AULA 8:
Introdução a JavaScript - JS

Características do JavaScript

- Linguagem poderosa, com sua grande aplicação do lado cliente, ou seja, executada pelo browser;
- É uma linguagem de scripts que permite interatividade nas páginas web;
- Podemos definir scripts como pequenas sequências de instruções ("miniprogramas") interpretados pelo browser e voltados para execução de tarefas específicas;
- Ele é incluído na página HTML e interpretada pelo navegador;
- Sua linguagem e estrutura são simples, porém pode-se criar construções complexas e criativas;
- **JavaScript não é Java**; são linguagens com aplicações e recursos totalmente distintos;
- Não é capaz de recuperar informações de outro arquivo ou salvar dados em um servidor da Web, ou no computador do usuário.

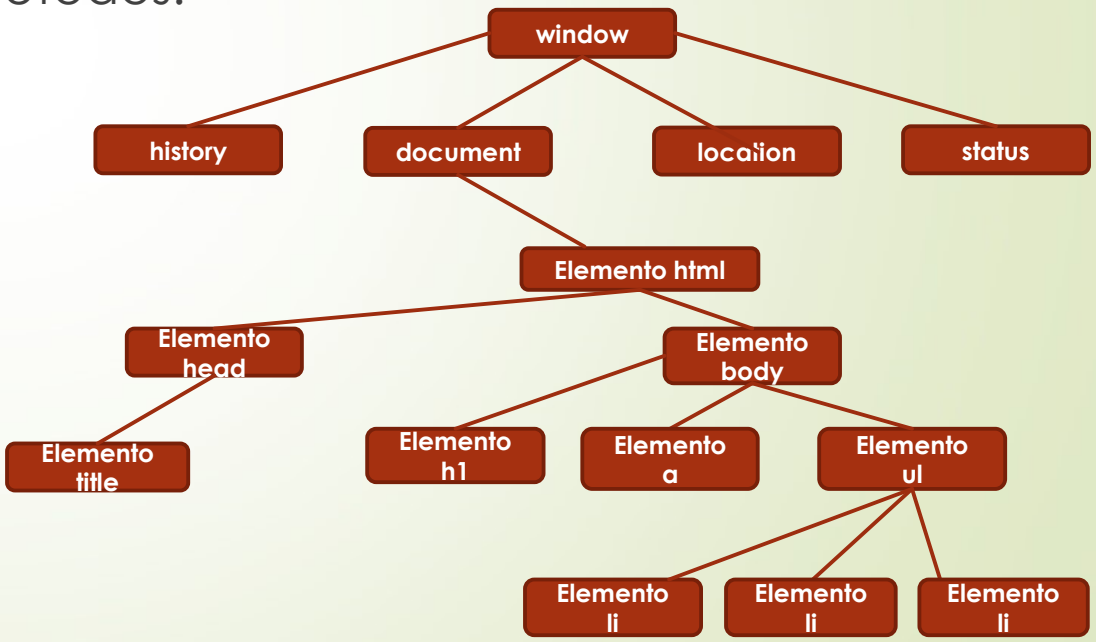
O que se pode fazer com JavaScript?

- Interagir com o código HTML, adicionando interatividade a sua página (em geral as páginas HTML são estáticas);
- Criar janelas popup;
- Interação com formulários (validando dados, por exemplo);
- Cálculos;
- Efeitos especiais;
- Atualização do conteúdo da página após sua visualização;
- Criar cookies (O que são cookies?);
- Abrir janelas do navegador, trocar informações entre janelas, manipular propriedades como histórico, barra de status, plug-ins, applets e outros objetos;
- Interagir com o usuário através do tratamento de eventos;
- Estes são apenas exemplos... Use sua imaginação!

