



iscte

INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA



emprego  
digital

Módulo 4: Introdução à programação em javascript

# Aula 10


## jQuery (continuação)



# Formulários

- O HTML fornece ferramentas para construir um formulário com **entrada de dados** nativamente:

```
<form method="GET">
  <input type="text" name="name" placeholder="Nome">
  <input type="text" name="age" placeholder="Idade">
  <button type="submit">Enviar</button>
</form>
```

A screenshot of a web browser window showing a simple HTML form. The browser's address bar displays 'localhost'. The form consists of two text input fields stacked vertically. The first field has the placeholder text 'Nome' and the second has 'Idade'. Below these fields is a button labeled 'Enviar'.

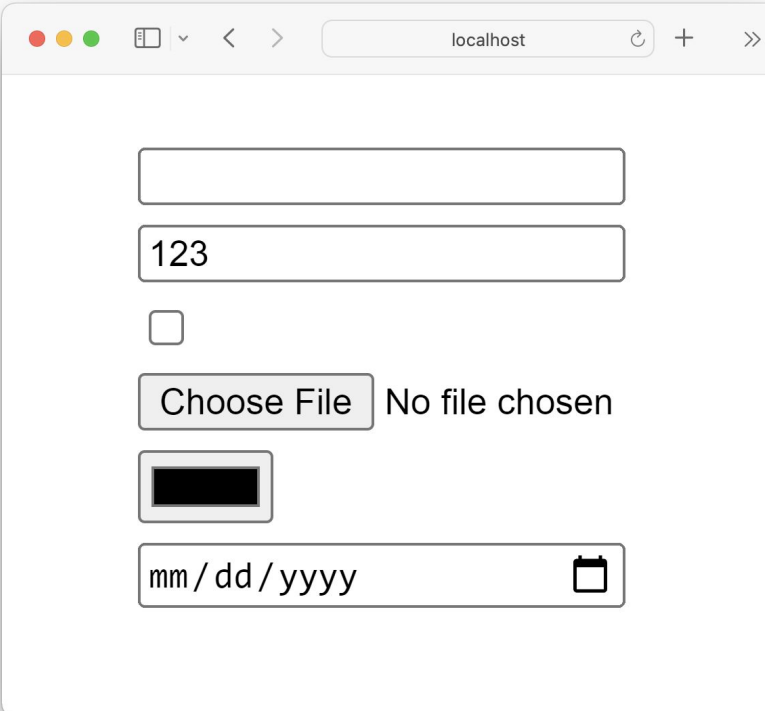
# Elemento <form>

- É utilizado como parente hierárquico dos elementos de *input* que permite agrupá-los e **submeter o formulário** a uma determinada página;
- Contém um atributo “**action**” que recebe um URL para o qual é enviado o formulário;
- Utilizado em conjunto com um `<button type="submit">` para, ao clicar, agregar a informação dos inputs e enviar.

# Elemento <input>

- Tem um comportamento especial porque permite a interação do utilizador e a inserção de informação, que pode ser utilizada nos formulários;

```
<input type="text">  
<input type="number">  
<input type="checkbox">  
<input type="file">  
<input type="color">  
<input type="date">
```



A screenshot of a web browser window displaying a series of HTML input elements. From top to bottom, the elements are: an empty text input field, a number input field containing the value '123', an unchecked checkbox, a file input field with a 'Choose File' button and the text 'No file chosen', a color input field showing a black color swatch, and a date input field with the placeholder text 'mm/dd/yyyy' and a calendar icon.

# Elemento <input>

- Atributos especiais do <input>:

```
//type: tipo de entrada
```

```
<input type="email">
```

```
//placeholder: valor apresentado quando o input está vazio
```

```
<input type="text" placeholder="Nome">
```

```
//name: representa o nome do atributo enviado no formulário
```

```
<input type="number" name="idade">
```

# Elemento <input>

- Atributos de **controlo** do <input>:

//required: representa um campo obrigatório

```
<input type="email" required>
```

//minlength/maxlength: obriga que o texto tenha mais ou menos de X caracteres

```
<input type="text" minlength="10">
```

//min/max: para o type number, obriga um número mínimo ou máximo

```
<input type="number" max="99">
```

# Elemento <input>

- Outros atributos do <input>:

```
//readonly: torna o campo não editável
```

```
<input type="email" readonly>
```

```
//disabled: parecido ao readonly, mas nem permite selecionar texto
```

```
<input type="text" disabled>
```

```
//step: para o type number, de quanto em quanto avança o número
```

```
<input type="number" step="5">
```

# .val()

- A função val() permite-nos obter o valor atual de um input.

```
<input type="email" id="email">  
<input type="password" id="password">
```

```
let email = $("#email");  
console.log(email.val());  
>> "joao@upskill.pt"
```



# Manipulação do Formulário

- Com o jQuery podemos **obter os valores** inseridos nos inputs, **detectar** a introdução de determinadas teclas e **validar** o que o utilizador colocou no formulário:

```
<input type="email" id="email">  
<input type="password" id="password">
```

```
.error {  
  background-color: #edd1d1;  
  color: #cd5c5c;  
  border: 1px solid #bf5b5b;  
}
```

bad@email

```
let email = $("#email");  
if(!email[0].checkValidity()) {  
  email.addClass("error");  
}  
  
let password = $("#password");  
if(password.val().length < 8) {  
  password.addClass("error");  
}
```

# .on("input")

O evento "input" permite-nos detetar alterações num determinado campo. Com a definição correta, poderemos executar uma função sempre que um valor é alterado.


```
<input type="email" id="email">  
<input type="password" id="password">
```

```
let email = $("#email");  
email.on("input", function(){  
    console.log(email.val());  
});
```

# Exercício 1

- Pretende-se criar um formulário que receba o **nome** e a **data de nascimento** da pessoa e coloque uma mensagem com o nome e idade por baixo (calculada pelo ano), num h2.

Dica: Podes usar o `.split("-")[0]` para ir buscar o **ano** a uma string com o formato "1997-06-22"



A screenshot of a web browser window displaying a form. The browser's address bar shows 'localhost'. The form consists of two input fields: the first contains 'Marisa Santos' and the second contains '06/22/1997' with a calendar icon. Below the fields, a message reads 'Nome: Marisa Santos, Idade: 25 anos'.

## Exercício 2

- Criar um ficheiro HTML com utilização do jQuery para gerar uma tabela com a informação dos vários alunos.

Nome	Apelido	Data Nascimento
Maria	Andrade	1990
João	Torres	1988
Sofia	Malhoa	1995

## Exercício 3

- Com base no exercício anterior, adicionar um formulário com inputs para o nome, apelido e data de nascimento, e com um botão, que ao clicar adiciona esse novo aluno à tabela.

## Exercício 4

- Dada uma lista de alunos, imprimir por cada aluno na consola uma mensagem de boas vindas.

Exemplo:

```
>> "O aluno Pedro Sousa com 36 anos chegou!"  
>> "O aluno Manuel Madeira com 31 anos chegou!"  
>> etc
```

## Exercício 5

- Criar uma função que dada uma frase, coloca em **letras maiúsculas** todas as palavras com mais de 4 caracteres.

Exemplo:

```
>> "uma frase com letras maiúsculas" -> "uma FRASE com LETRAS MAIÚSCULAS"  
>> "olá mundo" -> "olá MUNDO"
```

# O futuro profissional começa aqui

iscte

INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA



emprego  
digital



UPskill