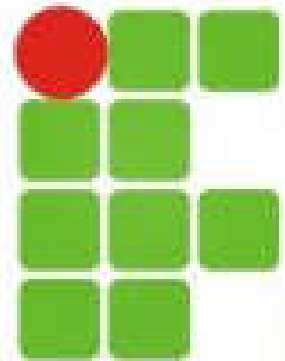


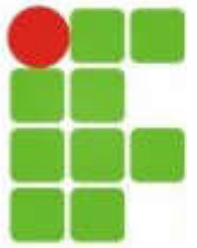
Fundamentos de Web Design 2

Professor Eng. Dr. Will Roger Pereira



Conteúdo

- Manipulação e verificação de formulários.





Introdução

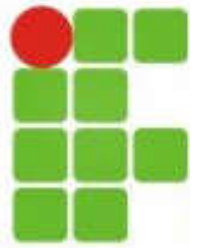
- Formulários são elementos importantes para submeter um conjunto de informações em uma aplicação Web;
- Um formulário é composto de outros elementos de entrada (input), sendo um do tipo “submit”:
 - Campos de texto: *text*;
 - Senha: *password*;
 - Opções: *radio*, *select*;
 - Datas;
 - Números;
 - Etc.



Introdução

- Quando o usuário submete tal formulário, é possível capturar o momento que este evento ocorre, com o manipulador **onsubmit**;
- O objetivo é mandarmos estes dados para o servidor. Porém, mandar formulários vazios ou sem nenhuma validação resulta em um excesso de requisições inválidas;
- Assim, para filtrar as informações que vão para o servidor, é realizada uma verificação no próprio navegador, com Javascript.

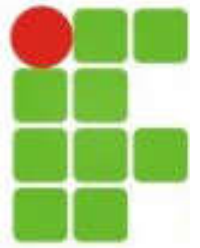
Identificando os elementos de um formulário



- **id:** Identifica, com exclusividade, qualquer elemento em uma página;
- **name:** Identifica, com exclusividade, qualquer elemento em um formulário;

```
<input type="text" id="idade" name="idade" />
```

Identificando os elementos de um formulário



- É sabido que todos os elementos de um formulário a serem enviados devem possuir o atributo **name**, para que seus dados sejam enviados em requisições HTTP;
- O atributo **name** será o nome do parâmetro passado na requisição;
- Isto será mantido. **Porém, inclusive o formulário deve ter o atributo name para fazermos facilmente sua manipulação.**



Selecionando um formulário

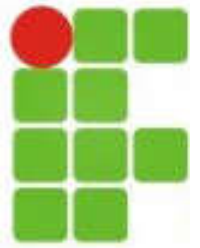
- Para seleção do formulário e seus elementos, o Javascript possui um seletor especial: **document.forms**;
- Este seletor contém um vetor todos os formulários dispostos na página;
- A seleção neste vetor, no entanto, ocorre pelo atributo **name**:

`<form name="form1">`



`document.forms["form1"]`

Selecionando os elementos de um formulário



- Quando o formulário é selecionado, nele há um vetor com todos os elementos em seu interior;
- A seleção dos elementos também ocorre pelo atributo **name**:

```
<form name="form1">
```

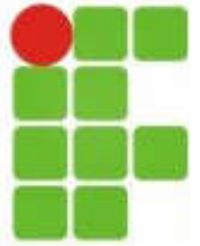
```
  <input type="text" name="idade" />
```



```
document.forms["form1"]["idade"]
```

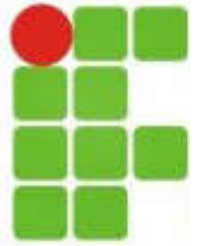
- Exemplo em **1-seleção.html**

Validação dos dados de um formulário



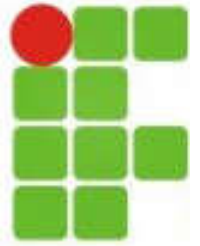
- A verificação de dados submetidos à um servidor deve ser realizada em duas frentes:
 1. No cliente Web, antes dos dados serem enviados;
 2. No servidor Web, após os dados serem enviados.
- **Deve ser feita sempre nas duas frentes:**
 - Evita que requisições com dados fora do formato sejam realizadas;
 - O código do navegador está disponível para o cliente;
 - As regras de negócio se encontram no servidor.

Validação dos dados de um formulário



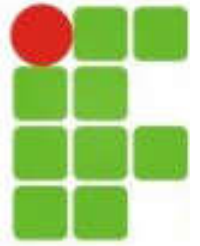
- O propósito da verificação dos dados é saber se o usuário inseriu os dados corretamente;
- Tarefas típicas de validação:
 - Preenchimento de todos os campos obrigatórios;
 - Verificação da forma e do sentido dos dados.
- Exemplo – Validação de um campo com data:
 - Verificação se há três números separados por “/”;
 - Verificação se o mês está entre 1 e 12, se o dia está entre 1 e 30/31 (depende do mês), e se o ano é adequado.

Atributos HTML5 para validação



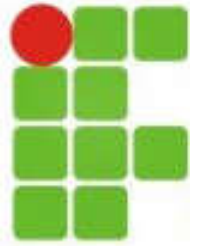
- **type:**
 - Especifica o tipo de dado que deve estar contido no campo;
 - E. g.: Text, password, number, range, date, radio, checkbox, etc.
- **required:**
 - Especifica que o campo deve ser obrigatoriamente preenchido;
 - Funciona com text, search, url, tel, email, password, date pickers, number, checkbox, radio, e file.

