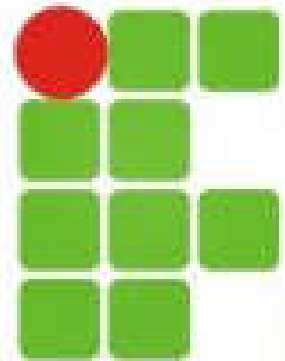
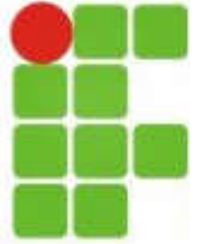


Fundamentos de Web Design 2

Professor Eng. Dr. Will Roger Pereira





Conteúdo

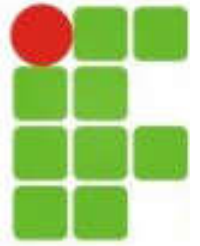
- Manipular os elementos do HTML DOM.
- Manipular código HTML de um elemento HTML;
- Manipular valores;
- Manipular atributos;
- Adicionar novo conteúdo HTML;
- Remover conteúdo HTML;
- Manipular CSS.



Introdução

- Com Javascript, é possível manipular diversos aspectos da página HTML, de modo a torná-la realmente dinâmica;
- Os atributos e métodos ensinados a seguir irão permitir ao desenvolvedor realizar diversas alterações, bem como adicionar e remover conteúdo da aplicação Web;
- Os seletores são primordiais para saber quais elementos serão manipulados, assim como os eventos, que servirão para trazer uma nova experiência de interação com os usuários, inexistente anteriormente.

Manipular HTML de um elemento

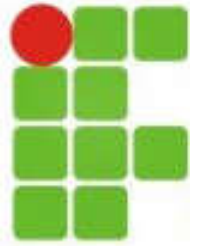


- Para realizar tal manipulação, selecione o elemento e utilize o atributo `innerHTML`;

`elemento.innerHTML`

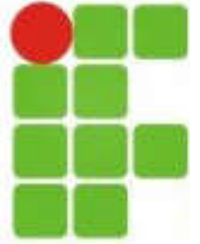
- Sua referência conterá o texto disposto dentro do elemento;
- Sua alteração substitui o texto disposto dentro do elemento, e é interpretado como HTML;
- Exemplos em **1-html.html**.

Manipular atributo HTML de um elemento



- Para manipular o atributo de um elemento, basta saber seu nome e sua função, dependendo do tipo do elemento;
- Atributos que não foram inicializados ou não existem possuem valor *undefined*;

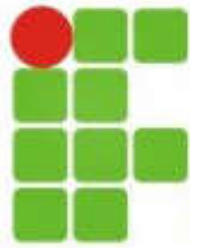
elemento.atributo



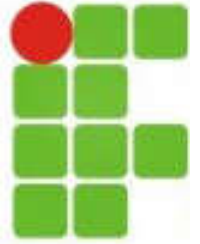
Exemplos de Atributos HTML

- **value:** Valor de elementos de entrada, e. g., input, select;
- **src:** Fonte de informação para elementos de imagem;
- **href:** Destino de elementos âncora;
- **className:** Classe do elemento;
- **checked:** Para input do tipo checkbox. Retorna **true** se estiver marcado;
- Todos foram aprendidos em FWD1;
- Exemplos em **2-attr.html**.

Manipulação de conteúdo HTML



- Esta funcionalidade permite a adição e remoção de conteúdo HTML de forma dinâmica em sua aplicação;
- Pode-se compor a página HTML conforme a aplicação é executada e os eventos vão disparando;
- A mudança só acontece na memória do navegador. Nenhuma mudança é refletida no código.



Criando elemento HTML

- Antes de ser possível adicionar um elemento HTML na página, é necessário criá-lo;

```
document.createElement(tag)
```

- Onde tag é o texto da tag, e. g., “li”, “p”, “table”. Retorna uma referência para o elemento criado
- O elemento é criado vazio. É necessário editar seus atributos, como texto interno, classe, valor, etc.



Criando elemento HTML

- Exemplo:

```
var node = document.createElement("li");
```

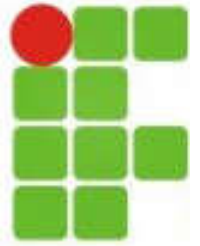
- Cria um elemento `` e o armazena na variável `node`;
- A referência será utilizada para adicionar este elemento à página posteriormente.

Adicionando elemento HTML criado anteriormente



- É possível adicionar um elemento HTML à página de duas maneiras diferentes:
 - Ao final, como elemento filho;
 - Antes de um elemento, no mesmo nível.
- Em qualquer situação, necessitar-se-á de:
 - Referência do elemento criado, a ser adicionado;
 - Elemento base para a adição do novo elemento previamente criado com “createElement”.

Adicionando elemento como filho, ao final

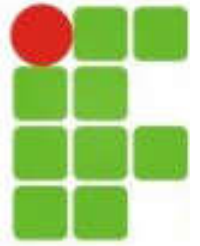


- Seja **elemento** o elemento já existente, base para a inserção, e **node** o elemento criado, a ser adicionado:

```
elemento.appendChild(node)
```

- O método **appendChild** irá inserir **node**, como filho de **elemento**, ao final.

