jQuery

Présentation

Le but de jQuery est de rendre beaucoup plus facile l'utilisation de JavaScript sur votre page web.

Qu'est-ce que vous devriez déjà savoir

Avant de commencer l'étude de jQuery, vous devriez avoir une connaissance de base:

* HTML
* CSS
* JavaScript

Qu'est-ce que jQuery?

jQuery est une bibliothèque JavaScript poids léger, "écrire moins, faire plus".

Le but de jQuery est de rendre beaucoup plus facile l'utilisation de

JavaScript sur votre site.

jQuery prend beaucoup de tâches courantes qui nécessitent de nombreuses lignes de code JavaScript pour les accomplir, et les enveloppe dans des méthodes que vous pouvez appeler avec une seule ligne de code.

jQuery simplifie aussi beaucoup de choses compliquées de JavaScript, comme les appels AJAX et la manipulation du DOM.

La bibliothèque jQuery contient les fonctionnalités suivantes:

* HTML / manipulation DOM
* la manipulation CSS
* méthodes d'événement HTML
* Effets et animations
* AJAX
* Utilitaires

Pourquoi jQuery?

Il y a beaucoup d'autres frameworks JavaScript, mais jQuery semble être le plus populaire, et aussi le plus extensible.

Bon nombre des plus grandes entreprises sur le Web utilisent jQuery, tels que:

* Google
* Microsoft
* IBM
* Netflix

|  |  |
| --- | --- |
| **Remarque** | **Est-ce que jQuery fonctionnera dans tous les navigateurs?**  L'équipe jQuery sait tout sur ​​les questions multi-navigateurs, et ils ont écrit cette connaissance dans la bibliothèque jQuery. jQuery fonctionnera exactement de la même façon dans tous les principaux navigateurs, y compris Internet Explorer 6! |

Ajout de jQuery à vos pages Web

Il existe plusieurs façons de commencer à utiliser jQuery sur votre site web. Vous pouvez:

* Télécharger la bibliothèque jQuery de jQuery.com
* Inclure jQuery à partir d'un CDN, comme Google

Téléchargement jQuery

Il existe deux versions de jQuery disponibles pour le téléchargement:

* Version de production - pour la publication votre site web - elle a été minimisée et comprimée
* La version de développement - pour le test et développement (code non compressé et lisible)

Les deux versions peuvent être téléchargées à partir de [jQuery.com](http://jquery.com/download/) .

La bibliothèque jQuery est un seul fichier JavaScript, que vous référencez avec le balise HTML <script> (notez que la balise <script> devrait être dans la section <head>):

<head>  
<script src="jquery-1.12.0.min.js"></script>  
</head>

**Astuce:** Placez le fichier téléchargé dans le même répertoire que les pages où vous souhaitez l'utiliser.

|  |  |
| --- | --- |
| **Remarque** | **Est - ce que vous vous demandez pourquoi nous ne disposons pas de type = "text / javascript" à l' intérieur du tag <script>?**  Ce n'est pas nécessaire en HTML5. JavaScript est le langage de script par défaut en HTML5 et dans tous les navigateurs modernes! |

jQuery CDN

Si vous ne souhaitez pas télécharger et héberger jQuery vous-même, vous pouvez l'inclure à partir d'un CDN (Content Delivery Network).

Google et Microsoft hôte de jQuery.

Pour utiliser jQuery de Google ou Microsoft, utilisez une des options suivantes:

Google CDN:

<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>  
</head>

Microsoft CDN:

<head>  
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.12.0.min.js"></script>  
</head>

|  |  |
| --- | --- |
| **Remarque** | **Un grand avantage d'utiliser le jQuery hébergée par Google ou Microsoft:**  Beaucoup d'utilisateurs ont déjà téléchargé jQuery de Google ou Microsoft en visitant un autre site. En conséquence, il sera chargé à partir du cache quand ils visitent votre site, ce qui conduit à accélérer le temps de chargement. Aussi, la plupart des CDN feront en sorte qu'une fois qu'un utilisateur demande un fichier à partir d'eux, il sera servi à partir du serveur le plus proche pour eux, ce qui conduit également à accélérer le temps de chargement. |

Syntaxe

Avec jQuery vous sélectionnez des éléments HTML et effectuez des "actions" sur eux.

Syntaxe jQuery

La syntaxe jQuery est fait sur​​ mesure pour la **sélection** des éléments HTML et en effectuant une certaine **mesure** sur l'élément (s).

La syntaxe de base est: **$ ( *sélecteur* ).*action* ()**

* Un signe $ pour définir / accès jQuery
* Un ( *sélecteur* ) pour "requête (ou trouver)" l'élément(s) HTML
* Une JQuery *action* () à effectuer sur l'élément (s)

Exemples:

$ (This).hide () - cache l'élément courant.

$ ( "P").hide () - Masquer <p> éléments.

$ ( ". Test").hide () - Cache tous les éléments avec class = "test".

$ ( "# Test").hide () - Cache l'élément avec id = "test".

|  |  |
| --- | --- |
| **Remarque** | **Êtes - vous familier avec les sélecteurs CSS?**  JQuery utilise la syntaxe CSS pour sélectionner des éléments. Vous en apprendrez plus sur la syntaxe de sélection dans le chapitre suivant de ce tutoriel. |

## L'événement 'Document Ready'

Toutes les méthodes jQuery sont à l'intérieur d'un événement 'document ready':

$(document).ready(function(){  
  
   *// jQuery methods go here...*  
  
});

Ceci afin d'éviter tout code jQuery de s'exécuter avant que le document est fini de se charger (est prêt).

Il est bon d'attendre que le document soit complètement chargé et prêt avant de travailler avec elle. Cela permet également d'avoir votre code JavaScript dans la section d'entête avant le corps de votre document.

Voici quelques exemples d'actions qui peuvent échouer si les méthodes sont exécutées avant que le document ait entièrement été chargé:

* Essayer de cacher un élément qui n'a pas encore été créé
* Essayer d'obtenir la taille d'une image qui n'a pas encore été chargée

**Astuce:** L'équipe jQuery a également créé une méthode encore plus courte pour l'événement 'document ready':

$(function(){  
  
   *// jQuery methods go here...*  
  
});

Utilisez la syntaxe que vous préférez. Nous pensons que l'événement 'document ready' est plus facile à comprendre à la lecture du code.

sélecteurs jQuery

Les sélecteurs jQuery sont l'une des parties les plus importantes de la bibliothèque jQuery.

sélecteurs jQuery

Les sélecteurs jQuery vous permettent de sélectionner et de manipuler l'élément (s) HTML.