Características Básicas: Objetos

- Ao renderizar uma página web, o navegador constrói o DOM (Document Object Model), uma coleção hierárquica de objetos que representam os elementos HTML presentes na página;
- Cada elemento é representado no DOM por um objeto diferente, incluindo objetos globais (como window e document) que não estão vinculados a elementos HTML. Um objeto é uma coleção de propriedades e que possui um comportamento definido por seus métodos.

```
<html>
        <head>
          <title>Aula 08 - CAW</title>
        </head>
        <body>
                <h1>Curso de JS</h1>
                <a href="#">Meu link</a>
                <ul>
                        <li>HTML</li>
                    <li>CSS</li>
                  <li>JScript</li>
                </ul>
        </body>
</html>
```



Características Básicas: Objetos

- Os atributos da tag viram propriedades do objeto;
- Algumas propriedades podem ter seu valor modificado;
- O navegador mantém a coerência entre o valor da propriedade e o que está sendo visualizado pelo usuário;

Exemplo - Objeto window (Manipula as janelas do navegador)
Algumas Propriedades:

- document: referência ao objeto Document, contendo todas as características da página HTML;
- history: referência ao objeto history, com a lista das URLs visitadas;
- innerHeight: define/obtém a altura da área onde o conteúdo é apresentado, não a altura do navegador em si;
- innerWidth: define/obtém a largura da área onde o conteúdo é apresentado, não a largura do navegador em si;
- location: informações referentes ao endereço corrente;
- fullscreen: indica se a janela é exibida em tela cheia ou não.

Características Básicas: Objetos

Exemplo - Objeto window (Manipula as janelas do navegador)
Alguns Métodos:

- Reload: Que força o navegador a carregar a página.
- Replace: Que carrega a URL informada.
- Alert: cria uma subjanela popup

Vamos testar???

1) No seu navegador, entre na página www.faeterj-rio.edu.br;
- Depois clique na barra lateral cinza (botão direito do mouse -> "Inspecionar");
- Na janela que se abrirá, selecione "Console" e entre com o comando: "document.body.style.background = "blue". O que acontece?

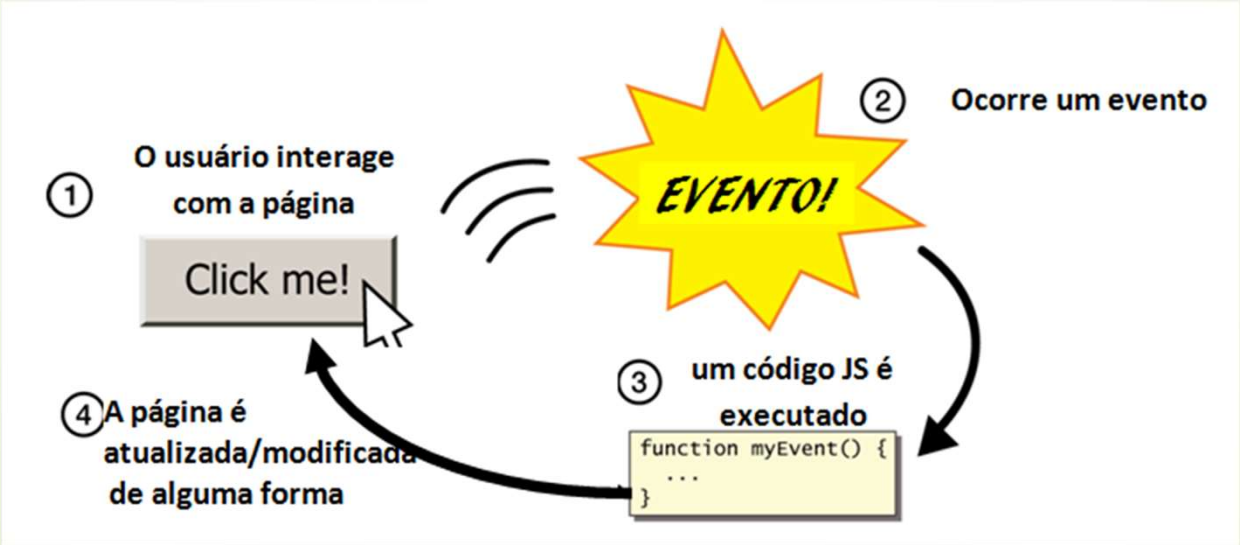
2) E se aplicarmos o comando:

window.location.href = "http://www.google.com"

