Javascript

Prof. Rodrigo Macedo

Escopo do Curso

- Histórico e característica da linguagem.
- Variáveis.
- Tipos primitivos.
- Array e Objetos.
- Funções.
- Eventos.
- Questões de concursos

Javascript

- Linguagem de script que trabalha para desenvolvimento web.
- Feita em 1995 para o Netscape 2.0 sob o nome de Mocha depois LiveScript e por fim, Javascript (segundo a especificação ECMA Script).
- O idealizador da linguagem foi Brendan Eich.
- A Microsoft adotou Javascript no IE 3.0.

Características da Linguagem Javascript

- Suporte universal aos navegadores web modernos.
- Linguagem multiparadigma: estruturada, orientada a objeto, funcional, etc.
- Linguagem fracamente tipada (problemas na comparação entre valores).
- Tipagem dinâmica.
- É uma linguagem case sensitive. Var1 é diferente de var1.
- Obs: Javascript e Java são linguagens totalmente diferentes. Linguagem fracamente tipada.

Características da Linguagem Javascript

- É uma das três linguagens mais importante no desenvolvimento web.
- São elas:

1. HTML para definir o conteúdo de uma página web.
2. CSS para especificar o layout de uma página web.
3. Javascript para desenvolver o comportamento de uma página web.

Com Javascript, podemos interagir entre os componentes de uma página web por meio do DOM (Document Object Model).

Integração HTML e Javascript

- Há duas formas de integrar código HTML com Javascript:

1. Interno: Nessa opção, o código Javascript é escrito dentro do arquivo HTML.
2. Externo: Nessa opção, o código Javascrípt é escrito num arquivo separado e linkado ao código HTML.

Exemplo 1

```
<!DOCTYPE html>
<html lang="pt-BR">
    <head>
        <meta charset="UTF-8" />
        <title>Wikipédia</title>
        <script>
            window.onload = function() {
                document.getElementById("hello").addEventListener("click", function() {
                    alert("Bem-vindo à Wikipédia!");
                }, false);
            };
        </script>
    </head>
    <body>
        <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
        <button id="hello">Dizer "Olá"</button>
    </body>
</html>
```

Exemplo 2

test.html | test.js

```
1  <html>
2      <head>
3      </head>
4      <body>
5          <script type="text/javascript" src="test.js"> </script>
6      </body>
7  </html>
```

Arquivo HTML

Arquivo Javascript

test.html | test.js

```
1  alert("Hello World");
```

Resultado da execução no navegador

Trabalhando com Variáveis

Variável é um nome qualquer para se atribuir um valor ou dado. Pode conter texto, um número inteiro, um número fracionário, um array, um valor booleano, etc.

- Regras na declaração de variáveis:

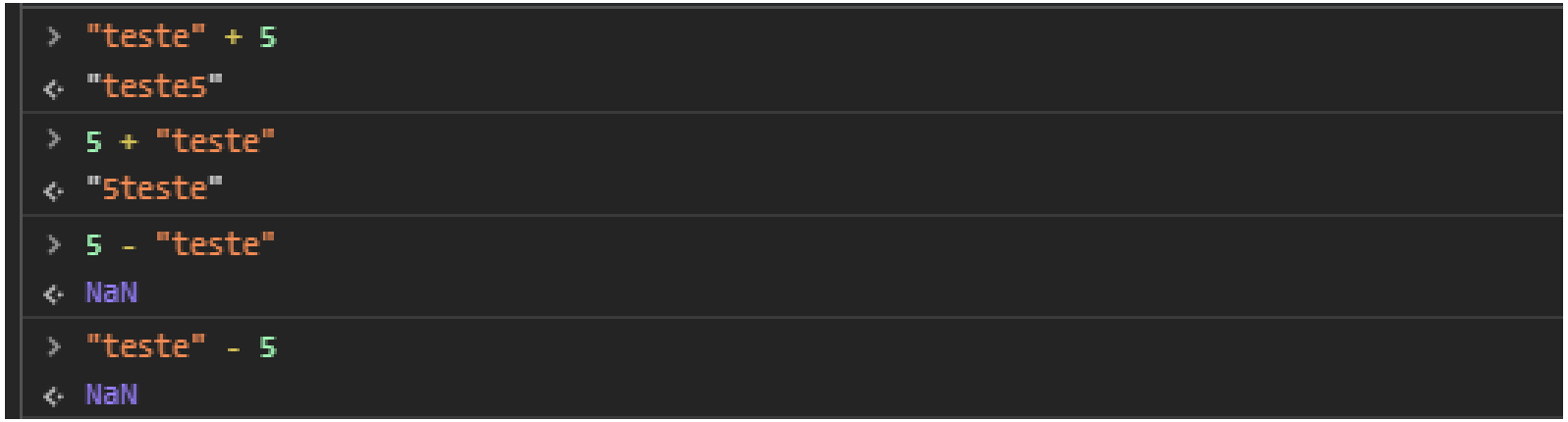
1. Deve iniciar com um dos três caracter:
   - Letra
   - Underscore (_)
   - Cifrão ($)
2. Os caracteres seguintes podem ser letras maiúscula ou minúscula, números, underscore ou caractere cifrão.

A partir da versão 1.5 do Javascript, permite-se o uso de letras acentuadas, dígitos e demais caracteres Unicode em nomes de variáveis.

Trabalhando com Variáveis

Exemplos:

- $v
- _teste
- nome_valido
- teste123
- Javascript é case sensitive. Variável teste é diferente de Teste.
- Javascript é uma linguagem fracamente tipada, ou seja, permite operações com variáveis de tipos diferentes.



Declaração de Variáveis

- É possível declarar variáveis usando três formas:

Forma 1:

var a = 30; var b= “Hello”; var c = 1.5;

Forma 2:

var a = 30;

var b = “Hello”;

var c = 1.5;

Forma 3:

var a = 30, b=“Hello”, c=1.5;

Escopo de Variáveis

- O escopo de uma variável, diz respeito a região do script no qual a variável assume o valor que foi a ela declarado.

Podem se dividir em:

1. Escopo local: Quando seu valor é reconhecido somente na região do script no qual ela foi declarado.
2. Escopo global: Quando seu valor é reconhecido em qualquer região do script.

Escopo de Variáveis - Exemplo

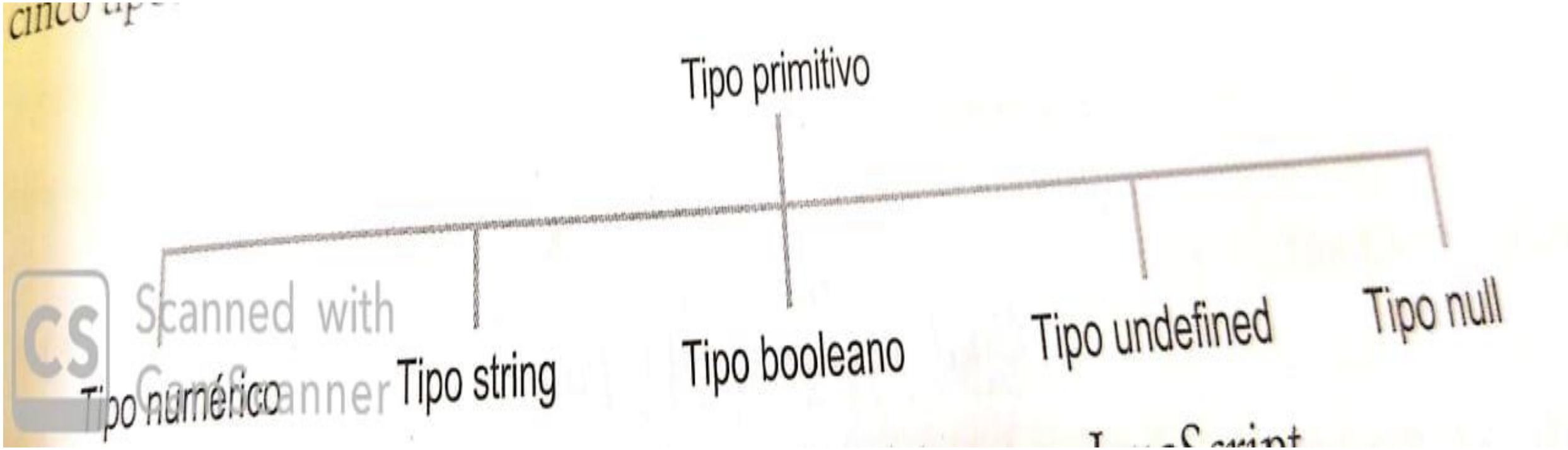
```
<script type="text/javascript">
    a=1;
    funcaoUm = function() {
        var a = 2;
        alert (a); //exibe o valor 2
    };
    funcaoUm(); //executa a funcaoUm

    funcaoDois = function(){
        alert(a); //exibe o valor 1
    };
    funcaoDois(); //executa a funcaoDois
        //faz alguma coisa....
        alert(a); //exibe o valor 1
</script>
```

Palavras Reservadas

abstract
boolean
break
byte
case
catch
char
class
const
continue
debugger
default
delete
do
double
else
enum
export
extends
false
final
finally
float
for
function
goto
if
implements
import
in
instanceof
int
interface
long
native
new
null
package
private
protected
public
return
short
static
super
switch
synchronized
this
throw
throws
transient
true
try
typeof
var
volatile
void
while
with

