

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas  
Prof. Felipe Scheidt – IFPR – Campus Foz do Iguaçu

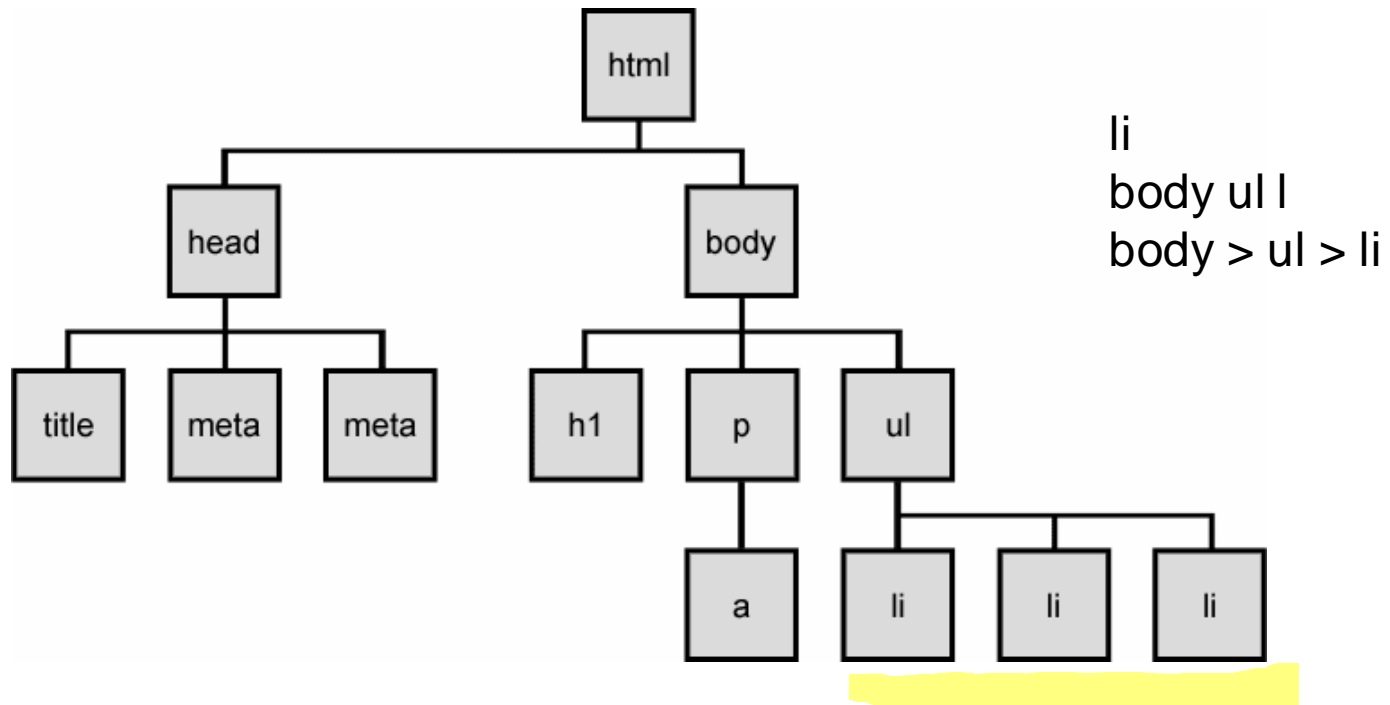
# DESENVOLVIMENTO WEB I

**JS**

Javascript: Eventos e funções

# DOM – Document object model

Páginas HTML são representadas através de uma árvore chamada DOM. Através do DOM pode-se manipular os elementos contidos no HTML.



