O objetivo desse curso é entender um pouco mais sobre o javascript em sí e sua sintaxe.

O javascript é uma linguagem de programação de alto nivel, se aproximando da linguagem humana, é uma linguagem interpretada.

O javascript serve para criar scripts dinâmicos que realizam a interação de apps com sites, podendo ser usado principalmente em paginas de HTML.

**Tipagem** é uma regra de uso de dados, quanto mais forte a tipagem, mais obrigatória a declaração de dados, no javascript possui uma tipagem fraca, que acontece de modo dinamico.

Variaveis de tipo primitivos são as **booleanas**, **null**, **undefined**, **number**, **string**, **array**, **object** e **function**.

**Variaveis** são dados que são modificados durante o script, são diferentes de **constantes**, que não variam. Tambem existe a **let**, uma variavel de escopo local, podendo ter seu valor alterado ou não, sem um valor inicial ela é tratada como null (nula). A constante precisa de um valor, se não gerará um erro.

Escopo se refere a delimitação que tal bloco de código possui, **global** é disponível em todo código, ficando visível para todo o código e podendo ser utilizada em qualquer bloco de código, **local** só funciona em dentro do bloco específicado, ele **funciona**, mas não pode ser visualizado e utilizado. Usa-se o let quando em se quer um escopo local.

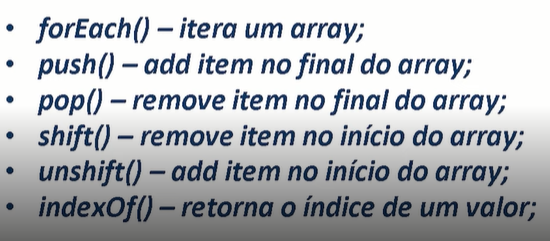
É pratica comum declarar variaveis no topo do código/do escopo, isso facilita a compreensão do código e evita possiveis erros de interpretação do mesmo, sobre sua nomeção, a primeira letra de uma variavel só pode ser letra, underscore ou $, pois o resto é utilizado como comando, naturalmente palavras reservadas não podem ser usadas como nome.

O sinal =não significa igual no js, mas **atribuição**, uma caracteristica que vai ser atribuida a a variavel, quando se quer dizer que um valor é igual a outro entao se coloca == em vez de 1 só. A comparação identica é feita com ===, note que isso não só leva em conta o que esta colocado, mas seu tipo ‘0’ === 0 daria falso pois ‘0’ é uma string e 0 sem ‘’ um numero.

\*\* significa potencialização

Essa parte tratou bastante da logica de programação do js, o que não difere tanto da lógica de programação padrão que já foi aprendida anteriormente, logo poucas anotações.

==

**Arrays** são um tipo de lista, ou matriz de variaveis, onde cada variavel possui um índice, *let array = [‘string’, 1, true, false, [array2], ... ]* o array pode guardar outros arrays. Os principais comandos relacionados a Array estão ao lado. Tambem temos o **splice** que remove ou substitui um item pelo índice e o **slice** que retorna parte do array existente.

==

**Objetos** são dados que possuem propriedades e vetores que definem suas caracteristicas, declaradas entre chaves {}.

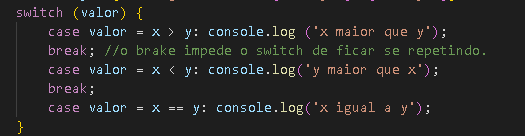
*Var = xicara {cor:’azul’, tamanho:’p’,função:tomarcafé()}*

Nessas condições, se colocarmos o comando *var xicara.cor =* o VSC vai lhe avisar que o único objeto (xicara) com a cor (atributo) é azul. *Var xicara.cor = azul*

*==*

Agora aprendemos sobre a estrutura de if e else, as estruturas condicionais.

O **nest if** é o uso de varios if else para circular todos os casos possíveis de acontecimento de nossa função, é considerado uma prática feia, mas funcional e facil de se montar.

O **if internário** é uma verificação if de uma única linha, seguindo o seguinte código

*Condição?instrução1:instrução2*

*X === y?console.log(‘sim’):console.log(‘não’);*

O **switch/case** também é uma estrutura condicional.

Estruturas de **repetição** são o **for**, **for/in** (para algo **dentro** de outro, in, geralmente para arrays), **for/of** (para algo de outro lugar, geralmente uma string, o for/of não funciona com objetos pois as propriedades variam, da para fazer, mas a propriedade sera passada dentro da condição, e cada letra ficada em uma linha diferente, o que não é prático), **while** e **do/while**.

==

**Função** é um bloco de comando e instrução para a execução de determinadas tarefas obviamente isso não se limita a funções matematicas claro.

*Function name(){[instrução]};*

*name();*

pode haver parametros dentro da função, servindo como variaveis, onde sua atribuição pode ser feita durante a função.

*function message(message1, message2){console.log(message1, message2)}*

*message(message1,message2);*

==

Funções **declarativas** são funções obrigatóriamente nomeadas, **expressões de função** são funções atribuidas em expressões, funções atribuidas a uma **variável**.

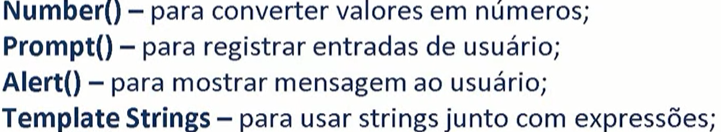
*var função = function(){console.log(‘LOL’)}*

*função();*

**arrow function** são funções de expressão de sintaxe curta, sempre **anonimas** (sem nome), declaradas entre parenteses seguido de => e {}.

*Var função=()=>{console.log(‘L’)}*

*função();*

agora vamos criar uma pequena calculadora utilizando somente o javascript. Utilizando principalmente os seguintes comandos ao lado.