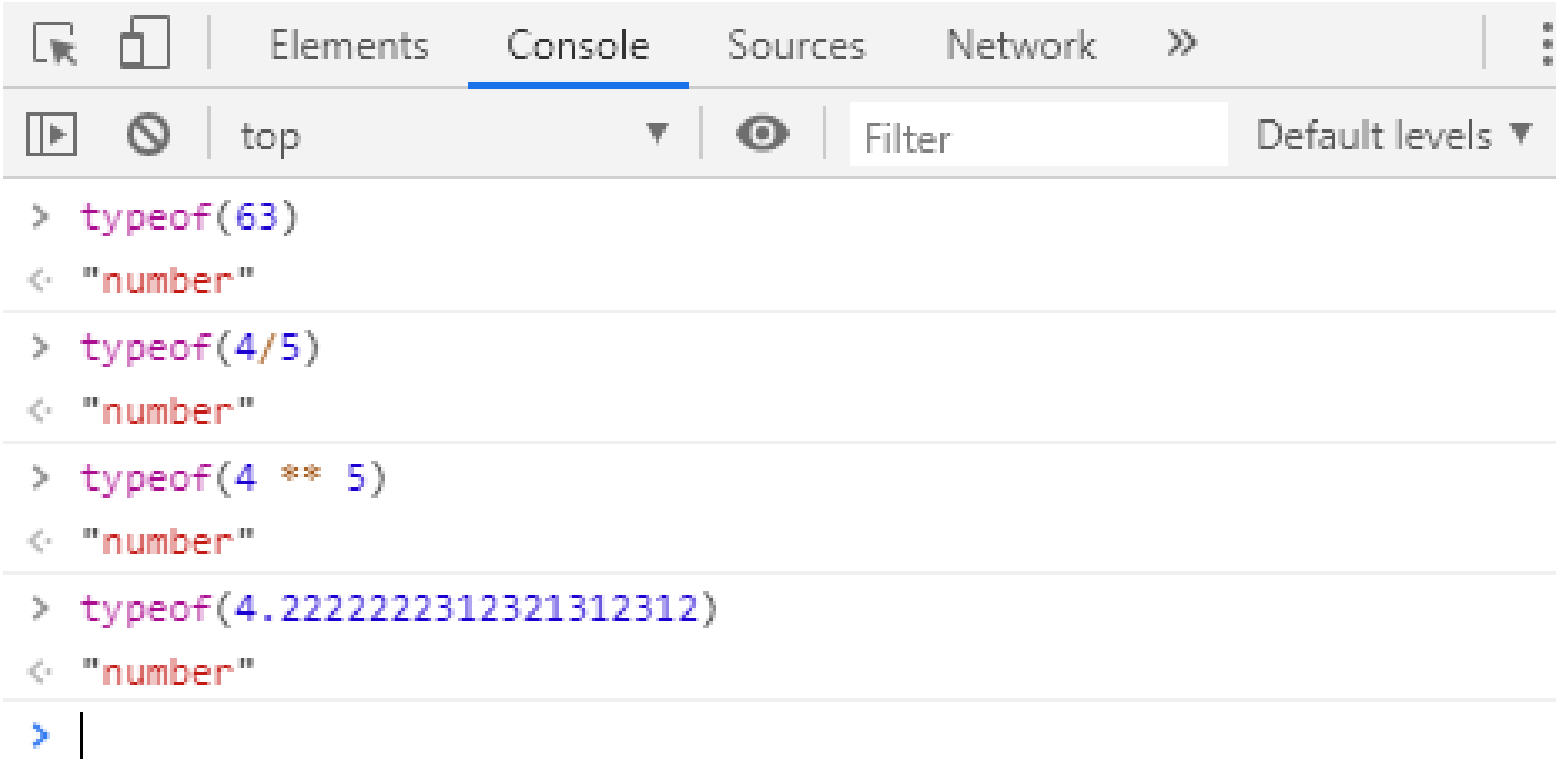
Tipos Primitivos

- Representam as possibilidade de tipificação nos dados em Javascript.



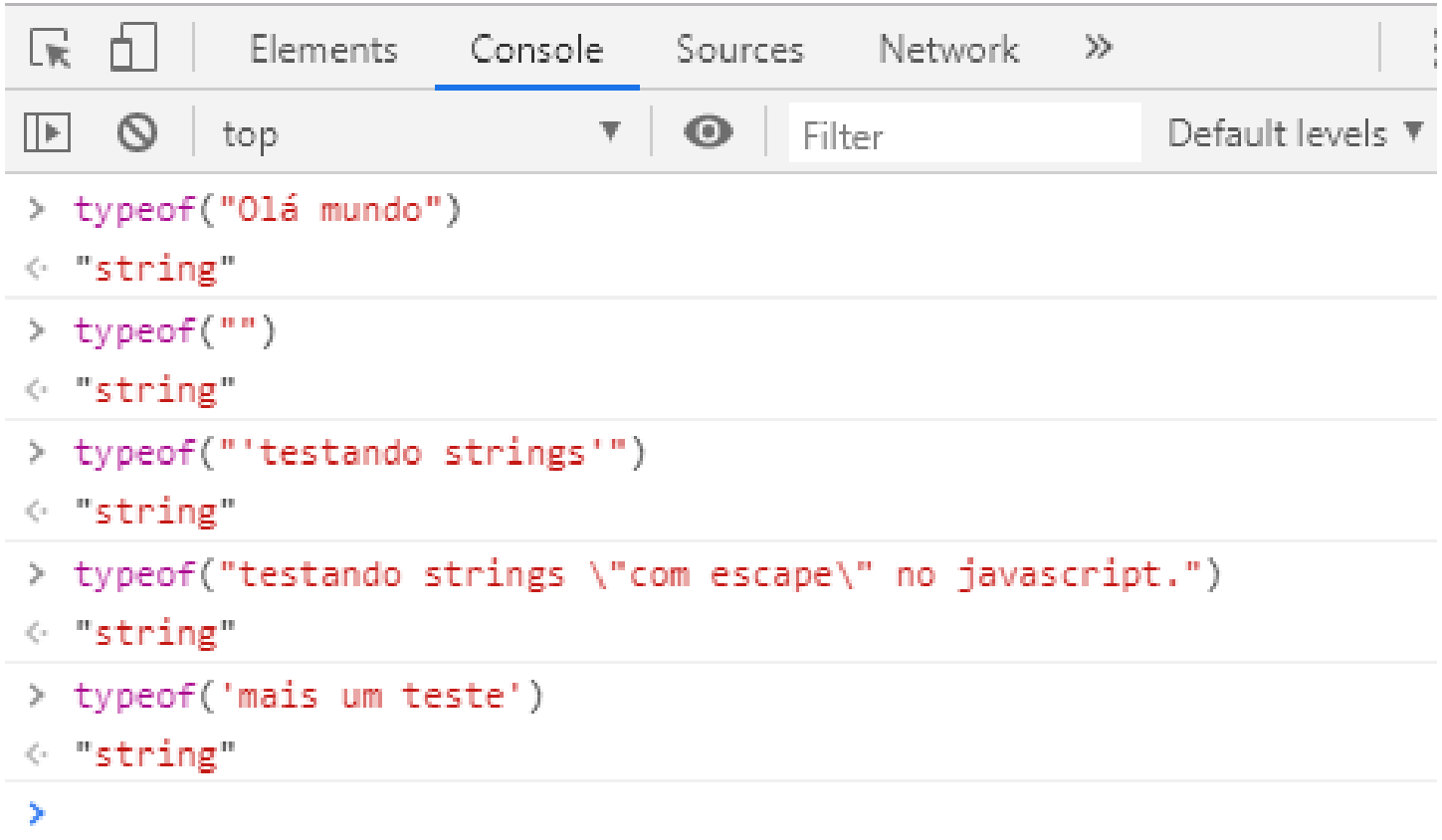
Numérico

- Representa todos os valores possíveis numéricos: frações, números inteiros, exponenciais, dentre outros.



Strings

- Corresponde a uma representação finita que inclui zero ou mais caracteres Unicode.



Métodos de String

Método	Descrição
charAt(indice)	retorna o caractere da string que ocupa a posição definida no parâmetro índice.
charCodeAt(indice)	retorna a codificação Unicode da string que ocupa a posição definida no parâmetro índice
concat(string1,string2,...,stringN)	concatena duas ou mais strings definidas nos parâmetros. A string resultante é formada pela junção das duas strings concatenadas.
fromCharCode(uni1,uni2,...,uniN)	retorna o caractere cuja codificação Unicode foi definida nos parâmetros passados.
indexOf(stringProcurada [,inicio])	retorna o índice da primeira ocorrência da string definida no parâmetro stringProcurada.
lastIndexOf(stringProcurada [,inicio])	retorna o índice da última ocorrência da string definida no parâmetro stringProcurada.

Métodos de String

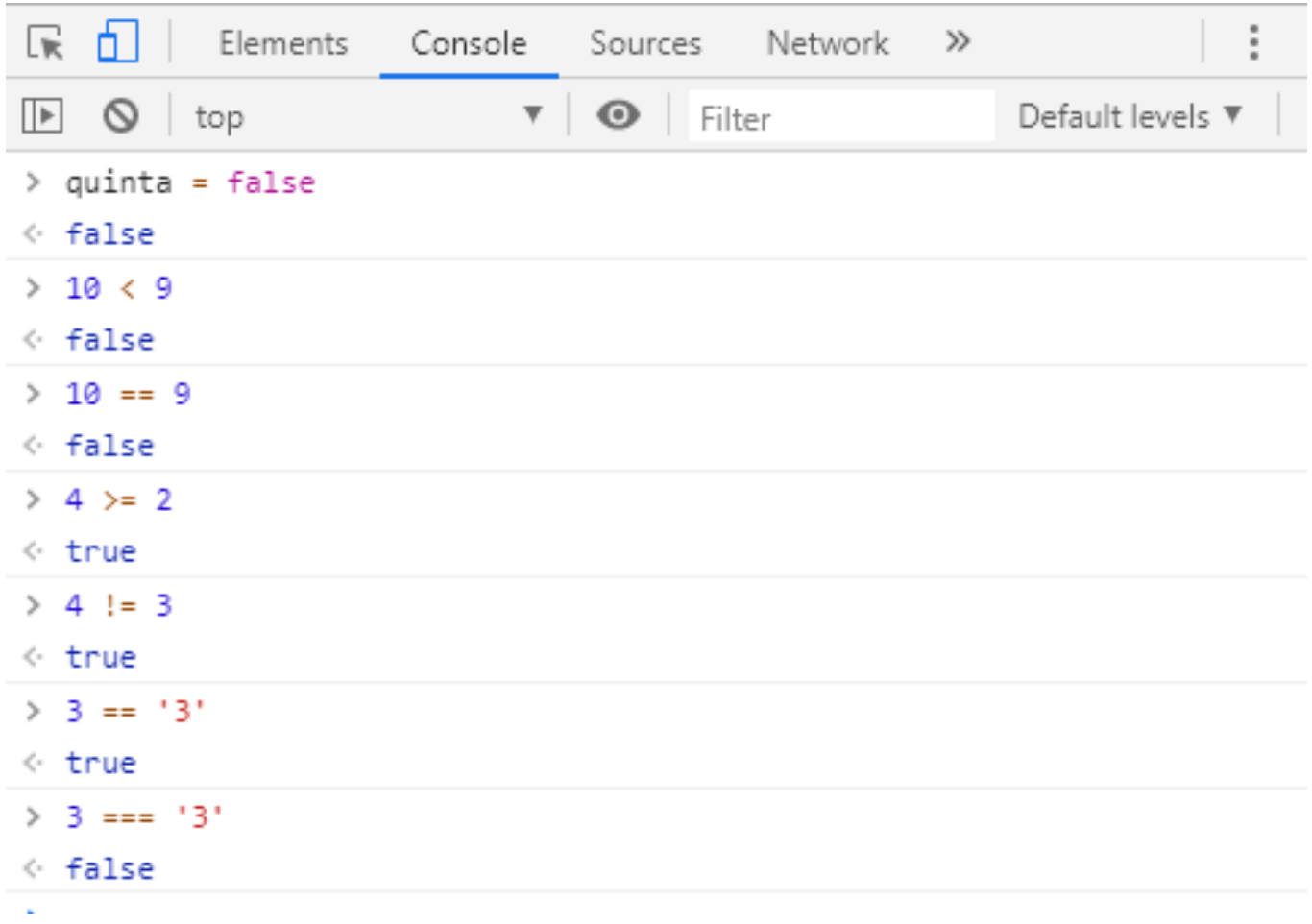
Método	Descrição
localCompare(string)	compara duas strings e define a ordem alfabética delas retornando um número negativo, o número 0 ou um número positivo.
match(expreg)	retorna as ocorrências da string definida no parâmetro expreg. Esse parâmetro é uma expressão regular supostamente contida na string em que se faz a procura.
replace(string | expreg, novastring | função)	encontra em uma string a primeira ocorrência da string definida no parâmetro string ou expreg e substitui essa ocorrência pela string definida no parâmetro novastring ou função.
search(expreg)	retorna a posição da primeira ocorrência do padrão definido no parâmetro expreg. Esse parâmetro é uma expressão regular supostamente contida na string em que se faz a procura.
slice(inicio [,fim])	permite criar uma nova string resultante da extração de parte da string original.

