Evolução da linguagem **JavaScrip**

* A internet surgiu em 1970 – EUA → Arpanet;
* 1993 surgiu o www;
* 1994 surgiu a empresa Netscape;
  + A empresa Netscape criou a linguagem JavaScript devido o sucesso da linguagem Java na época (imitação);
  + Alguns comandos do Java e do JavaScript são parecidos porque as duas linguagens foram inspiradas na linguagem C;
  + Java e JavaScript são totalmente diferentes
* A Microsoft fez uma imitação do JavaScript chamando-a de Jscript;
* A Netscape, pensando na bagunça que poderia virar, contratou a empresa ECMA em 1995 para fazer uma padronização da linguagem JavaScript;
* A linguagem padronizada pela ECMA passou a ser chamada de ECMAScript;
* ECMAscrip é o JavaScript padronizado;
* 2002 cria-se o Firefox;
* 2008 o Google criou o Google Chrome;
  + Usa o V8, motor por baixo do navegador que roda JavaScript
* 2010 → Node JS= usado para rodar JavaScript fora do navegador

ECMAScript

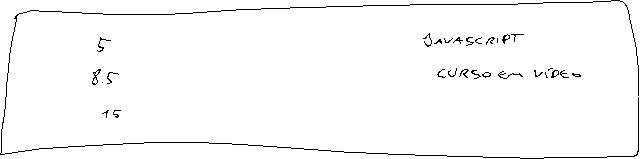
* 1.0 (1997)
* 2.0 (1998)
* 3.0 (1999)
* ES5 (2009)
* ES6 (2015)
* ES (2016) e por aí vai



Aula 5 – Curso JS



* Estabelecer variáveis



* + Identificadores → nomes das variáveis
    - Podem começar com letra, $ ou \_



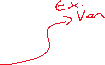
* + - Não podem começar com números
    - É possível usar letrar ou números
    - É possível usar acentos ou números
      * Pode-se criar uma variável com o nome contendo acentos ou símbolos
    - Não pode conter espaços
    - Não podem ser palavras reservadas
      * Ex.: não é possível criar uma variável com o nome “var”
* Comandos Node.JS
  + Limpar a tela
    - Ctrl + L
  + Sair do node
    - .exit
  + Sair do prompt do VS code
    - exit
      * É obrigatório executar para não deixar vários prompts abertos em background



* Dicas



* + Maiúsculas e minúsculas fazem diferença



* + Tente escolher nomes coerentes para as variáveis



* + Evite se tornar um ‘programador alfabeto’ ou um ‘programador contador’



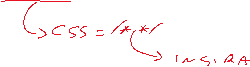
* + Para fazer um comentário na parte do HTML = <!-- -- >



* Data Types



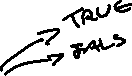
* + Number



* + - Infinity



* + - NaN
  + String
  + Boolean



* + Null



* + Undefined
  + Object



* + - Array
  + Function



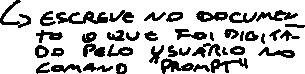
Comando utilizado: typeof



* Conversão de string para number



* + Number.parseInt(n)
  + Number.parseFloat(n)



* Conversão de number para string
  + String(n)



* + n.toString()
* Tamanho de uma string
  + s.length → retorna quantos caracteres tem a string
* Jogar para maiúscolas
  + s.toUpperCase()
* jogar para minúsculas
  + s.toLowerCase()
* formatar números



* + ex.: var n1 = 1545.5
    - > n1.toFixed(2)

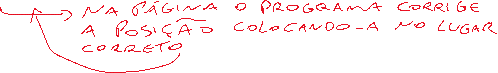


* + - * 1545,50
  + Ex 2.: > n1.toFixed(2).replace(‘.’, ‘,’) → substitui o ponto pela vírgula
    - * 1545,50

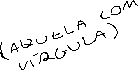


* + n1.toLocaleString(‘pt-BR’, {style: ‘currency’, currency: ‘BRL’})
    - R$ 1,545.50

Aula 7



Operadores do JavaScript



* Aritméticos



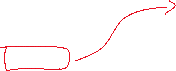
* + ( + )
  + ( - )



* + ( \* )



* + ( / ) → 5 / 2 = 2.5



* + ( % ) → divisão inteira. Ex.: 5 % 2 = 1



* + ( \*\* ) → exponenciação



**Ordem de precedência de operadores**



* ()
* \*\*



* (\*) (/) (%) → se aparecerem mais de um destes operadores, da esquerda para a direita de quem aparecer primeiro



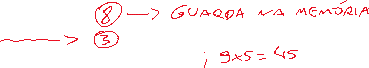
* (+) (-)



* Atribuição
  + var a = 5 + 3



* + var b = a % 5

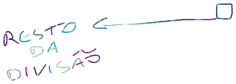


* + var c = 5 \* b \*\* 2



* + var d = 10 – a / 2
  + var e = 6 \* 2 / d
  + var f = b % e + 4 / e



* + Auto atribuições



* + - var n = 3



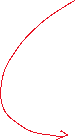
* + - n = n + 4
    - n = n – 5



* + - n = n \* 4



* + - n = n / 2



* + - n = n \*\* 2
    - n = n % 5



* + Simplificando...



