





EXERCÍCIOS DE FIXAÇÃO

1. Crie um elemento input do tipo texto e um elemento label. Ao elemento input deve ser adicionado um evento denominado “onblur” (disparado quando o foco do elemento é retirado) que chame uma função javascript que leia o valor contido no input e o coloque no elemento label conforme ilustrado abaixo.

Estado I	Estado II
Antes de disparar o evento	Após o evento ser disparado (foco do input é retirado)
<p style="text-align: center;">Exercise 1</p> <div>teste </div>	<p style="text-align: center;">Exercise 1</p> <div>teste teste </div>

2. Crie uma estrutura similar à do exercício anterior, isto é, um input com evento *onblur* e um label. No evento deve ser chamada um função javascript que acrescente à tag label o valor digitado, ou seja, haverá concatenação dos valores apresentados em tela.

Estado I	Estado II
Antes de disparar o evento	Após o evento ser disparado (foco do input é retirado)
<p style="text-align: center;">Exercise 2</p> <div>teste teste </div>	<p style="text-align: center;">Exercise 2</p> <div>teste2 teste teste2 </div>

3. Faça o que se pede:
 - a. Crie um elemento do tipo ComboBox (select) que conterà as opções: *Escolha um estado; São Paulo; Minas Gerais; Rio de Janeiro e Espírito Santo*. Atribua um valor inteiro para o value de cada opção.
 - b. Crie um novo ComboBox que conterà a opção: *Escolha uma cidade*. Este elemento deve receber, inicialmente, o estilo *display: none* para que não seja exibido na página.
 - c. Crie uma função que vai ser disparada no evento *onchange* do primeiro ComboBox criado (estados). Caso a opção escolhida seja diferente da correspondente a “Escolha um estado”, o ComboBox criado no item **b** deve ser apresentado. Caso contrário, ele deve ser ocultado.

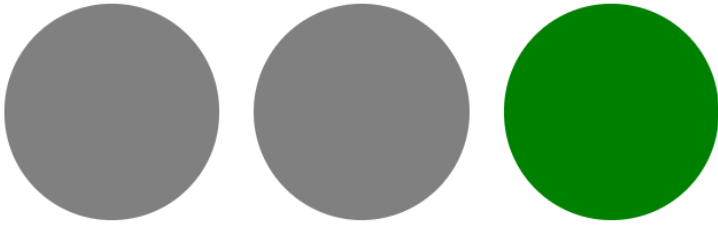
Estado I	Estado II
Com a opção “Escolha um estado” selecionada	Com uma opção diferente de “Escolha um estado” selecionada
<p>Exercise 4</p> <p>Escolha um estado: ▼</p>	<p>Exercise 4</p> <p>Minas Gerais ▼</p> <p>Escolha uma cidade: ▼</p>

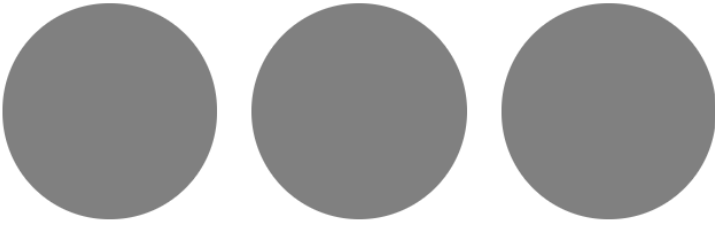
4. Partindo do mesmo princípio adotado no exercício 4, isto é, criar um ComboBox para selecionar um estado e um ComboBox para Selecionar uma cidade, considere que agora, de acordo com o estado escolhido, deve ser apresentado no segundo ComboBox (de cidades) uma lista de cidades (pelo menos três) referente a cada estado.

