**Modulo 1**

**Aula - Primeiros passos com javascript**

**Variáveis e operadores**

CTRL + ‘ para abrir o terminal

Para imprimir algo no terminal eu uso console que é a interface para acessar o terminal e o .log(‘’); que quer dizer imprima.

No javascript temos 2 formar para declarar variáveis. O let e o const (constante).

**Desafio**: Faça um programa para calcular o valor gasto de combustível em uma viajem. Você terá 3 variáveis, sendo:

1 – Preço do combustível.

2 – Gasto médio de combustível do carro por KM.

3 – Distância em KM da viajem.

Imprima o valor que será gosto para realizar a viajem.

Para casas decimais com 2 unidade eu chamo a função .toFixed(2), passando o parâmetro 2 entre parênteses e a função ainda faz o arredondamento automático caso seja necessário.

**Estruturas Condicionais**

CTRL + l para limpar o terminal

Os sinais de == ou === significa igualdade.

Para variáveis do tipo booleana eu tenho que colocar o prefixo de **is** antes do nome da variável. Ex: **isNumeroPar**.

**Desafio**: Faça um programa para calcular o valor de uma viajem.

Você terá cinco variáveis, sendo ela:

1 – Preço do etanol;

2 – Preço da gasolina;

3 – Tipo de combustível que está no seu carro;

4 – Gasto médio do combustível do seu carro por KM

5 – Distância em KM da viajem.

Imprima no console o valor que será gasto para realizar esta viajem.

Com o código selecionado eu dou o comando ALT + SHIFT + F para formatar o código.

**Praticando com variáveis, operadores e estruturas condicionais**

Biblioteca nativa de funções matemáticas do javascript para fazer cálculo ao quadrado da maneira correta. Ex: Math.pow(altura,2); que no caso é a potência ao quadrado da minha variável altura.

**Desafio3**: Faça um algoritmo que dada as três notas tiradas por um aluno em um semestre da faculdade calcule e imprima a sua média e a sua classificação conforme a tabela abaixo.

Media = (nota1 + nota2 + nota3) / 3;

Classificação:

1 – Media menor que 5 reprovado;

2 – media entre 5 e 7 recuperação;

3 – media acima de 7, passou de semestre.

**Desafio4**: O IMC - Indice de massa corporal é um criterio da organização mundial de saúde para dar uma indicação sobre a condição de peso de uma pessoa adulta.

Formulado IMC:

IMC = peso / (altura \* altura)

Elabore um algoritmo que dado o peso e altura de um adulto moatre sua condição de acordo com a tabela abaixo:

IMC em adultos, condição:

- Abaixo de 18.5 abaixo d peso;

- Entre 18.5 e 25 peso normal;

- Entre 25 e 30 acima do peso;

- Entre 30 e 40 obeso;

- Acima de 40 obesidade grave.

**Desafio5**: Elabora um algoritmo que calcule o que deve ser pago por um produto, considerando o preço normal da etiqueta e a escolha da condição de pagamento.

Utilize os códigos da tabela a seguir para ler qual condição de pagamento escolhida e efetuar o calculo adequado.

código condição de pagamento:

À vista debito, recebe 10% de desconto;

À vista no dinheiro ou pix, recebe 15% de desconto;

Em duas vezes, preço normal de etiqueta sem juros;

Acima de duas vezes, preço normal de etiqueta mais juros de 10%.

**Aula - Conhecendo funções javascript**

Funções são pequenos trechos de códigos que podemos chamar a qualquer momento.

Eu declaro uma função com a palavra *function* o *nome da função* com *(parametro){código}.*

Ex: function carro(){

Código;

}

Formar de chamar uma função: carro();

Uma função é como se ela fosse uma ‘variável’.

Eu posso colocar uma função dentro de uma variável.

Ex: main = function() {

Console.log(código);

}

No javascript as funções são como se fossem objetos.

Eu posso passar qual função vai ser resolvida primeiro (função imediatamente invocada) usando os parênteses (função) entre a função como é feito nos cálculos matemáticos usando a precedência para dizer qual cálculo vai ser executado primeiro.

Exemplo usando precedência:

(function main() {

Console.log(código);

})();

**Aula – Criando objetos e classes em javascript**

**Entendendo os objetos e classes**

Para atribuir um objeto eu uso os colchetes {} passando o valor do objeto.

Ex: let variável = {

nome: ‘Fabio Oliveira’,

idade: 30

};

Um objeto é uma coleção de valores. Ele agrupa valores e funciona com chave/valores.

EX: nome: ‘Fabio Oliveira’ que é chave/valor.

E para imprimir eu uso o console.log(variável.chave);.

Eu posso fazer incrementações ao objeto passando o nome do objeto.nome do incremento recebendo o incremento.

Ex: pessoa.altura = 1.60;

Para remover um valor basta passar delete objeto.chavepraremover;

Função dentro de objeto chamamos de método.

O this é uma palavra reservada de função e ela desrespeito ao contexto aonde esta função esta sendo executada. O this dentro do objeto ele assume o objeto.

Outro meio de acessar os valores do objeto através de uma sintaxe mais dinamicamente é criando uma variável recebendo o valor que eu quero pegar de dentro do objeto e eu imprimo colocando o nome do objeto com o nome da variável dentro de colchetes. Ou também passando a chave do objeto entre ‘’ direto dentro de colchetes ao invés da variável.

Ex: let atributo = ‘idade’;

Console.log(‘pessoa[atributo]’);

**Classes**

Eu defino classe passando a palavra class e o nome da classe.

Classe é uma definição do que deveria ser, e, uma instancia é uma ocorrência de uma classe.

O constructor é o que acontece quando uma classe é instanciada. Eu posso colocar parâmetros dentro do construtor passando esses parâmetros dentro dos parênteses. Ex: constructor(parâmetros){};.

Sempre que uma classe é instanciada ela passa pelo construtor.

**Aula Arrays e estruturas de repetição**

**Arrays**

As listas (arrays) são representadas por colchetes [].

O Método .push(); para adicionar item ao array.

O método .pop(); remove o ultimo item do array mas não remove o item, apenas tira ele do array.

O método .shift(); remove o primeiro item do array mas não remove o item, apenas tira ele do array.

Para eu saber o tamanho da lista eu uso o .length();.

**Estruturas de repetição**

**For**

O for é usado para percorrer uma quantidade específica de vezes.

Para percorrer eu preciso de uma variável para controlar qual e a interação que eu estou.

for (1-variável de controle; 2-estrutura condicional; 3-o que vai acontecer depois da execução do for) {

}

Depuração de código é percorrer o código para ver se existe algum erro ou algo feito errado e se a execução do código está correta. Para executar a depuração eu vou na opção do vscode em depurar e executar>node.js e colocar o break point (pontinho vermelho ao lado do número de linhas do código).

**Aula Importação e exportação com javascript**

O termo mocado significa simulação.

O gets é para entrada de dados.

O print é saída de dados.

A função require(‘./nome do arquivo’); serve para fazer a importação (o uso) de código de outro arquivo. É basicamente importação do que esta sendo exportado pelo outro arquivo.

O module.exports = {} serve para fazer a exportação.