**jQuery**

**Aprende como funciona la etiqueta script y la función "document ready"**

Ahora estamos listos para aprender jQuery, la herramienta de JavaScript más popular de todos los tiempos.

Antes de empezar a usar jQuery, necesitamos agregar algunas cosas a nuestro HTML.

Primero, agrega un elemento scripten la parte superior de tu página. Asegúrese de cerrarlo en la siguiente línea.

Tu navegador ejecutará cualquier JavaScript dentro de un elemento script, incluido jQuery.

Dentro de tu elemento script, agrega este código: $(document).ready(function() {a tu script. Luego ciérralo en la siguiente línea (todavía dentro de tu elemento script) con:});

Aprenderemos más sobre functionsmás tarde. Lo importante a saber es que el código que pongas dentro de esta function(función) se ejecutará tan pronto como tu navegador haya cargado tu página.

Esto es importante porque sin tu document ready function, tu código podría ejecutarse antes de que se procese tu HTML, lo que causaría errores.

**<script>**

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

**});**

**</script>**

**Elementos HTML de destino con selectores usando jQuery**

Ahora tenemos una document readyfunción.

Ahora escribamos nuestra primera instrucción jQuery. Todas las funciones de jQuery comienzan con un $, generalmente denominado operador de signo de dólar o bling.

jQuery a menudo selecciona un elemento HTML con un selector y luego hace algo con ese elemento.

Por ejemplo, hagamos que todos sus buttonelementos reboten. Simplemente agregue este código dentro de su función lista para documentos:

**$("button").addClass("animated bounce");**

Tenga en cuenta que ya hemos incluido la biblioteca jQuery y la biblioteca Animate.css en segundo plano para que pueda usarlas en el editor. Entonces, está utilizando jQuery para aplicar la bounceclase Animate.css a sus buttonelementos.

**<script>**

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

**$("button").addClass("animated bounce");**

**});**

**</script>**

**Elementos de destino por clase usando jQuery**

¿Ves cómo hicimos que todos tus buttonelementos reboten? Los seleccionamos con $("button"), luego les agregamos algunas clases de CSS con .addClass("animated bounce");.

Acaba de usar .addClass()la función de jQuery, que le permite agregar clases a los elementos.

Primero, apuntemos a sus divelementos con la clase wellusando el $(".well")selector.

Tenga en cuenta que, al igual que con las declaraciones CSS, escribe a .antes del nombre de la clase.

Luego use .addClass()la función de jQuery para agregar las clases animatedy shake.

Por ejemplo, podría hacer que todos los elementos con la clase text-primarytiemblen agregando lo siguiente a su document **ready function:**

**$(".text-primary").addClass("animated shake");**

**<script>**

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

**$(".well").addClass("animated bounce");**

**$(".well").addClass("shake");**

**});**

**</script>**

**Elementos de destino por id usando jQuery**

También puede orientar elementos por sus atributos de identificación.

Primero apunte a su buttonelemento con la identificación target3usando el $("#target3")selector.

Tenga en cuenta que, al igual que con las declaraciones CSS, escribe a #antes del nombre de la identificación.

Luego use .addClass()la función de jQuery para agregar las clases animatedy fadeOut.

Así es como haría que el buttonelemento con la identificación target6se desvaneciera:

**$("#target6").addClass("animated fadeOut");**

**Solution**

**$("#target3").addClass("fadeOut");**

**Eliminar sus funciones jQuery**

Estas animaciones eran geniales al principio, pero ahora empiezan a distraer un poco.

Elimine estas tres funciones de jQuery de su archivo document ready function, pero déjelo document ready functionintacto.

**<script>**

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

**});**

**</script>**

**Apunte al mismo elemento con múltiples selectores de jQuery**

Ahora conoce tres formas de segmentar elementos: por tipo: $("button"), por clase: $(".btn")y por id $("#target1").

Aunque es posible agregar varias clases en una sola .addClass()llamada, agréguelas al mismo elemento de tres formas distintas .