Métodos de String

Método	Descrição
split([separador] [,limite])	cria um array de substrings extraídas da string original.
substring(indiceA [,indiceB])	é semelhante ao método slice(). A diferença é o fato de que para esse método, se o primeiro parâmetro for maior que o segundo, ocorrerá uma inversão automática dos parâmetros.
toLowerCase()	retorna uma string igual à string original, mas com todos os caracteres minúsculos.
toUpperCase()	retorna uma string igual à string original, mas com todos os caracteres maiúsculos.
valueOf()	retorna o valor primitivo de um objeto.

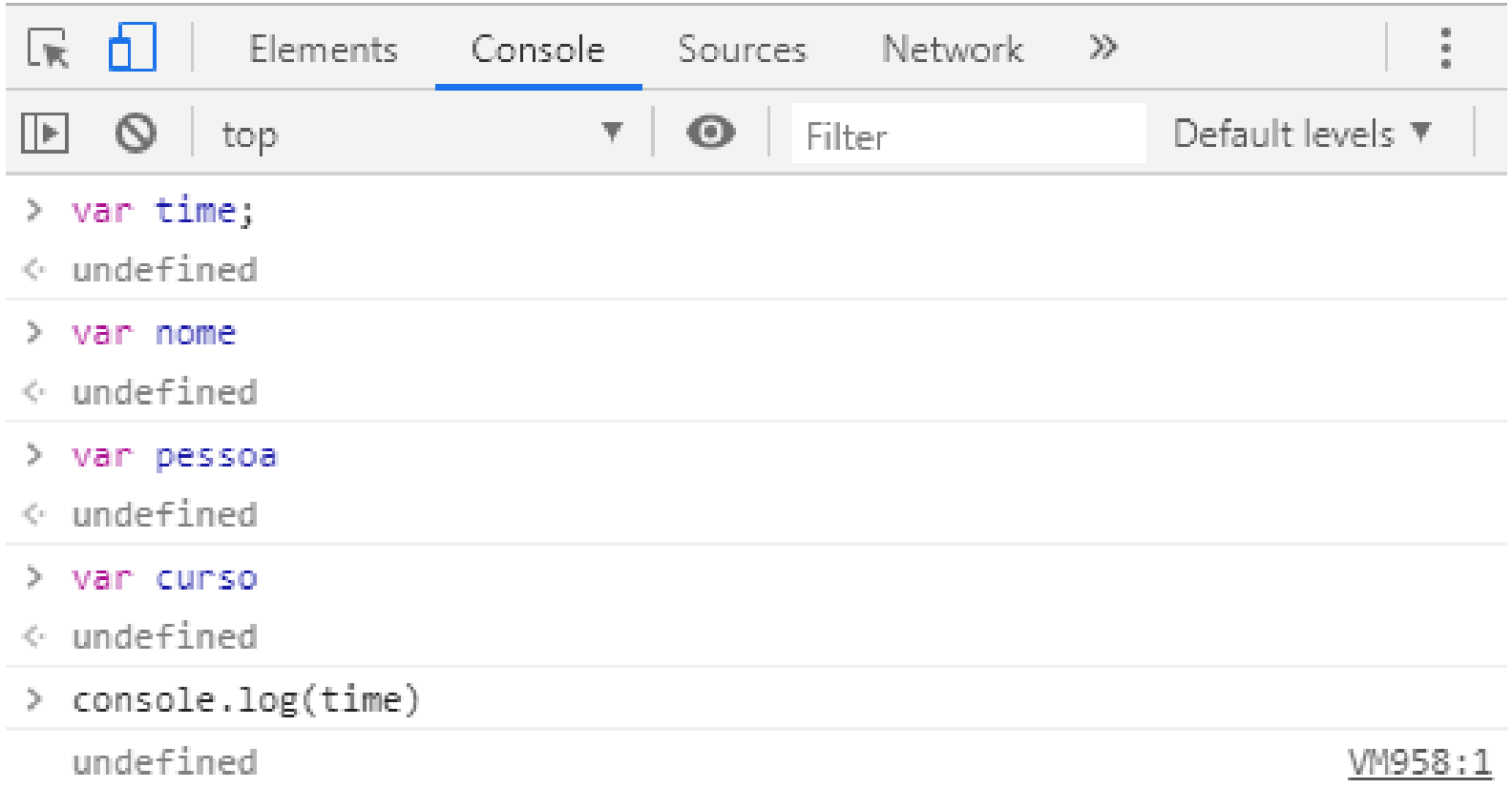
Booleano

- Representa um valor lógico, constituindo-se de dois valores possíveis: verdadeiro e falso.



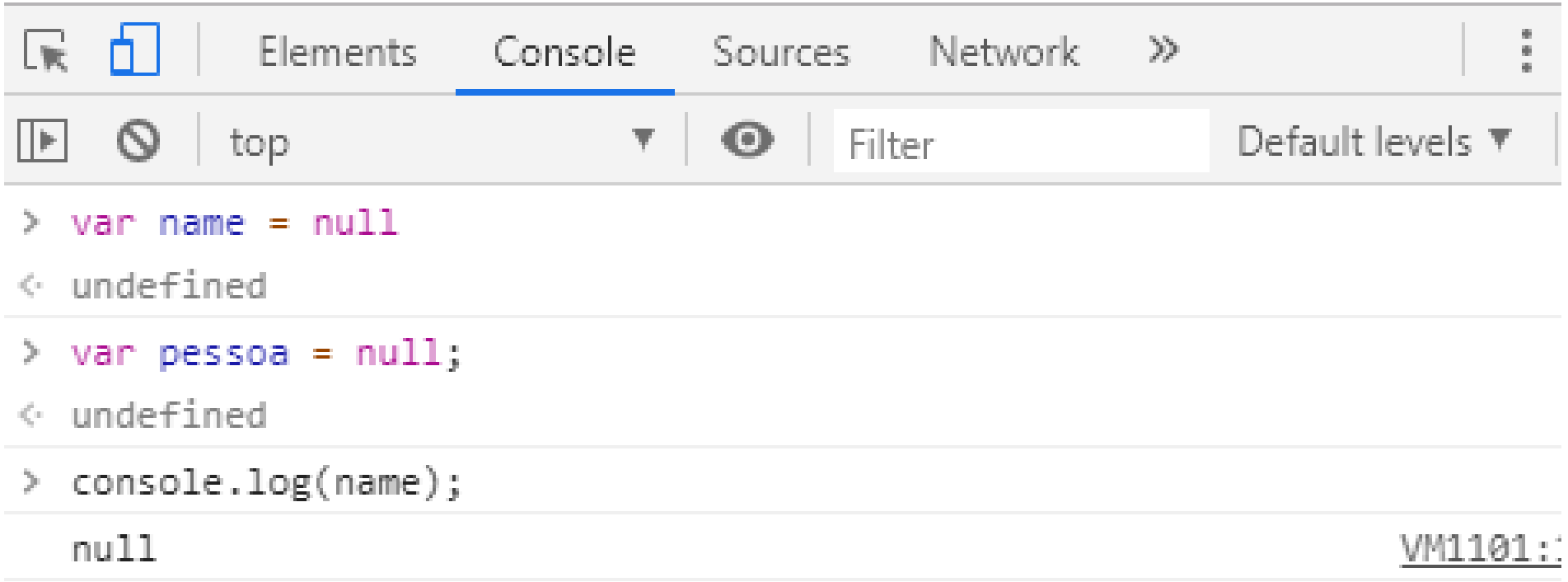
Undefined

- É usado para representar o caso em que um identificador ainda não tem nenhum valor atribuído.



Null

- Representa a ausência intencional de um valor. É usado explicitamente para fazer referência vazia ou inexistente.



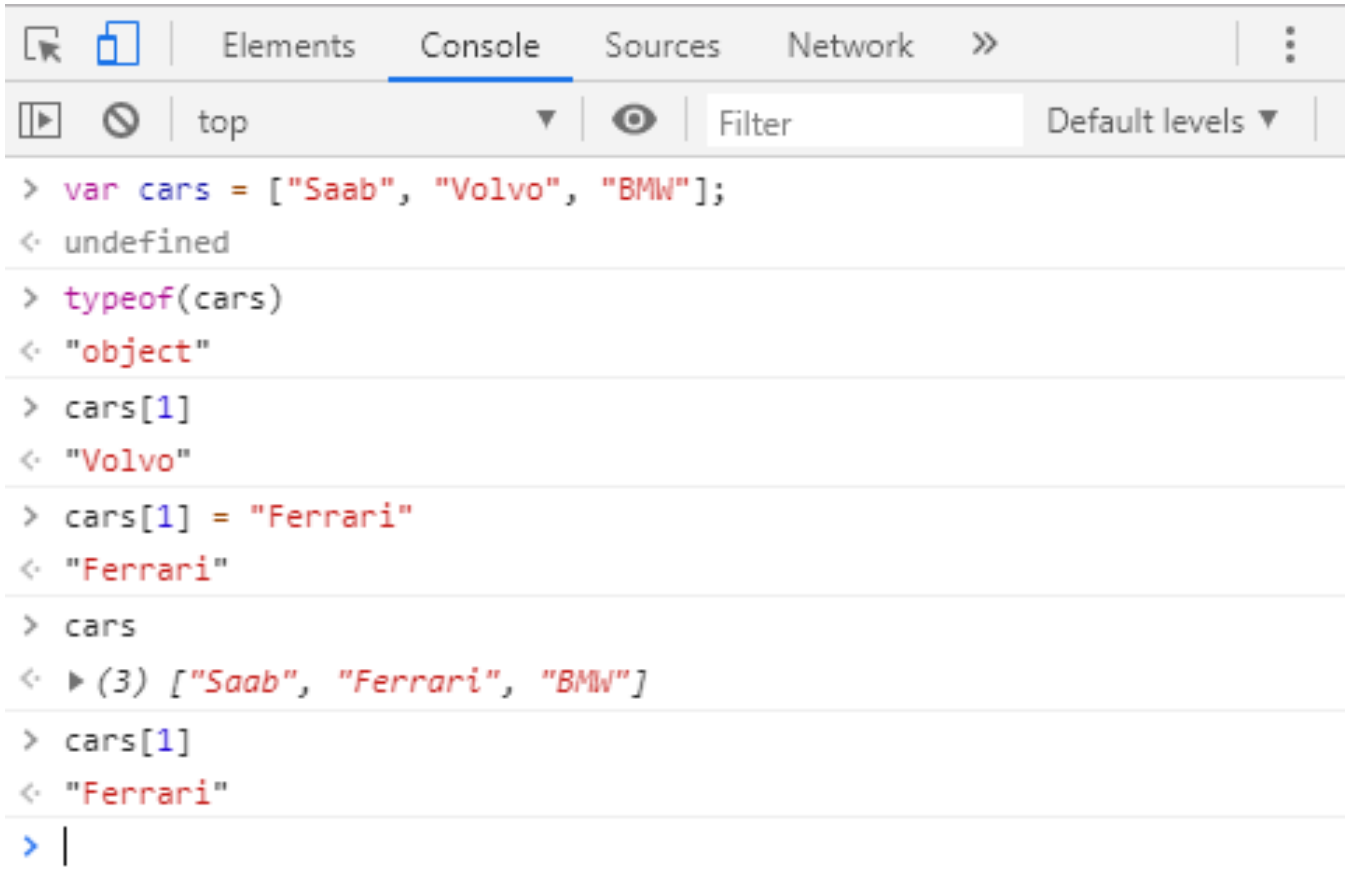
Operadores de Comparação

Operador de comparação	Operador	Descrição
Menor que	<	Usado para determinar se o operando da esquerda é menor que o operando da direita.
Maior que	>	Usado para determinar se o operando da esquerda é maior que o operando da direita.
Menor ou igual a	<=	Usado para determinar se o operando da esquerda é menor ou igual ao operando da direita.
Maior ou igual a	>=	Usado para determinar se o operando da esquerda é maior ou igual ao operando da direita.
Igual	==	Usado para determinar se o operando da esquerda é igual ao operando da direita.
Diferente de	!=	Usado para determinar se o operando da esquerda é diferente do operando da direita.
Estritamente igual	===	Verifica se os dois operandos são iguais sem permitir que haja qualquer coerção de tipo.
Estritamente diferente	!==	Verifica se os dois operandos são diferentes sem permitir que haja qualquer coerção de tipo.