Les sélecteurs jQuery sont utilisés pour «trouver» (ou sélectionnez) les éléments HTML en fonction de leurs id, des classes, des types, des attributs, des valeurs d'attributs et bien plus encore. Il est basé sur les [sélecteurs CSS](http://www.w3schools.com/cssref/css_selectors.asp) existants , et plus encore, jQuery possède quelques propres sélecteurs personnalisés.

Tous les sélecteurs jQuery commencent avec le signe du dollar et entre parenthèses: $ ().

Le sélecteur d'élément

Le sélecteur d'élément jQuery sélectionne des éléments basés sur le nom de l'élément.

Vous pouvez sélectionner tous les éléments <p> sur une page comme ceci:

$("p")

**Exemple**

Lorsqu'un utilisateur clique sur un bouton, tous les éléments <p> seront cachés:

Exemple

$(document).ready(function(){  
    $("button").click(function(){  
        $("#hhh").hide();  
    });  
});

Le sélecteur #id

jQuery utilise l'attribut id d'une balise HTML pour trouver l'élément spécifique.

Un id doit être unique dans une page, vous devez donc utiliser le sélecteur de #id lorsque vous souhaitez trouver un seul élément unique.

Pour trouver un élément avec un id spécifique, écrire un caractère dièse, suivi par l'identifiant de l'élément HTML:

$("#test")

**Exemple**

Lorsqu'un utilisateur clique sur un bouton, l'élément avec id = "test" sera caché:

Exemple

$(document).ready(function(){  
    $("button").click(function(){  
        $("#test").hide();  
    });  
});

Le sélecteur .class

Le sélecteur de classe jQuery trouve des éléments avec une classe spécifique.

Pour trouver des éléments avec une classe spécifique, écrire le caractère ".", suivi du nom de la classe:

$(".test")

**Exemple**

Lorsqu'un utilisateur clique sur un bouton, les éléments avec class = "test" seront cachés:

Exemple

$(document).ready(function(){  
    $("button").click(function(){  
        $(".test").hide();  
    });  
});

Autres exemples de sélecteurs jQuery

|  |  |
| --- | --- |
| **Syntax** | **Description** |
| $("\*") | Selects all elements |
| $(this) | Selects the current HTML element |
| $("p.intro") | Selects all <p> elements with class="intro" |
| $("p:first") | Selects the first <p> element |
| $("ul li:first") | Selects the first <li> element of the first <ul> |
| $("ul li:first-child") | Selects the first <li> element of every <ul> |
| $("[href]") | Selects all elements with an href attribute |
| $("a[target='\_blank']") | Selects all <a> elements with a target attribute value equal to "\_blank" |
| $("a[target!='\_blank']") | Selects all <a> elements with a target attribute value NOT equal to "\_blank" |
| $(":button") | Selects all <button> elements and <input> elements of type="button" |
| $("tr:even") | Selects all even <tr> elements |
| $("tr:odd") | Selects all odd <tr> elements |

Fonctions dans un fichier séparé

Si votre site contient un grand nombre de pages, et vous voulez que vos fonctions jQuery soient faciles à entretenir, vous pouvez mettre vos fonctions jQuery dans un fichier .js séparé.

Lorsque nous démontrons jQuery dans ce tutoriel, les fonctions sont ajoutées directement dans la section <head>. Cependant, il est parfois préférable de les placer dans un fichier séparé, comme celui-ci (utiliser l'attribut src pour désigner le fichier .js):

Exemple

<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js">  
</script>  
<script src="my\_jquery\_functions.js"></script>  
</head>

Méthodes D’Evenement jQuery

jQuery est fait sur mesure pour répondre aux événements dans une page HTML.

Quels sont les événements?

Toutes les différentes actions d'un visiteur auxquelles une page Web peut réagir, sont appelées des événements.

Un événement représente le moment précis où quelque chose se passe.

Exemples:

* le déplacement d'une souris sur un élément
* la sélection d'un bouton radio
* en cliquant sur un élément

Voici quelques événements DOM communs:

|  |  |  |  |
| --- | --- | --- | --- |
| **Mouse Events** | **Keyboard Events** | **Form Events** | **Document/Window Events** |
| click | Keypress | submit | load |
| dblclick | Keydown | change | resize |
| mouseenter | Keyup | focus | scroll |
| mouseleave |  | blur | unload |

Syntaxe jQuery Pour les méthodes d'événements

La plupart des événements DOM ont une méthode jQuery équivalent.

Pour attribuer un événement clic à tous les paragraphes sur une page, vous pouvez faire ceci:

$("p").click();

L'étape suivante consiste à définir ce qui doit se produire lorsque l'événement se déclenche. Vous devez passer une fonction à l'événement:

$("p").click(function(){  
  // action goes here!!  
});

Méthodes d'événement jQuery couramment Utilisées

**$ (Document) .ready ()**

La méthode $ (document) .ready () nous permet d'exécuter une fonction lorsque le document est entièrement chargé.

click **()**

La méthode click () attache une fonction de gestionnaire d'événement à un élément HTML.

La fonction est exécutée lorsque l'utilisateur clique sur l'élément HTML.

L'exemple suivant dit: Quand un événement cliquer se déclenche sur un élément <p>; cacher la <p> élément courant:

Exemple

$("p").click(function(){  
    $(this).hide();  
});

**dblclick ()**

La méthode dblclick () attache une fonction de gestionnaire d'événement à un élément HTML.

La fonction est exécutée lorsque l'utilisateur double-clique sur l'élément HTML:

Exemple

$("p").dblclick(function(){  
    $(this).hide();  
});

**mouseenter ()**

La méthode mouseenter () attache une fonction de gestionnaire d'événement à un élément HTML.

La fonction est exécutée lorsque le pointeur de la souris entre dans l'élément HTML:

Exemple

$("#p1").mouseenter(function(){  
    alert("You entered p1!");  
});

**mouseleave ()**

La méthode mouseleave () attache une fonction de gestionnaire d'événement à un élément HTML.

La fonction est exécutée lorsque le pointeur de la souris quitte l'élément HTML:

Exemple

$("#p1").mouseleave(function(){  
    alert("Bye! You now leave p1!");  
});

mousedown **()**

La méthode mousedown () attache une fonction de gestionnaire d'événement à un élément HTML.