# Função getElementById()

Esta função permite obter uma referência a determinado elemento/tag que possua um id.

```
function calcula(){  
    var div = document.getElementById("total");  
    var input = document.getElementById("nome");  
}
```

Para isso é preciso definir o atributo id:

```
<input type="text" id="nome"/>  
<div id="total"></div>
```

# Propriedade innerHTML

Permite alterar o conteúdo de um elemento HTML.

```
function alteraConteudo(){  
    var div = document.getElementById("total");  
    div.innerHTML = "novo conteúdo";  
}
```

Para a tag <input> usar a propriedade value

# Propriedade display

A propriedade display permite ocultar ou exibir um elemento.

```
// função recebe uma tag e esconde a mesma  
function ocultar(tag){  
    tag.style.display = "none";  
}  
// função recebe uma tag e exhibe a mesma  
function exibir(tag){  
    tag.style.display = "block";  
}
```

↳ altera também outras propriedades CSS

# Eventos

---

- Um evento pode ser descrito como "algo" que acontece a um elemento HTML. Nesses casos é possível programar o javascript para responder de acordo com um conjunto de ações.
- Geralmente estamos interessados em eventos disparados pelo usuário, oferecendo assim mais interatividade ao site.
- Que tipos de eventos podemos tratar no javascript?

# Exemplos de Eventos

Exemplos de eventos produzidos pelo usuário:

- Clique do **mouse**
- Quando o mouse passa por um elemento.
- Quando o **valor** de um elemento é alterado
- Uma tecla é pressionada
- Quando um formulário é enviado ao servidor.

Evento não relacionado ao usuário:

- Quando a página termina de carregar
- Quando uma resposta retorna do servidor

# Eventos de mouse

## ■ Eventos associados a uma ação do mouse:

Função Javascript	Descrição
onclick	quando um clique é detectado
onmouseover	quando o ponteiro do mouse passa em cima de um elemento
onmouseout	quando o mouse sai de cima de um elemento
onmousedown	Aperta o botão do mouse
onmouseup	Solta o botão do mouse



# Exemplo onclick

Quando clicar num botão mostrar uma caixa de alert.

```
<input onclick="alert('ok')" type="button" value="clique aqui"/>
```

Código javascript



# Exemplo onmouseover()

Altera a cor de fundo da body para vermelho, quando o mouse passa por cima de uma div:

```
<div onmouseover="document.bgColor='red';">  
    VERMELHO  
</div>
```

# Eventos de teclado

Eventos associados a uma ação do teclado:

Função Javascript	Descrição
onkeypress()	usuário aperta uma tecla
onkeydown()	quando o usuário está pressionando uma tecla
onkeyup()	quando o usuário solta a tecla

# Exemplo de onkeypress()

exibe o valor digitado numa caixa alert.

```
<input type="input" onkeypress="alert(this.value);"/>
```

# Evento de mudança

---

- **onfocus()** => elemento ganha o foco
- **onblur()** => elemento perde o foco
- **onchange()** => elemento tem seu valor alterado.

# Definição de funções

Encapsular o código em funções é essencial para *reaproveitamento* de código e organização. No javascript uma função é definida:

```
function bemVindo(){  
    alert("Bom dia! Seja bem vindo.");  
}
```

# Chamando a função

Para chamar a função, podemos associar um evento qualquer e passar o nome da função que será executada. Por exemplo, para chamar a função bemVindo

```
<input type="button" value="Clique aqui" onClick="bemVindo()"/>
```

# Funções de seleção

Javascript fornece diversas funções para seleção de elementos:

ATRIBUTO	NOME DA FUNÇÃO
id	getElementById()
classe	getElementsByClassName()
tipo da tag	getElementsByTagName()
nome do elemento	getElementByName()
selector	querySelector()



# getElementsByTagName

Retorna um vetor de elementos de acordo com o nome da tag (atributo name).

```
var inputs = document.getElementsByTagName("idade");  
// percorrendo elementos cujo nome é idade  
for(i=0;i<inputs.length;i++){  
    console.log(inputs[i]);  
}
```

# getElementsByClassName

retorna uma lista de elementos de acordo com o nome da classe

```
var inputs = document.getElementsByClassName("required");  
// percorrendo elementos cuja classe é required  
for(i=0;i<inputs.length;i++){  
    console.log(inputs[i]);  
}
```

# getElementsByTagName

retorna uma *lista* de elementos de acordo com o nome da tag.

```
var inputs = document.getElementsByTagName("input");  
// percorrendo as inputs  
for(i=0;i<inputs.length;i++){  
    console.log(inputs[i]);  
}
```