Array

- O objeto Array do JavaScript é um objeto global usado na construção de 'arrays': objetos de alto nível semelhantes a listas.

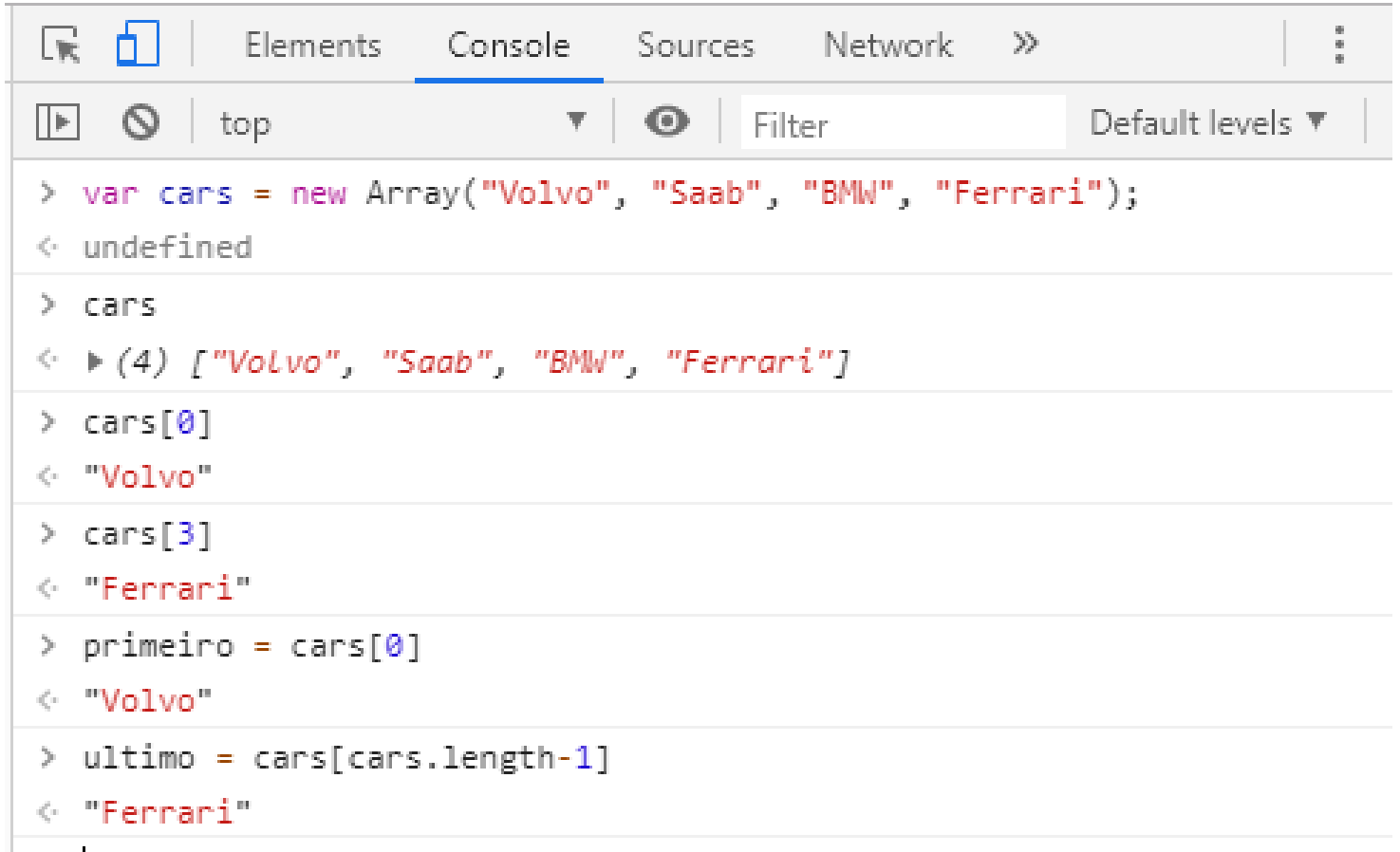
Atenção a função typeof



Array

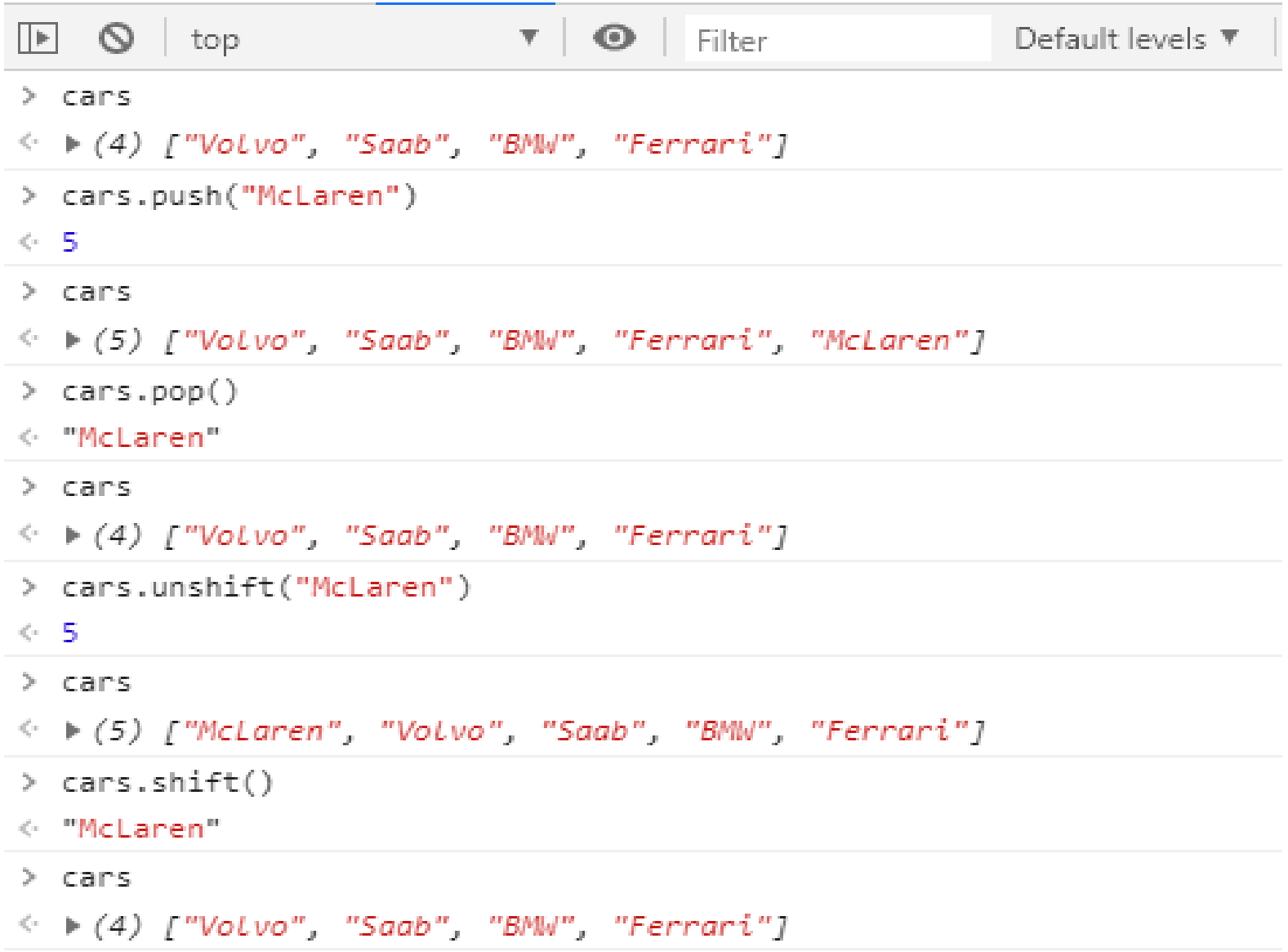
- O objeto Array do JavaScript é um objeto global usado na construção de 'arrays': objetos de alto nível semelhantes a listas.

Atenção a função length



Array

Atenção a função push, pop, shift e unshift().



Métodos do Array

Método	Descrição
concat(arg1,arg2,...,argn)	acrescenta os elementos definidos nos seus argumentos.
every(função(elemen,ind,obj) [,thisObjeto])	percorre cada um dos elementos do array e executa uma função callback. Assim que a função encontrar um elemento que não satisfaça a função, retorna false.
filter(elemen,ind,obj) [,thisObjeto])	aplica o filtro em um array e retorna o array após a aplicação;
forEach(elemen,ind,obj) [,thisObjeto])	percorre cada um dos elementos do array e executa uma função callback. Não altera o array original, e admite um argumento obrigatório, que é a função callback, e um argumento opcional.
indexOf(elemento, [arg2, true ou false])	retorna o índice do elemento de um array. Caso o elemento não exista, retorna -1.

Métodos do Array

Método	Descrição
lastIndexOf(elemento,[arg2, true ou false])	faz a mesma coisa que o método indexOf, mas se existir mais de uma ocorrência, o retorno será o último índice.
join([arg])	transforma os elementos do array em uma string. Admite um argumento opcional que se destina a criar um separador para os elementos do array. O padrão é a vírgula.
map(função(elem,ind,obj) [.thisObjeto])	percorre cada um dos elementos do array e modifica-o confome definido em uma função callback.
pop()	remove o último elemento do array.
push(arg1, arg2, …, argn)	Acrescenta elementos no final de um array.
reduce(função(v1,v2,ind,arr)[,vInicial])	percorre cada um dos elementos do array e executa uma função callback. Será retornado um valor único para cada elemento.

Métodos do Array

Método	Descrição
reverse()	inverte a ordem dos elementos do array
shift()	remove o primeiro elemento do array e retorna o valor que foi removido
slice(arg1[,arg2])	retorna um "subarray" do array original. Os dois argumentos recebidos são o índice inicial e final.
some(função(elem,ind,obj)[,thisObjeto])	percorre cada um dos elementos do array e executa uma função callback. Assim que a função encontra um elemento que satisfaça as condições, retorna true.
sort([função])	ordena os elementos de um array. Permite a passagem de um argumento opcional que é uma função de ordenação.
splice(arg1[,arg2,arg3,...,argn])	serve para inserir e/ou remover elementos de um array.
unshift(args)	insere elementos no início de um array e retorna a nova quantidade de elementos.