Mais detalhes podem ser encontrados em: https://www.w3schools.com/js/js_htmldom.asp

Características Básicas: Eventos

- Sempre que algo acontece em uma página Web, ocorre um evento;
- Eventos podem ser qualquer coisa: clique do mouse, o mouse é arrastado, uma página é carregada, uma imagem recebe o foco, a abertura de uma página web ou imagem, o envio de um formulário HTML, uma tecla pressionada, entre outras;
- Com o Javascript, é possível realizar ações quando um evento ocorre.



Código JavaScript no HTML

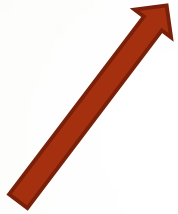
- Nesse contexto, utilizam-se alguns termos:
  - Objeto: dados e funcionalidade juntos;
  - Propriedade: atributos (valores) que são associados a alguma coisa;
  - Método: uma ação que um objeto pode realizar;
  - Evento: uma ação iniciada por um usuário ou pelo computador;
  - Variável: um lugar para armazenar valores em um computador (propriedade está relacionada a objeto);
  - Função: uma rotina ou procedimento que realiza uma ação (método está relacionado a objeto).

- Para inserir JavaScript na página HTML, deve se utilizar a tag <script></script> - delimita o código em JavaScript

```
Exemplo: <script type="text/javascript" >
                    // código javascript;
                </script>
```

Código JavaScript no HTML

```
<!DOCTYPE html>
<html>
  <head>
      <meta charset="utf-8">
  </head>
  <body>
      <p id='idLugar'></p>
      <script type="text/javascript">
          var lugar;
          lugar=
document.getElementById('idLugar');
          lugar.innerHTML = "Alô mundo!";
      </script>
  </body>
</html>
```



document . getElementById('idLugar');

Elemento HTML identificado pela string idLugar

Método() do Objeto document

Acessa métodos disponíveis de um objeto

Objeto raiz de todos os objetos do documento HTML

Código JavaScript no HTML

```
<!DOCTYPE html>
<html>
  <head>
      <meta charset="utf-8">
  </head>
  <body>
      <script type="text/javascript">
            var lugar;
            lugar =
document.getElementById('idLugar');
            lugar.innerHTML = ”<strong>Alô
mundo!</strong>";
      </script>
      <p id='idLugar'></p>
  </body>
</html>
```

Há um erro no código ao lado...
O que você acha que acontecerá?

Código JavaScript no HTML

- Onde devemos incluir o código JavaScript?
  - Depende de quando a ação deve acontecer:
    - ✓ quando o usuário entrar no site?
    - ✓ quando o usuário clicar em um botão?
    - ✓ quando o usuário preencher um formulário?
    - ✓ ...
  - Há mais de uma maneira de embutir os códigos:
    - ✓ scripts no <head>
    - ✓ scripts no <body>
    - ✓ scripts no <head> e <body>
    - ✓ scripts externos
- Na seção head, os scripts podem ser executados quando chamados ou quando algum evento ocorre;
- Na seção body, os scripts são executados uma vez ao carregar a página web;
- Para definição externa, um arquivo ".js" precisa ser fornecido com as funções necessárias.

