Aula 06 de Linguagem Java Script

O simbolo de + no JS é a ação de concatenação (ligação ou ato de Juntar), usaremos sempre que quisermos unir um comando com outro;

EX: var nome =window.prompt(‘Qual é o seu nome?’)

window.alert(‘é um prazer conheçe-lo’ + nome)

o resultado fica assim...

é um prazer conhece-lo Thainã(ou o nome que for digitado na caixa)

Sendo assim temos o entendimento do uso da concatenação, e podemos usar nos momentos que forem necessarios.

Ja para fazermos soma de numeros dentro do programa, existem dois codigos que fazem esse calculo para nós, que são o “parseint” e o “parsefloat”, mas qual é a diferença desses dois?

, Parseint: é usado para soma de numeros inteiros Ex: 5, 7, 10...

Parsefloat: é feito para calculo de numeros quebrados. Ex: 5.5, 7.25, 8.70...

E o codigo completo para funcionar essa soma é:

Number.parseint

Number.parsefloat

Assim saberemos que estamos avisando ao programa que queremos somar esses dois valores, vamos usar uma codificação completa para compreendermos melhor;

Ex: var nome = Number.parseint(window.prompt(‘Digite um numero’))

Nós começamos avisando o programa que iremos querer adicionar uma variavel com o comando “var” logo apos nomeando essa variavel de nome para que tenhamos um espaço na memoria do computador que salve os dados digitados, logo apos iremos adicionar o “Recebe (=)”, para avisar o que iremos colocar dentro da variavel, assim podemos por em pratica os comandos novos que aprendemos nesta aula que usaremos para somar os valores aplicados aqui, como o “Number.parseint”,então apos definir a variavel, vamos digitar o comando Number.parseint, pois queremos somar numeros inteiros neste caso, logo após iremos abrir parenteses e chamar o prompt com o “window.prompt”, para gerar uma caixa que nos permite digitar um valor, abrir parenteses novamente e abrir aspas colocando o valor que aparecerá como pergunta dentro do prompt, ficando:

var nome =Number.parseint(window.prompt(‘Digite um valor’))

lembrando que para conseguirmos somar esses numeros precisaremos criar duas variaveis com os mesmos comandos, e nesse caso indicamos identificar essas variaveis com “N1” e “N2” para não dar erro de syntax,ficando assim:

var n1 = Number.parseint(window.prompt(‘Digite um numero’))

var n2 = Number.parseint(window.prompt(‘Digite outro numero’))

sendo assim temos duas variaveis capazes de gerar soma entre os dois numeros digitados pelo usuario, mas para que isso funciona mais uma variavel precisa ser criada.

Agora quase finalizando esse script de soma, precisamos justamente indicar para o programa que queremos que ele some essas duas variaveis,como faremos isso?

Simples criaremos uma variavel com o atributo “S”:

Var s = n1 + n2

Criando a variavel de soma, indicaremos ao programa quais variaveis ele deve somar, por isso indicaremos as variaveis n1 e n2 que é o nosso objetivo.

Feito isso iremos para o ultimo passo desse script de soma entre variaveis, que é disparar um alerta mostrando o valor dessa soma para o usuario:

window.alert(‘A soma dos valores é’ + s)

O que fizemos aqui foi criar um comando de alerta, dizendo que o valor dessa soma foi tanto, para isso vamos pegar o comando window.alert, entre aspas e parenteses digitar a frase que nos indicará o valor dessa soma.

Gerando assim o script completo , que ficara assim:

Var n1 = Number.parseint(window.prompt(‘Digite um numero’))

Var n2 = Number.parseint(window.alert(‘Digite outro numero’))

Var s = n1 + n2

window.alert(‘A soma dos valores é’ + s)

Lembrando sempre do sinal de concatenação para indicar que o programa deve pegar a variavel “s” para mostrar o resultado desta soma.

Para somar numeros quebrados que são chamados de “Float”, usaremoso Number.parsefloat que ele gerará essa soma de numeros quebrados.

Var n1 = Number.parsefloat(window.prompt(‘Digite um numero’))

Var n2 = Number.parsefloat(window.alert(‘Digite outro numero’))

Var s = n1 + n2

window.alert(‘A soma dos valores é’ + s).

Se ao invés de numeros eu quiser converter esse resultado para string, basta colocar

window.alert(‘A soma dos valores é’ + s.tostring()).

Com isso ele converterá esse resultado para uma string.

Como formatar uma nova string?

Existem diversos tipos de formatação mas nesta aula citaremos apenas 3 tipos:

s.length: é o comando para usar para saber quantos caracteres existem na palavra que o usuario digitar. Ex: nome.length

s.toUppercase(): é o comando para usar pra deixar todas as letras maiusculas. Ex: nome.toUppercase

s.toLowercase(): é o comando usado para deixar todas as letras minusculas. Ex: nome.toLowercase

Lembrando que não podemos nos esquecer dos () no final do Uppercase e do Lowercase, pois se não adicionarmos esses parenteses no final o comando se torna invalido ou incompleto.

Usamos tambem dois comandos novos:

Document.write: Serve para adicionarmos um texto na pagina em que estamos codificando.

Agora vamos aprender como formatar numeros.

Para fazer isso temos que aprender alguns comandos:

Vamos criar uma variavel...

Var n1 = 1545.5

Para arrumar esse 5 do final e tornar ele de fato decimal utilizaremos um comando:

toFixed(2): esse comando solicita ao programa que adicione o 0 para que ele fique decimal, então o comando completo fica assim:

var n1 = 1545.5

n1.toFixed(2)

então ele dara o resultado de 1545.50

Mas você deseja colocar uma virgula no lugar do ponto, que é a forma convencional que utilizamos aqui no Brasil, existe um comando pra isso:

n1.toFixed(2).replace(‘.’ , ‘,’)

ele nos dara esse resultado:

1545,50

Mas se quisermos adicionar um valor monetario a esse numero, como fazemos?:

Usaremos um comando que é bem longo portanto é importante decorar bem e treinar bastante com ele.

n1.toLocaleString(‘pt-BR’, {style: ‘currency’, currency: ‘BRL’)

n1.toLocaleString(‘pt-BR’, {style: ‘currency’, currency: ‘USD’)

n1.toLocaleString(‘pt-BR’, {style: ‘currency’, currency: ‘EUR’)

Resumidamente, esse comando converterá esse numero em um valor de moeda, no exemplo acima, usamos três moedas Real, Dolar, Euro;

Então sempre que precisarmos adicionar um valor monetario a nosso programa essa linha deve ser usada.