

# jQuery Mobile

## Lesson 04 : Creating jQuery Mobile Forms

# Lesson Objectives

- **In this lesson, you will learn :**
- Understanding form basics
  - Using text and search inputs
  - Using radio buttons, checkboxes, and toggles
  - Using select lists and sliders



# Form structure

- **Forms in JQuery Mobile are similar to traditional HTML-coded forms in many ways.**
  - All forms should be wrapped in a form tag that has an action and method that will handle the form data processing on the server. Eg:  
`<form id="myform1" method="post" action="someaction">`
- **However, JQuery Mobile forms are also different from traditional HTML-coded forms:**
  - Since jQuery Mobile makes use of a single-page navigation model all forms (and form elements) must have a different ID to prevent any kind of trouble.
  - All elements in the framework are designed to be flexible so that they can comfortably fit the width of any mobile device screen.

# Form structure

## Field containers

- To visually group form elements together within a form, apply the **data-role="fieldcontain"** attribute to an element wrapping the elements in question. Fieldsets or divs are ideal for this

```
<div data-role="fieldcontain">  
  <label for="name">Textbox: </label>  
  <input type="text" name="name" id="name" value="" />  
</div>
```

## Mini sized elements

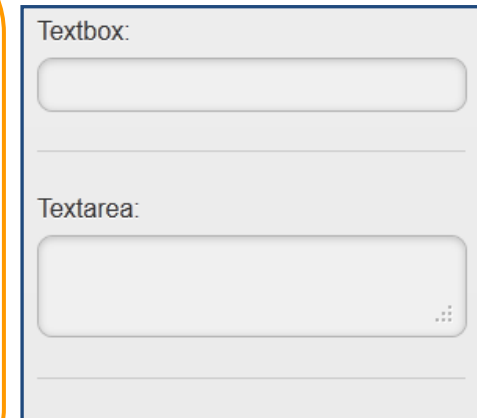
- Add the **data-mini="true"** attribute to the element to create a more compact version of all form elements and buttons

```
<input type="text" name="name" id="basic" data-mini="true" />
```

# Text inputs

- Text inputs are the primary way in which users can enter information and interact with the form.
  - They include actual text inputs, password inputs, and text areas
- `<input type="text">` and `<input type="password">` are used to accept text and display a password filed respectively.
- For multi-line text inputs, use a `textarea` element. Eg:
  - `<textarea cols="20" rows="10" name="ta1" id="mytextarea1" />`

```
<div data-role="fieldcontain">  
  <label for="textbox">Textbox:</label>  
  <input type="text" name="textbox" id="textbox" value="" />  
</div>  
<div data-role="fieldcontain">  
  <label for="textarea">Textarea:</label>  
  <textarea name="textarea" id="textarea" value="" />  
</div>
```



## Text input methods

- **jQuery Mobile uses the textinput plug-in to handle textareas and input fields**
- **The textinput plugin can be directly called on any selector**

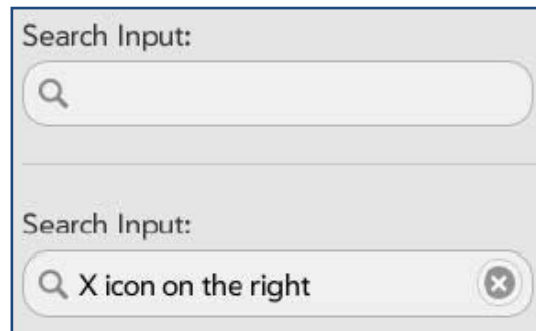
Example : `$('input').textinput();`

```
<script>
$(document).ready(function() {
    // Disable an input
    $("#myinput").textinput('disable');
    // Enable an input
    $("#myotherinput").textinput("enable");
})
</script>
```

## Search inputs

- **Search inputs were introduced in HTML5 and are created using `<input type="search">`**
  - They are styled with pill-shaped corners, a magnifier icon and an "x" icon to clear the field on the right.

