

Módulo 02 - Javascript

Aula 08 – Manipulação básica de tags

- Clonando objetos
- Propriedade Style
- Propriedade innerText / innerHTML
- Monitorando Eventos via Tag
- Monitorando Eventos via Script

Clonando Objetos

Comportamento de Objetos em memória

- As variáveis em JavaScript possuem a habilidade de armazenar valores dentro de si, gravando assim uma quantidade de dados X em um espaço da memória.
- Quando se trata de Objetos, o que é armazenado dentro da variável não é o objeto em si, mas sim seu endereçamento de memória (vale lembrar que Arrays e Dicionário também são objetos).
- Isso faz com que criar cópias de objetos seja um pouco mais complexo do que copiar dados primitivos.

Copiando dados primitivos

- No script “ let x = 10 ; let y = x ” o valor de y é uma cópia direta do valor armazenado em x, por ser uma cópia eles podem ser alterados de forma independente :

```
let x = 10
```

```
let y = x
```

```
y += 2
```

```
x » 10
```

```
y » 12
```

Clonando Objetos

Copiando objetos

- Por x armazenar um **endereço de memória** que aponta para um objeto, quando copiamos esse endereço para y e alteramos ele, ambas a variáveis retornam o mesmo resultado modificado

```
let x = [ 1, 2, 3]
```

```
let y = x
```

```
y.push(4)
```

```
x » [1,2,3,4] // y » [1,2,3,4]
```

Clonando Objetos

Copiando objetos

- Dado a esse comportamento, se quisermos que Y seja uma cópia dos dados de X mas sem modificar o valor original, devemos usar outros métodos para realizar essa clonagem (como por exemplo, atribuindo a Y cada posição de x)

```
for( var i of x) {  
    y.push( i )  
}
```

Clonando Objetos

- Tendo esse conceito em mente, fica mais claro o conceito de manipular os objetos da arvore DOM :

```
let a = document.getElementById("titulo")
```

```
let b = document.getElementById("titulo")
```

- Toda alteração feita em A, será feita em B pois ambos retornam o mesmo endereço de alocação na memória

Atributos do Objeto Style

Objeto Style

- Agora que já sabemos como acessar uma Tag usando os métodos do objeto document, precisamos falar um pouco sobre o objeto style (filho da tag, seus atributos são filhos da tag – filhos de style).
- Quando se trata de estilização de tags, esse é o objeto que concentra todas as características HTML de uma Tag, e esse objeto possui uma infinidade de atributos os quais veremos alguns exemplos.
- https://www.w3schools.com/jsref/dom_obj_style.asp

Atributos do Objeto Style

Objeto Style – Principais Atributos :

- `tag.style.backgroundColor` // Muda cor de fundo
- `tag.style.color` // Muda cor do texto
- `tag.style.height` // Altura da Tag
- `tag.style.border` // Config básica de borda
- `tag.style.borderRadius` // Config básica de borda radial
- `tag.style.textAlign` // Alinhamento de texto
- `tag.style.padding` // Espaçamento interno
- `tag.style.margin` // Espaçamento externo