La fonction est exécutée, lorsque le bouton du milieu, droit ou gauche de la souris est pressé vers le bas, tandis que la souris survole l'élément HTML:

Exemple

$("#p1").mousedown(function(){  
    alert("Mouse down over p1!");  
});

**mouseup ()**

La méthode mouseup () attache une fonction de gestionnaire d'événement à un élément HTML.

La fonction est exécutée, lorsque le bouton du milieu, droit ou gauche de la souris est relâché, alors que la souris survole l'élément HTML:

Exemple

$("#p1").mouseup(function(){  
    alert("Mouse up over p1!");  
});

hover **()**

La méthode hover () prend deux fonctions et est une combinaison des méthodes mouseenter () et mouseleave ().

La première fonction est exécutée lorsque la souris entre l'élément HTML, et la seconde fonction est exécutée lorsque la souris quitte l'élément HTML:

Exemple

$("#p1").hover(function(){  
    alert("You entered p1!");  
},  
function(){  
    alert("Bye! You now leave p1!");  
});

**focus ()**

La méthode **focus** () attache une fonction de gestionnaire d'événement à un champ de formulaire HTML.

La fonction est exécutée lorsque le champ de formulaire reçoit le focus:

Exemple

$("input").focus(function(){  
    $(this).css("background-color", "#cccccc");  
});

blur **()**

La méthode blur () attache une fonction de gestionnaire d'événement à un champ de formulaire HTML.

La fonction est exécutée lorsque le champ de formulaire perd le focus:

Exemple

$("input").blur(function(){  
    $(this).css("background-color", "#ffffff");  
});

on ()

La méthode on () attache un ou plusieurs gestionnaires d'événements pour les éléments sélectionnés.

Joindre un événement de clic sur un élément <p>:

Exemple

$("p").on("click", function(){  
    $(this).hide();  
});

Joindre plusieurs gestionnaires d'événements à un élément <p>:

Exemple

$("p").on({  
    mouseenter: function(){  
        $(this).css("background-color", "lightgray");  
    },   
    mouseleave: function(){  
        $(this).css("background-color", "lightblue");  
    },   
    click: function(){  
        $(this).css("background-color", "yellow");  
    }   
});

Effets jQuery - Hide and Show

Masquer, Afficher, Basculer, glisser, Fade et Animation. (Hide, Show, Toggle, Slide, Fade, and Animate).

Exemples

jQuery hide () et show ()

Avec jQuery, vous pouvez masquer et afficher les éléments HTML avec les méthodes hide () et show ():

Exemple

$("#hide").click(function(){  
    $("p").hide();  
});  
  
$("#show").click(function(){  
    $("p").show();  
});

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>**

**<script>**

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

**$("#hide").click(function(){**

**$("p").hide();**

**});**

**$("#show").click(function(){**

**$("p").show();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p>If you click on the "Hide" button, I will disappear.</p>**

**<button id="hide">Hide</button>**

**<button id="show">Show</button>**

**</body>**

**</html>**

**Syntaxe:**

$(*selector*).hide(*speed,callback*);  
  
$(*selector*).show(*speed,callback*);

Le paramètre de vitesse optionnel spécifie la vitesse de la cachette / montrant, et peut prendre les valeurs suivantes: "slow", "fast", ou en millisecondes.

Le paramètre de rappel optionnel est une fonction à exécuter après que la méthode hide () ou show () se complète (vous en apprendrez plus sur les fonctions de rappel dans un chapitre ultérieur).

L'exemple suivant illustre le paramètre de vitesse avec hide ():

Exemple

$("button").click(function(){  
    $("p").hide(1000);  
});

jQuery toggle ()

Avec jQuery, vous pouvez basculer entre les méthodes hide () et show () avec la méthode toggle ().

Montrer les éléments qui sont cachés et cacher les éléments qui sont visibles:

Exemple

$("button").click(function(){  
    $("p").toggle();  
});

**Syntaxe:**

$(*selector*).toggle(*speed,callback*);

Le paramètre de vitesse en option peut prendre les valeurs suivantes: "lent", "rapide", ou en millisecondes.

Le paramètre de rappel optionnel est une fonction à exécuter après que toggle () se complète.

Effets jQuery - fading

Avec jQuery, vous pouvez faire estomper ou adoucir et afficher des éléments.

**jQuery a les méthodes fade() suivantes:**

* fadeIn ()
* fadeOut ()
* fadeToggle ()
* fadeTo ()

Méthode fadeIn ()

La méthode jQuery fadeIn () est utilisée pour afficher un élément caché.

**Syntaxe:**

$(*selector*).fadeIn(*speed,callback*);

Le paramètre de vitesse optionnel spécifie la durée de l'effet. Il peut prendre les valeurs suivantes: "lent", "rapide", ou en millisecondes.

Le paramètre de rappel optionnel est une fonction à exécuter après la disparition complète.

L'exemple suivant illustre la méthode fadeIn () avec des paramètres différents:

Exemple

$("button").click(function(){  
    $("#div1").fadeIn();  
    $("#div2").fadeIn("slow");  
    $("#div3").fadeIn(3000);  
});

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("#div1").fadeIn();

$("#div2").fadeIn("slow");

$("#div3").fadeIn(3000);

});

});

</script>

</head>

<body>

<p>Demonstrate fadeIn() with different parameters.</p>

<button>Click to fade in boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;display:none;background-color:red;"></div><br>

<div id="div2" style="width:80px;height:80px;display:none;background-color:green;"></div><br>

<div id="div3" style="width:80px;height:80px;display:none;background-color:blue;"></div>

</body>

</html>

Méthode fadeOut ()

La méthode jQuery fadeOut () est utilisée pour faire disparaître un élément visible.

**Syntaxe:**

$(*selector*).fadeOut(*speed,callback*);

Le paramètre de vitesse optionnel spécifie la durée de l'effet. Il peut prendre les valeurs suivantes: "lent", "rapide", ou en millisecondes.

Le paramètre de rappel optionnel est une fonction à exécuter après la disparition complète.

L'exemple suivant illustre la méthode fadeOut () avec des paramètres différents:

Exemple

$("button").click(function(){  
    $("#div1").fadeOut();  
    $("#div2").fadeOut("slow");  
    $("#div3").fadeOut(3000);  
});

Méthode fadeToggle ()

La méthode jQuery fadeToggle () permet de basculer entre le fadeIn () et fadeOut (méthodes).

