

## TP6 JQUERY

**Exercice1 :** Ecrire un exemple, pour détecter l'évènement focus à partir d'un élément fieldset qui regroupe plusieurs champs de notre formulaire. Grâce à focusin(), cela va être possible lorsqu'on va donner le focus à l'un de nos champs de formulaire

### Focusin.html

```
<!DOCTYPE html><html>    <head>        <title> jQuery</title>        <meta charset='utf-8'>    </head>

    <body>

        <h1>évènements en jQuery Focusin</h1>

        <fieldset id='fld'>

            <label for='prenom'>Entrez votre prénom</label>

            <input type='text' id='prenom' name='prenom'>

            <label for='nom'>Entrez votre nom de famille</label>

            <input type='text' id='nom' name='nom'>

        </fieldset>

        <script src='jquery-V-3.3.1.js'></script>

        <script>

            $(document).ready(function(){

                $('#fld').focusin(function(){

                    $(this).css('background-color', 'yellow');

                });

            });

        </script></body></html>
```

## Exercice 2 :Gérer l'évènement "perte de focus" en jQuery

Le jQuery nous fournit également deux méthodes pour gérer l'évènement « perte de focus » d'un élément : les méthode `blur()` et `focusout()`. La différence entre ces deux méthodes va être la même que précédemment : la méthode `blur()` ne va s'exécuter que si un élément bien précis perd le focus tandis que la méthode `focusout()` va s'exécuter si l'élément ou n'importe lequel de ses éléments enfants perd le focus.

### Focusinout.html

```
<!DOCTYPE html><html>          <head><title> jQuery</title>          <meta
charset='utf-8'>  </head>

  <body>

    <h1>évènements en jQuery focus in out</h1>

    <fieldset id='fld'>

      <label for='prenom'>Entrez votre prénom</label>

      <input type='text' id='prenom' name='prenom'>

      <label for='nom'>Entrez votre nom de famille</label>

      <input type='text' id='nom' name='nom'>

    </fieldset>


    <script src='jquery-V-3.3.1.js'></script>
```

```
<script>

    $(document).ready(function(){

        $('#fld').focusin(function(){

            $(this).css('background-color', 'yellow');

        });

        $('#fld').focusout(function(){

            $(this).css('background-color', 'purple');

        });

    });

</script>

</body>

</html>
```

## FocusBlur.html

second avec les méthodes focus() et blur() :

```
<!DOCTYPE html><html>
```

```
<head>  <title> jQuery</title><meta charset='utf-8'>  </head>
```

```
<body>
```

```
  <h1>évènements en jQuery focus blur</h1>
```

```
  <fieldset id='fld'>
```

```
    <label for='prenom'>Entrez votre prénom</label>
```

```
    <input type='text' id='prenom' name='prenom'>
```

```
    <label for='nom'>Entrez votre nom de famille</label>
```

```
    <input type='text' id='nom' name='nom'>
```

```
  </fieldset>
```

```
  <!--On inclut la librairie jQuery-->
```

```
    <script src='jquery-V-3.3.1.js'></script>
```

```
<script>
```

```
  $(document).ready(function(){
```

```
    $('#prenom').focus(function(){
```

```
      $(this).css('background-color', 'yellow');
```

```
    });
```

```
    $('#prenom').blur(function(){
```

```
      $(this).css('background-color', 'purple');
```

```
    });
```

```
  });
```

```
</script>
```

```
</body>
```

```
</html>
```

### Exercice 3 : La méthode jQuery change()

Le jQuery va encore nous permettre de détecter un changement de valeur dans un champ d'un formulaire et de pouvoir réagir à cet évènement avec la méthode change().l'évènement change ne va bien fonctionner qu'avec les éléments input, textarea et select. De plus, notez que l'évènement se déclenche immédiatement avec les cases à cocher, boutons radio et dans le cas de l'utilisation avec l'élément select, mais ne va se déclencher qu'après la perte de focus pour tous les autres éléments.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title> jQuery</title>
```