Código JavaScript no HTML

- Incluir o script no cabeçalho (<head>) de uma página garante que ele será executado antes desta carregar:

```
<!DOCTYPE html>
<html>
  <head>
      <script type="text/javascript">
            window.alert("Interrompe Carregamento!");
             console.log("Verique esse texto no console, apareceu?");
      </script>
  </head>
  <body>
      Blah Blah Blah
  </body>
</html>
```

Código JavaScript no HTML

- O comando document.write() é um comando padrão do JavaScript para escrever em uma página HTML. O uso deve ser feito **com cautela** e, normalmente, para teste. Após carregar a página, o uso deste comando sobrescreve todo o conteúdo já disponibilizado:

```
<!DOCTYPE html>
<html>
  <head>
      <script type="text/javascript">
            function myFunc(){
              document.write("<p><h1>Sobrescreve o conteúdo anterior</h1></p>");
            }
      </script>
  </head>
  <body onload="myFunc()">
      Blah Blah Blah
  </body>
</html>
```

Código JavaScript no HTML

- O que acontecerá nesses dois casos? Responda mentalmente e depois confirme ;)

```
<html>
  <head>
      <script type="text/javascript">
          document.write("<title>Título da página</title>");
      </script>
  </head>
  <body>
      <script type="text/javascript">
          document.write("<h1>Conteúdo da página</h1>");
      </script>
  </body>
</html>
```

```
<html>
  <head>
      <title>Página com Javascript externo</title>
      <script type="text/javascript" src="arquivo.js">
      </script>
  </head>
  <body>
      Corpo da Página
  </body>
</html>

Arquivo "arquivo.js"
// código javascript
/* aqui eu posso colocar qualquer comando JS */
alert("Alo mundo!");
```

Comando e Código JavaScript

- A linguagem Javascript é case-sensitive (ou seja, há distinção entre representação com letras maiúsculas e minúsculas);
- Um comando Javascript é executado pelo navegador;
- Um comando Javascript termina com ";" (boa prática de programação);
- O uso de ";" permite escrever mais de um comando por linha;
- Código Javascript é uma sequência de comandos Javascript. O código é executado sequencialmente, comando por comando;

```
<!DOCTYPE html>
<html>
  <head>
        <title>Page Title</title>
          <meta charset="utf-8"/>
          <style>
                h1 {color:#aaa;}
                body {background-color:#ffa;margin:8px;}
          </style>
  </head>
  <body>
        <div id='idLugar'> </div>
        <script type="text/javascript">
          var lugar;
          lugar = document.getElementById('idLugar');
          lugar.innerHTML = "<p><h1>Oi Turma!</h1></p>";
        </script>
  </body>
</html>
```

Comando e Código JavaScript

Bloco de Comandos:

- Comandos Javascript podem ser agrupados em blocos;
- Um bloco é iniciado por um { e termina por um };
- A finalidade do bloco é agrupar os comandos para que eles sejam tratados como se fossem apenas um único comando (comando composto);

Comentários:

- Finalidade: documentação;
- Devem ser incluídos para explicar o código e facilitar a sua manutenção;
- Tipos de comentários:
  - em uma única linha: inicia com os caracteres //. Indica ao interpretador de JavaScript que o resto da linha é um comentário. Ele ignora o resto da linha, continuando a interpretar o código na linha seguinte;
  - com várias linhas: inicia com "/*" e termina com a sequência de caracteres "*/";

Comando e Código JavaScript

```
<!DOCTYPE html>
<html>
	<head>
		<title>Page Title</title>
		<meta charset="utf-8">
	</head>
	<body>
		<div id='lugar'>Modificando o documento pelo JavaScript</div>
		<script type="text/javascript">
		{
			var doc = document.getElementById("lugar");
			doc.innerHTML = "<h1>Oi Turma!!!</h1>"
			doc.innerHTML =doc.innerHTML +"<ol>Aprenderemos:" // cria lista
			doc.innerHTML =doc.innerHTML +"<li>Variaveis</li>"; // opção 1
			doc.innerHTML =doc.innerHTML +"<li>Tipos de Dados</li> "; // opção 2
			doc.innerHTML =doc.innerHTML + "<li>Entrada e Saida</li>";//opção 3
			doc.innerHTML =doc.innerHTML +"</ol>"; // Finalização da lista
		}
		</script>
	</body>
</html>
```

Comando e Código JavaScript

- Para entender o uso de JavaScript (entrada e saída, variáveis), usaremos um exercício:

  Construa um script em JavaScript que permita o usuário introduzir:

  - Nome;
  - Idade;
  - Gasto diário com locomoção: (em R$);
  - Tempo semanal de estudo extraclasse (em h);
  - Tempo semanal livre (em h) e mostrar seus dados; e
  - Relação do tempo livre com o tempo total, dada por: tempo livre*100/(tempo livre + tempo estudo)

- Para resolver esse problema, devemos:

  1) Criar variáveis necessárias;
  2) Perguntar ao usuário seu Nome, Idade, Gasto diário com locomoção: (em R$), Tempo semanal de estudo extraclasse (em h) e Tempo semanal livre (em h), armazenando as respostas do usuário em variáveis;
  3) Calcular a relação de tempo, armazenando o resultado em uma variável;
  4) Mostrar as respostas no documento.

Variáveis

- Identificador capaz de guardar um valor;
- As variáveis podem ser:
  - Globais: são declaradas fora de qualquer função (veremos mais tarde) e podem ser utilizadas em qualquer parte do script (permanentemente acessível);
  - Locais: declaradas dentro de uma função e só podem ser utilizada dentro dessa função e identificadas pelo uso da palavra reservada var;
- Atenção: Variáveis criadas sem o uso de var são sempre globais, mesmo quando criadas dentro de uma função!

Declaração e uso de variáveis

- Nomes de variáveis: uso de Letras, números, _ e $. Não podem começar por números e nem ser iguais a palavras reservadas

Exemplo: var peso; // cria a variável local peso
idade = 18; //cria e inicializa a variável global idade com o valor 18

- Os tipos de dados são assumidos dinamicamente: valor = 25;
valor = "vinte e cinco";

Variáveis

- Cuidado: não utilize nomes com escritas diferentes mas pronúncias iguais ou parecidas.

  Exemplo: não crie as seguintes variáveis:

  - Nome e nome
  - Idade e idade
  - Num1 e num1

  } Javascript é case-sensitive e você pode confundir as variáveis

- **Não** utilize as palavras reservadas como nome de variável, função ou objeto, tais como:

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

Variáveis

- JavaScript é capaz de reconhecer os seguintes tipos de dados:
  - Números: 12; 3.14159
  - Texto (variáveis de tipo string): "Oi Turma!"
  - Valores lógicos: true; false
  - Tipos especiais:
    - null: palavra reservada que indica que a variável não guarda nenhum valor
    - undefined: valor de variável não inicializada
    - NaN: valor numérico que representa um não-número (erro - Not a Number)
- Atribui-se um valor a uma variável: identificador da variável = expressão
- Primeiro, resolve-se a expressão e depois guarda-se o resultado na variável;

Exemplo:

Valor constante	Valor de 1 variável	Operações
total = 100;	numero2 = numero1;	soma = numero1 + numero2;
peso1 = 10;	peso1 = numero2;	total = total + 1;
peso2 = 10.		

Variáveis

Operadores

- Soma + : a + b
- Subtração – : a – b
- Multiplicação * : a * b
- Divisão / : a / b
- Resto da divisão de inteiros % : a % b
- Incremento de uma unidade ++ : ++a a++
- Decremento de uma unidade – – : --a a--

Adição e concatenação

- Se ambos os operadores são numéricos então soma os números. Senão, converte ambos os valores para string e concatena ambos. Exemplo: '$' + 3 + 4 -> '$34'
- Observação: +"42" -> 42 +"3" + (+"4") -> 7 +"3" + "4" -> "34“

Divisão A divisão de dois valores inteiros pode resultar em um valor real.