# getElementById()

Função que obtém o elemento a partir do id.  
Seja o código html:

```
<div id="mensagem"></div>
```

Para obter este elemento no JS:

```
document.getElementById( "mensagem" );
```

# querySelector

É uma função mais genérica que permite selecionar elementos usando **seletores** semelhantes ao que usamos nas regras do CSS. Exemplos:

```
var el = document.querySelector("#mensagem")  
var el = document.querySelector(".header h1")  
var el = document.querySelector("div.card span.red")
```

# Acessando atributos

Atributos agregam informações sobre os elementos e muitas vezes precisamos acessá-los para verificar seus valores. Exemplos de atributos:

- id
- class
- style
- name
- type

# Ler atributos

Para obter o valor de um atributo, usamos a função `getAttribute()`, passando como parâmetro o nome do atributo desejado:

```
var div = document.getElementById("total");  
// obtem a classe da div:  
var classe = div.getAttribute("class");
```

# Alterar valor do atributo

Para definir um novo atributo usamos a função `setAttribute()` passando como parâmetro o nome do atributo e qual o seu valor:

```
var input = document.getElementById("total");  
div.setAttribute("value", "200");
```



# Formulários com javascript

Usando eventos no formulário podemos tornar o resultado mais dinâmico. Por exemplo:

## Orçamento

Item:	<input type="text" value="Leite"/>	valor:	<input type="text" value="3.00"/>	quantidade:	<input type="text" value="4"/>	=	<div>R\$ 12.00</div>
Item:	<input type="text" value="Sabão"/>	valor:	<input type="text" value="2.00"/>	quantidade:	<input type="text" value="10"/>	=	<div>R\$ 20.00</div>
Item:	<input type="text" value="Iogurte"/>	valor:	<input type="text" value="4.50"/>	quantidade:	<input type="text" value="10"/>	=	<div>R\$ 45.00</div>

calcular

Valor total: R\$ 77.00

# Ideia de Validação

Para implementar um validação de formulário, precisamos definir antes:

- quais campos são obrigatórios
- como notificar o usuário sobre o erro
- impedir que o formulário seja enviado ao servidor caso contenha erros

# Atividade

---

- 1) Defina 3 divs no HTML e faça uma função JS que localiza todas as divs e muda a cor de fundo para laranja (orange). O gatilho dessa ação ocorre no click de um button

# Exercícios

1. Faça um script que detecta quando o mouse clica num input tipo texto, exibindo uma mensagem "você clicou num input".
2. Faça um script que possui 1 imagem, que quando o usuário usuário clica na mesma, sua largura aumenta para 300px;
  1. Conforme o script anterior, faça agora um script que utiliza onmouseover e onmouseout. Quando o mouse passa em cima da imagem, coloca tamanho em 400px; quando mouse sai da imagem, volta tamanho original.
3. Faça uma função que obtém a cor selecionada de uma <select> e altera a cor de fundo da body usando esta cor.
4. Faça um script que detecta quando um input perde o foco, e automaticamente calcula o juros do empréstimo com base no valor digitado no input. Use juros de 10%.
5. Faça uma função que calcula a área de um quadrado. Leia o valor de um input e exiba o resultado numa div.

# Exercícios

---

6. Defina 3 botões (azul, laranja e verde), que quando clicados alterar a cor de fundo da página para as cores citadas. Faça uma única função para os 3 botões.

# Atividade

- 2) Faça um script, que possui um vetor de imagens, e toda vez que o usuário clica no botão próximo, uma nova imagem é carregada. Dica: use a propriedade `src` para modificar o caminho da imagem. O vetor possui 5 imagens.

próximo



# Atividade

4. Modifique o exercício 2, e adicione um botão voltar, e um contador (em vermelho) que marca o número do carro (0 até 4).

voltar <<

>> avançar

0



# Atividade

3. Faça a validação do formulário abaixo. Os três campos são obrigatórios:

**Preencha o formulário:**

Nome:	<input type="text"/>
CPF:	<input type="text"/>
Email:	<input type="text"/>
<input type="button" value="enviar"/>	