> n1.toFixed(2)

'1545.40'

> //Se quisermos aumentar as casas decimais usamos esse truque.

 //O mais pode servir, para adição ou concatenação se forem string

      //Serve para adição se ambas variaveis forem number

      //O window.prompt, ele recebe um valor por padrão string, o que devo fazer é converter em number. A primeira maneira é Number.parseInt(n) ou simplismente parseInt, se for para converter para números decimais é Number.parseFloat(n);

      //É importante saber qual é o tipo de dados que você esta precisando. Mas se quiser que o javascript decida deixe apenas Number(n)

      //Para converter um number para string, você pode usar String(o valor que quero converter), ou apenas coloco  variavel.toString()

      //Podemos treinar com template string e usando aspas simples vamos treinar isso.

      //Uma das formas de formatar uma string é nomeDavariavel.length ele vai informar quantos caracteres a string possui

      //nomedaVariavel.toUpperCase() coloca tudo para maiuscula

      //nomedaVariavel.toLowerCase() coloca tudo para minúscula

> n1.toFixed(2).replace(',','.')

'1545.40'

n1.toLocaleString('pt-br', {style:'currency',currency:'USD'}).

**Operadores em javascript**

Em javascript ele carreta vários operadores, mas que nós abordaremos são: aritméticos, atribuição, relacionais, lógicos, ternário.

Os operadores aritméticos usados para fazer cálculos em javascript são : +, -, \*,/,%,\*\*.

Operando são algarismo que vão fazer nossos operadores funcionarem.

Operadores binários são operadores de dois operandos. E a % é usada para calcular o resto da divisão inteira. Tome bastante cuidado no uso de operadores porque existe uma coisa chamada de precedência. Quando você programa a linguagem só da erro, se tiver um erro sintático.

Em qualquer linguagem nós temos a seguinte ordem de precedência : (), \*\*, \*,/,% e por fim temo as + e -.

O sinal de atribuição é igual =. E temos as auto-atribuição.

> n=n+4 exemplo de uma auto-atribuição.

Temos a possibilidade de simplificar as nossas auto-atribuição

> n+=8 exemplo simplificado.

Temos operadores de incremento > x=x+3 assim incrementou, lembrando que antes de usamosr, precisamos de uma variável simples.

> ++n

11

> --n

Temos esta operação que da o resultado na hora . E temos o resultado abaixo que da algo diferente:

N++, ele vai mostrar alteração depois.

Os operadores relacionais de javascript são o mesmo com o da maioria das linguagens são:<,>,>=, <=,==, !=. A igualdade é ==, != diferente.

Quando temos o operadores aritméticos e relacionais na mesma linha, primeiro fazemos o aritméticos e depois fazemos os relacionais.

5=='5' Para o javascript isso é a mesma coisa, o sinal de igualdade não verifica o tipo, mais sim o numero. Temos o operador de identidade restrita que são no total três sinais de iguais.

No javascript temos os seguintes operadores relacionais:

! = Quer dizer negação.

&&= Isso é uma conjução (é o nosso e lógico)

||=disjunção (é o nosso ou lógico)

! é conhecido como operador onario ou seja ele tem apenas um operando, isso é depois da exclamação ou eu coloque true ou false.

Uma coisa verdadeira é uma coisa falsa.

Uma coisa falsa é uma coisa verdadeira.

&& para a conjução ele é um operador binário,

True && true ex: pode ser os dois verdadeiros

True && false ex: resultado falso

False && true ex: resultado falso

False && false ex resultado falso

No operador de conjunção só me satisfaz se as duas forem verdadeira.

A disjunção também é um operador binário:

True || true resultado verdadeiro

True || false resultado verdadeiro

False || true resultado verdadeiro

False || false resultado falso

Na disjunção basta que uma das opções seja verdadeira vamos ter um resultado verdadeiro.

Quando temos operadores aritméticos, relacionais e lógicos, primeiro é prioriado operadores aritméticos, segundo relacionais e ultimo lógicos.

Se numa expressão agente tiver um ! (não), && (e) e um ||(não) primeiro vai ser priorizado o não.

Normalmente os operadores lógicos funcionam com operadores relacionais.

Os operadores tem a seguinte ordem de precedência:

Aritmetico

Relacionais:eles não têm uma preferência o que chegar primeiro é que executa.

Temos os lógicos que primeiro vem o não !, e &&, || ou.

O operador ternário é representado por dois sinais o ? e o :. Ele se chama ternário porque ele tem 3 partes.

r=idade >=18?'Maior':'Menor'

Entendendo DOM

Node exec vai fazer o javascript rodar pelo node.

DOM é um acronomo para Document Object Model, em português modelo de objectos para documentos, são conjuntos de objectos no navegador que vão abrir conjuntos de objectos internos no nosso web site. Ele não funciona dentro do node.js ele funciona quando estou rodando o site dentro do navegador.

Ele começa tudo em uma raiz que é chamada de window e dentro do window temos três tipos de objectos: location, document, history.

Location : qual é a localização do seu site, inclusive a URL.

Document: que é o document atual.

History: que vai guardando de onde vem e onde vai.

Dentro do document existe a parte de html, no html temos o head e body.

Dentro de head nós temos meta title e entre outros.

E dentro do body temos h1 p e entre outras coisas que podemos colocar.

window.document.write(window.document.charset)

desse jeito mostra o caractere que é usado no site.

window.document.write(window.navigator.appName)

O nome do navegador, ele vai dizer Netscape.

window.document.write(window.document.URL)

Ele vai mostrar a sua URL.

Com o Dom nós podemos selecionar de duas formas:

Por Marrca, por ID, por Nome, por classe, por selector.

Para selecionar por marca, nós usamos getElementsByTagName().

//Isso vai fazer apenas que eu marque um paragráfo

p1=document.getElementsByTagName('p')[0]

Essa concheta indica qual é o paragrafo que pretendemos acessar. O innerText

O innerText ele reproduz o texto, mais o innerHTML ele coloca o texto do jeito que foi criado.

Podemos identificar por Id (getElementById() , por nome getElementsByName() então podemos usar o conchete quando tenho mais de um objecto , por classe getElementsByClassName().

d.innerText='Estou amando javascript Manuel' //Originalmente esta click em min,mas eu alterei com dom e atualizando nomedaVariavel.innerText='AS Strings que desejo'

 var d=window.document.querySelector('div.msg').style.background='red'

O método mais adequado é o document.querySelector(digite o que você quer manipular ).

Lembrando é um comando novo, portanto para navegadores mais antigos ele não vai rodar.

Eventos de DOM

O Dom facilita o javascript ter acesso aos elementos html.

Eventos é qualquer coisa que pode acontecer com um elemento.

Por exemplo quando o mouse esta dentro da div,nós podemos colocar mouse enter, o que significa que mouse esta dentro, se nós movermos o mouse na div podemos colocar mousemove, se clicar e segurar a o elemento temos mousedown (ele dispara). No momento em que nós soltamos a div temos mouseup, temos também o evento click, se eu disparar o mouse para fora agente vai ver mouseout.

Existem vários eventos, tem eventos especifico para cada situação e nem todos eventos funciona para todos elementos.

Para trabalharmos com eventos precisamos compreender o que são funções. São conjuntos de códigos que vão acontecer quando aquela div vai disparar ou começar a trabalhar.