Objetos

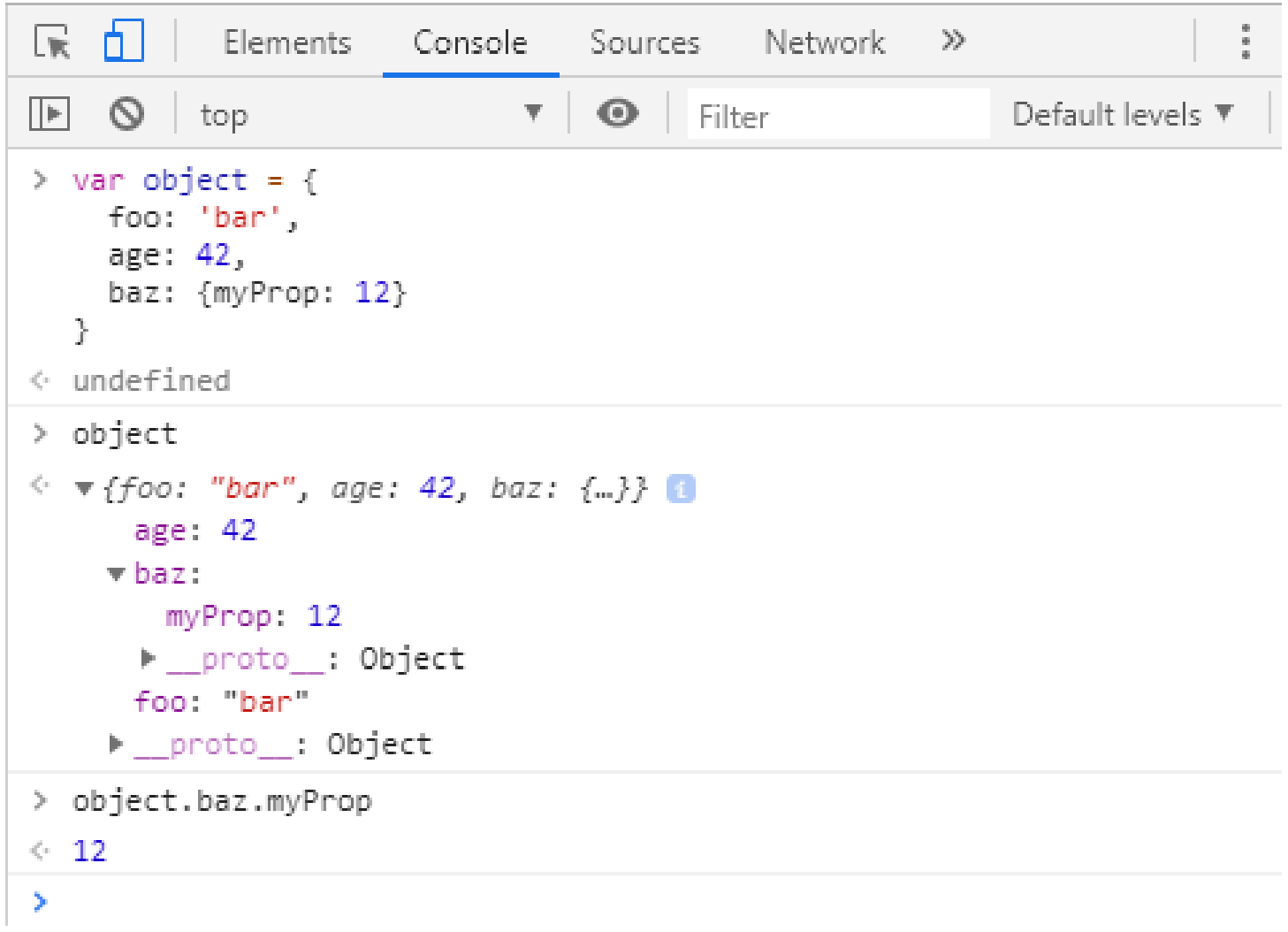
- Um inicializador de objetos é uma lista delimitada por vírgula de zero ou mais pares de nomes de **propriedades** e **valores** associados a um objeto, entre chaves ({}).

O acesso a um elemento, pode ser feito por meio de sua propriedade (ou chave, também chamada).

Objetos

- Um inicializador de objetos é uma lista delimitada por vírgula de zero ou mais pares de nomes de **propriedades** e **valores** associados a um objeto, entre chaves ({}).

Podemos ter também, propriedades, dentro de outras propriedades.

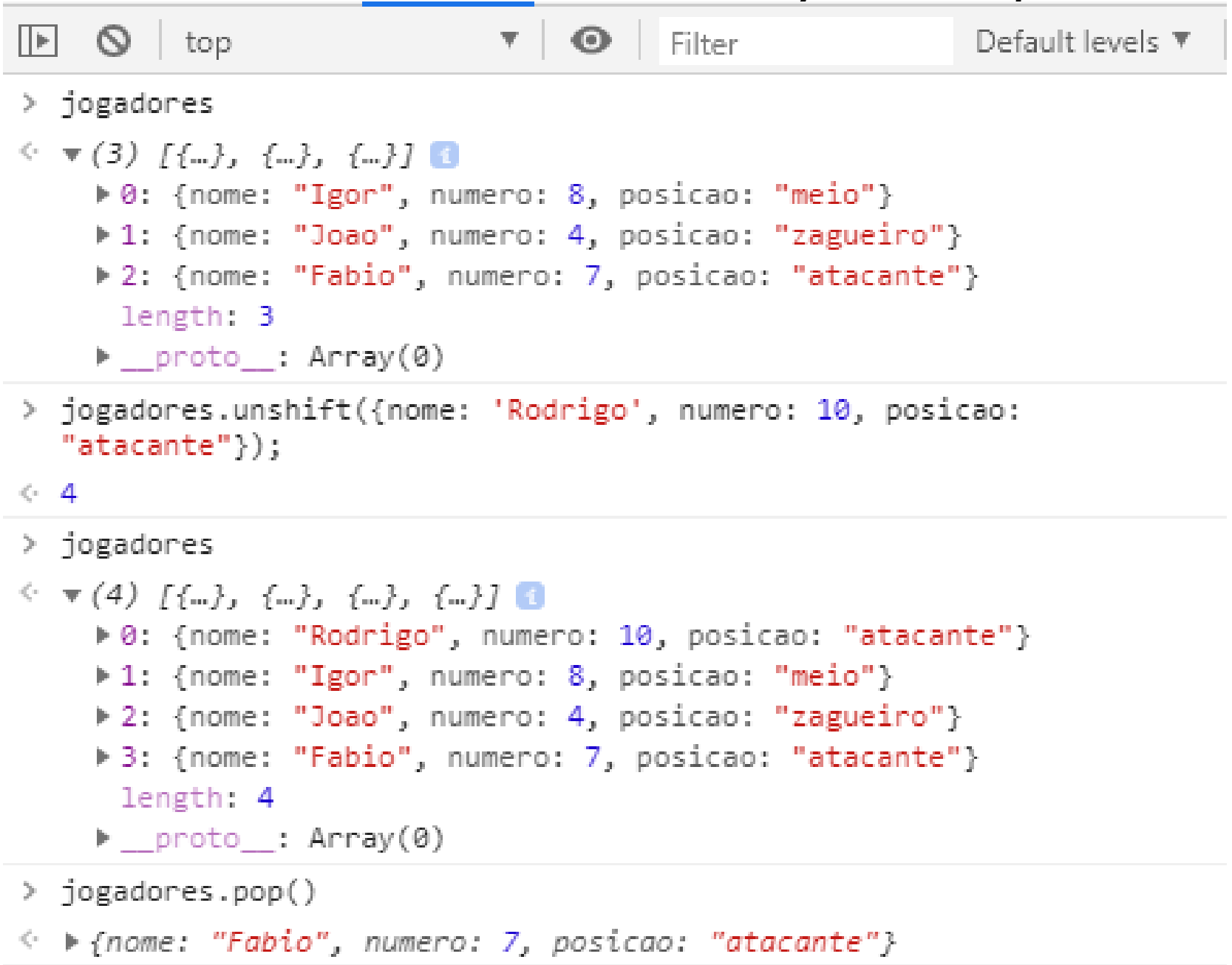


Array X Objetos

```
> var jogadores = [
      {nome: "Igor", numero: 8, posicao: "meio"},
      {nome: "Joao", numero: 4, posicao: "zagueiro"},
      {nome: "Fabio", numero: 7, posicao: "atacante"},
  ];
< undefined
> jogadores
< ▼(3) [{…}, {…}, {…}]
    ▶0: {nome: "Igor", numero: 8, posicao: "meio"}
    ▶1: {nome: "Joao", numero: 4, posicao: "zagueiro"}
    ▶2: {nome: "Fabio", numero: 7, posicao: "atacante"}
     length: 3
    ▶__proto__: Array(0)
> jogadores[0]
< ▶{nome: "Igor", numero: 8, posicao: "meio"}
>
```

Podemos ter também, objetos dentro de array, bem como o inverso, também é possível.

Array X Objetos



Podemos ter também, objetos dentro de array, bem como o inverso, também é possível.

Funções

- Funções são blocos de construção fundamentais em JavaScript.
- Uma função é um procedimento de JavaScript - um conjunto de instruções que executa uma tarefa ou calcula um valor.
- Para usar uma função, você deve defini-la em algum lugar no escopo do qual você quiser chamá-la.
- A definição de uma função, é feita pela palavra reservada **function** seguida por:

1. Nome da Função.
2. Lista de argumentos para a função, entre parênteses e separados por vírgulas.
3. Declarações JavaScript que definem a função, entre chaves { }.

Criação de Função

```
1  function square(numero) {
2    return numero * numero;
3  }
4  square(4);
```

Função estática

```
6  var square = function(numero) {
7      return numero * numero
8  }
9  var x = square(4)
```

Função expressão e também anônima.

Ambas as funções, utilizam parâmetros na declaração da função.

Escopo de Função

- As variáveis definidas no interior de uma função não podem ser acessadas de nenhum lugar fora da função, porque a variável está definida apenas no escopo da função.

```
num1 = 20, num2 = 3, nome = "Fulano"; // variável global

function multiplica() {
  return num1 * num2;
}

multiplica(); // Retorna 60
```

Nesse exemplo, temos a utilização de duas variáveis locais: num1 e num2. Além disso, temos uma função dentro de outra (aninhada).

```
function getScore () {
  var num1 = 2,
      num2 = 3;

  function add() {
    return nome + " scored " + (num1 + num2);
  }

  return add();
}

getScore(); // Retorna "Rodrigo scored 5"
```

Arrow Functions

- A nova versão do JavaScript, a ES6, trouxe novas features e dentre elas uma nova forma de criar funções usando o operador =>. Esta nova forma de se trabalhar com funções são chamadas **Arrow Functions**.

```
1  const materials = [
2    'Hydrogen',
3    'Helium',
4    'Lithium',
5    'Beryllium'
6  ];
7
8  console.log(materials.map(material => material.length));
9
10 // retorna Array [8. 6. 7. 9]
```

Função Callback

- Callback é uma função passada como parâmetro para outra função.
- A função setTimeout recebe dois parâmetros: o primeiro é uma função callback, e o segundo é o tempo que o interpretador irá esperar até executar essa função.

```
1  setTimeout(() =>
2      alert("Hello World"),
3      5000);
4
```

Funções Pré-Definidas

- eval(): O método eval() avalia código JavaScript representado como uma string.
- uneval(): O método uneval() cria uma representação de string do código-fonte de um Object.
- isFinite(): A função global isFinite() determina se o valor passado é um número finito. Se necessário, o parâmetro é primeiro convertido para um número.
- isNaN(): A função isNaN() determina se um valor é um número ou não.
- parseFloat(): A função parseFloat() analisa um argumento do tipo string e retorna um número de ponto flutuante.
- parseInt(): A função parseInt() analisa um argumento do tipo string e retorna um número inteiro.

