História e Conceitos

* Lançado em setembro de 1995;
* Criado por Brendan Eich;

# ECMAScript

Padronização do JavaScript

TC39 é o comitê responsável pela evolução do Javacript, matém e atualiza os padrões do ECMAScript

# Fluxo de Proposta

* Stage 0: strawman;
* Stage 1: proposal – terá um campeão;
* Stage 2: draft – descrição das alterações na proposta;
* Stage 3: Candidate – alterações por outra pessoa sem ser o campeão;
* Stage 4: Finished

# ES2018

* Operadores rest/spread;
* Iteração assíncrona;
* Promise.prototype.finally();

# ES.Next

* Alterações futuras;
* Podem ser testadas no Babeljs.io

Conceitos

# Linguagem interpretada – código executado de cima pra baixo e o resultado dessa execução é imediatamente retornado ;

# Linguagem de tipagem fraca e dinâmica – Significa que não há verificação em todas as operações no JS ex: é possível utilizar uma string e um número em uma operação sem apresentar um erro;

# TypeScript

Linguagem typeScript com tipagem forte

# Funções de primeira classe e ordem maior

* *Função de primeira classe e ordem maior:* A função pode ser atribuida a uma variável, estrutura de dados(array, objetos), podem ser passadas por argumentos;

# Closure

* *Closure :*  escopo léxico, é a capacidade de uma função lembrar do ambiente onde foi criada.
* *3 tipos de escopo :* Global, função, bloco(com a inclusão do let e const).

Currying, Hoisting, imutabilidade, Tipos e Variáveis

# Currying – É a técnica de tranformar uma função com “n” parametros, seja 1;2;3 ou 4, em apenas uma função que receba um parametro

Para cada parametro a gente vai criar uma função

Ex: function soma( n1, n2 ){  
 return n1 + n2

}

soma(2, 2);

soma(2, 3);

soma(2, 4);

soma(2, 5);

para não ficar colocando o número “2” sempre, trasnformar em(reaproveitamento de código) :

function soma(a){  
 return function(b){  
 return a + b

}

}

const soma2 = soma(2); // o parametro “a” sempre será 2

// quando a soma2() for instanciada o parametro “b” será inserido.

soma2(2);

soma2(3);

soma2(4);

soma2(5);

# Hoisting – Significa levantar ou suspender algo

Ocorre no JS nas declarações de variáveis e funções. As declarações são elevadas ao escopo em que elas estão seja de bloco, função ou global

Os dois tipos de hoisting são de variável e função

A grande diferença é : O hoisting de variável só eleva a criação de variável e não a sua atribuição, já a função é elevada ao topo como um todo.

# Imutabilidade – Os que dados que criarmos nunca mudarão, a variével nunca vai mudar, caso seja necessário alterá-la , nós criaremos uma nova variável.

Ao invés da gente alterar um array, seja deletar ou inserir um novo dado, a gente vai criar um novo array baseado nesse que queremos alterar, caso a gente queira adicionar um novo item a gente pega todo o conteúdo desse array e concatena um novo. Caso queiramos deletar um item do array a gente pega filtra pra remover esse item.

A mesma coisa com um objeto. Um objeto nunca é atualizado, ele vai ser copiado e você altera só o que voce quer.

# Tipos e variáveis

* Var
* Let – respeita escopo de bloco
* Const - respeita escopo de bloco

Tipos e Variáveis

# Tipos primitivos

* String;
* Number;
* Boolean – true or false;
* Null;
* Undefined;
* Symbol – criar valores únicos;
* Object;
* Function;
* Array.

OBS: abrir arquivos no vscode

Operadores

* Aritméticos;
* Atribuição;
* Comparação;
* Condicional;
* Lógicos;
* Spread.

Estruturas condicionais

* If, else e else if

Estruturas de repetição

* For;
* While;
* Do...while;
* For...in;
* Continue;
* Break.