Propriedade innerText

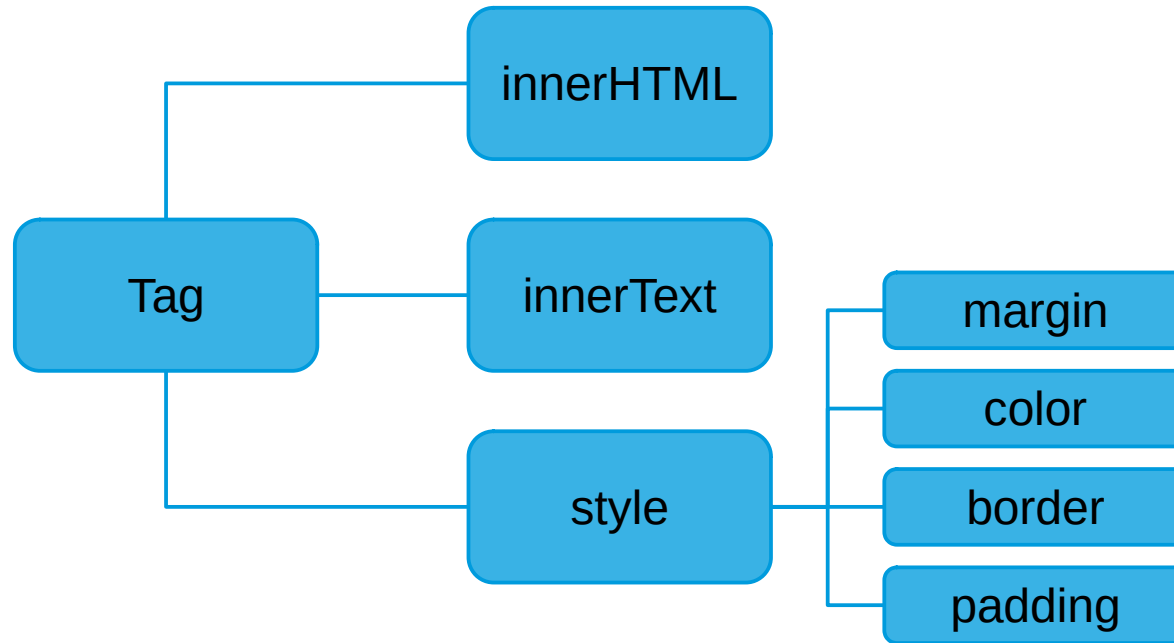
- Tão importante quanto o atributo Style(que leva consigo todas as configurações de estilo CSS) também existe o atributo innerText, que ao invés de ser uma coleção de atributos é um único valor de string que representa o conteúdo dentro da tag, e podemos passar valores para dentro desse atributo de diversas maneiras.

Propriedade innerText

Propriedade innerText - Passando como valor :

- String Pura
- Concatenação (com citação de variáveis)
- Template-String (crases e a marcação especial \$ { })
- Operadores de incremento
- Não é Possível passar Tags dentro da string e esperar que o js reconheça, para isso é necessário o uso do innerHTML

Arquitetura de uma tag



O que são os eventos ?

- Existem duas principais formas de amarrar um evento a uma Tag, a mais primitiva (que é configurando os atributos de uma tag dentro do HTML e dizendo qual função essa tag deve chamar quando acontecer evento X atrelado a ela) e a forma mais adequada (configurando via script, que é a forma majoritária como iremos lidar com eventos dentro do curso).

Inserindo um monitor de Eventos dentro de uma tag

- Assim como quaisquer características relacionadas ao estilo de uma tag, esses valores podem ser configurados dentro da tag, ou em um documento separado de CSS (lembrando que existem algumas divergências na nomenclatura de alguns atributos entre o HTML e o CSS).
- O mesmo acontece para a relação evento função, podemos adicionar na tag um atributo com o nome de um evento e seu valor ser a função de javascript.

```
<div onclick="minha_funcao()"> </div>
```

Inserindo um monitor de Eventos dentro do script

- A forma mais adequada de amarrar um evento associado a uma tag a uma função javascript é através do método `addEventListener` (esse método é comum a quase todas as tags), onde passamos 2 argumentos, o primeiro sendo uma string do evento que queremos monitorar e o segundo sendo o nome da função sem os parênteses.

```
let my_tag = document.getElementById("my_tag")
```

```
my_tag.addEventListener("click", nomedafuncao)
```

Nomenclatura de eventos :

- Diferente dos conflitos de nomenclatura que os atributos possuem entre o HTML e o CSS (onde não existe um padrão), a nomenclatura de eventos entre HTML e JS é bem simples, basta cortar o “on” no início do nome do evento e você terá a conversão de todos os eventos HTML para o JS.
- Segue a lista de eventos da documentação do HTML :
- https://www.w3schools.com/tags/ref_eventattributes.asp

Arquitetura de uma tag