Estado I	Estado II
Exemplo da seleção do estado “São Paulo”	Exemplo da seleção do estado “Minas Gerais”
<p>Exercise 5</p> <p>São Paulo ▼</p> <p>Escolha uma cidade: ▼</p> <p>Escolha uma cidade: —</p> <p>São Paulo</p> <p>Campinas</p> <p>Ribeirão Preto</p>	<p>Exercise 5</p> <p>Minas Gerais ▼</p> <p>Escolha uma cidade: ▼</p> <p>Escolha uma cidade: —</p> <p>Belo Horizonte</p> <p>Ituiutaba</p> <p>Uberaba</p>

5. Faça três div's com a propriedade CSS *border-radius* com valor elevado de modo que elas fiquem com aparência de um círculo. Cada div deve ter a cor de fundo cinza. A primeira div irá ter uma classe chamada “alert”, a segunda uma classe chamada “warning” e a terceira div uma classe chamada “free”. Cada div irá disparar um evento *onmouseover* e um evento *onmouseleave*.
- O evento *onmouseover* deve chamar uma função denominada *changeSemaphoreStyleOn* de modo que o evento disparado será tratado da seguinte forma:
 - Se for a div com a classe “alert”, deve ser alterada a propriedade *background-color* do atributo style da div para “red”.

- ii. Se for a div com a classe “warning”, deve ser alterada a propriedade *background-color* contida no atributo style da div para “orange”
- iii. Se for a div com a classe “free”, deve ser alterada a propriedade *background-color* contida no atributo style da div para “green”.
- b. O evento *onmouseleave* deve chamar uma função javascript denominada *changeSemaphoreStyleOff* de modo que altere a propriedade *background-color* da div que disparou o evento para “gray”.

Estado I
Exemplo do comportamento quando houver o evento <i>onmouseover</i>
<p>Exercise 6</p>  <pre> <section id="exercise6"> <header>...</header> <div id="semaphore"> <div class="alert" onmouseover="changeSemaphoreStyleOn(this);" on <div class="warning" onmouseover="changeSemaphoreStyleOn(this);" <div class="free" onmouseover="changeSemaphoreStyleOn(this);" on </div> <hr> </section> </pre>

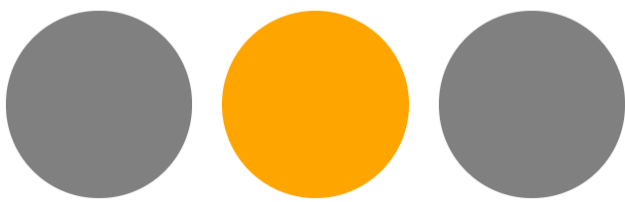
Estado II
Exemplo do comportamento quando houver o evento <i>onmouseleave</i>
<p>Exercise 6</p> 

```

<section id="exercise6">
  <header>...</header>
  <div id="semaphore">
    <div class="alert" onmouseover="changeSemaphoreStyleOn(this);" onmouseleave="changeSemaph
    <div class="warning" onmouseover="changeSemaphoreStyleOn(this);" onmouseleave="changeSema
    <div class="free" onmouseover="changeSemaphoreStyleOn(this);" onmouseleave="changeSemapho
  </div>
  <hr>
</section>

```

6. Faça três div's com a propriedade CSS *border-radius* com valor elevado de modo que elas fiquem com aparência de um círculo. Cada div deve ter a cor de fundo cinza. A primeira div irá ter uma classe chamada "alert", a segunda uma classe chamada "warning" e a terceira div uma classe chamada "free". Cada div irá disparar um evento *onmouseover* e um evento *onmouseleave*.
- O evento *onmouseover* chamará uma função javascript *changeSemaphore*. Esta função deve incluir na div que disparou o evento uma classe chamada *on*. Desta forma:
 - Se for a div "alert" quem disparou o evento, deve ser adicionada a classe *on* na div. Além disso, sua respectiva cor deve ser alterada para vermelho.
 - Se for a div "warning" quem disparou o evento, deve ser adicionada a classe *on* à div e a sua cor deve ser alterada para amarelo.
 - Se for a div "free" quem disparou o evento, deve ser adicionada a classe *on* à div e a sua cor deve ser alterada para verde.
 - O evento *onmouseleave* deve chamar a mesma função javascript *changeSemaphore* de tal modo a remover a classe *on* da div que disparou o evento.