Exemplo: 10 / 3 = 3.3333333333333335

Variáveis

String Exemplo: nome = "João"; sobrenome = "Silva";
nomeCompleto = nome + " " + sobrenome;

- JavaScript converte os valores para realizar as operações:

Exemplos:
"Agosto " + 1977 torna o nº uma string e concatena as strings -> Agosto 1977
7 * "3" : primeiro converte a string em nº e depois realiza a multiplicação -> 21

- Em um contexto numérico: null é convertido para 0
  undefined é convertido para NaN
- Valor 0 é interpretado como "false". Todos os outros números são interpretados como "true";

- Atribuição Composta: utilizada para alterar o valor de uma variável, usando seu antigo valor

Operador	Uso Compacto	Uso Explícito
+=	a+=b	a=a+b
-=	a-=b	a=a-b
*=	a*=b	a=a*b
/=	a/=b	a=a/b
%=	a%=b	a=a%b

Variáveis – Formatação de Saída

- toFixed():
  - Converte número para string, mantendo o número de casas decimais especificadas

    var num = 5.56789;

    var n = num.toFixed(2); -> 5.57
- toPrecision():
  - Formata um número para um comprimento específico.

    var num = 15.56789;

    var n = num.toPrecision(2); -> 15

    var n = num.toPrecision(4); -> 15.57

Janela de Saída

window.alert() ou apenas alert()

- Abre uma janela para exibir um aviso ao usuário;
- Atenção: A saída do alert não é HTML. Assim, para uma nova linha use \n em vez de
;

Exemplo: window.alert(" Senha incorreta, acesso negado ");

window.confirm() ou apenas confirm()

- Abre uma janela para o usuário confirmar ou não
- Tem dois botões: OK e Cancel: (1) Se o usuário pressiona OK, retorna valor "true"; (2) Se o usuário pressiona Cancel, retorna valor "false";

Exemplo: var question = confirm("Quer continuar com este download?");

Entrada de Dados

window.prompt() ou prompt()

- Abre uma janela para pedir uma string ao usuário.

Exemplo: nome=prompt("Qual o seu nome?");

- Um valor padrão (default) pode ser fornecido. Isso evita que apareça "undefined" na linha da resposta.

Exemplo: nome=prompt("Qual o seu nome?", "(sem nome)");

- Como o comando prompt retorna uma string,

  (a) para perguntar por um valor inteiro é necessário convertê-la para inteiro => **parseInt (str)**: converte uma string para um número inteiro

  Exemplo: id=parseInt(prompt("Idade?"));

  (b) para perguntar por um valor real é necessário convertê-la para real => **parseFloat (str)**: converte uma string para um número real

  Exemplo: peso=parseFloat(prompt("Peso?"));

Relembrando Exercício

Utilizando os conceitos aprendidos até agora, resolva o exercício:

Construa um script em JavaScript que permita o usuário introduzir:

- Nome;
- Idade;
- Gasto diário com locomoção: (em R$);
- Tempo semanal de estudo extraclasse (em h);
- Tempo semanal livre (em h);
- Mostre seus dados; e
- Relação do tempo livre com o tempo total: tempo livre*100/(tempo livre + tempo estudo)

E apresente numa página HTML

Referenciando Elementos

- A estrutura de objetos que representa cada uma das tags na página HTML é chamada de DOM (Document Object Model) e pode ser acessada através da variável global document;
- Cada elemento do documento HTML torna-se um objeto (chamado de nó) e o JS pode acessá-los independentemente.
- O nó de um elemento é distinguido pelo nome do elemento: head, body, h1...;
- Porém, os elementos não são sempre únicos => identificar pelo id;
- Manipular elementos por meio do DOM é semelhante a aplicar estilos CSS:
  - Especificar o elemento (ou grupo);
  - Especificar o efeito a ser aplicado.
- Para acessar um elemento pelo id em JS:

  variável = elemento.getElementById("string_com_id");

```
<div id='idLugar'> </div>
    <script type="text/javascript">
        var lugar;
        lugar = document.getElementById('idLugar');
        ...
    </script>
```

