***JavaScript II – Estrutura de dados***

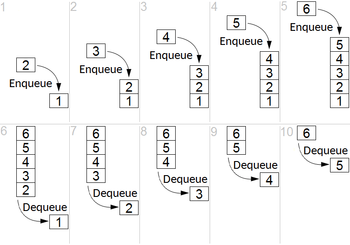
São um modo particular de armazenar e organizar dados de modo que possam ser usados eficientemente, facilitando sua busca e modificação.

Estruturas de dados clássicas:

Lista ligada – Coleção ordenada de valores onde cada elemento aponta para o próximo(ordem).

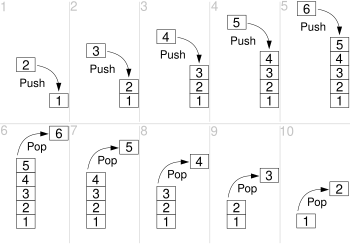
Array (vetor) – Coleção de elementos que podem ser identificados por um índice.

Fila (Queue) – Coleção de elementos onde o primeiro elemento a ser retirado deve ser o primeiro elemento adicionado (funcionamento específico). Também chamadas de listas FIFO(first-in / first-out, ou primeiro a entrar / primeiro a sair).



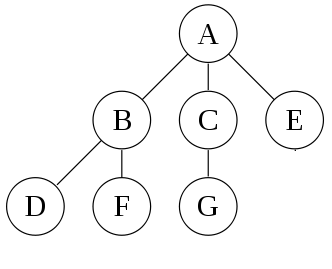
Pilha (Stack) – Coleção de elementos onde o primeiro elemento a ser retirado deve ser o último elemento adicionado.

Também chamadas de lista LIFO (last-in /first-out, ou último a entrar / primeiro a sair).



Árvore (Tree) – Conjunto de elementos chamados de nós (ou nodes) organizados em estrutura hierárquica (não sequencial), ou seja, podendo estar “abaixo” ou “acima” de outros nós.

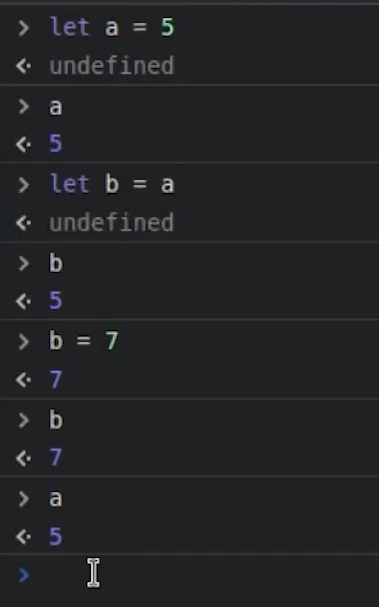
Pense em uma árvore genealógica, na estrutura das pastas no seu computador ou na estrutura de uma página HTML.



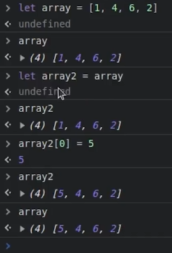
Estruturas de dados no javascript (e outras linguagens de alto-nível) >> Listas(Arrays) e Dicionários (Objetos).

***Valor e Referência.***

Tipos primitivos tem valores individuais para cada variável que está sendo criada; Valores não alteram o objeto original porque o valor da variável é um clone do valor original, e não um ponteiro:



Tipos não-primitivos(objetos) eles utilizam as referências; Referências apontam para o objeto original como um ponteiro, então alterações em qualquer variável com esse ponteiro, vão alterar o objeto original também.



***Arrays***

[***https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array***](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array)

É uma estrutura do JavaScript para se trabalhar com todo tipo de lista.

Seus elementos são identificados por um índice inteiro começando pelo 0.

Pode armazenar qualquer tipo de dado e ter qualquer tamanho.

Pode ser criada e utilizado através de colchetes [ ].

*let array = [ ]* *let array = [ 1, 2, 4, 7 ]*

*array [0] = “Olá”* *array [1] = 42*

Possuem funções para vários casos de uso:

*Enfileirar e desenfileirar*

*Empilhar e desempilhar*

*Achar o índice de um valor*

*Cortar e concatenar*

*Etc...*

Tem um PDF na mesma pasta desse arquivo com métodos dos arrays, quando precisar acessse.