

Os exemplos de código usados neste material
foram baseados no material da W3C

Sistemas Web I

Aula 3

JavaScript

Tiago Cruz de França

tcruz.franca@gmail.com

Entendendo como Interagir com HTML

@ Mudando o conteúdo HTML

- @ Utiliza o método “getElementById” para encontrar o elemento do HTML com id “teste”

```
document.getElementById("teste").innerHTML = "Web I UFRRJ";
```

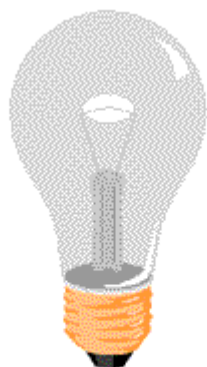
- @ Ao achar o elemento, muda o seu conteúdo escrevendo “Web I UFRRJ” na página

Entendendo como Interagir com HTML

@ Mudando atributos HTML

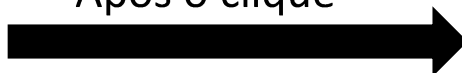
@ Clicar em uma imagem e mudá-la no evento do clique

► Ver código no próximo slide



Click the light bulb to turn on/off the light

Após o clique



Click the light bulb to turn on/off the light

```
<!DOCTYPE html>
<html>
<body>

<h1>JavaScript Can Change Images</h1>



<p>Click the light bulb to turn on/off the light.</p>

<script>
function changeImage() {
    var image = document.getElementById('myImage');
    if (image.src.match("bulbon")) {
        image.src = "pic_bulboff.gif";
    } else {
        image.src = "pic_bulbon.gif";
    }
}
</script>

</body>
</html>
```

Fonte: W3C

Entendendo como Interagir com HTML

@ Mudando o estilo (CSS) do HTML

@ Semelhante a mudança de um atributo HTML

```
document.getElementById("demo").style.fontSize = "25px";
```

```
<!DOCTYPE html>
<html>
<body>
<p id="atributo">mudando estilo do elemento HTML com JavaScript.</p>
<script>
function mudaEstilo() {
    var x = document.getElementById("atributo");
    x.style.fontSize = "25px";
    x.style.color = "red";
}
</script>
<button type="button" onclick="mudaEstilo ()">Clique Aqui</button>
</body>
</html>
```



Teste

Baseado em
conteúdo da W3C

Entendendo como Interagir com HTML

```
<!DOCTYPE html>
<html>
<body>
<input id="numb" type="number">
<button type="button" onclick="myFunction()">Submit</button>
<p id="demo"></p>
<script>
function myFunction() {
  var x, text;
  // Get the value of the input field with id="numb"
  x = document.getElementById("numb").value;
  // If x is Not a Number or less than one or greater than 10
  if (isNaN(x) || x < 1 || x > 10) {
    text = "Input not valid";
  } else {
    text = "Input OK";
  }
  document.getElementById("demo").innerHTML = text;
}
</script>
</body>
</html>
```

@ Validando dados com JavaScript

Teste

Input OK

Input not valid

Saídas JavaScript

🕒 Opções para apresentar conteúdo no documento

🕒 `window.alert`

▶ Conteúdo apresentado em uma caixa de alerta

🕒 `document.write()`

▶ Escreve direto na página HTML

🕒 `innerHTML`

▶ Escreve dentro de um elemento HTML

🕒 `console.log()`*

▶ Escreve no console do navegador

Saídas JavaScript

@ Escrevendo no console do navegador

@ Ative o console pressionando F12 e escolha “console”

```
<!DOCTYPE html>
<html>
<body>

<script>
console.log(7 + 8);
</script>

</body>
</html>
```

Baseado no conteúdo W3C

Objetos JavaScript

```
var aula = {professor:"Tiago França",  
            qtdeAlunos:50  
            tempoAula : function(c) { return 50 * 2; }  
};
```

- ▶ Pode quebrar linha a cada propriedade
- ⌚ Acessando as propriedades dos objetos
 - ▶ Ex: `aula.professor` ou `aula[professor]`
- ⌚ É possível definir e usar funções nos objetos
 - ▶ Acessando: `aula.tempoAula()`
 - Retorno: 100

Eventos HTML

Ⓢ Existe alguns eventos que podem ser utilizados para disparar ações

Ⓢ São declarados como atributos dos elementos HTML

▶ Principais eventos HTML

- onchange, onclick, onmouseover, onmouseout, onkeydown, onload, etc.

