

JavaScript



ANA CAROLINA N R GRACIOSO
carol.nrg@gmail.com

Agenda

- Objetos JavaScript;
- DOM;
- Render Tree;
- Eventos;
- Objeto Date;
- Manipulação de Strings;

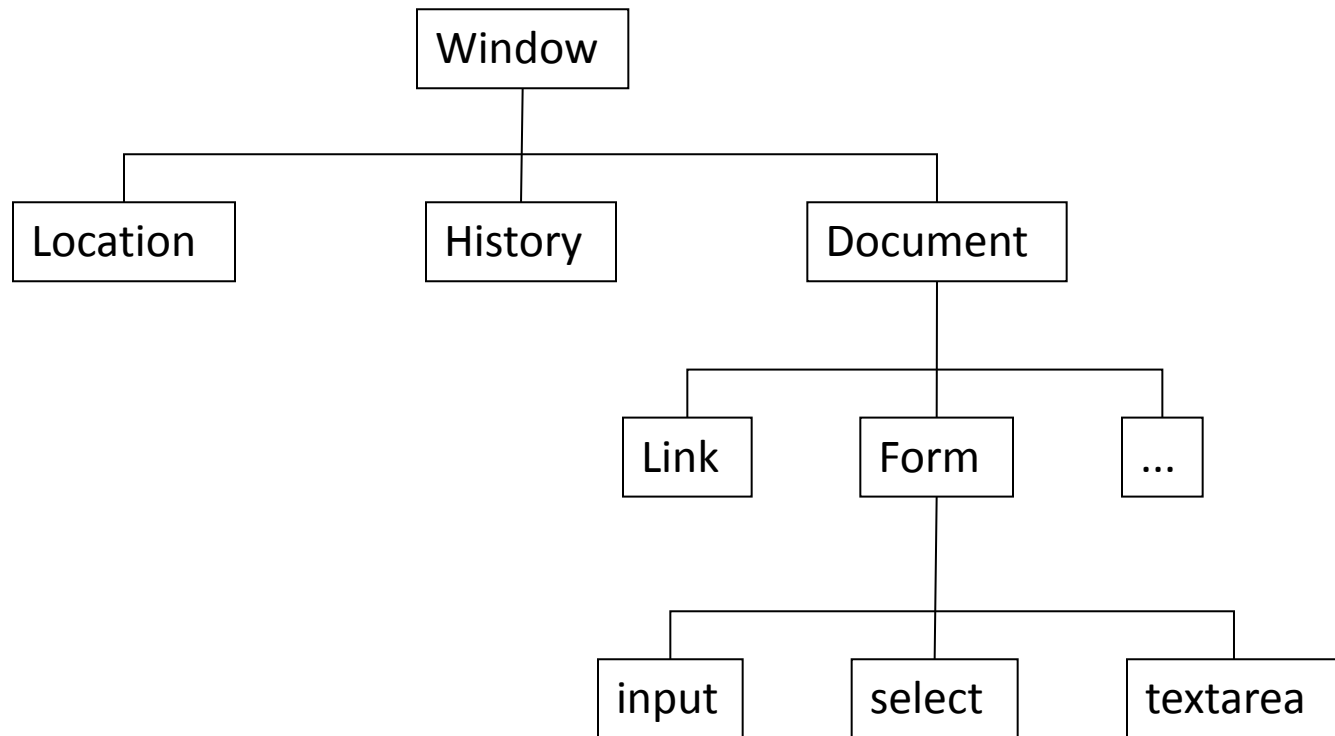
JavaScript

- **Orientação a Objetos:**

- **Objeto:** no mundo real, um objeto é alguma coisa que existe como por exemplo uma cadeira, uma bolsa, etc. Para o JavaScript, existem os objetos com que ele lida em navegadores Web, como janelas, formulários, elementos do formulário, dentre outros.
- **Propriedades:** os objetos possuem propriedades. As propriedades podem modificar os objetos, e a mesma propriedade pode se aplicar a objetos completamente diferentes. Nome, tamanho, cor são propriedades que podem ser encontradas na maioria dos objetos JavaScript.
- **Métodos:** os métodos são as ações que os objetos podem realizar. Imprimir, abrir uma janela, fechar uma janela, são exemplos de ações (métodos) que um objeto JavaScript pode realizar.
- **Eventos:** os são ações que o usuário executa enquanto visita a página. Exemplo: entrar na página, sair da página, clicar, passar o mouse, etc.