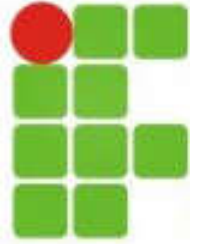
Adicionando elemento no mesmo nível, anteriormente



- Seja **elemento** o elemento já existente, base para a inserção, **container** onde **elemento** está contido, e **node** o elemento criado, a ser adicionado:

```
container.insertBefore(elemento, node)
```

- O método **insertBefore** irá inserir **node** imediatamente antes de **elemento**, dentro de **container**, no mesmo nível.



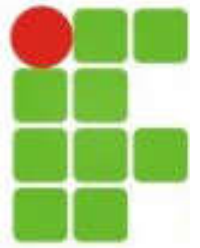
Removendo elemento HTML

- Seja **elemento** o elemento já existente para remoção:

```
elemento.remove()
```

- O método **remove** irá remover **elemento** e tudo que estiver dentro do mesmo;
- Exemplos em **3-add-remove.html**.

Manipulação do estilo de um elemento HTML



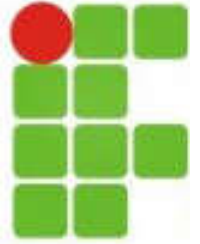
- Esta funcionalidade permite manipular valores de todos os atributos CSS, e. g., cor, cor de fundo, etc;
- Para manipular o atributo CSS de um elemento, basta saber seu nome e sua função, dependendo do tipo do elemento;

`elemento.style.atributoCSS`

- Todos os atributos CSS estão dentro do atributo style dos elementos.

Propriedade - CSS	Referência - JavaScript
background	background
background-attachment	backgroundAttachment
background-color	backgroundColor
background-image	backgroundImage
background-position	backgroundPosition
background-repeat	backgroundRepeat
border	border
border-bottom	borderBottom
border-bottom-color	borderBottomColor
border-bottom-style	borderBottomStyle
border-bottom-width	borderBottomWidth
border-color	borderColor
border-left	borderLeft
border-left-color	borderLeftColor
border-left-style	borderLeftStyle
border-left-width	borderLeftWidth
border-right	borderRight
border-right-color	borderRightColor
border-right-style	borderRightStyle
border-right-width	borderRightWidth
border-style	borderStyle
border-top	borderTop
border-top-color	borderTopColor
border-top-style	borderTopStyle
border-top-width	borderTopWidth
border-width	borderWidth
clear	clear
clip	clip
color	color
cursor	cursor
display	display
filter	filter
font	font

font-family	fontFamily
font-size	fontSize
font-variant	fontVariant
font-weight	fontWeight
height	height
left	left
letter-spacing	letterSpacing
line-height	lineHeight
list-style	listStyle
list-style-image	listStyleImage
list-style-position	listStylePosition
list-style-type	listStyleType
margin	margin
margin-bottom	marginBottom
margin-left	marginLeft
margin-right	marginRight
margin-top	marginTop
overflow	overflow
padding	padding
padding-bottom	paddingBottom
padding-left	paddingLeft
padding-right	paddingRight
padding-top	paddingTop
page-break-after	pageBreakAfter
page-break-before	pageBreakBefore
position	position
float	styleFloat
text-align	textAlign
text-decoration	textDecoration
text-decoration: blink	textDecorationBlink
text-decoration: line-through	textDecorationLineThrough
text-decoration: none	textDecorationNone
text-decoration: overline	textDecorationOverline
text-decoration: underline	textDecorationUnderline
text-indent	textIndent
text-transform	textTransform
top	top
vertical-align	verticalAlign
visibility	visibility
width	width
z-index	zIndex



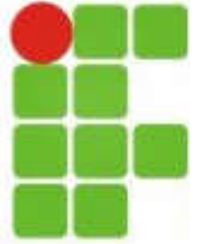
Exemplos de Atributos CSS

- **color:** Cor da letra;
 - **background:** Cor de fundo;
 - **fontSize:** Tamanho da fonte (letra);
 - **opacity:** Nível de transparência;
 - **border:** Características de borda do elemento.
-
- Todos foram aprendidos em FWD1.
 - Referência:
https://www.w3schools.com/jsref/dom_obj_style.asp



Atributos customizados

- Esta funcionalidade permite manipular atributos customizados, ou seja, criados pelo desenvolvedor;
- É uma forma interessante de guardar valores nos elementos HTML para a tomada de decisões;
- A manipulação de atributos HTML é livre em Javascript, e **requer o uso de nomes já não existentes**. Tente nomes em português.



Atributos customizados

- Qualquer atributo poderá ser manipulado através disto, uma vez que o controle dos atributos e de seus valores é de responsabilidade do desenvolvedor;
- Para um atributo ser utilizado, um valor deve ser atribuído a ele inicialmente, uma vez que atributos não inicializados possuem valor **undefined**;
- Caso o atributo não seja inicializado, seu valor é **undefined**.



Atributos customizados

- Sintaxe:

elemento.nomeatributo

- Exemplo:

elemento.ligado = true

- Exemplos em **4-custom.html**.