▶ Lista de eventos

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

Formulário

```
<!DOCTYPE html>
<html>
<head>
<script>
function validateForm() {
    var x = document.forms["myForm"]["fname"].value;
    if (x == null || x == "") {
        alert("Name must be filled out");
        return false;
    }
}
</script>
</head>
<body>
<form name="myForm" onsubmit="return validateForm()" method="post">
Name: <input type="text" name="fname">
<input type="submit" value="Submit">
</form>
</body>
</html>
```

Validações de Formulário

@ Verificações comuns

- @ Todos os campos foram validados?
- @ Os valores com os quais os campos foram preenchidos são válidos?
- @ Os tipos de dados dos campos estão corretos?
 - ▶ Ex: existe texto onde deveria ter número?

@ Objetivo

- @ Verificar se os dados estão corretos
 - ▶ No cliente: realizado no navegador antes de enviá-los ao servidor
 - ▶ No servidor: realizado antes de inserir/alterar/consultar o banco

API de Validação JavaScript

Validações JavaScript

@ Métodos DOM de validação de restrição

Propriedade	Descrição
checkValidity()	Retorna “true” se o conteúdo do elemento de entrada for um dado válido
setCustomValidity()	Configura a mensagem de validação de uma elemento

@ Propriedades DOM de validação de restrição

Propriedade	Descrição
validity	Propriedade booleana relacionadas a validação de uma elemento de entrada
validationMessage	Mensagem exibida quando a validade é falsa
willValidate	Indica se um elemento será validado

Validações JavaScript

- 🕒 The validity property of an input element contains a number of properties related to the validity of data

Property	Description
customError	Set to true, if a custom validity message is set.
patternMismatch	Set to true, if an element's value does not match its pattern attribute.
rangeOverflow	Set to true, if an element's value is greater than its max attribute.
rangeUnderflow	Set to true, if an element's value is less than its min attribute.
stepMismatch	Set to true, if an element's value is invalid per its step attribute.
tooLong	Set to true, if an element's value exceeds its maxLength attribute.
typeMismatch	Set to true, if an element's value is invalid per its type attribute.
valueMissing	Set to true, if an element (with a required attribute) has no value.
valid	Set to true, if an element's value is valid.

The checkValidity() Method

```
<!DOCTYPE html>
<html>
<body>
<p>Enter a number and click OK:</p>
<input id="id1" type="number" min="100" max="300">
<button onclick="myFunction()">OK</button>
<p>If the number is less than 100 or greater than 300, an error message will be
displayed.</p>
<p id="demo"></p>
<script>
function myFunction() {
  var inpObj = document.getElementById("id1");
  if (inpObj.checkValidity() == false) {
    document.getElementById("demo").innerHTML = inpObj.validationMessage;
  } else {
    document.getElementById("demo").innerHTML = "Input OK";
  }
}
</script>
</body>
</html>
```


[...]

```
<p>Enter a number and click OK:</p>
```

```
<input id="id1" type="number" max="100">
```

```
<button onclick="myFunction()">OK</button>
```

```
<p>If the number is greater than 100 (the input's max attribute), an error message will be displayed.</p>
```

```
<p id="demo"></p>
```

```
<script>
```

```
function myFunction() {
```

```
    var txt = "";
```

```
    if (document.getElementById("id1").validity.rangeOverflow) {
```

```
        txt = "Value too large";
```

```
    } else {
```

```
        txt = "Input OK";
```

```
    }
```

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

```
}
```

```
</script>
```

[...]

```
<script>
function myFunction() {
  var txt = "";
  if (document.getElementById("id1").validity.rangeUnderflow) {
    txt = "Value too small";
  } else {
    txt = "Input OK";
  }
  document.getElementById("demo").innerHTML = txt;
}
```