* + - n += 4



* + - n -= 5
    - n \*= 4
    - n /= 2
    - n \*\*= 2
    - n %= 5



* + incremento
    - var x = 5
    - x = x + 1 x++



* + - x = x – 1 x--



* Relacionais
* Lógicos
* Ternário

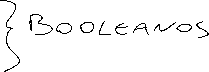
**Aula 8**



* Operadores Relacionais
  + >
    - 5 > 2 → true



* + <



* + - 7 < 4 → false



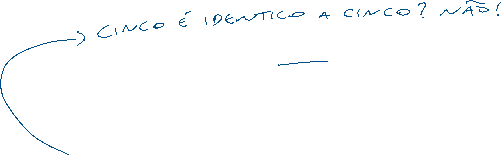
* + >=



* + - 8 >= 8 → true
  + <=
    - 9 <= 7 → false
  + ==



* + - 5 == 5 → true
  + !=
    - 4 != 4 → true
* **Operadores de Identidade**
  + 5 == 5 → true
  + 5 == ‘5’ → true



* + - Para o JavaScript os dois valores têm a mesma grandeza
    - Para saber se os dois valores são “idênticos”, utiliza-se === (três sinais de igual)



* + 5 === ‘5’ → false



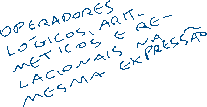
* + 5 === 5 → true
  + 5 != ‘5’ → false
  + 5 !== ‘5’ → true



* **Operadores Lógicos**
  + **!** → negação (não)
    - Operador unário. Só aceita true ou false
  + **&&** → conjunção (e)
    - Operador binário. Aceita dois valores
      * True && true → só me satisfaz se os dois valores forem true
      * True && false
      * False && true
      * False && false
  + **||** → disjunção (ou)
    - Também operador binário. Aceita dois valores.
      * True || true → true
      * True || false → true
      * False | | true → true



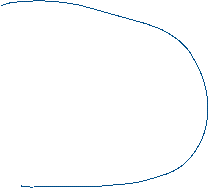
* + - * False ||false → false
      * Basta que um valor seja true que o resultado é true
    - Exemplos:
      * var a = 5
      * var b = 8



* + - * a > b && b % 2 == 0 → false



* + - * + 1º fazer os operadores aritméticos (%)
        + 2º fazer os operadores relacionais (>, ==)
        + 3º fazer os operadores lógicos (&&)
        + a > b && b % 2 == 0



* **Ordem de precedência dos operadores lógicos**
  + 1º → ! (não)



* + 2º → && (e)
  + 3º → || (ou)



* + Em uma expressão completa



* + - A precedência é



* + - * Aritméticos



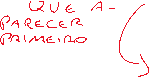
* + - * Relacionais



* + - * Lógicos



* + - * + !



* + - * + &&
        + || → (Alt + 124)



* Operador ternário
  + ? e : na mesma expressão



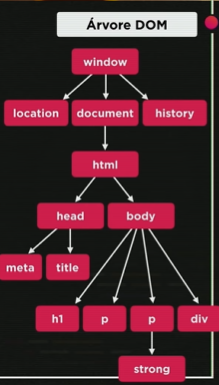
* + Junta três operandos
  + Exemplo: (fazer no terminal nodejs)
    - var média = 5.5



* + - média > 7?’Aprovado’:’Reprovado’
    - → ‘Reprovado’

Aula 9



* O que é DOM?
  + Document Object Model
  + Conjunto de objetos dentro do navegador para acessar objetos
  + O DOM está presente quanto estamos rodando JavaScript dentro do navegador.
    - Não funciona (DOM) no NodeJS
  + É importante fazer a Árvore DOM
    - 



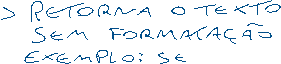
* + Selecionando os elementos da árvore DOM
    - Por Marca



* + - * getElementsByTagName()
      * Exemplos:
* var p1 = window.document.getElementsByTagName('p')[0]
* var p2 = window.document.getElementsByTagName('p')[1]
* window.document.write(p2.innerText)
  + - * + innerText → escreve o texto que está dentro do parágrafo



* + - Por ID
      * getElementById()
    - Por Nome



* + - * getElementsByName()



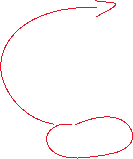
* + - Por Classe



* + - * getElementsByClassName()



* + - Por Seletor



* + - * querySelector
        + faz a seleção através de elementos CSS

Aula 10

* Eventos DOM
  + Evento é tudo que pode acontecer com a div
  + mouseenter
  + mousemove
  + mousedown
  + mouseup
  + click
  + mouseout
* funções



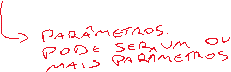
* + function ação(param) {



bloco



}



* rever as aulas 9 e 10

Aula 11 – Condições



* para imprimir um texto na tela com o NodeJS
  + console.log()
* pedir dados ao usuário

name="txtvel" id="txtvel"> km/h

    <input type="button" value="Verificar" onclick="calcular()">

