

JavaScript – parte 4

Profa. Me. Andréia Rodrigues Casare

Email: casareandreia@gmail.com

Função isNaN

- ▶ A função **isNaN** tem a finalidade de testar um número para determinar se é ou não um número.
- ▶ É usada para comparar valores do tipo número ou string.
- ▶ Sintaxe:
 - `isNaN(valor);`

Exemplo:

```
<!DOCTYPE html>
<html>
  <script>
    function condicao() {
      valor=document.form1.nome.value;
      if(isNaN(valor)){
        alert("Não é um número!!!");
      }
    }
  </script>
  <form name="form1">
    Nome: <input type="text" name="nome" >
    <input type="submit" value="testar" onclick="condicao()">
  </form>
</html>
```

Função parseFloat

- ▶ É usada para converter uma string em um valor de ponto flutuante.
- ▶ É convertido uma string em um valor numérico equivalente. Se a função encontrar um caractere diferente de um sinal (+ ou -), números (0 à 9), ponto decimal ou expoente, será retornado o valor até aquele ponto e ignorado o restante. Caso o primeiro caractere não puder ser convertido para um número, parseFloat irá retornar os valores 0 para a plataforma Windows e NaN para as outras plataformas.
- ▶ Sintaxe:
 - parseFloat(string);

Exemplo:

```
<!DOCTYPE html>
<html>
  <script>
    function converter() {
      num=parseFloat(document.form1.valor.value);
      alert(num);
    }
  </script>
  <form name="form1">
    Valor: <input type="text" name="valor">
    <input type="submit" value="Converter" onclick="converter()">
  </form>
</html>
```

Função parseInt

- ▶ A função **parseInt** converte valores de string em valores numéricos equivalentes.
- ▶ É possível a conversão de números de bases como hexadecimal ou octal em valores decimais.
- ▶ Se a função encontrar um caractere diferente de um sinal (+ ou -), números (0 à 9), ponto decimal ou expoente, será retornado o valor até aquele ponto e ignorado o restante. Caso o primeiro caractere não puder ser convertido para um número, a função parseInt irá retornar o valor 0 para Windows e NaN para outros sistemas.
- ▶ Sintaxe:
 - `parseInt(string,radix);`

Exemplo:

```
<script>  
    valor=parseInt("123.456");  
    alert(valor);  
</script>
```

Função parseInt

- ▶ Através do parâmetro radix, é possível a conversão de um número de uma base para decimal, já no caso contrário isto não é possível.
- ▶ **Exemplos:**
- ▶ `valor=parseInt("ff",16); //Conversão hexadecimal, retorna 255`
- ▶ `valor=parseInt("0xff",16); //Conversão hexadecimal, retorna 255`
- ▶ `valor=parseInt("1111",2); //Conversão binário, retorna 15`
- ▶ `valor=parseInt("765",8); //Conversão octal, retorna 501`
- ▶ A omissão do parâmetro radix, a linguagem JavaScript assume que o valor definido está no formato decimal.

Função print()

- ▶ Através da função print(), o usuário poderá executar a função de imprimir evitando caminhos longos.

- ▶ **Sintaxe:**

`<p onMouseDown='self.print()>Imprimir Documento</p>`

- ▶ Neste exemplo foi usado o evento onMouseDown que avisa ao navegador para imprimir o documento atual quando o usuário clicar sobre o texto de parágrafo.

Método open

- ▶ Este método tem como objetivo abrir uma nova janela do browser. Com este método o usuário poderá abrir uma nova janela em branco ou uma janela que contém um documento específico.
- ▶ Sintaxe:
 - **NomeJanela=window.open(URL,alvo,opções);**
- ▶ NomeJanela é uma variável que será uma referência a nova janela.
- ▶ URL é o endereço da janela a ser aberta.
- ▶ Alvo é o nome da janela para ser usado no atributo target dos tag's <FORM> ou <A>.
- ▶ Opções é o parâmetro que controla o comportamento da janela.

