obtenção do feedback do usuário e é realizada a documentação;
  + Stage 4: Finished – a proposta está pronta ara ser implementada ao ECMAScript;
* Como o JS é uma linguagem que possui várias versões diferentes, existe um aplicativo que converte versões mais recentes para versões mais antigas, chamado <https://babeljs.io/>;

Conceitos:

* JS é uma linguagem interpretada, ou seja, não precisa ser compilada antes de ser executada, como o python por exemplo.
* Possui tipagem fraca e dinâmica, fraca pois permite operações entre diferentes tipos de variáveis e dinâmica porque não é preciso indicar o tipo da variável no momento de sua declaração, ele é atribuído quando atribui-se um valor à ela;
* É interessante observar que existe ainda a “linguagem” TypeScript, é um superste de JS, ou seja, possui basicamente a mesma sintaxe porem adiciona funcionalidades extras e algumas outras features interessantes [https://www.typescriptlang.org/];
* JS é uma linguagem com funções e primeira classe e ordem maior, isto é, pode ser atribuída a uma variável ou a uma estrutura de dados;
* Possui Closure (ou escopo léxico) que é a capacidade de uma função ou variável “lembrar” do ambiente em que foi criada;

Currying, Hoisting, Imutabilidade, Tipos e Variáveis:

* Currying é uma técnica em JS que transforma uma função de vários parâmetros em uma função que lembra algum ou alguns destes parâmetros para a próxima execução, veja uma exemplo:

//Aplicação do Currying:

function soma(a, b){

    return a + b;

}

soma(2, 3);

soma(2, 4);

soma(2, 5);

soma(2, 6); //Perceba que o primeiro parâmetro é igual em todas as chamadas

//Aplicando Currying poderiamos fazer o seguinte:

const soma2 = soma(2);

soma2(3);

soma2(4);

soma2(5);

soma2(6); //Perceba que agora só foi neceário digitar um parâmetro

* Hoisting (içar, leantar) é uma característica do JS que diz que as funções independentemente de onde foram declaradas são elevadas par ao início do script, o mesmo acontece com variáveis mas apenas com sua declaração, seu valor é dado como undefined.
* Imutabilidade: um objeto nunca é alterado, ele é copiado e então alterado, veja:

 //Demonstração de imutabilidade

 const students = [{

    name: 'Leo',

    grade: 10

  },

  {

    name: 'Nicolly',

    grade: 6

  },

];

function getApproved(studentList) {

  return studentList.filter(student => student.grade >= 7);

}

console.log('Lista de alunos aprovados:');

console.log(getApproved(students));

console.log('\nLista de todos os alunos:');

console.log(students); // Perceba que a lista de estudantes original não é alterado após aplicar a função

parei na aula 1 em 13:28 da parte de currying