Objetos

- Hierarquia dos objetos JavaScript:



http://www.w3schools.com/jsref/obj_window.asp
http://www.w3schools.com/jsref/obj_location.asp
http://www.w3schools.com/jsref/obj_history.asp

Exemplo

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Exemplo JavaScript</title>
  <script type="text/javascript">
    alert(window.location);
  </script>
</head>
<body>
</body>
</html>
```

Exemplo

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Exemplo JavaScript</title>
  <script type="text/javascript">
    alert(location.pathname);
  </script>
</head>
<body>
</body>
</html>
```

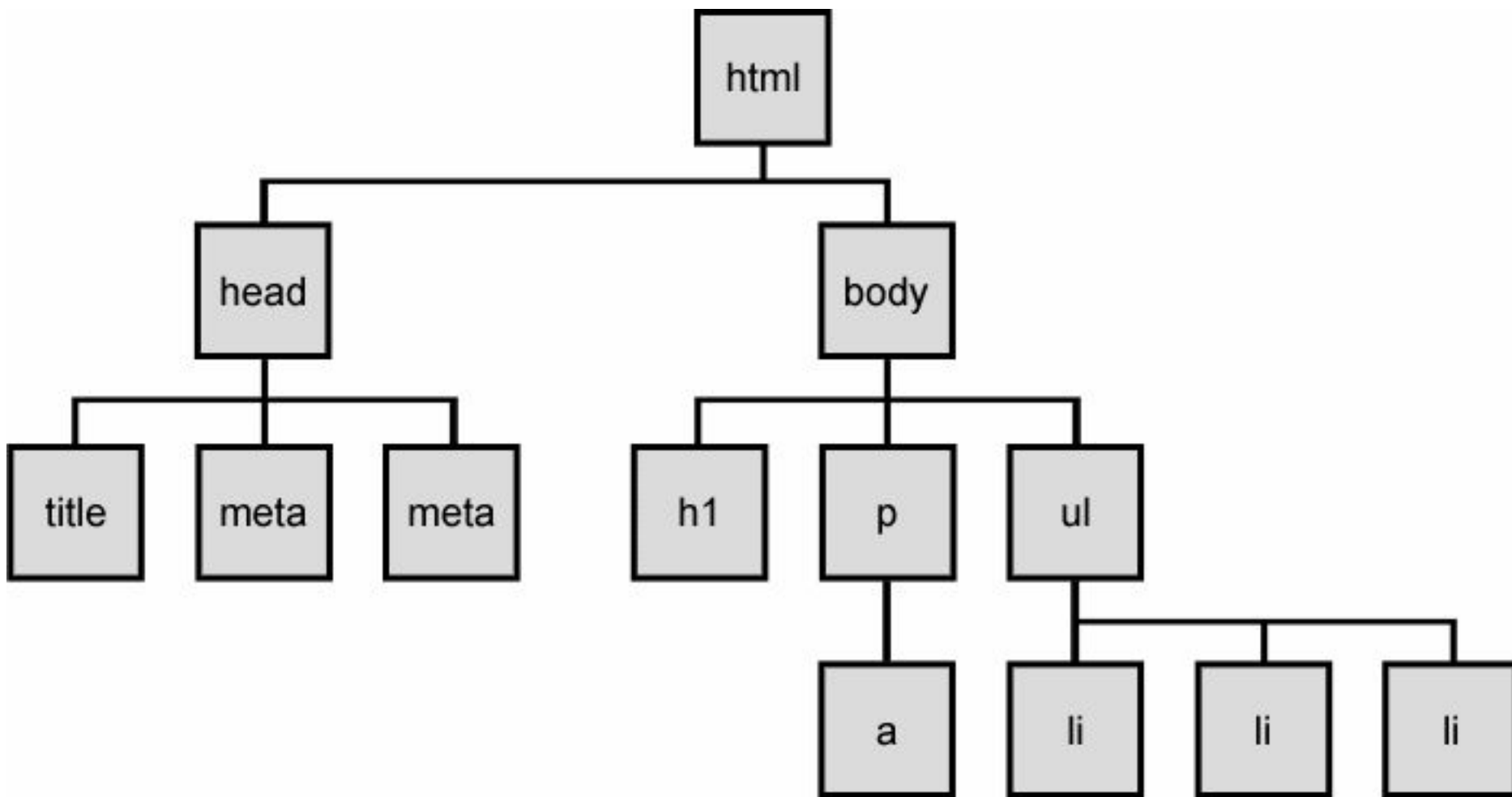
Exemplo

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Exemplo JavaScript</title>
  <script type="text/javascript">
    alert(history.length);
  </script>
</head>
<body>
</body>
</html>
```

DOM (Document Object Model)