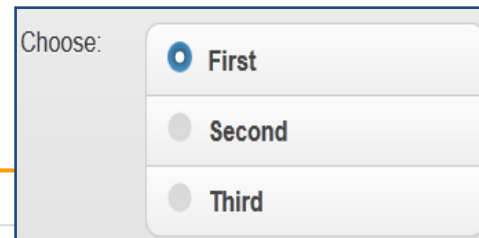
```
<div data-role="fieldcontain">  
  <label for="mysearch">Search Input:</label>  
  <input type="search" name="mysearch" id="mysearch" value="" />  
</div>
```



## Radio buttons

- To create a set of radio buttons, add an input with a `type="radio"` attribute and a corresponding label.
  - Wrap the radio buttons in a **fieldset** element with a legend that serve as a label for the radio buttons group.
  - wrap the fieldset in a div with **data-role="controlgroup"** attribute so it can be styled similar to other elements
  - jQuery Mobile automatically remove margins & borders from the buttons

```
<div data-role="fieldcontain">
  <fieldset data-role="controlgroup"> <legend>Choose:</legend>
    <input type="radio" name="r1" id="r1" value="radio1" checked="true" />
    <label for="myradio1">First</label>
    <input type="radio" name="r2" id="r2" value="radio2" />
    <label for="myradio2">Second</label>
    <input type="radio" name="r3" id="r3" value="radio3" />
    <label for="myradio3">Third</label>
  </fieldset>
</div>
```





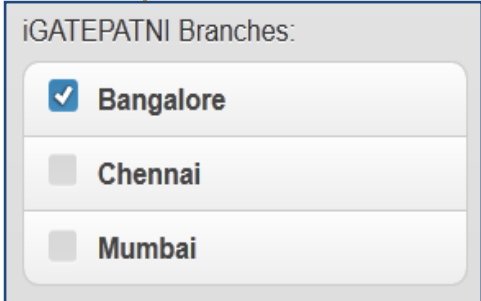
## Check Boxes

- To create a single checkbox, add an input with a type="checkbox" attribute and a corresponding label.

```
<input type="checkbox" name="mycb1" id="mycb1" />  
<label for="mycb1">Yes, I agree!</label>
```

- We can also group any checkboxes vertically

```
<fieldset data-role="controlgroup">  
  <legend>Capgemini Branches:</legend>  
  <label for="pf1">Bangalore</label>  
  <input type="checkbox" name="pf1" id="pf1" />  
  <label for="pf2">Chennai</label>  
  <input type="checkbox" name="pf2" id="pf2" />  
  <label for="pf3">Mumbai</label>  
  <input type="checkbox" name="pf3" id="pf3" />  
</fieldset>
```



iGATEPATNI Branches:	
<input checked="" type="checkbox"/>	Bangalore
<input type="checkbox"/>	Chennai
<input type="checkbox"/>	Mumbai

# Sliders

- To create a slider, add a type="range" attribute to an input element.

```
<div data-role="fieldcontain">  
  <label for="slider">Input slider:</label>  
  <input type="range" name="myslider1" id="myslider1" value="50"  
    min="0" max="100" />  
</div>
```

## Select menus

- A Select menu is created exactly as we do in plain HTML.

```
<label for="select-restaurants">Select Your Restaurants:</label>
<select id="select-restaurants" name="select-restaurants">
  <option value="lecentral">Le Central</option>
  <option value="bistrovandome">Bistro Vandome</option>
  <option value="antoines">Antoine's</option>
</select>
```

- Options can be disabled by adding a disabled attribute to the option tag.
  - Eg : <option value="value2" disabled="true">Value 2</option>

# Summary

- **In this lesson, you learnt:**
- Understanding form basics
  - Using text and search inputs
  - Using radio buttons, checkboxes, and toggles
  - Using select lists and sliders