```
    <meta charset='utf-8'>
```

```
  </head>
```

```
  <body>
```

```
    <h1>évènements en jQuery change</h1>
```

```
    <select>
```

```
      <option value='c1' selected='selected'>Choix 1</option>
```

```
      <option value='c2'>Choix 2</option>
```

```
    </select>
```

```
    <script src='jquery-V-3.3.1.js'></script>
```

```
<script>

$(document).ready(function(){

    $('select').change(function(){

        alert('Nouvelle option sélectionnée');

    });

});

</script>

</body>

</html>
```

#### Exercice 4 La méthode jQuery submit()

L'évènement submit va se déclencher lorsqu'un utilisateur tente d'envoyer un formulaire.

##### **submit.html**

```
<!DOCTYPE html>

<html>

  <head>

    <title> jQuery</title>

    <meta charset='utf-8'>

  </head>

  <body>
```

```
<h1>évènements en jQuery submit</h1>
```

```
<form>
```

```
  <select>
```

```
    <option value='c1' selected='selected'>Choix 1</option>
```

```
    <option value='c2'>Choix 2</option>
```

```
  </select>
```

```
  <input type='submit' value='Envoyer'>
```

```
</form>
```

```
<script src='jquery-V-3.3.1.js'></script>
```

```
<!--On écrit notre code JavaScript / jQuery-->
```

```
<script>
```

```
  $(document).ready(function(){
```

```
    $('form').submit(function(){
```

```
      alert('Formulaire envoyé');
```

```
    });
```

```
  });
```

```
</script>
```

```
</body>
```

```
</html>
```

## Exercice 6 .Gérer le déclenchement des évènements

Le jQuery met à notre disposition les méthodes `trigger()` et `triggerHandler()` qui vont nous permettre de forcer le déclenchement d'un évènement.

Entre autres, retenez que la méthode `trigger()` va déclencher l'évènement spécifié en argument ainsi que le comportement par défaut de l'évènement (comme la prise de focus dans le cas de l'évènement `focus` par exemple)

tandis que `triggerHandler()` ne va pas déclencher le comportement par défaut de l'évènement.

Voici comment nous allons utiliser ces deux méthodes en pratique :

### trigger.html

```
<!DOCTYPE html>

<html>

  <head>

    <title> jQuery</title>

    <meta charset='utf-8'>

  </head>

  <body>

    <h1>évènements en jQuery trigger</h1>

    <label for='fcs'>Champ à focus</label>

    <input type='text' placeholder='Je veux le focus' id='fcs'>

    <button id='trg'>Forcer le focus avec trigger()</button>

    <button id='trgh'>Forcer le focus avec triggerHandler()</button>

    <span id='msg'></span>

    <!--On inclut la librairie jQuery-->

    <script src='jquery-V-3.3.1.js'></script>
```



```

<!--On écrit notre code JavaScript / jQuery-->
<script>
    $(document).ready(function(){
        //trigger() s'exécute si on clique sur le bouton trg
        $('#trg').click(function(){
            $('#fcs').trigger('focus');
        });
        //triggerHandler() s'exécute si on clique sur le bouton trgh
        $('#trgh').click(function(){
            $('#fcs').triggerHandler('focus');
        });
        //On attache un gestionnaire d'évènement focus à notre input
        $('#fcs').focus(function(){
            $('#msg').text('Evènement focus déclenché');
        })
    });
</script>
</body>
</html>

```

Ici, on force le déclenchement de l'évènement focus sur notre élément input lorsque l'on clique sur l'un des deux boutons. Le gestionnaire d'évènement relatif à l'évènement focus est alors exécuté.

On peut alors observer la différence entre `trigger()` et `triggerHandler()` : si l'on clique sur le bouton qui possède `trigger()`, le comportement par défaut de l'évènement focus (c'est-à-dire donner le focus à un élément) s'exécute également, ce qui n'est pas le cas lorsque l'on clique sur le bouton auquel est accroché `triggerHandler()`.