Método open

- ▶ Exemplo:

```
<!DOCTYPE html>
<html>
  <script>
    window.open("https://www.google.com", "", "width=550,height=420,menubar=no");
  </script>
</html>
```

Método close

- ▶ O método close do objeto window tem a finalidade de fechar uma janela.
- ▶ Normalmente utiliza-se este método atribuído à um botão de ação criado com os formulários.
- ▶ Sintaxe:
 - window.close()
- ▶ Exemplo:

```
<!DOCTYPE html>
<html>
  <form>
    <input type="submit" value="Fechar" onclick="window.close()">
  </form>
</html>
```

Objeto location

- ▶ Armazena as informações referentes aos endereços de URLs.
- ▶ Propriedade **href**:
 - Armazena a url da página atual
 - Exemplo:

```
<html>  
  <script>  
    document.write(window.location.href) ;  
  </script>  
  
</html>
```

- Este exemplo mostrou o endereço atual da página

Objeto location

- ▶ Podemos também atribuir uma nova URL.
- ▶ Exemplo:

```
<!DOCTYPE html>  
<html>  
  <script>  
    window.location.href="exercicio_1.html";  
  </script>  
</html>
```

Método reload

- ▶ Recarrega/atualiza a página atual.
- ▶ Exemplo:

```
<!DOCTYPE html>
<html>
  <body>
    <a href="javascript:window.location.reload();" > Atualizar página </a>
  </body>
</html>
```

Objeto history

- ▶ Armazena o histórico das URLs visitadas.
- ▶ Método go()
 - Permite a navegação entre as páginas já visitadas.
 - Argumento com valor positivo avança para próxima página já visitada (next do navegador).
 - Argumento com valor negativo volta para página anterior (back do navegador).

```
<!DOCTYPE html>
<html>
  <body>
    <p><a href="javascript:window.history.go(1);"> Avançar </a>
    <p><a href="javascript:window.history.go(-1);"> Voltar </a>
  </body>
</html>
```


Objeto history

- ▶ Método back()
 - Retorna a página anterior (back do navegador).

```
<!DOCTYPE html>
<html>
  <body>
    <p><a href="javascript:window.history.back();" >Voltar </a>
  </body>
</html>
```

Método toLowerCase

- ▶ Converte uma string em letras minúsculas.
- ▶ Exemplo:

```
<!DOCTYPE html>
<html>
  <script>
    texto="FATEC - CURSO ADS";
    document.write(texto.toLowerCase());
  </script>
</html>
```

Método to UpperCase

- ▶ Converte uma string em letras maiúsculas.
- ▶ Exemplo:

```
<!DOCTYPE html>
<html>
<script>
    texto="fatec - curso de ads";
    document.write(texto.toUpperCase());
</script>
</html>
```

Método length

- ▶ Retorna o número de caracteres de uma string.
- ▶ Exemplo:

```
<script>  
    var frase="Fatec Itapetininga";  
    document.write(frase.length) ;  
</script>  
head</head>
```

Substring

- ▶ No Javascript podemos também trabalhar com substrings.
- ▶ O método substring retorna parte de uma string, formando uma nova string.
- ▶ Sintaxe:
 - `objeto.substring(inicio, fim+1);`
 - Onde:
 - inicio é a posição inicial da string
 - Fim é a posição final + 1

Substring

- ▶ Exemplo:

```
<script>  
    var frase="Fatec Itapetininga";  
    document.write(frase.substring(6,11));  
</script>
```

- ▶ Imprime Itape

Exercícios

1. Crie um forms com caixas de texto para informar: nome, cidade. No evento onclick do botão submit, crie uma função que verifique o tamanho do nome e da cidade, e mostre na tela usando document.write.
2. Crie outra função para este mesmo forms, que transforme em letras maiúsculas o nome e cidade;
3. Crie uma terceira função que retorne os 10 primeiros dígitos do nome;

Método split

- ▶ Divide uma string a partir de um caracter especificado, gerando várias strings que serão armazenadas em um array.
- ▶ Exemplo:
 - nome="Andreia Rodrigues Casare";
 - partes= nome.split(" ");
 - document.write(partes[0]);

Método split

```
<script>
```

```
nome="Andreia Rodrigues Casare";  
partes= nome.split(" ");  
document.write(partes[0]+"<br>");  
document.write(partes[1]+"<br>");  
document.write(partes[2]);
```

```
</script>
```

Tratadores de eventos

- ▶ Existe uma ampla variedade de eventos suportados pelo Javascript.
- ▶ Eventos são aqueles que o usuário pode gerar através do mouse, teclado.
- ▶ Os tratadores de eventos devem ser usados com moderação, já que eles podem atrapalhar a navegação e irritar os usuários.

Evento onClick

- ▶ Este evento acontece quando o usuário clica no botão esquerdo do mouse sobre algum elemento do documento.