Usando .addClass(), agregue solo una clase a la vez al mismo elemento, de tres maneras diferentes:

Agrega la animatedclase a todos los elementos con tipo button.

Agregue la shakeclase a todos los botones con clase .btn.

Agregue la btn-primaryclase al botón con id #target1.

Nota: solo debe apuntar a un elemento y agregar solo una clase a la vez. En conjunto, sus tres selectores individuales terminarán agregando las tres clases shake, animatedy btn-primarya #target1.

**<script>**

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

**$("button").addClass("animated");**

**$(".btn").addClass("shake");**

**$("#target1").addClass("btn-primary");**

**});**

**Eliminar clases de un elemento con jQuery**

De la misma manera que puede agregar clases a un elemento con addClass()la función de jQuery, puede eliminarlas con removeClass()la función de jQuery.

Así es como haría esto para un botón específico:

**$("#target2").removeClass("btn-default");**

Eliminemos la btn-defaultclase de todos nuestros buttonelementos.

**$("button").removeClass("btn-default");**

**Cambiar el CSS de un elemento usando jQuery**

**También podemos cambiar el CSS de un elemento HTML directamente con jQuery.**

**jQuery tiene una función llamada .css()que te permite cambiar el CSS de un elemento.**

Así es como cambiaríamos su color a azul:

**$("#target1").css("color", "blue");**

Esto es ligeramente diferente de una declaración CSS normal, porque la propiedad CSS y su valor están entre comillas y separados por una coma en lugar de dos puntos.

Elimine sus selectores de jQuery, dejando un archivo document ready function.

Seleccione target1y cambie su color a rojo.

**$("#target1").css("color", "red");**

**Deshabilitar un elemento usando jQuery**

También puede cambiar las propiedades que no son CSS de los elementos HTML con jQuery. Por ejemplo, puede deshabilitar botones.

Cuando deshabilita un botón, se atenuará y ya no se podrá hacer clic en él.

jQuery tiene una función llamada .prop()que te permite ajustar las propiedades de los elementos.

Así es como deshabilitaría todos los botones:

**$("button").prop("disabled", true);**

Deshabilite solo el target1botón.

**$("#target1").prop("disabled", true);**

**Cambiar texto dentro de un elemento usando jQuery**

Usando jQuery, puede cambiar el texto entre las etiquetas de inicio y final de un elemento. Incluso puede cambiar el marcado HTML.

jQuery tiene una función llamada .html()que le permite agregar etiquetas HTML y texto dentro de un elemento. Cualquier contenido previamente dentro del elemento será reemplazado por completo con el contenido que proporcione utilizando esta función.

Así es como reescribiría y enfatizaría el texto de nuestro encabezado:

**$("h3").html("<em>jQuery Playground</em>");**

jQuery también tiene una función similar llamada .text()que solo modifica el texto sin agregar etiquetas. En otras palabras, esta función no evaluará ninguna etiqueta HTML que se le haya pasado, sino que la tratará como el texto con el que desea reemplazar el contenido existente.

Cambie el botón con id target4enfatizando su texto.

Vea nuestro artículo de noticias para <em> para conocer la diferencia entre <i>y <em>y sus usos.

Tenga en cuenta que, si bien la <i>etiqueta se ha utilizado tradicionalmente para enfatizar texto, desde entonces se ha adoptado como etiqueta para iconos. La <em>etiqueta ahora es ampliamente aceptada como la etiqueta para enfatizar. Cualquiera de los dos funcionará para este desafío.

**$("#target4").html("<em>#target4</em>");**

**Eliminar un elemento usando jQuery**

Ahora eliminemos un elemento HTML de su página usando jQuery.

jQuery tiene una función llamada .remove()que eliminará un elemento HTML por completo.

Elimina el #target4elemento de la página usando la .remove()función.

**$("#target4").remove();**

**Use appendTo para mover elementos con jQuery**

Ahora intentemos mover elementos de uno diva otro.

jQuery tiene una función llamada appendTo()que le permite seleccionar elementos HTML y agregarlos a otro elemento.