Estado I
Exemplo do comportamento quando houver o evento <i>onmouseover</i>
<p>Exercise 7</p> 

```

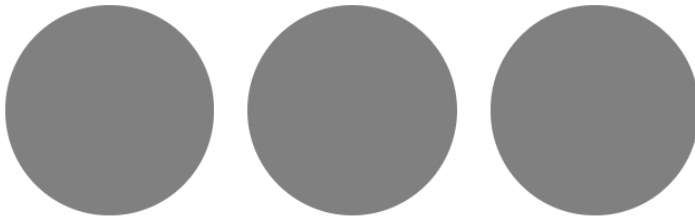
</section>
▼ <section id="exercise7">
  ▶ <header>...</header>
  ▼ <div id="semaphore">
    <div class="alert" onmouseover="changeSemaphore(this);" onmo
    <div class="warning on" onmouseover="changeSemaphore(this);"
    <div class="free" onmouseover="changeSemaphore(this);" onmou
  </div>
  <hr>
</section>

```

Estado II

Exemplo do comportamento quando houver o evento *onmouseleave*

Exercise 7



```

▼ <section id="exercise7">
  ▶ <header>...</header>
  ▼ <div id="semaphore">
    <div class="alert" onmouseover="changeSemaphore(this);" onmouseleave
    <div class="warning" onmouseover="changeSemaphore(this);" onmouseleave
    <div class="free" onmouseover="changeSemaphore(this);" onmouseleave
  </div>
  <hr>
</section>

```

7. Faça um botão que contenha um evento *onclick*. Tal evento deve chamar uma função javascript chamada *addButton*. Desta forma, um novo botão deve ser acrescentado ao lado do botão anterior. Os botões devem ser numerados, sendo assim, o primeiro botão se chamará "Botão 1", o próximo "Botão 2" e assim sucessivamente.

Observação: Para que os botões fiquem posicionados lado a lado, utilize uma lista com com a propriedade CSS *display* com valor *inline-block*.

Estado I	Estado II	Estado III
Botão inicial	Após clicar uma vez no Botão 1.	Após clicar em qualquer um dos botões.
Exercise 8 <div>Botão 1</div>	Exercise 8 <div>Botão 1 Botão 2</div>	Exercise 8 <div>Botão 1 Botão 2 Botão 3</div>

8. Crie um botão chamado “Remove Button” que contenha um evento *onclick*. Este evento deve chamar uma função javascript chamada *RemoveLastButton* que deverá remover o último botão adicionado na lista de botões criada no exercício 8. Caso reste apenas um botão, deve ser apresentada uma mensagem a partir de um *alert* informando que é impossível remover tal botão, pois resta apenas ele.

Estado I	Estado II	Estado III
Botão inicial	Após clicar uma vez no Remove Button.	Quando resta apenas um botão (do exercício 8) e clica-se novamente em Remove Button.
Exercise 8 <div>Botão 1 Botão 2 Botão 3</div> <hr/> Exercise 9 <div>Remove Button</div>	Exercise 8 <div>Botão 1 Botão 2</div> <hr/> Exercise 9 <div>Remove Button</div>	<div>localhost:8080 diz: Resta apenas um botão. Não é possível remover.</div> <div>OK</div> <div>Botão 1</div> <hr/> Exercise 9 <div>Remove Button</div>

9. Crie um input do tipo text, um botão e um textarea. Ao botão adicione um evento *onclick* de modo que quando o evento for disparado será chamada a função *concatText*. Esta função será responsável por ler o valor contido no input text e adicionar ao textarea sem apagar o texto já existente. Não será permitido a inclusão apenas de espaços ou de texto vazio (""), sendo que nestes casos uma mensagem a partir de um *alert* deve ser emitida ao usuário informando que não é possível a inserção de texto vazio.

Estado I
Ilustração dos componentes pedidos neste exercício.

Exercise 10

Add