Evento onDbIClick

- ▶ Ocorre quando o usuário dá um duplo clique sobre algum elemento do documento.
- ▶ Exemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Testando eventos no Javascript </title>
    <meta charset="UTF-8">
  </head>
  <body>
    <input type="button" value="Não clique aqui !!" onclick="alert('Teimosoooo vc clicou !!')">
    <input type="button" value="Dê um duplo clique aqui !!" onDbIClick="alert('Você clicou 2 vezes!!')">
  </body>
</html>
```

Evento onMouseDown e onMouseUp

- ▶ **onMouseDown** => o evento acontece quando o usuário pressiona o botão esquerdo do mouse
- ▶ **onMouseUp** => o evento acontece quando o usuário libera o botão do mouse que estava pressionado sobre um objeto

```
<body>  
  <input type="button" value="Botão" onMouseDown="alert('Botão do mouse pressionado !!')">  
  <input type="button" value="Botão liberado" onMouseUp="alert('Botão liberado!!')">  
</body>
```

Evento onBlur

- ▶ O evento acontece quando elementos de um formulário (text, select ou textarea) perdem o foco.

```
<script>
    function teste(){
        if (form1.campo1.value=="") {
            alert("FAVOR PREENCHER");
        }
    }
</script>
<body>
    <form name="form1">
        <p> Digite seu Nome:
        <input type="text" name="campo1" onBlur="teste()">
        <p> Digite sua Idade:
        <input type="text" name="campo2">
    </form>
</body>
```

Palavra-chave THIS

- ▶ Esta palavra-chave é utilizada para fazer referência ao objeto corrente.
- ▶ Sintaxe:
 - `this.propriedade`
- ▶ Exemplo:

Digite seu Endereço:

```
<input type="text" name="campo2" onChange="alert('testando') "  
onFocus="this.value='R. Castelo da Beira'">
```

Evento onFocus

- ▶ Com o evento onFocus o usuário poderá criar uma ação sempre que os elementos SELECT, TEXT ou TEXTAREA receberem o foco.
- ▶ Exemplo:

```
<body>
  <form name="form1">
    <p> <label> Digite seu Nome: </label>
    <input type="text" name="campo1" onBlur="teste()" onFocus="alert ('Digite seu nome completo')">
    <p><label> Digite seu Endereço: </label>
    <input type="text" name="campo2" onChange="alert('testando')" onFocus="this.value='R. Castelo da Beira'">

  </form>
</body>
```


Evento onLoad

- ▶ O evento onLoad executa alguma ação assim que o documento é carregado no browser. Este objeto é aplicado diretamente ao objeto window.
- ▶ Exemplo:
 - `<body onLoad="alert('Seja Bem Vindo')">`

Evento onSelect

- ▶ Ocorre sempre quando o usuário seleciona um texto dos elementos de formulário text ou textarea.
- ▶ Exemplo:

```
<body >  
  <form name="form1">  
    Digite seu Nome:  
    <input type="text" name="campo1" onSelect="alert('O usuário selecionou '+this.value) ">  
  </form>  
</body>
```

Evento onSubmit

- ▶ O evento onSubmit ocorre sempre que o usuário envia o formulário. Com este evento o usuário poderá determinar ou não o envio do formulário.
- ▶ Com este evento o usuário poderá verificar a validação de dados, como por exemplo se todos os campos do formulário estão preenchidos.

Evento onSubmit

```
        function teste() {
            if (form1.nome.value=="") {
                alert("FAVOR PREENCHER NOME");
                return false;
            }
            if (form1.idade.value=="") {
                alert("FAVOR PREENCHER IDADE");
                return false;
            }
        }
    }
</script>
</head>
<body>
    <form name="form1" onSubmit="teste()">
        <p><label> Digite seu Nome: </label>
            <input type="text" name="nome">
        <p><label> Digite sua Idade: </label>
            <input type="text" name="idade">
        <p><input type="submit" value="Enviar">
    </form>
</body>
```

Exercícios

1. Crie um formulário com os seguintes dados:
 - a) Nome
 - b) Endereco
 - c) Cidade
 - d) Telefone
 - e) Data de Nascimento

Todos esses campos são obrigatórios e devem ser validados usando o evento onBlur.

Exercícios

2. Crie um formulário com botões conforme figura abaixo, ao clicar no botão, deve ser aberto uma nova página.

Dica: use o evento onclick para cada botão e use o método window.open



Exercícios

3. Crie um botão em html, que ao ser clicado aparece um número aleatório entre 0 e 50.
4. Crie uma caixa de texto e um botão em html que chama código em Javascript que dependendo do valor introduzido mostra uma mensagem diferente.

Exemplo:

- a. Entre 0 e 10 – mostra “Insuficiente”
- b. Entre 11 e 20 – mostra “Bom”
- c. Maior que 20, mostra “Muito Bom”