- Criado pelo W3C, O DOM é uma **multi-plataforma que representa como as marcações em HTML, XHTML e XML** são organizadas e lidas pelo navegador que você usa.
- Uma vez indexadas, estas marcações se transformam em elementos de uma árvore que você pode manipular – que é o que fazemos quando usamos programas ou scripts para alterar funcionalidades de uma página: conteúdo, estrutura ou folha de estilo.

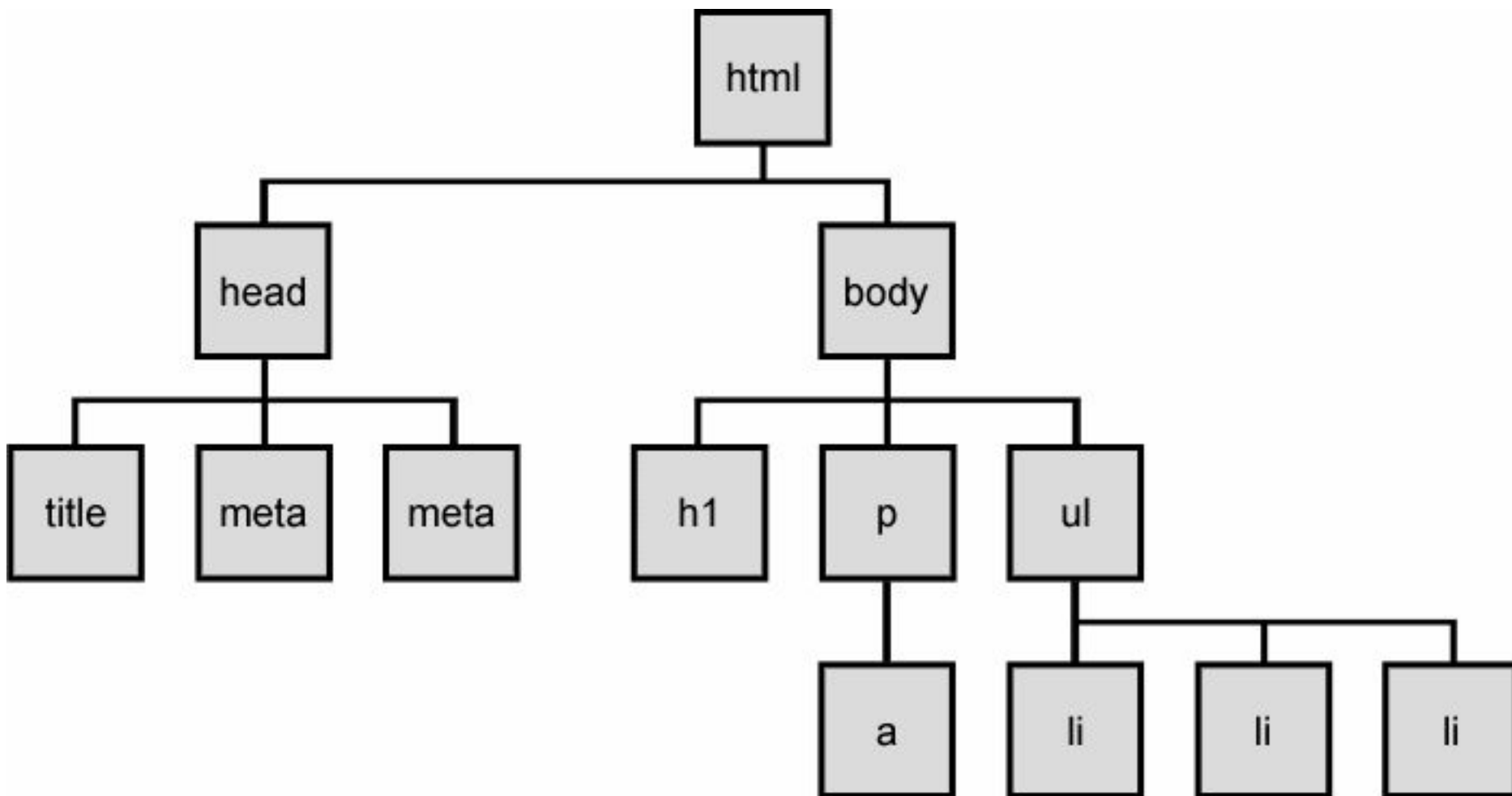
DOM nos Navegadores



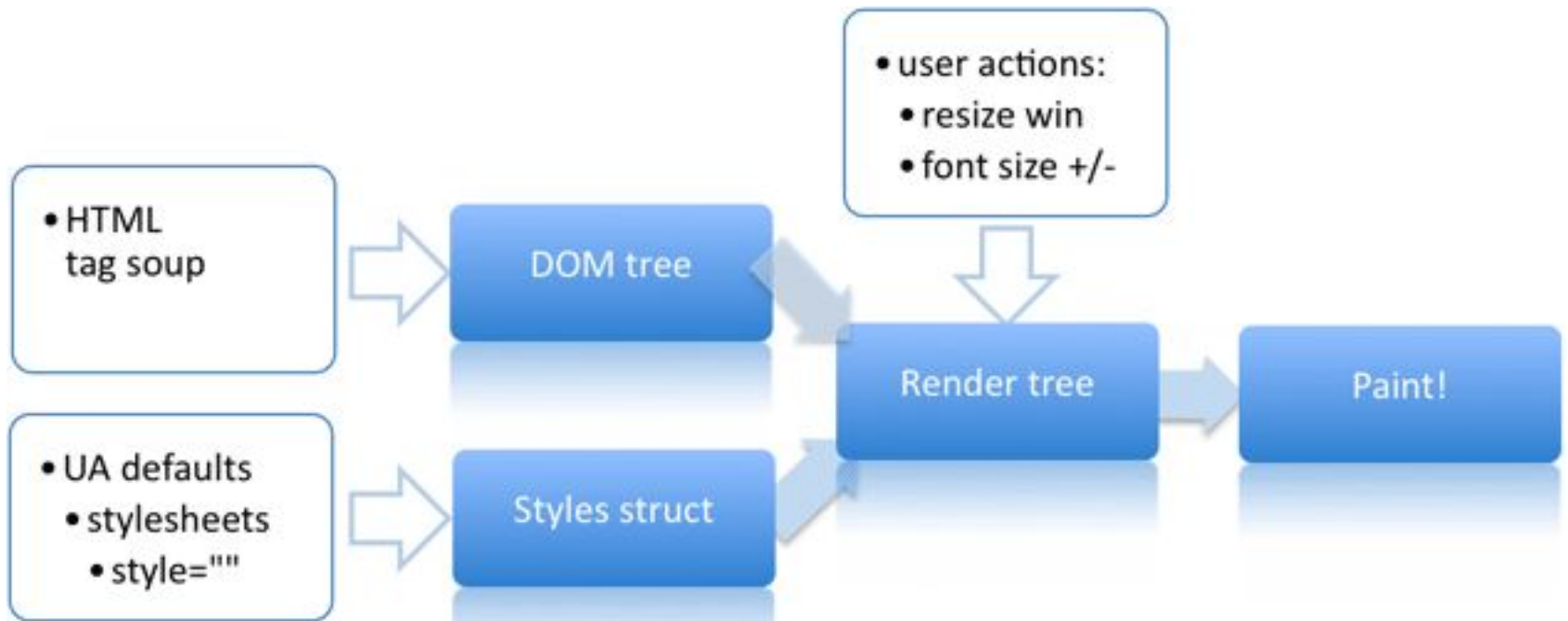
DOM nos Navegadores

- Esta estrutura não será o que veremos no navegador – o layout em si.
- O DOM é a base para uma outra árvore que é o que realmente um browser monta na tela, a Árvore de Renderização – Render Tree.

DOM nos Navegadores



Render Tree



Render Tree

- A Render Tree é a parte mais importante do processo de renderização.
- Bem parecida com a árvore DOM, cada objeto corresponde a nós de Documentos, Elementos ou Texto.
- A diferença é que a Render Tree possui também objetos que não possuem nós na árvore DOM, como **scripts** e folhas de estilos.

Eventos

- **OnLoad:** Ocorre quando a página é carregada pelo browser.
- **OnUnload:** Ocorre quando o usuário sai da página.
- **OnChange:** Ocorre quando o valor de um campo é modificado.
- **OnBlur:** Ocorre quando o elemento perde o foco.
- **OnFocus:** Ocorre quando o elemento recebe o foco.
- **OnClick:** Ocorre quando o elemento é clicado.
- **OnMouseover:** Ocorre quando o cursor do mouse passa sobre o elemento.
- **OnSelect:** Ocorre quando o elemento é selecionado.
- **OnSubmit:** Ocorre quando o formulário é submetido.

http://www.w3schools.com/jsref/dom_obj_event.asp

Funções

- As funções, também conhecidas como sub-rotinas, são muito utilizadas em programação.
- Um dos grandes benefícios é não precisar copiar o código todas as vezes que precisar executar aquela operação, além de deixar a leitura do código mais intuitiva.

Criar Função:

```
function NomeFuncao(){  
    // instruções  
}
```

Chamar Função:

```
NomeFuncao()
```

Funções

- As funções, também podem retornar valores...

```
function NomeFuncao(){  
    // instruções  
    return valor;  
}
```


Funções

- Ou receber valores...

```
function NomeFuncao(a, b){  
    // instruções  
    return x;  
}
```

Métodos

- `parseInt()`
- `document.getElementById('IdElemento')`
 - `document.getElementById('IdElemento').value`
 - `document.getElementById('IdElemento').style`
- `addEventListener`

`document.getElementById('IdElemento').addEventListener("evento",Funcao)`

Exemplo (má prática)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Exemplo JavaScript</title>
</head>
<body>
  <form>
    <input type="button" value="Botão"
      onClick="alert('O botão foi clicado')">
  </form>
</body>
</html>
```

Exemplo (pq não funciona?)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Exemplo JavaScript</title>
  <script>
    document.getElementById("meuBt").addEventListener("click",
    minhaFuncao);

    function minhaFuncao(){
      alert("O botão foi clicado");
    }
  </script>
</head>
<body>
  <form>
    <input type="button" value="Botão" id="meuBt">
  </form>
</body>
</html>
```

Soluções

- Colocar script no final do body;
- Colocar script externo e usar atributo **defer**
- Utilizar DOMContentLoaded

```
document.addEventListener("DOMContentLoaded", function() {  
    ...  
});
```

Exemplo

(função anônima)

```
<script>  
    document.getElementById("meuBt").addEventListener("  
click",function(){ alert("O botão foi clicado"); });  
</script>
```

Exemplo

(arrow function)