Funções Pré-Definidas

- encodeURI(): Reescreve uma URI substituindo alguns caracteres especiais por sua codificação UTF-8.
- decodeURI(): Recupera uma URI que tenha sido codificada com a função encodeURI().
- encodeURIComponent(): O método encodeURIComponent() codifica um componente URI, substituindo cada ocorrência de determinados caracteres por um, dois, três, ou quatro sequências de escape que representa a codificação UTF-8 do caractere.
- decodeURIComponent(): O método decodeURIComponent() decodifica um componente URI criado anteriormente por encodeURIComponent ou por uma rotina similar.

Eventos

- Eventos são ações ou ocorrências que acontecem no sistema que estamos desenvolvendo, no qual este te alerta sobre essas ações para que você possa responder de alguma forma, se desejado.
- Por exemplo, se o usuário clica em um botão numa pagina web, você pode querer responder a esta ação mostrando na tela uma caixa de informações.
- Todas as vezes que ocorrem interações com um documento web, eventos são disparados.

Categoria de Eventos

- Os eventos Javascript podem ser agrupados em algumas categorias:

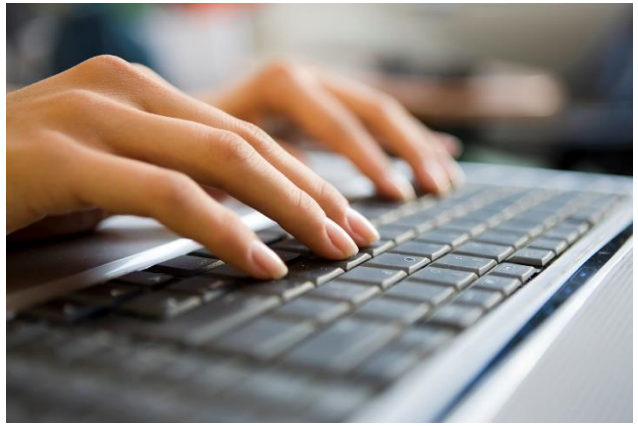
1. Eventos de Mouse: São eventos que ocorrem quando o usuário interage com a página web, usando o cursor do mouse.
2. Eventos de Teclado: São eventos que ocorrem quando o usuário interage com a página web usando o teclado.
3. Eventos HTML: São eventos que ocorrem quando há uma mudança na janela do navegador.

Eventos de Mouse:

- click: Ocorre quando o usuário pressiona o botão esquerdo do mouse.
- dblclick: Um elemento foi clicado duas vezes em um intervalo de tempo curto.
- mousedown: Ocorre quando o usuário qualquer um dos botões do mouse sobre um elemento.
- mouseup: Ocorre quando o usuário solta o botão, após ter pressionado um dos botões do mouse.
- mouseenter: O mouse está acima de um elemento que tem um listener ativo.
- wheel: Foi detectada rotação no scroll do mouse.

Eventos de Teclado:

- keydown: Ocorre quando qualquer tecla foi pressionada
- keypress: Ocorre quando pressiona qualquer tecla, com exceção de Shift, Fn e Caps Lock está pressionada, que resulte em um caractere ou na alteração da apresentação de um texto na tela.
- keyup: Ocorre quando o usuário solta uma tecla de teclado que foi anteriormente pressionada.



Eventos HTML:

- load: Ocorre quando há o carregamento completo de uma janela do navegador, de um conjunto de frames (<frameset>), de uma imagem (<img>) ou de um elemento objeto (<object>).
- unload: Ocorre quando há o fechamento completo de uma janela do navegador, de um conjunto de frames (<frameset>), de uma imagem (<img>) ou de um elemento objeto (<object>).
- abort: Ocorre quando há interrupção no carregamento de um recurso.
- error: Ocorre quando houver falha no carregamento do recurso.
- resize: Ocorre quando uma janela do navegador ou frame é redimensionada.

Eventos HTML:

- scroll: Ocorre quando há um elemento que possui barra de rolagem sofre uma rolagem.
- focus: Ocorre um elemento recebeu o foco.
- blur: Ocorre quando um elemento perdeu o foco.
- submit: Ocorre quando o botão submit de um formulário é clicado.
- reset: Ocorre quando o botão reset de um formulário é clicado.
- fullscreenchange: Ocorre quando um elemento alterna para o modo de visualização para fullscreeen ou normal.
- drag: Ocorre quando um elemento ou uma seleção de texto está sendo arrastado.
- drop: Ocorre quando um elemento ou uma seleção de texto foi solto em um destino válido.

Q1) [IBFC TRE PA 2020] Um identificador é uma palavra usada para identificar variáveis, constantes e funções criadas pelo programador. Um identificador JavaScript deve começar com:

a) uma letra (somente maiúscula), sustenido (#), ou cifrão ($)

b) uma letra (somente minúscula), underline (_), ou porcentagem (%)

c) uma letra (maiúscula ou minúscula), underline (#), ou cifrão (%)

d) uma letra (maiúscula ou minúscula), underline (_), ou cifrão ($)

Q1) [IBFC TRE PA 2020] Um identificador é uma palavra usada para identificar variáveis, constantes e funções criadas pelo programador. Um identificador JavaScript deve começar com:

a) uma letra (somente maiúscula), sustenido (#), ou cifrão ($)

b) uma letra (somente minúscula), underline (_), ou porcentagem (%)

c) uma letra (maiúscula ou minúscula), underline (#), ou cifrão (%)

d) uma letra (maiúscula ou minúscula), underline (_), ou cifrão ($)

Q2) [IBFC EBSERH 2020] Em relação às linguagens XML e JavaScript, analise as afirmativas abaixo e dê valores Verdadeiro (V) ou Falso (F).

I. o código fonte JavaScript é incluído no próprio arquivo HTML com a tag <script>.

II. existe a possibilidade de abrir arquivos XML por meio de um simples editor de texto.

III. tanto JavaScript como Java necessitam da máquina virtual Java para funcionarem.

Assinale a alternativa que apresenta a sequência correta de cima para baixo.

a) V, F, F

b) V, F, V

c) F, V, V

d) V, V, F

e) F, F, V

Q2) [IBFC EBSERH 2020] Em relação às linguagens XML e JavaScript, analise as afirmativas abaixo e dê valores Verdadeiro (V) ou Falso (F).

I. o código fonte JavaScript é incluído no próprio arquivo HTML com a tag <script>.

II. existe a possibilidade de abrir arquivos XML por meio de um simples editor de texto.

III. tanto JavaScript como Java necessitam da máquina virtual Java para funcionarem.

Assinale a alternativa que apresenta a sequência correta de cima para baixo.

a) V, F, F

b) V, F, V

c) F, V, V

d) V, V, F

e) F, F, V

Q3) [COMPERVE TJ RN 2020] No javascript é possível interagir com o console dos navegadores. O comando para imprimir o texto “TJ-RN” no console é:

a) console.dump("TJ-RN ");

b) console.print("TJ-RN ");

c) console.log("TJ-RN ");

d) console.echo("TJ-RN ");

Q3) [COMPERVE TJ RN 2020] No javascript é possível interagir com o console dos navegadores. O comando para imprimir o texto “TJ-RN” no console é:

a) console.dump("TJ-RN ");

b) console.print("TJ-RN ");

c) console.log("TJ-RN ");

d) console.echo("TJ-RN ");

Q4) [VUNESP FITO 2020] Um Técnico elaborou o programa a seguir, na Linguagem HTML com JavaScript.

<!DOCTYPE html> <html> <script> function botao ( ) { alert ("B") ; } </script> <body> <input type="button" onclick="botao ( ) " value = "A"/> </body> </html>

Sobre esse programa é correto afirmar que, ao ser aberto por um navegador que suporte HTML e JavaScript, como o Chrome e o Edge, um botão será exibido na tela com o conteúdo

a) A, que, ao ser pressionado com um click, provocará a exibição de uma janela.
b) A, que, ao ser pressionado com dois clicks, trocará o seu valor para B.
c) B, que, ao ser pressionado com um click, provocará a exibição de uma janela.
d) B, que, ao ser pressionado com dois clicks, trocará o seu valor para A.
e) B, que, ao ser pressionado com um click, apresentará na tela o valor A.

Q4) [VUNESP FITO 2020] Um Técnico elaborou o programa a seguir, na Linguagem HTML com JavaScript.

<!DOCTYPE html> <html> <script> function botao ( ) { alert ("B") ; } </script> <body> <input type="button" onclick="botao ( ) " value = "A"/> </body> </html>

Sobre esse programa é correto afirmar que, ao ser aberto por um navegador que suporte HTML e JavaScript, como o Chrome e o Edge, um botão será exibido na tela com o conteúdo

a) A, que, ao ser pressionado com um click, provocará a exibição de uma janela.

b) A, que, ao ser pressionado com dois clicks, trocará o seu valor para B.

c) B, que, ao ser pressionado com um click, provocará a exibição de uma janela.

d) B, que, ao ser pressionado com dois clicks, trocará o seu valor para A.

e) B, que, ao ser pressionado com um click, apresentará na tela o valor A.

Q5) [FCC AFAP 2019] Considere o fragmento de código abaixo, retirado do corpo de uma página web que utiliza JavaScript.

```
<body>
<img id="evento" src="grupo1.gif" onclick="trocar()">
<script>
function trocar() {


}
</script>
</body>
```

Para que ao clicar na imagem grupo1.gif ela seja substituída pela imagem grupo2.gif no interior da função trocar deverá ser incluído o comando

a) # ('$evento ') .src ('grupo2.gif ‘) ;
b) document.getElementById("evento").src = "grupo2.gif";.
c) $ ('. evento ') . attr ('src ', 'grupo2.gif ‘) ;
d) document.img.src = "grupo2.gif";
e) document.getElementById("evento").change = "grupo2.gif";

Q5) [FCC AFAP 2019] Considere o fragmento de código abaixo, retirado do corpo de uma página web que utiliza JavaScript.

```
<body>
<img id="evento" src="grupo1.gif" onclick="trocar()">
<script>
function trocar() {


}
</script>
</body>
```

Para que ao clicar na imagem grupo1.gif ela seja substituída pela imagem grupo2.gif no interior da função trocar deverá ser incluído o comando

a) # ('$evento ') .src ('grupo2.gif ‘) ;
b) document.getElementById("evento").src = "grupo2.gif";
c) $ ('. evento ') . attr ('src ', 'grupo2.gif ‘) ;
d) document.img.src = "grupo2.gif";
e) document.getElementById("evento").change = "grupo2.gif";

Q6) [FGV DPE RJ 2019] Analise o código JavaScript a seguir.

```
var pessoa = {
  nome: "Carlos",
  sobreNome: "Moreira",
  rg: 12125566,
  nomeCompleto : function() {
    return this.nome + " " + this.sobreNome;
  }
};
```

Analise ainda o comando de atribuição a seguir.

document.getElementById("34").innerHTML=...;

No trecho pontilhado, a expressão que retorna corretamente a concatenação do nome com o sobrenome de pessoa é:

('$evento ') .src ('grupo2.gif ‘) ;

a) pessoa.nomeCompleto()

b) pessoa:nomeCompleto

c) nomeCompleto()

d) pessoa.nomeCompleto.

e) pessoa.nomecompleto

Q6) [FGV DPE RJ 2019] Analise o código JavaScript a seguir.

```
var pessoa = {
  nome: "Carlos",
  sobreNome: "Moreira",
  rg: 12125566,
  nomeCompleto : function() {
    return this.nome + " " + this.sobreNome;
  }
};
```

Analise ainda o comando de atribuição a seguir.

document.getElementById("34").innerHTML=...;

No trecho pontilhado, a expressão que retorna corretamente a concatenação do nome com o sobrenome de pessoa é:

('$evento ') .src ('grupo2.gif ‘) ;

a) pessoa.nomeCompleto()
b) pessoa:nomeCompleto
c) nomeCompleto()
d) pessoa.nomeCompleto.
e) pessoa.nomecompleto

Q7) [FGV DPE RJ 2019] Sobre a função alert do JavaScript, analise as afirmativas a seguir.