Por ejemplo, si quisiéramos movernos target4de nuestro pozo derecho a nuestro pozo izquierdo, usaríamos:

$("#target4").appendTo("#left-well");

Mueve tu target2elemento de tu left-wella tu right-well.

**$("#target2").appendTo("#right-well");**

**Clonar un elemento usando jQuery**

Además de mover elementos, también puedes copiarlos de un lugar a otro.

jQuery tiene una función llamada clone()que hace una copia de un elemento.

Por ejemplo, si quisiéramos copiar target2de nuestro left-wella nuestro right-well, usaríamos:

**$("#target2").clone().appendTo("#right-well");**

¿Notaste que esto implica unir dos funciones jQuery? Esto se llama encadenamiento de funciones y es una forma conveniente de hacer las cosas con jQuery.

Clone su target5elemento y agréguelo a su archivo left-well.

**$("#target5").clone().appendTo("#left-well");**

**Diríjase al padre de un elemento usando jQuery**

Cada elemento HTML tiene un parentelemento a partir del cual inheritspropiedades.

Por ejemplo, el h3elemento en your jQuery Playgroundtiene el elemento principal de <div class="container-fluid">, que a su vez tiene el elemento principal de body.

jQuery tiene una función llamada parent()que le permite acceder al padre de cualquier elemento que haya seleccionado.

Aquí hay un ejemplo de cómo usaría la parent()función si quisiera darle al elemento principal del left-wellelemento un color de fondo azul:

**$("#left-well").parent().css("background-color", "blue")**

Dale al padre del #target1elemento un color de fondo rojo.

**$("#target1").parent().css("background-color", "red");**

**Apuntar a los hijos de un elemento usando jQuery**

Cuando los elementos HTML se colocan un nivel debajo de otro, se denominan elementos secundarios de ese elemento. Por ejemplo, los elementos de botón en este desafío con el texto #target1, #target2y #target3son todos elementos secundarios del <div class="well" id="left-well">elemento.

jQuery tiene una función llamada children()que le permite acceder a los elementos secundarios de cualquier elemento que haya seleccionado.

Aquí hay un ejemplo de cómo usaría la children()función para dar a los niños de su left-wellelemento el color blue:

**$("#left-well").children().css("color", "blue")**

**$("#right-well").children().css("color", "orange")**

**Dirigirse a un elemento secundario específico de un elemento mediante jQuery**

Ha visto por qué los atributos de identificación son tan convenientes para la orientación con selectores de jQuery. Pero no siempre tendrás identificaciones tan ordenadas con las que trabajar.

Afortunadamente, jQuery tiene otros trucos para apuntar a los elementos correctos.

jQuery usa selectores de CSS para seleccionar elementos. El target:nth-child(n)selector de CSS le permite seleccionar todos los elementos n con la clase de destino o el tipo de elemento.

Así es como le daría al tercer elemento en cada pocillo la clase de rebote:

**$(".target:nth-child(3)").addClass("animated bounce");**

Haz rebotar al segundo niño en cada uno de tus elementos de pozo. Debes seleccionar los hijos de los elementos con la targetclase.

**$(".target:nth-child(2)").addClass("animated bounce");**

**Apunta a elementos pares usando jQuery**

También puede orientar elementos en función de sus posiciones mediante los selectores :oddo :even.

Tenga en cuenta que jQuery está indexado a cero, lo que significa que el primer elemento en una selección tiene una posición de 0. Esto puede ser un poco confuso ya que, de manera contraria a la intuición, :oddselecciona el segundo elemento (posición 1), el cuarto elemento (posición 3) y pronto.

Así es como apuntaría a todos los elementos impares con clase targety les daría clases:

**$(".target:odd").addClass("animated shake");**

Prueba a seleccionar todos los targetelementos pares y dales las clases de animatedy shake. Recuerde que incluso se refiere a la posición de los elementos con un sistema de base cero en mente.

**$(".target:even").addClass("animated shake");**

**$("body").addClass("animated hinge");**