# Java Script – Anotações

## ATALHOS

Ctrl + k + c -> transforma o conteúdo selecionado em comentário

## NODE NO PROMPT DE COMANDO

node –v: verifica a versão dentro do Prompt de Comando

node: entra dentro do node para execução no Prompt de Comando

node .\nomedoprojeto

## CONSOLE LOG E OPERAÇÕES ARITMÉTICAS

console.log(“Meu Primeiro programa”); -> “printf” \*O ; na linguagem JS é facultativo

const idade = 19;

const idadeSomada = idadeSomada + 2;

console.log(idade);

comentários -> //comentário

\*JS é uma linguagem *case sensitive*, ou seja, diferencia maiúsculas de minúsculas

Idade = 29; e idade = 19; são variáveis diferentes e nesse caso estão sem tipagem de declaração, o que não é interessante porque dificulta em achar e diferenciar os estados ao debugar um código.

console.log(parseInt("2") + parseInt("2")); -> converte a variável string em int

O operador + para texto no console.log concatena os valores (junta); os demais operadores a própria linguagem subentende que se trata de uma operação aritmética e realiza a operação: console.log(“10” / “2”);

const nome = "Anna";

const sobrenome = "Luiza";

console.log(nome + " " + sobrenome);

console.log(nome, sobrenome);

console.log(`Meu nome é ${nome}`);

## VARIÁVEIS

const -> constante e não varia ao longo do programa, ou seja, não pode receber atribuições posteriores

let -> variável que pode ser alterada ao longo do programa

\_variavel / #variavel -> representa as variáveis privadas do código, ou seja, as que não podem ser acessadas fora de uma classe

## LISTAS

const listaDestinos = new Array(

    `Salvador`,

    `Belo Horizonte`,

    `Montes Claros`

);

listaDestinos.push(`Curitiba`); //adiciona um item na lista após ela ser delcarada

listaDestinos.splice(1,1); //retira um item na lista após ser declarada, nesse caso o 2 é a posição de início e

//o 1 é a quantidade a ser deletada

console.log(listaDestinos);

console.log(listaDestinos[0]); //mostra o item na posição 0

## CLASSE

É o molde do programa, ou seja, sempre que precisar de um certo conjunto de informações a classe é utilizada. Exemplo: classe Cliente que contém os atributos padrões que todos as variáveis cliente1, cliente2.... precisam ter.

class Cliente{

    nome;

    cpf;

    agencia;

    saldo;

}

const cliente1 = new Cliente();

cliente1.nome = "Anna";

cliente1.cpf = 12345678923;

cliente1.agencia = 3044;

cliente1.saldo = 0;

console.log(cliente1);

A classe possui então o conjunto de atributos.

Cliente1 é um objeto e a classe o molde.

Quando um arquivo for criado para representar uma classe, o arquivo deve ser nomeado começando com letra maiúscula.

## FUNÇÕES

São os métodos da programação orientada a objetos, assim nelas são adicionados comportamentos para definir/manipular as variáveis do código.

Um ***return*** dentro de uma função sem nada atribuído é utilizado para “finalizar” o uso do método quando ele for aplicado, ex.: dentro de um ***if*** que representa o que eu não quero que a função execute caso seja verdadeiro.

## MÓDULOS NO JAVASCRIPT

Terminal -> npm init -> nome do pacote (pasta do projeto)

Isso cria o package.json dentro da pasta do projeto