**Syntaxe:**

$(*selector*).fadeToggle(*speed,callback*);

Le paramètre de vitesse optionnel spécifie la durée de l'effet. Il peut prendre les valeurs suivantes: "lent", "rapide", ou en millisecondes.

Le paramètre de rappel optionnel est une fonction à exécuter après la disparition complète.

L'exemple suivant illustre la méthode fadeToggle () avec des paramètres différents:

Exemple

$("button").click(function(){  
    $("#div1").fadeToggle();  
    $("#div2").fadeToggle("slow");  
    $("#div3").fadeToggle(3000);  
});

Méthode fadeTo ()

La méthode jQuery fadeTo () permet la décoloration à une opacité donnée (valeur entre 0 et 1).

**Syntaxe:**

$(*selector*).fadeTo(*speed,opacity,callback*);

Le paramètre de vitesse requise spécifie la durée de l'effet. Il peut prendre les valeurs suivantes: "lent", "rapide", ou en millisecondes.

Le paramètre d'opacité nécessaire dans la méthode fadeTo () spécifie la décoloration à une opacité donnée (valeur entre 0 et 1).

Le paramètre de rappel optionnel est une fonction à exécuter après la fonction complète.

L'exemple suivant illustre la méthode fadeTo () avec des paramètres différents:

Exemple

$("button").click(function(){  
    $("#div1").fadeTo("slow", 0.15);  
    $("#div2").fadeTo("slow", 0.4);  
    $("#div3").fadeTo("slow", 0.7);  
});

jQuery - Sliding

Avec jQuery, vous pouvez créer un effet de glissement sur les éléments.

jQuery a les méthodes de glissement suivantes:

* slideDown()
* slideUp()
* slideToggle()

Méthode slideDown ()

La méthode jQuery slideDown () est utilisée pour glisser vers le bas un élément.

**Syntaxe:**

$(*selector*).slideDown(*speed,callback*);

Le paramètre de vitesse optionnel spécifie la durée de l'effet. Il peut prendre les valeurs suivantes: "lent", "rapide", ou en millisecondes.

Le paramètre de rappel optionnel est une fonction à exécuter après le glissement terminée.

L'exemple suivant illustre la méthode slideDown ():

Exemple

$("#flip").click(function(){  
    $("#panel").slideDown();  
});

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#flip").click(function(){

$("#panel").slideDown("slow");

});

});

</script>

<style>

#panel, #flip {

padding: 5px;

text-align: center;

background-color: #e5eecc;

border: solid 1px #c3c3c3;

}

#panel {

padding: 50px;

display: none;

}

</style>

</head>

<body>

<div id="flip">Click to slide down panel</div>

<div id="panel">Hello world!</div>

</body>

</html>

jQuery slideUp () Méthode

Le procédé jQuery slideUp () est utilisée pour glisser vers le haut un élément.

**Syntaxe:**

$(*selector*).slideUp(*speed,callback*);

Le paramètre de vitesse optionnel spécifie la durée de l'effet. Il peut prendre les valeurs suivantes: "lent", "rapide", ou en millisecondes.

Le paramètre de rappel optionnel est une fonction à exécuter après le glissement terminée.

L'exemple suivant illustre la méthode slideUp ():

Exemple

$("#flip").click(function(){  
    $("#panel").slideUp();  
});

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#flip").click(function(){

$("#panel").slideUp("slow");

});

});

</script>

<style>

#panel, #flip {

padding: 5px;

text-align: center;

background-color: #e5eecc;

border: solid 1px #c3c3c3;

}

#panel {

padding: 50px;

}

</style>

</head>

<body>

<div id="flip">Click to slide up panel</div>

<div id="panel">Hello world!</div>

</body>

</html>

Méthode slideToggle ()

La méthode jQuery slideToggle () permet de basculer entre le slideDown () et slideUp (méthodes).

Si les éléments ont été glissé vers le bas, slideToggle () va glisser vers le haut.

Si les éléments ont été glissé vers le haut, slideToggle () va les glisser vers le bas.

$(*selector*).slideToggle(*speed,callback*);

Le paramètre de vitesse en option peut prendre les valeurs suivantes: "lent", "rapide", millisecondes.

Le paramètre de rappel optionnel est une fonction à exécuter après le glissement terminée.

L'exemple suivant illustre la méthode slideToggle ():

Exemple

$("#flip").click(function(){  
    $("#panel").slideToggle();  
});

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#flip").click(function(){

$("#panel").slideToggle("slow");

});

});

</script>

<style>

#panel, #flip {

padding: 5px;

text-align: center;

background-color: #e5eecc;

border: solid 1px #c3c3c3;

}

#panel {

padding: 50px;

display: none;

}

</style>

</head>

<body>

<div id="flip">Click to slide the panel down or up</div>

<div id="panel">Hello world!</div>

</body>

</html>

Effets jQuery - Animations

La méthode jQuery animate () vous permet de créer des animations personnalisées.

**Syntaxe:**

$(*selector*).animate({*params*}*,speed,callback*);

Le paramètre params requis définit les propriétés CSS pour être animés.

Le paramètre de vitesse optionnel spécifie la durée de l'effet. Il peut prendre les valeurs suivantes: "lent", "rapide", ou en millisecondes.

Le paramètre de rappel optionnel est une fonction à exécuter après l'animation complète.

L'exemple suivant illustre une utilisation simple de la méthode animate(); elle déplace un élément <div> vers la droite, jusqu'à ce qu'il ait atteint une propriété gauche de 250px:

Exemple

$("button").click(function(){  
    $("div").animate({left: '250px'});  
});

|  |  |
| --- | --- |
| **Remarque** | Par défaut, tous les éléments HTML ont une position statique, et ne peuvent pas être déplacés.  Pour manipuler la position, souvenez - vous d' abord de définir la propriété CSS de position de l'élément à relative, fixe ou absolue! |

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("div").animate({left: '250px'});

});

});

</script>

</head>

<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>

</body>

</html>

jQuery animate () - Manipuler Propriétés multiples

Notez que plusieurs propriétés peuvent être animées en même temps:

Exemple

$("button").click(function(){  
    $("div").animate({  
        left: '250px',  
        opacity: '0.5',  
        height: '150px',  
        width: '150px'  
    });  
});

|  |  |
| --- | --- |
| **Remarque** | **Est - il possible de manipuler toutes les propriétés CSS avec la méthode animée ()?**  Oui, presque! Cependant, il y a une chose importante à retenir: tous les noms de propriété doivent être CamelCase lorsqu'ils sont utilisés avec la méthode animate (): Vous aurez besoin d'écrire paddingLeft au lieu de padding-left, marginRight au lieu de margin-Right, et ainsi de suite. en outre, l'animation de couleur n'est pas incluse dans le noyau de bibliothèque jQuery. Si vous souhaitez animer la couleur, vous devez télécharger le plugin couleur Animations de jQuery.com. |

jQuery animate () - Utilisation des valeurs relatives

Il est également possible de définir des valeurs relatives (dont la valeur est alors par rapport à la valeur actuelle de l'élément). Ceci est réalisé en plaçant ou + = - = en face de la valeur:

Exemple

$("button").click(function(){  
    $("div").animate({  
        left: '250px',  
        height: '+=150px',  
        width: '+=150px'  
    });  
});

jQuery animate () - Utilisation des valeurs pré-définies

Vous pouvez même spécifier la valeur de l'animation d'une propriété comme "show", "hide", ou "toggle":

Exemple

$("button").click(function(){  
    $("div").animate({  
        height: 'toggle'  
    });  
});

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("div").animate({

height: 'toggle'

});

});

});

</script>

</head>

<body>

<p>Click the button multiple times to toggle the animation.</p>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>

</body>

</html>

jQuery animate () - Utilise la Fonctionnalité de la file d'attente

Par défaut, jQuery est livré avec la fonctionnalité de file d'attente pour les animations.

Cela signifie que si vous appelez de multiple animate () l'une après l'autre, jQuery crée une file d'attente «interne» avec ces appels de méthode. Ensuite, il exécute les animations un par un.

Donc, si vous souhaitez effectuer différentes animations l'une après l'autre, nous profitons de la fonctionnalité de file d'attente:

Exemple 1

$("button").click(function(){  
    var div = $("div");  
    div.animate({height: '300px', opacity: '0.4'}, "slow");  
    div.animate({width: '300px', opacity: '0.8'}, "slow");  
    div.animate({height: '100px', opacity: '0.4'}, "slow");  
    div.animate({width: '100px', opacity: '0.8'}, "slow");  
});

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

var div = $("div");

div.animate({height: '300px', opacity: '0.4'}, "slow");

div.animate({width: '300px', opacity: '0.8'}, "slow");

div.animate({height: '100px', opacity: '0.4'}, "slow");

div.animate({width: '100px', opacity: '0.8'}, "slow");

});

});

</script>

</head>

<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>

</body>

</html>

L'exemple ci-dessous déplace l'élément <div> à droite, puis augmente la taille de la police du texte:

exemple 2

$("button").click(function(){  
    var div = $("div");  
    div.animate({left: '100px'}, "slow");  
    div.animate({fontSize: '3em'}, "slow");  
});

Arrêt des Animations

La méthode jQuery stop() est utilisée pour arrêter les animations ou les effets avant qu'ils ne soient terminés.

La méthode stop() fonctionne sur toutes les fonctions d'effet jQuery, y compris de glissement, de fading et des animations.

**Syntaxe:**

$(*selector*).stop(*stopAll,goToEnd*);

Le paramètre optionnel stopAll précise si aussi la file d'attente d'animation doit être effacée ou non. Par défaut il est à faux, ce qui signifié que l'animation seulement active s'arrête, ce qui permet aux animations en attente d'être exécutées par la suite.

Le paramètre optionnel gotoEnd précise si oui ou non l'animation en cours se termine immédiatement. Par défaut il est à faux.

L'exemple suivant illustre la méthode stop(), sans paramètres:

exemple

$("#stop").click(function(){  
    $("#panel").stop();  
});

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#flip").click(function(){

$("#panel").slideDown(5000);

});

$("#stop").click(function(){

$("#panel").stop();

});

});

</script>

<style>

#panel, #flip {

padding: 5px;

font-size: 18px;

text-align: center;

background-color: #555;

color: white;

border: solid 1px #666;

border-radius: 3px;

}

#panel {

padding: 50px;

display: none;

}

</style>

</head>

<body>

<button id="stop">Stop sliding</button>

<div id="flip">Click to slide down panel</div>

<div id="panel">Hello world!</div>

</body>

</html>

jQuery Callback Fonctions

Une fonction de rappel est exécutée après que l'effet actuel soit terminé à 100%.

Fonctions jQuery Callback

Les instructions JavaScript sont exécutés ligne par ligne. Cependant, avec des effets, la prochaine ligne de code peut être exécutée même si l'effet n'est pas terminé. Cela peut créer des erreurs.

Pour éviter cela, vous pouvez créer une fonction de rappel.

Une fonction de rappel est exécutée après que l'effet en cours soit terminé.

Syntaxe typique: **$ ( *sélecteur* ) .hide ( *vitesse, rappel* );**

**Exemples**

L'exemple, ci-dessous, présente un paramètre de rappel qui est une fonction qui sera exécutée après que l'effet hide soit terminé:

Exemple avec Callback

$("button").click(function(){  
    $("p").hide("slow", function(){  
        alert("The paragraph is now hidden");  
    });  
});

L'exemple ci-dessous n'a pas de paramètre de rappel, et la boîte d'alerte sera affichée avant que l'effet hide ne soit terminée:

Exemple sans Callback

$("button").click(function(){  
    $("p").hide(1000);  
    alert("The paragraph is now hidden");  
});

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("p").hide(1000);

alert("The paragraph is now hidden");

});

});

</script>

</head>

<body>

<button>Hide</button>

<p>This is a paragraph with little content.</p>

</body>

</html>

jQuery - Enchaînement

Avec jQuery, vous pouvez enchaîner les actions / méthodes.

L'enchaînement nous permet d'exécuter plusieurs méthodes jQuery (sur le même élément) dans une seule instruction.

Jusqu'à présent, nous avons écrit des déclarations jQuery une à la fois (l'un après l'autre).

Cependant, il existe une technique appelée chaînage, qui nous permet d'exécuter des commandes jQuery multiples, l'un après l'autre, sur le même élément (s).

**Astuce:** De cette façon, les navigateurs ne chercheront pas le même élément (s) plusieurs fois.

Pour la chaîne d'actions, vous ajoutez simplement l'action à l'action précédente.

Les exemples de chaînes ensemble les css (), slideUp (), et slideDown () méthodes suivantes. L'élément "p1" change d'abord rouge, puis il glisse vers le haut, puis il glisse vers le bas:

Exemple

$("#p1").css("color", "red").slideUp(2000).slideDown(2000);

Nous pouvons aussi ajouter plus d'appels de méthode si nécessaire.