I. Requer apenas um parâmetro.

II. Oferece apenas o botão OK para o usuário.

III. Retém a execução do código até que o usuário responda.

Está correto o que se afirma em:

a) somente I;

b) somente II;

c) somente I e III;

d) somente II e III;

e) I, II e III.

Q7) [FGV DPE RJ 2019] Sobre a função alert do JavaScript, analise as afirmativas a seguir.

I. Requer apenas um parâmetro.

II. Oferece apenas o botão OK para o usuário.

III. Retém a execução do código até que o usuário responda.

Está correto o que se afirma em:

a) somente I;

b) somente II;

c) somente I e III;

d) somente II e III;

e) I, II e III.

Q8) [CCV-UFC UFC 2019] Utilizando JavaScript , qual é a sintaxe correta para alterar o conteúdo do elemento HTML de id="teste", por meio do clique do botão abaixo?

```
<html>
<head></head>
<body>
   <p id="teste">UFC</p>
   <button type="button" onclick="ALTERAR AQUT">
      UFC
   </button>
</body>
</html>
```

a) document.getElementByName("p").innerHTML = "Universidade Federal do Ceará“

b) document.getElementById('teste').innerHTML = 'Universidade Federal do

c) document.getElement("p").innerHTML = "Universidade Federal do Ceará“

d) document.p.innerHTML = "Universidade Federal do Ceará"

e) #demo.innerHTML = "Universidade Federal do Ceará"

Q8) [CCV-UFC UFC 2019] Utilizando JavaScript , qual é a sintaxe correta para alterar o conteúdo do elemento HTML de id="teste", por meio do clique do botão abaixo?

```
<html>
<head></head>
<body>
   <p id="teste">UFC</p>
   <button type="button" onclick="ALTERAR AQUT">
      UFC
   </button>
</body>
</html>
```

a) document.getElementByName("p").innerHTML = "Universidade Federal do Ceará“

b) document.getElementById('teste').innerHTML = 'Universidade Federal do

c) document.getElement("p").innerHTML = "Universidade Federal do Ceará“

d) document.p.innerHTML = "Universidade Federal do Ceará"

e) #demo.innerHTML = "Universidade Federal do Ceará"

Q9) [CCV-UFC UFC 2019] Qual a função da linguagem JavaScript responsável por avaliar um texto contendo um código JavaScript passado como parâmetro e retornar o resultado da execução?

a) eval

b) isFinite

c) execute

d) unescape

e) encodeURI

Q10) [CESPE SLU/DF 2019] Julgue o próximo item, relativos à linguagem de programação JavaScript e às ferramentas Node e React.

Uma função JavaScript é um bloco de código utilizado para executar tarefas repetidas e é definida pela palavra-chave public seguida por um nome seguido por parênteses ( ).

Q9) [CCV-UFC UFC 2019] Qual a função da linguagem JavaScript responsável por avaliar um texto contendo um código JavaScript passado como parâmetro e retornar o resultado da execução?

a) eval

b) isFinite

c) execute

d) unescape

e) encodeURI

Q10) [CESPE SLU/DF 2019] Julgue o próximo item, relativos à linguagem de programação JavaScript e às ferramentas Node e React.

Uma função JavaScript é um bloco de código utilizado para executar tarefas repetidas e é definida pela palavra-chave public seguida por um nome seguido por parênteses ( ). ERRADO.

Q11) [FCC SEMEF 2019] Em um site desenvolvido utilizando como uma das linguagens a JavaScript, um Web Designer possui uma variável chamada mensagem contendo um fragmento de texto. Deseja-se trocar, no conteúdo dessa variável, a palavra Parintins por Manaus e exibir o fragmento de texto já com a troca realizada em um elemento HTML cujo atributo id="texto ". Para isso, terá que utilizar o comando

a) document.write.getElementById(‘texto') = mensagem.change("Paritins", "Manaus");

b) document.getElementById('#texto').innerHTML = mensagem.replace("Paritins", "Manaus");

c) document.getElementById('texto').value = mensagem.replace("Paritins", "Manaus");

d) document.getElementById('#texto').value = mensagem.change("Paritins", "Manaus");

e) document.getElementById('texto').innerHTML = mensagem.replace("Paritins", "Manaus");

Q11) [FCC SEMEF 2019] Em um site desenvolvido utilizando como uma das linguagens a JavaScript, um Web Designer possui uma variável chamada mensagem contendo um fragmento de texto. Deseja-se trocar, no conteúdo dessa variável, a palavra Parintins por Manaus e exibir o fragmento de texto já com a troca realizada em um elemento HTML cujo atributo id="texto ". Para isso, terá que utilizar o comando

a) document.write.getElementById(‘texto') = mensagem.change("Paritins", "Manaus");

b) document.getElementById('#texto').innerHTML = mensagem.replace("Paritins", "Manaus");

c) document.getElementById('texto').value = mensagem.replace("Paritins", "Manaus");

d) document.getElementById('#texto').value = mensagem.change("Paritins", "Manaus");

e) document.getElementById('texto').innerHTML = mensagem.replace("Paritins", "Manaus");

Q12) [FCC SEMEF 2019] Em uma página web que utiliza JavaScript um Programador deseja aumentar alguns valores de salário que estão em um vetor em 10% e armazenar estes valores ajustados em outro vetor, utilizando o fragmento de código abaixo.

```
<body>
  <script>
    var salarios = [1050.00, 2000.00, 5000.00, 1000.00];
    var v2 = salarios .....I..... (opera);
    function opera(v, i, a) {
      return v * 1.1;
    }
  </script>
</body>
```

Para realizar a operação, a lacuna I deverá ser preenchida por

a) map

b) Math

c) split

d) Clone

e) calc

Q12) [FCC SEMEF 2019] Em uma página web que utiliza JavaScript um Programador deseja aumentar alguns valores de salário que estão em um vetor em 10% e armazenar estes valores ajustados em outro vetor, utilizando o fragmento de código abaixo.

```
<body>
  <script>
    var salarios = [1050.00, 2000.00, 5000.00, 1000.00];
    var v2 = salarios .....I..... (opera);
    function opera(v, i, a) {
      return v * 1.1;
    }
  </script>
</body>
```

Para realizar a operação, a lacuna I deverá ser preenchida por

a) map

b) Math

c) split

d) Clone

e) calc

Q13) [UFMG UFMG 2019] Considere as seguintes operações na linguagem JavaScript:

```
<script>
var i = 2;
i +=3;
i *= 4;
document.writeln(i)
</script>
```

O valor exibido na tela será

a) 4

b) 20

c) 9

d) 10

Q13) [UFMG UFMG 2019] Considere as seguintes operações na linguagem JavaScript:

```
<script>
var i = 2;
i +=3;
i *= 4;
document.writeln(i)
</script>
```

O valor exibido na tela será

a) 4

b) 20

c) 9

d) 10

Q14) [VUNESP UFABC 2019] Observe o trecho de código JavaScript a seguir:

```
x = "10";
if (x === 10)
  alert("SIM");
else
  alert("NÃO");
```

Esse código exibe a mensagem “NÃO”, pois

a) a variável “x” não é do tipo numérico

b) o operador comparado testa se a variável “x” é diferente de 10.

c) faltou delimitar os blocos da instrução “if” com chaves.

d) não é possível utilizar acentos em strings.

e) não é permitido o uso de ponto-e-vírgula antes de “else”.

Q14) [VUNESP UFABC 2019] Observe o trecho de código JavaScript a seguir:

```
x = "10";
if (x === 10)
  alert("SIM");
else
  alert("NÃO");
```

Esse código exibe a mensagem “NÃO”, pois

a) a variável “x” não é do tipo numérico

b) o operador comparado testa se a variável “x” é diferente de 10.

c) faltou delimitar os blocos da instrução “if” com chaves.

d) não é possível utilizar acentos em strings.

e) não é permitido o uso de ponto-e-vírgula antes de “else”.

Q15) [CESPE TJ-AM 2019] Com relação a desenvolvimento em Java para Web, julgue o item que se segue.

JavaScript é uma linguagem de programação orientada ao desenvolvimento da interface para aplicações web, cujo código-fonte é compilado pelo servidor antes de sua execução.

Q16) [VUNESP Câmara Piracicaba] Na linguagem JavaScript, o operador === (três sinais de igualdade) realiza a comparação apenas do

a) tipo dos operandos.

b) conteúdo dos operandos.

c) valor dos operandos

d) valor lógico dos operandos.

e) valor e do tipo dos operandos.

Q15) [CESPE TJ-AM 2019] Com relação a desenvolvimento em Java para Web, julgue o item que se segue.

JavaScript é uma linguagem de programação orientada ao desenvolvimento da interface para aplicações web, cujo código-fonte é compilado pelo servidor antes de sua execução. ERRADO

Q16) [VUNESP Câmara Piracicaba] Na linguagem JavaScript, o operador === (três sinais de igualdade) realiza a comparação apenas do

a) tipo dos operandos.

b) conteúdo dos operandos.

c) valor dos operandos

d) valor lógico dos operandos.

e) valor e do tipo dos operandos.

Q17) [FEPESE CELESC 2019] Qual operador Javascript permite concatenar strings?

a) .

b) !

c) &&

d) &

e) +

Q18) [CCV-UFC UFC 2019] Para o desenvolvimento de aplicações Web, qual item abaixo contém apenas frameworks/bibliotecas/plataformas que foram desenvolvidas ou que dependem de JavaScript ou TypeScript:

a) Node.js, CSS, Java.

b) React, Node.js, Scala.

c) Angular, React, Vue.js.

d) Angular, Node.js, Java.

e) Java AWT, Angular, Scala.

Q17) [FEPESE CELESC 2019] Qual operador Javascript permite concatenar strings?

a) .

b) !

c) &&

d) &

e) +

Q18) [CCV-UFC UFC 2019] Para o desenvolvimento de aplicações Web, qual item abaixo contém apenas frameworks/bibliotecas/plataformas que foram desenvolvidas ou que dependem de JavaScript ou TypeScript:

a) Node.js, CSS, Java.

b) React, Node.js, Scala.

c) Angular, React, Vue.js.

d) Angular, Node.js, Java.

e) Java AWT, Angular, Scala.

Q19) [FEPESE CELESC 2019] Quais métodos Javascript abaixo constituem métodos válidos de strings?