Atributos HTML5 para validação



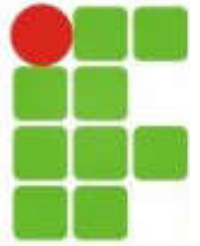
- **min e max:**
 - Especifica os valores mínimo e máximo do campo;
 - Funciona com number, range, date, datetime-local, month, time e week.
- **step:**
 - Especifica o intervalo numérico válido do campo;
 - Funciona com number, range, date, datetime-local, month, time e week.

Atributos HTML5 para verificação



- **minlength e maxlength:**
 - Especifica as quantidades mínima e máxima de caracteres em um campo;
 - Geralmente utilizado para text.
- **pattern:**
 - Especifica a **expressão regular** a ser utilizada para validar o campo;
 - A expressão regular é a maneira mais poderosa de verificar qualquer característica em um dado.

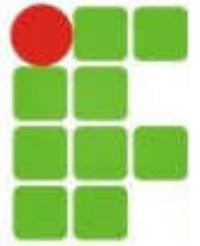
Validando os dados em um formulário HTML



- Em um formulário, cujo seus elementos possuam os atributos acima dispostos corretamente, é possível realizar sua validação;
- A priori, sua validação é feita por HTML5, que é interessante, porém não muito personalizável;
- Para podermos fazê-la via JS, devemos suprimir esta validação, alterando o atributo **novalidate** do formulário:

```
<form novalidate="novalidate" />
```

Checando a validade de um formulário HTML

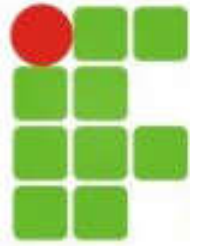


- Para checar a validade de um formulário, selecione-o, e utilize o método **checkValidity()**:

```
formulario.checkValidity()
```

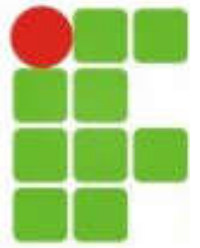
- Retornará **true** caso o formulário todo, i.e., todos os seus campos, estiverem preenchidos corretamente, segundo seus próprios atributos, e **false**, caso contrário;
- Ótimo para uma verificação inicial do todo.

Checando a validade de um campo



- Esta funcionalidade pode ser utilizada para checar campos em qualquer contexto, e não somente dentro de um formulário;
- Pode incluir outros botões ou eventos como blur, quando o usuário acaba de preencher determinado campo;
- Ela pode ser feita de maneira mais personalizada, identificando a origem do erro e sugerindo ao usuário uma correção na entrada de dados.

Checando a validade de um campo



- Para checar a validade de um campo, selecione-o, e utilize o método **checkValidity()**:

```
campo.checkValidity()
```

- Retornará **true** caso o campo estiver preenchido corretamente, segundo seus próprios atributos, e **false**, caso contrário.

Mensagem de validação de um campo



- Para obter a mensagem de validação de um campo, selecione-o, e utilize o atributo **validationMessage**:

`campo.validationMessage`

- Retornará uma string com a mensagem de validação padrão, caso haja, no idioma configurado no navegador.

Estado de validação de um campo

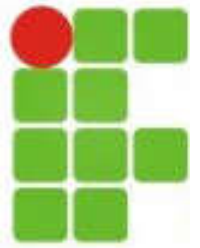


- Através do atributo **validity**, é possível obter detalhes sobre as causas de um campo não estar válido;

`campo.validity.estado`

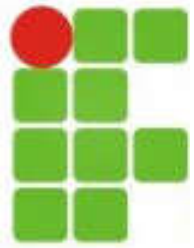
- Os estados guardarão valores booleanos, indicando a confirmação da razão da inconsistência nos dados de entrada.

Estados de validação de um campo

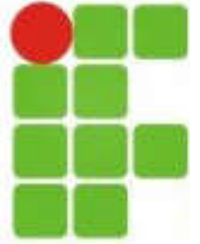


Estado	Validação
valueMissing	Indica se o campo possui atributo required , mas nenhum valor.
typeMismatch	Indica se o valor preenchido não possui a sintaxe adequada para o tipo do campo, e.g., number , email .
rangeUnderflow	Indica se o valor preenchido está abaixo do especificado no atributo min .
rangeOverflow	Indica se o valor preenchido está abaixo do especificado no atributo max .
stepMismatch	Indica se o valor preenchido não é divisível pelo valor especificado no atributo step .

Estados de validação de um campo



Estado	Validação
tooLong	Indica se a quantidade de caracteres excede o especificado no atributo maxlength .
tooShort	Indica se a quantidade de caracteres é inferior ao especificado no atributo minlength .
badInput	Indica se o usuário especificou entrada que o navegador é incapaz de converter/compreender.
patternMismatch	Indica se o valor preenchido não corresponde ao padrão Regex especificado no atributo pattern .



Conclusão

- Com as ferramentas aprendidas aqui, temos uma maneira bem mais personalizada de validar formulários;
- Para impedir que um formulário seja submetido caso seja detectada falha em sua validação, retorne **false** na função de callback;
- Para permitir que o mesmo seja submetido normalmente, retorne **true**;
- Exemplos em **2-validação.html**