**Astuce** : Lors du chaînage, la ligne de code pourrait devenir assez longue. Cependant, jQuery n'est pas très stricte sur la syntaxe; vous pouvez formater comme vous voulez, y compris les sauts de ligne et des indentations.

Cela fonctionne aussi très bien:

Exemple

$("#p1").css("color", "red")  
  .slideUp(2000)  
  .slideDown(2000);

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("#p1").css("color", "red").slideUp(2000).slideDown(2000);

});

});

</script>

</head>

<body>

<p id="p1">jQuery is fun!!</p>

<button>Click me</button>

</body>

</html>

jQuery - Obtenir du contenu et des attributs

jQuery contient des méthodes puissantes pour changer et manipuler les éléments et attributs HTML.

jQuery Manipulation DOM

Une partie très importante de jQuery est la possibilité de manipuler le DOM.

jQuery est livré avec un tas de méthodes DOM connexes qui font qu'il est facile d'accéder et de manipuler les éléments et attributs.

|  |  |
| --- | --- |
| **Remarque** | **DOM = Document Object Model**  Le DOM définit une norme pour l'accès aux documents HTML et XML: *"Le modèle objet de document de W3C (DOM) est une interface de plate - forme et un langage neutre qui permet aux programmes et aux scripts d'accéder dynamiquement et mettre à jour le contenu, la structure et le style d'un document ".* |

Obtenez le Contenu - text (), html (), et val ()

Trois méthodes jQuery simples, mais utiles, pour la manipulation du DOM, qui sont:

* text () - Définit ou retourne le contenu du texte des éléments sélectionnés
* html () - Définit ou retourne le contenu des éléments sélectionnés (y compris la balise HTML)
* val () - Définit ou retourne la valeur des champs de formulaire

L'exemple suivant montre comment obtenir le contenu avec le jQuery méthodes text () et html ():

Exemple

$("#btn1").click(function(){  
    alert("Text: " + $("#test").text());  
});  
$("#btn2").click(function(){  
    alert("HTML: " + $("#test").html());  
});

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#btn1").click(function(){

alert("Text: " + $("#test").text());

});

$("#btn2").click(function(){

alert("HTML: " + $("#test").html());

});

});

</script>

</head>

<body>

<p id="test">This is some <b>bold</b> text in a paragraph.</p>

<button id="btn1">Show Text</button>

<button id="btn2">Show HTML</button>

</body>

</html>

L'exemple suivant montre comment obtenir la valeur d'un champ de saisie avec la méthode jQuery val ():

Exemple

$("#btn1").click(function(){  
    alert("Value: " + $("#test").val());  
});

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

alert("Value: " + $("#test").val());

});

});

</script>

</head>

<body>

<p>Name: <input type="text" id="test" value="Mickey Mouse"></p>

<button>Show Value</button>

</body>

</html>

Obtenez les attributs - attr ()

La méthode jQuery attr () est utilisée pour obtenir les valeurs d'attribut.

L'exemple suivant montre comment obtenir la valeur de l'attribut href dans un lien:

Exemple

$("button").click(function(){  
    alert($("#w3s").attr("href"));  
});

jQuery - Content Set et Attributs

Set contenu - texte (), html (), et val ()

Nous allons utiliser les trois mêmes méthodes de la page précédente pour **définir le contenu** :

* text () - Définit ou retourne le contenu du texte des éléments sélectionnés
* html () - Définit ou retourne le contenu des éléments sélectionnés (y compris la balise HTML)
* val () - Définit ou retourne la valeur des champs de formulaire

L'exemple suivant montre comment définir le contenu avec le jQuery text (), html (), et val () méthodes:

Exemple

$("#btn1").click(function(){  
    $("#test1").text("Hello world!");  
});  
$("#btn2").click(function(){  
    $("#test2").html("<b>Hello world!</b>");  
});  
$("#btn3").click(function(){  
    $("#test3").val("Dolly Duck");  
});

Une fonction de rappel pour text (), html (), et val ()

Toutes les trois méthodes jQuery ci-dessus: text (), html (), et val (), viennent également avec une fonction de rappel. La fonction de rappel a deux paramètres: l'indice de l'élément courant dans la liste des éléments sélectionnés et la valeur d'origine. Vous retournez ensuite la chaîne que vous souhaitez utiliser comme nouvelle valeur de la fonction.

L'exemple suivant illustre text () et html () avec une fonction de rappel:

Exemple

$("#btn1").click(function(){  
    $("#test1").text(function(i, origText){  
        return "Old text: " + origText + " New text: Hello world!  
        (index: " + i + ")";   
    });  
});  
  
$("#btn2").click(function(){  
    $("#test2").html(function(i, origText){  
        return "Old html: " + origText + " New html: Hello <b>world!</b>  
        (index: " + i + ")";   
    });  
});

Set Attributs - attr ()

La méthode jQuery attr () est également utilisée pour définir / modifier les valeurs des attributs.

L'exemple suivant montre comment modifier (set) la valeur de l'attribut href dans un lien:

Exemple

$("button").click(function(){  
    $("#w3s").attr("href", "http://www.w3schools.com/jquery");  
});

La méthode attr () vous permet également de définir plusieurs attributs en même temps.

L'exemple suivant montre comment définir à la fois le href et l'attributs de titre dans en même temps:

Exemple

$("button").click(function(){  
    $("#w3s").attr({  
        "href" : "http://www.w3schools.com/jquery",  
        "title" : "W3Schools jQuery Tutorial"  
    });  
});

Une fonction de rappel pour attr ()

La méthode jQuery attr (), viennent également avec une fonction de rappel. La fonction de rappel a deux paramètres: l'indice de l'élément courant dans la liste des éléments sélectionnés et la valeur l'original d'attribut. Vous retournez ensuite la chaîne que vous souhaitez utiliser comme nouvelle valeur d'attribut de la fonction.

L'exemple suivant montre attr () avec une fonction de rappel:

Exemple

$("button").click(function(){  
    $("#w3s").attr("href", function(i, origValue){  
        return origValue + "/jquery";   
    });  
});

jQuery - Ajouter Elements

Avec jQuery, il est facile d'ajouter de nouveaux éléments / contenu.