Referenciando Elementos

- Observe a construção HTML:

```
<body>
        <div id='idLugar'> </div>
```

- O documento tem um elemento chamado 'idLugar' não é uma variável, ou seja, fazer algo do tipo idLugar.style.color='red' não é possível;
- Para fazer alguma operação com os atributos do elemento é preciso "copiá-lo" para uma variável: lugar = document.getElementById('idLugar');

- Modificando o conteúdo: **document.getElementById(id).innerHTML = novo HTML**
- Modificando um estilo: **document.getElementById(id).style.propriedade = novo estilo**
- Modificando um atributo: **document.getElementById(id).atributo = novo valor**

Exemplos:

```
{ var lugar;
  lugar = document.getElementById('idLugar');
  lugar.innerHTML = "<p><h1>Oi Turma!</h1></p>";
  lugar.style.color = "red";
  document.getElementById("img1").src =
http://www.imagensanimadas.com/data/media/1214/garfield-imagem-animada-0032.gif }
```

HTML, CSS, Javascript, Eventos

- Algumas propriedades:

Propriedades CSS	Propriedades Javascript
backgroud-color	backgroundColor
border-radius	borderRadius
font-size	fontSize
list-style-type	listStyleType
word-spacing	wordSpacing
z-index	zIndex

- Eventos são ações que ocorrem como resultado de atividades do browser ou interações do usuário com a página Web

  Exemplos de eventos:
  - Uso do mouse: clique, duplo clique, movimento,...;
  - Carga de uma página ou figura, ...;
  - A seleção de um campo de entrada em um formulário HTML;
  - O envio de um formulário;
  - Entrada de dados: apertar uma tecla, digitar, soltar uma tecla, ...;

HTML, CSS, Javascript, Eventos

- Eventos Comuns:

Evento	Ocorrência
onChange	Quando muda o valor do elemento
onClick	Quando o elemento é clicado pelo usuário
onFocus	Quando o elemento é focado
onKeyPress	Quando o usuário pressiona uma tecla sobre o elemento
onLoad	Quando o elemento foi carregado por completo
onMouseOver	Quando o usuário passa o mouse sobre o elemento
onMouseOut	Quando o usuário retira o mouse sobre o elemento
onResize	quando a exibição do documento é redimensionada
onUnLoad	Quando a página é descarregada (p/ <body>)

HTML, CSS, Javascript, Eventos

- Quando um evento ocorre, o segmento de código que é executado em resposta ao evento específico é chamado de manipulador do evento (handler do evento);
- Formato geral: <tagHTML evento = "código Javascript">

Exemplo:

```
<!DOCTYPE html>
<html>
  <head>
      <title>Usando o evento onClick no Javascript Inline</title>
  </head>
  <body>
      <h1 onclick="this.style.color='red'> Clique para testar o evento onClick! </h1>
  </body>
</html>
```

this refere-se ao objeto atual, o objeto do evento

Outros Manipuladores de Eventos

- Evento OnLoad:

```
<!DOCTYPE html>
<html>
  <head>
      <title>Usando o evento onLoad no
Javascript Inline</title>
  </head>
  <body onload="alert('Bem Vindo')"> >
      <h1> Boas Vindas ao carregar a
página.</h1>
  </body>
</html>
```

- Evento onMouseOver e Out :

```
<!DOCTYPE html>
<html>
  <head>
      <title>Usando eventos onmouseover e
onmouseout no Javascript</title>
  </head>
  <body>
      <div onmouseover =
"this.innerHTML='Obrigado'“ onmouseout =
"this.innerHTML='Passe o mouse em mim'"
style="background-color:blue; width:180px
;height:20px;">
            Passe o mouse em mim
      </div>
  </body>
</html>
```

Tutoriais e Exercícios de Fixação

- https://www.w3schools.com/js/js_variables.asp
- https://www.w3schools.com/js/js_strings.asp
- https://www.w3schools.com/js/js_numbers.asp
- https://www.w3schools.com/js/js_operators.asp
- https://www.w3schools.com/js/js_events.asp
- https://github.com/gabrieldarezzo/helpjs-ravi