1. slice()
2. indexOf()
3. substr()
4. search()

Assinale a alternativa que indica todas as afirmativas corretas.

a)São corretas apenas as afirmativas 1, 2 e 3.

b)São corretas apenas as afirmativas 1, 2 e 4.

c)São corretas apenas as afirmativas 1, 3 e 4.

d)São corretas apenas as afirmativas 2, 3 e 4.

e)São corretas as afirmativas 1, 2, 3 e 4.

Q19) [FEPESE CELESC 2019] Quais métodos Javascript abaixo constituem métodos válidos de strings?

1. slice()
2. indexOf()
3. substr()
4. search()

Assinale a alternativa que indica todas as afirmativas corretas.

a)São corretas apenas as afirmativas 1, 2 e 3.

b)São corretas apenas as afirmativas 1, 2 e 4.

c)São corretas apenas as afirmativas 1, 3 e 4.

d)São corretas apenas as afirmativas 2, 3 e 4.

e)São corretas as afirmativas 1, 2, 3 e 4.

Q20) [IBADE IF-RO 2019] A linguagem JavaScript suporta o uso de operadores relacionais, que possibilitam a análise de duas situações:

(1) O uso do operador retorna VERDADEIRO caso os operandos sejam iguais e do mesmo tipo.

(2) O uso do operador retorna VERDADEIRO se os operandos não são iguais.

Para as situações descritas em (1) e em (2) são empregados para os operadores, respectivamente, os seguintes símbolos:

a)== e <>

b)== e !=

c)=== e ?=

d)=== e <>

e)=== e !=

Q20) [IBADE IF-RO 2019] A linguagem JavaScript suporta o uso de operadores relacionais, que possibilitam a análise de duas situações:

(1) O uso do operador retorna VERDADEIRO caso os operandos sejam iguais e do mesmo tipo.

(2) O uso do operador retorna VERDADEIRO se os operandos não são iguais.

Para as situações descritas em (1) e em (2) são empregados para os operadores, respectivamente, os seguintes símbolos:

a)== e <>

b)== e !=

c)=== e ?=

d)=== e <>

e)=== e !=

Q21) [FEPESE CELESC 2019] Qual propriedade HTML DOM pode ser utilizada em Javascript para recuperar o conteúdo de um elemento HTML?

a)HTMLID

b)innerHTML

c)DOMContent

d)ElementHTML

e)getElementById

Q22) [CESPE MPC PA 2019] Considerando as linguagens de programação que têm recursos dinâmicos, a linguagem que é totalmente processada no computador-cliente é denominada

a) Java.

b) PHP (hypertext preprocessor).

c) ASP (active server pages).

d) CGI (Common Gateway Interface).

e) JavaScript.

Q21) [FEPESE CELESC 2019] Qual propriedade HTML DOM pode ser utilizada em Javascript para recuperar o conteúdo de um elemento HTML?

a)HTMLID

b)innerHTML

c)DOMContent

d)ElementHTML

e)getElementById

Q22) [CESPE MPC PA 2019] Considerando as linguagens de programação que têm recursos dinâmicos, a linguagem que é totalmente processada no computador-cliente é denominada

a) Java.

b) PHP (hypertext preprocessor).

c) ASP (active server pages).

d) CGI (Common Gateway Interface).

e) JavaScript.

Q23) [FEPESE CELESC 2019] Considere a página web abaixo.

```
<!DOCTYPE html>
<html>
   <head><title>Home</title></head>
   <body>
      <p>Texto 1</p>
      <p>Texto 2</p>
      <p>Texto 3 </p>
      <p id="x"></p>
      <script>
      var elementos = document.getElementsByTagName("p");
       I
      </script>
   </body>
</html>
```

Para exibir o conteúdo do segundo parágrafo ("Texto 2") no parágrafo cujo id="x", utiliza-se na lacuna I a instrução

a)document.write(x.elementos[2]);

b)document.getElementById("x").innerHTML = elementos[1];

c)document.write("#x").innerHTML= elementos[1].toString();

d)document.getElementById("x").innerHTML = elementos[1].innerHTML;

e)document.getElementById("x").add = elementos[1].innerHTML;

Q23) [FEPESE CELESC 2019] Considere a página web abaixo.

```
<!DOCTYPE html>
<html>
   <head><title>Home</title></head>
   <body>
      <p>Texto 1</p>
      <p>Texto 2</p>
      <p>Texto 3 </p>
      <p id="x"></p>
      <script>
      var elementos = document.getElementsByTagName("p");
       I
      </script>
   </body>
</html>
```

Para exibir o conteúdo do segundo parágrafo ("Texto 2") no parágrafo cujo id="x", utiliza-se na lacuna I a instrução

a)document.write(x.elementos[2]);

b)document.getElementById("x").innerHTML = elementos[1];

c)document.write("#x").innerHTML= elementos[1].toString();

d)document.getElementById("x").innerHTML = elementos[1].innerHTML;

e)document.getElementById("x").add = elementos[1].innerHTML;

Q24) [FUMARC COPASA 2018] Analise o seguinte código escrito na linguagem Javascript:

```
var frutas = new Array("banana", "laranja");

document.write(frutas[2]);
```

O resultado correspondente apresentado como saída é:

a) banana laranja

b) laranja

c)null

d) undefined

Q24) [FUMARC COPASA 2018] Analise o seguinte código escrito na linguagem Javascript:

```
var frutas = new Array("banana", "laranja");

document.write(frutas[2]);
```

O resultado correspondente apresentado como saída é:

a) banana laranja

b) laranja

c)null

d) undefined

Q25) [FGV MPE AL 2018] Considere os operadores a seguir. Igual a Não igual a Módulo (resto da divisão) Ou lógico And lógico Assinale a opção que indica a lista dos símbolos que, respectivamente, representam esses operadores no JavaScript.

a) == != % || &&

b) == <> %% || &&

c) = != & or and

d) == <> mod || &&

e) = != & or and

Q25) [FGV MPE AL 2018] Considere os operadores a seguir. Igual a Não igual a Módulo (resto da divisão) Ou lógico And lógico Assinale a opção que indica a lista dos símbolos que, respectivamente, representam esses operadores no JavaScript.

a) == != % || &&

b) == <> %% || &&

c) = != & or and

d) == <> mod || &&

e) = != & or and

Q26) [FGV AL RO 2018] Assinale a forma correta para inserir JavaScript em uma página, por meio de arquivos.

a) <script load("xxx.txt")></script>

b) <inputJava>xxx.js</inputJava>

c) <script onload="xxx.txt"></script>

d) <src file="xxx.js"></src>

e) <script src="xxx.js"></script>

Q26) [FGV AL RO 2018] Assinale a forma correta para inserir JavaScript em uma página, por meio de arquivos.

a) <script load("xxx.txt")></script>

b) <inputJava>xxx.js</inputJava>

c) <script onload="xxx.txt"></script>

d) <src file="xxx.js"></src>

e) <script src="xxx.js"></ script >

Q27) [FGV AL RO 2018] O comando JavaScript exibido a seguir escreve o texto “XXX” no interior de um elemento HTML.

```
document.getElementById("demo").innerHTML = "XXX";
```

Sabe-se que o elemento a ser modificado é localizado pelo valor de certo atributo. Assinale o nome do atributo HTML que deve possuir o valor “demo”.

a) id

b) identifier

c) key

d) name

e) value

Q27) [FGV AL RO 2018] O comando JavaScript exibido a seguir escreve o texto “XXX” no interior de um elemento HTML.

```
document.getElementById("demo").innerHTML = "XXX";
```

Sabe-se que o elemento a ser modificado é localizado pelo valor de certo atributo. Assinale o nome do atributo HTML que deve possuir o valor “demo”.

a) id

b) identifier

c) key

d) name

e) value

Q28) [Gestão Concurso EMATER-MG 2018] Existe uma linguagem de programação da Web que a maioria dos sites modernos usa e todos os navegadores atuais – em computadores de mesa, consoles de jogos, tablets e smartphones – a incluem, tornando-a a linguagem de programação mais onipresente da história.

Qual é a linguagem de programação descrita?

a) CSS.

b) HTML.

c) Ruby.

d) JavaScript.

Q29) [QUADRIX CRM-PR 2018] Acerca dos conceitos e padrões Java e JavaScript, julgue o item a seguir.

A linguagem JavaScript é uma extensão da plataforma Java Standard Edition destinada à criação de códigos interpretados em máquinas virtuais e navegadores web.

Q28) [Gestão Concurso EMATER-MG 2018] Existe uma linguagem de programação da Web que a maioria dos sites modernos usa e todos os navegadores atuais – em computadores de mesa, consoles de jogos, tablets e smartphones – a incluem, tornando-a a linguagem de programação mais onipresente da história.

Qual é a linguagem de programação descrita?

a) CSS.

b) HTML.

c) Ruby.

d) JavaScript.

Q29) [QUADRIX CRM-PR 2018] Acerca dos conceitos e padrões Java e JavaScript, julgue o item a seguir.

A linguagem JavaScript é uma extensão da plataforma Java Standard Edition destinada à criação de códigos interpretados em máquinas virtuais e navegadores web. ERRADO.

Q30) [Instituto AOCP PRODEB 2018] JavaScript é uma linguagem de programação que permite implementar itens complexos em páginas web, como toda vez que uma página da web faz mais do que simplesmente mostrar informação estática. Desta forma, o JavaScript é largamente utilizado para a validação de campos de entrada do usuário. Um dos comandos mais utilizados é aquele que promove o foco a um campo input quando o usuário preenche uma entrada de dados errada. A sintaxe correta do comando que realiza esse foco em um determinado campo é

a) HTMLElementObject.focus();

b) HTMLElementObject.onfocus();

c) HTMLElementObject.infocus();

d) HTMLElementObject.ondFocus();

e) HTMLElementObject.select();

Q30) [Instituto AOCP PRODEB 2018] JavaScript é uma linguagem de programação que permite implementar itens complexos em páginas web, como toda vez que uma página da web faz mais do que simplesmente mostrar informação estática. Desta forma, o JavaScript é largamente utilizado para a validação de campos de entrada do usuário. Um dos comandos mais utilizados é aquele que promove o foco a um campo input quando o usuário preenche uma entrada de dados errada. A sintaxe correta do comando que realiza esse foco em um determinado campo é

a) HTMLElementObject.focus();

b) HTMLElementObject.onfocus();

c) HTMLElementObject.infocus();

d) HTMLElementObject.ondFocus();

e) HTMLElementObject.select();

GABARITO

Q1 – LETRA D.
Q2 - LETRA D.
Q3 – LETRA C.
Q4 - LETRA A.
Q5 - LETRA B.
Q6 - LETRA A.
Q7 – LETRA E.
Q8 – LETRA A.
Q9 - LETRA A.
Q10 – ERRADO.
Q11 – LETRA E.
Q12 - LETRA A.
Q13 - LETRA B.
Q14 – LETRA A.
Q15 – ERRADO.
Q16 – LETRA E.
Q17 – LETRA E.
Q18 – LETRA C.
Q19 - LETRA E.
Q20 - LETRA E.
Q21 - LETRA B.
Q22 - LETRA E.
Q23 - LETRA D.
Q24 – LETRA D.
Q25 – LETRA A.
Q26 - LETRA E.
Q27 – LETRA A.
Q28 – LETRA D.
Q29 – ERRADO.
Q30 – LETRA A