Ajouter le nouveau contenu HTML

Nous allons examiner quatre méthodes jQuery qui sont utilisés pour ajouter du nouveau contenu:

* append () - contenu Inseré à la fin des éléments sélectionnés
* prepend () - contenu Inseré au début des éléments sélectionnés
* after () - Insère contenu après les éléments sélectionnés
* before () - contenu Inserts avant les éléments sélectionnés

jQuery append ()

La méthode jQuery append () insère un contenu à la fin de l'élément HTML sélectionnés.

Exemple

$("p").append("Some appended text.");

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#btn1").click(function(){

$("p").append(" <b>Appended text</b>.");

});

$("#btn2").click(function(){

$("ol").append("<li>Appended item</li>");

});

});

</script>

</head>

<body>

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

<ol>

<li>List item 1</li>

<li>List item 2</li>

<li>List item 3</li>

</ol>

<button id="btn1">Append text</button>

<button id="btn2">Append list items</button>

</body>

</html>

jQuery prepend ()

La méthode jQuery prepend () insère un contenu AU DÉBUT des éléments HTML sélectionnés.

Exemple

$("p").prepend("Some prepended text.");

Ajouter plusieurs nouveaux éléments Avec append () et prepend ()

Dans les deux exemples ci-dessus, nous avons seulement inséré un texte / HTML au début / fin des éléments HTML sélectionnés.

Cependant, le append () et prepend () peuvent prendre un nombre infini de nouveaux éléments comme paramètres. Les nouveaux éléments peuvent être générés avec le texte / HTML (comme nous l'avons fait dans les exemples ci-dessus), avec jQuery, ou avec le code JavaScript et des éléments DOM.

Dans l'exemple suivant, nous créons plusieurs nouveaux éléments. Les éléments sont créés avec le texte / HTML, jQuery et JavaScript / DOM. Ensuite, nous ajoutons les nouveaux éléments au texte avec la méthode append () (cela aurait fonctionné pour prepend () aussi):

Exemple

function appendText() {  
    var txt1 = "<p>Text.</p>";               // Create element with HTML    
    var txt2 = $("<p></p>").text("Text.");   // Create with jQuery  
    var txt3 = document.createElement("p");  // Create with DOM  
    txt3.innerHTML = "Text.";  
    $("p").append(txt1, txt2, txt3);         // Append the new elements   
}

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

function appendText() {

var txt1 = "<p>Text.</p>"; // Create text with HTML

var txt2 = $("<p></p>").text("Text."); // Create text with jQuery

var txt3 = document.createElement("p");

txt3.innerHTML = "Text."; // Create text with DOM

$("body").append(txt1, txt2, txt3); // Append new elements

}

</script>

</head>

<body>

<p>This is a paragraph.</p>

<button onclick="appendText()">Append text</button>

</body>

</html>

Méthodes jQuery after () et before ()

La méthode jQuery after () insère un contenu après les éléments HTML sélectionnés.

La méthode jQuery before () insère le contenu avant les éléments HTML sélectionnés.

Exemple

$("img").after("Some text after");  
  
$("img").before("Some text before");

Ajouter plusieurs nouveaux éléments Avec après () et avant ()

En outre, les deux après () et avant () peuvent prendre un nombre infini de nouveaux éléments comme paramètres. Les nouveaux éléments peuvent être générés avec le texte / HTML (comme nous l'avons fait dans l'exemple ci-dessus), avec jQuery, ou avec le code JavaScript et des éléments DOM.

Dans l'exemple suivant, nous créons plusieurs nouveaux éléments. Les éléments sont créés avec le texte / HTML, jQuery et JavaScript / DOM. Ensuite, nous insérons les nouveaux éléments du texte avec la méthode after () (cela aurait fonctionné pour before () aussi):

Exemple

function afterText() {  
    var txt1 = "<b>I </b>";                    // Create element with HTML    
    var txt2 = $("<i></i>").text("love ");     // Create with jQuery  
    var txt3 = document.createElement("b");    // Create with DOM  
    txt3.innerHTML = "jQuery!";  
    $("img").after(txt1, txt2, txt3);          // Insert new elements after <img>  
}

jQuery - Supprimer Elements

Avec jQuery, il est facile de supprimer des éléments HTML existants.

Supprimer Elements / Contenu

Pour supprimer des éléments et le contenu, il existe principalement deux méthodes jQuery:

* remove () - Supprime l'élément sélectionné (et ses éléments enfants)
* empty () - Supprime les éléments enfants de l'élément sélectionné

jQuery remove () Méthode

La méthode jQuery remove () supprime l'élément sélectionné (s) et ses éléments enfants.

Exemple

$("#div1").remove();

jQuery empty () Méthode

La méthode jQuery empty () supprime les éléments enfants de l'élément sélectionné.

Exemple

$("#div1").empty();

Filtrer les éléments à être supprimés

La méthode jQuery remove () accepte également un paramètre qui vous permet de filtrer les éléments à supprimer.

Le paramètre peut être l'une des syntaxes de sélecteur jQuery.

L'exemple suivant supprime tous les éléments <p> avec class = "test":

Exemple

$("p").remove(".test");

Cet exemple supprime tous les éléments <p> avec class = "test" et class = "demo":

Exemple

$("p").remove(".test, .demo");

jQuery - Get et Set Classes CSS

Avec jQuery, il est facile de manipuler des éléments CSS. jQuery a plusieurs méthodes pour la manipulation des CSS. Nous allons examiner les méthodes suivantes:

* addClass () - Ajoute une ou plusieurs classes aux éléments sélectionnés
* removeClass () - Supprime une ou plusieurs classes à partir des éléments sélectionnés
* toggleClass () - Bascule entre l'ajout / suppression des classes à partir des éléments sélectionnés
* css () - Définit ou retourne l'attribut de style

Exemple Stylesheet

La feuille suivante sera utilisée pour tous les exemples sur cette page:

.important {  
    font-weight: bold;  
    font-size: xx-large;  
}  
  
.blue {  
    color: blue;  
}

jQuery addClass () Méthode

L'exemple suivant montre comment ajouter des attributs de classe à différents éléments. Bien sûr, vous pouvez sélectionner plusieurs éléments, lors de l'ajout des classes:

Exemple

$("button").click(function(){  
    $("h1, h2, p").addClass("blue");  
    $("div").addClass("important");  
});

Vous pouvez également spécifier plusieurs classes dans le addClass () méthode:

Exemple

$("button").click(function(){  
    $("#div1").addClass("important blue");  
});

jQuery removeClass () Méthode

L'exemple suivant montre comment supprimer un attribut de classe spécifique à partir de différents éléments:

Exemple

$("button").click(function(){  
    $("h1, h2, p").removeClass("blue");  
});

jQuery toggleClass () Méthode

L'exemple suivant montre comment utiliser la méthode jQuery toggleClass (). Cette méthode permet de basculer entre l'ajout / suppression des classes à partir des éléments sélectionnés:

Exemple

$("button").click(function(){  
    $("h1, h2, p").toggleClass("blue");  
});

jQuery - css () Méthode

La méthode css () définit ou retourne une ou plusieurs propriétés de style pour les éléments sélectionnés.