```
<script>  
    document.getElementById("meuBt").addEventListener("  
click",()=>{ alert("O botão foi clicado"); });  
</script>
```

Objeto Date

- **date()** retorna a data e a hora correntes do computador do usuário no formato:
Dia da Semana, Nome do mês, Dia do mês, Hora:minuto:segundo.
Exemplo: Mon Sep 15 18:27:55 2008
- Para se obter os dados separadamente, existem os seguintes métodos:
 - **getDate()**: dia do mês (1 a 31)
 - **getDay()**: dia da semana (0 a 6)
 - **getMonth()**: mês do ano (0 a 11)
 - **getFullYear()**: ano (-1900)
 - **getHours()**: hora (0 a 23)
 - **getMinutes()**: minutos (0 a 59)
 - **getSeconds()**: segundos (0 a 59)

Exemplo

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Exemplo JavaScript</title>
  <script type="text/javascript">
    let data = new Date();
    document.write(data);
    let dia = data.getDate();
    document.write("<br>" + dia);
  </script>
</head>
<body>
</body>
</html>
```

Exemplo

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"><title>Exemplo JavaScript</title>
</head>
<body>
  <input type="text" id="n1"> + <input type="text" id="n2">
  <button id="somar"> = </button> <input type="text" id="res">
  <script type="text/javascript">
    function Soma(){
      let n1 =
parseInt(document.getElementById('n1').value);
      let n2 =
parseInt(document.getElementById('n2').value);
      let res = n1 + n2;
      document.getElementById('res').value = res;
      document.getElementById('res').disabled = true;
    }

document.getElementById("somar").addEventListener("click",Soma);
  </script>
</body>
</html>
```

Manipulando Strings

- **string.length**: retorna o tamanho da string;
- **string.charAt(posicao)**: retorna o caracter da posição especificada;
- **string.indexOf("string")**: retorna o numero da posição onde começa a primeira ocorrência da string;
- **string.lastIndexOf("string")**: retorna o numero da posição onde começa a ultima ocorrência da string;
- **string.substring(x,y)**: retorna o conteúdo da string que corresponde ao intervalo: x a y-1.
- **string.toUpperCase()**: transforma o conteúdo da string em maiúsculo;
- **string.toLowerCase()**: transforma o conteúdo da string em minúsculo.

Exemplo

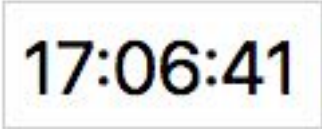
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Exemplo JavaScript</title>
</head>
<body>
  <input type="text" id="texto">
  <button onClick="Exemplo()">Clique Aqui</button>
  <script type="text/javascript">
    function Exemplo(){
      let texto = document.getElementById('texto').value;
      let aux = texto.length;
      alert(aux);
    }
  </script>
</body>
</html>
```

Atividades

1. Implemente um script JavaScript que exiba a data atual no formato:

Quarta-feira, 15 de fevereiro de 2023

2. Pesquise a utilização do método `setTimeout()` e implemente um relógio que atualize o horário a cada segundo:

A digital clock display showing the time 17:06:41. The time is displayed in a black, sans-serif font within a light gray rectangular border.

3. Implemente um script que leia uma palavra ou frase de um campo de texto e exiba em uma janela de alert se o texto é ou não um palíndromo.

ATUALIZE SEU PROJETO NO GIT E COLOQUE EM PRODUÇÃO

Referências

- http://www.w3schools.com/jsref/obj_window.asp
- http://www.w3schools.com/jsref/obj_location.asp
- http://www.w3schools.com/jsref/obj_history.asp
- http://www.w3schools.com/jsref/dom_obj_event.asp
- http://www.w3schools.com/js/js_events.asp
- http://www.w3schools.com/jsref/jsref_obj_string.asp
- http://www.w3schools.com/js/js_functions.asp
- http://www.w3schools.com/js/js_date_methods.asp
- http://www.w3schools.com/js/js_date_methods.asp

Leitura Complementar

- [https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What is JavaScript](https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript)
- [https://developer.mozilla.org/pt-BR/docs/Learn/Getting started with the web/JavaScript basics](https://developer.mozilla.org/pt-BR/docs/Learn/Getting_started_with_the_web/JavaScript_basics)
- https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/Variables
- https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/Math
- https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/Strings
-