**ENTRADAS E FUNÇÕES**

*Organizando o entendimento em 3 passos*

1 )Agora que nós já sabemos o que é dado e o que é informação,

pense em uma receita de bolo. Os ingredientes dessa receita são os dados da receita e a receita é o algoritmo que vai criar o comportamento de cozinhar. Mas primeiro, precisamos saber dos ingredientes da receita.

2) Após tomarmos conhecimento dos ingredientes, a gente identifica esses itens a serem utilizados na receita e os junta por categoria. Por exemplo, os vegetais de um lado, laticínios de outro, farinhas em outro, etc ...

3) Após essa categorização dos itens será preciso obter o comportamento de cozinhar.

Trazendo para o entendimento de um algoritmo, o passo 1 seria relacionar as entradas e saídas da receita com dados. Esses são os dados da receita. Cada item que será utilizado e cozinhado.

O passo 2 cria a relação entre a representação de dados no mundo real e sua categorização em tipos da computação. Da mesma forma que separamos os vegetais para serem tratados como vegetais, os números inteiros serão tratados como números inteiros por um algoritmo, ou seja, não busque saber quantas casas decimais tem um número inteiro porque números inteiros não tem casas decimais.

O passo 3 organiza e processa as entradas em uma função. Seria justamente o comportamento de cozinhar da nossa receita. Os itens da receita são identificados, categorizados e processados de forma organizada durante o cozimento.

Sobre as funções, a gente pode observar essa imagem: Uma função é um algoritmo representado na sistemática computacional, mas como tudo surge do mundo real, podemos dizer que uma função é um espremedor de laranja. Porque? Porque ele só sabe espremer laranjas .. Ok, você pode espremer limões com ele, mas apenas porque limões e laranjas possuem o mesmo formado. Então podemos dizer que esse objeto tem a função de espremer qualquer coisa que se pareça com laranja. Essa é a função dele. Essa é a responsabilidade que ele precisa cumprir com maestria. Tente espremer bananas e verá que o objeto não trará o resultado esperado.

Podemos dizer que um algoritmo (função) NÃO é um canivete suíço. Canivete suíço corta unha, papel, abre garrafa, rasgar coisas grossas, coisas finas, desaparafusar, etc ...

Mas podemos dizer que um algoritmo (função) SIM seja cada parte de um canivete suíço. Exemplo: Você só conseguirá desaparafusar com a chave de fenda contida nele e não poderá utilizá-la para cortar papéis, pois essa função é realizada pela pequena tesoura que existe nele.

Perceberam a diferença?

Até o próximo vídeo.

**Representando funções no computador**

A função que está sendo exibida existe para nos dizer que ela é uma função. Ou seja, ela imprime uma frase “Eu sou uma função” no console do navegador. Essa sintaxe *function* + *nome da função + abre e fecha parenteses + abre e fecha chaves* é exigencia da linguagem que estamos utilizando. Assim como o *console.log* é uma propriedade que é utilizada para imprimir dados no console no navegador.

function euSouUmaFuncao () {

console.log(“Eu sou uma função”)

}

euSouUmaFuncao();

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Associando entradas com variáveis**

Estamos diante de uma função que existe para juntar uma palavra desconhecida com uma frase pré definida pela função. Ou seja, essa entrada será uma variável criada durante a chamada dessa função. Nesse momento estamos diante de um estudo sobre variáveis que nós já sabemos o que significa.

Podemos observar que o valor dessa variável é fornecido quando o uso dessa função for conveniente, ou seja, quando alquém desejar somar uma palavra nessa frase “Especialista avante ao “.

Podemos atribuir qualquer valor para essa variável: “Ar”, “Vento”, “Espaço”, “Futuro”, etc ...

function juntarUmaPalavraComUmaFrase (palavra) {

console.log("Especialista avante ao " + palavra)

}

juntarUmaPalavraComUmaFrase("ar");

**Escopo de variáveis**

O escopo de variáveis diz respeito ao acesso ao valor de uma variável. Por exemplo: Para a função abaixo exibir o resultado do cálculo, ela precisa localizar as variáveis na memória para chegar a valor de cada variável. Uma variável é declarada justamente para informar à função que ela existe e qual o valor dela.

Perceba que no código abaixoduas variáveis são declaradas cada uma com um valor. O escopo dessas variáveis é global por não pertencer à função e mesmo assim conseguirem ser acessíveis por ela.

var primeiroNumero = 5;

var segundoNumero = 6;

function somarDoisNumeros () {

console.log(primeiroNumero + segundoNumero)

}

Se fizermos uma modificação neste código e declararmos as variáveis dentro da função, o mesmo resultado será exibido, só que agora, as variáveis passam a ser reconhecidas pela estrutura somente dentro da função.

function somarDoisNumeros () {

var primeiroNumero = 5;

var segundoNumero = 6;

console.log(primeiroNumero + segundoNumero)

}

Se tentarmos utilizar o valor dessas variáveis fora da função com um *console.log(primeiroNumero + segundoNumero)* já não mais poderão ser acessadas e um erro será exibido no console.