Retour d'une propriété CSS

Pour avoir la valeur d'une propriété CSS spécifié, utilisez la syntaxe suivante:

css("*propertyname*");

L'exemple suivant retourne la valeur background-color du premier l'élément <p> rencontré:

Exemple

$("p").css("background-color");

Définir une propriété CSS

Pour définir une propriété CSS spécifié, utilisez la syntaxe suivante:

css("*propertyname*","*value*");

L'exemple suivant définit la valeur background-color pour tous les éléments correspondants:

Exemple

$("p").css("background-color", "yellow");

Définir les propriétés CSS multiples

Pour définir plusieurs propriétés CSS, utilisez la syntaxe suivante:

css({"*propertyname*":"*value*","*propertyname*":"*value*",...});

L'exemple suivant établira un background-color et une police de taille pour tous les éléments correspondants:

Exemple

$("p").css({"background-color": "yellow", "font-size": "200%"});

jQuery - Dimensions

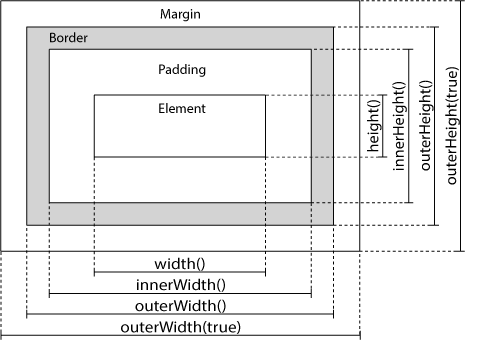
Avec jQuery, il est facile de travailler avec les dimensions des éléments et la fenêtre du navigateur.

jQuery Méthodes Dimension

jQuery a plusieurs méthodes importantes pour travailler avec les dimensions:

* width()
* height()
* innerWidth()
* innerHeight()
* outerWidth()
* outerHeight()

jQuery Dimensions



La width () méthode définit ou retourne la largeur d'un élément (exclut padding, border et marge).

La height ()méthode définit de ou retourne la hauteur d'un élément (exclut padding, border et marge).

L'exemple suivant renvoie la largeur et la hauteur d'un élément <div> spécifiée:

Exemple

$("button").click(function(){  
    var txt = "";  
    txt += "Width: " + $("#div1").width() + "</br>";  
    txt += "Height: " + $("#div1").height();  
    $("#div1").html(txt);  
});

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

var txt = "";

txt += "Width of div: " + $("#div1").width() + "</br>";

txt += "Height of div: " + $("#div1").height();

$("#div1").html(txt);

});

});

</script>

<style>

#div1 {

height: 100px;

width: 300px;

padding: 10px;

margin: 3px;

border: 1px solid blue;

background-color: lightblue;

}

</style>

</head>

<body>

<div id="div1"></div>

<br>

<button>Display dimensions of div</button>

<p>width() - returns the width of an element.</p>

<p>height() - returns the height of an element.</p>

</body>

</html>

jQuery innerWidth () et innerHeight () Méthodes

La méthode innerWidth () retourne la largeur d'un élément (comprend le padding).

La méthode innerHeight () retourne la hauteur d'un élément (comprend le padding).

L'exemple suivant renvoie le inner-width/height  d'un élément <div> spécifiée:

Exemple

$("button").click(function(){  
    var txt = "";  
    txt += "Inner width: " + $("#div1").innerWidth() + "</br>";  
    txt += "Inner height: " + $("#div1").innerHeight();  
    $("#div1").html(txt);  
});

jQuery outerWidth () et outerHeight () Méthodes

La méthode outerWidth () retourne la largeur d'un élément (comprend le padding et la bordure).

La méthode outerHeight () retourne la hauteur d'un élément (comprend le padding et la bordure).

L'exemple suivant retourne la hauteur et la largeur extérieure d'un élément <div> spécifiée:

Exemple

$("button").click(function(){  
    var txt = "";  
    txt += "Outer width: " + $("#div1").outerWidth() + "</br>";  
    txt += "Outer height: " + $("#div1").outerHeight();  
    $("#div1").html(txt);  
});

La méthode outerWidth (true) retourne la largeur d'un élément (comprend le padding, la bordure et la marge).

La méthode outerHeight (true) retourne la hauteur d'un élément (comprend le padding, la bordure et la marge).

Exemple

$("button").click(function(){  
    var txt = "";  
    txt += "Outer width (+margin): " + $("#div1").outerWidth(true) + "</br>";  
    txt += "Outer height (+margin): " + $("#div1").outerHeight(true);  
    $("#div1").html(txt);  
});

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

var txt = "";

txt += "Width of div: " + $("#div1").width() + "</br>";

txt += "Height of div: " + $("#div1").height() + "</br>";

txt += "Outer width of div (margin included): " + $("#div1").outerWidth(true) + "</br>";

txt += "Outer height of div (margin included): " + $("#div1").outerHeight(true);

$("#div1").html(txt);

});

});

</script>

<style>

#div1 {

height: 100px;

width: 300px;

padding: 10px;

margin: 3px;

border: 1px solid blue;

background-color: lightblue;

}

</style>

</head>

<body>

<div id="div1"></div>

<br>

<button>Display dimensions of div</button>

<p>outerWidth(true) - returns the width of an element (includes padding, border, and margin).</p>

<p>outerHeight(true) - returns the height of an element (includes padding, border, and margin).</p>

</body>

</html>

L'exemple suivant renvoie la largeur et la hauteur du document (le document HTML) et la fenêtre (la fenêtre du navigateur):

Exemple

$("button").click(function(){  
    var txt = "";  
    txt += "Document width/height: " + $(document).width();  
    txt += "x" + $(document).height() + "\n";  
    txt += "Window width/height: " + $(window).width();  
    txt += "x" + $(window).height();  
    alert(txt);  
});

L'exemple suivant définit la largeur et la hauteur d'un élément <div> spécifiée:

Exemple

$("button").click(function(){  
    $("#div1").width(500).height(500);  
});