# JQuery:

1.jQuery is a JavaScript Library.

2.jQuery greatly simplifies JavaScript programming.

3.jQuery is easy to learn.

# What is JQuery:

1.jQuery is a lightweight, "write less, do more", JavaScript library.

2.The purpose of jQuery is to make it much easier to use JavaScript on your website.

3.jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

4.jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

5.The jQuery library contains the following features:

HTML/DOM manipulation

CSS manipulation

HTML event methods

Effects and animations

AJAX

Utilities

Tip: In addition, jQuery has plugins for almost any task out there.

# Why jQuery:

1.There are lots of other JavaScript frameworks out there, but jQuery seems to be the most popular, and also the most extendable.

2.Many of the biggest companies on the Web use jQuery, such as:

Google

Microsoft

IBM

Netflix

# 1.Program to disappear the lines:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js "></script>

<script>

$(document).ready(function(){

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

$(this).hide();

});

});

</script>

</head>

<body>

<p>If you click on me, I will disappear.</p>

<p>Click me away!</p>

<p>Click me too!</p>

</body>

</html>

# 2.Program to hide the lines:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

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

$("p").hide();

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

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

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

<button>Click me</button>

</body>

</html>

# Adding jQuery to Your Web Pages

There are several ways to start using jQuery on your web site. You can:

1.Download the jQuery library from jQuery.com

2.Include jQuery from a CDN, like Google

# Downloading jQuery

There are two versions of jQuery available for downloading:

1.Production version - this is for your live website because it has been minified and compressed

2.Development version - this is for testing and development (uncompressed and readable code)

Both versions can be downloaded from jQuery.com.

The jQuery library is a single JavaScript file, and you reference it with the HTML <script> tag (notice that the <script> tag should be inside the <head> section):

<head>

<script src="jquery-1.12.4.min.js"></script>

</head>

# jQuery Syntax

With jQuery you select (query) HTML elements and perform "actions" on them.

# jQuery Syntax

The jQuery syntax is tailor-made for selecting HTML elements and performing some action on the element(s).

Basic syntax is:

$(selector).action()

1.A $ sign to define/access jQuery

2.A (selector) to "query (or find)" HTML elements

3.A jQuery action() to be performed on the element(s)

Examples:

1.$(this).hide() - hides the current element.

2.$("p").hide() - hides all <p> elements.

3.$(".test").hide() - hides all elements with class="test".

4.$("#test").hide() - hides the element with id="test".

# The Document Ready Event

You might have noticed that all jQuery methods in our examples, are inside a document ready event:

$(document).ready(function()

{

// jQuery methods go here...

});

This is to prevent any jQuery code from running before the document is finished loading (is ready).

It is good practice to wait for the document to be fully loaded and ready before working with it. This also allows you to have your JavaScript code before the body of your document, in the head section.

Here are some examples of actions that can fail if methods are run before the document is fully loaded:

1.Trying to hide an element that is not created yet

2.Trying to get the size of an image that is not loaded yet

Tip: The jQuery team has also created an even shorter method for the document ready event:

$(function()

{

// jQuery methods go here...

});

# jQuery Selectors

jQuery selectors are one of the most important parts of the jQuery library.

# jQuery Selectors

jQuery selectors allow you to select and manipulate HTML element(s).

jQuery selectors are used to "find" (or select) HTML elements based on their name, id, classes, types, attributes, values of attributes and much more. It's based on the existing CSS Selectors, and in addition, it has some own custom selectors.

All selectors in jQuery start with the dollar sign and parentheses: $().

# The element Selector

The jQuery element selector selects elements based on the element name.

You can select all <p> elements on a page like this:

$("p")

# 3.Example:

When a user clicks on a button, all <p> elements will be hidden:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js "></script>

<script>

$(document).ready(function(){

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

$("p").hide();

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

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

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

<button>Click me to hide paragraphs</button>

</body>

</html>

# The #id Selector

The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.

An id should be unique within a page, so you should use the #id selector when you want to find a single, unique element.

To find an element with a specific id, write a hash character, followed by the id of the HTML element:

$("#test")

# 4.Example:

When a user clicks on a button, the element with id="test" will be hidden:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js "></script>

<script>

$(document).ready(function(){

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

$("#test").hide();

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

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

<p id="test">This is another paragraph.</p>

<button>Click me</button>

</body>

</html>

# The .class Selector

The jQuery class selector finds elements with a specific class.

To find elements with a specific class, write a period character, followed by the name of the class:

$(".test")

# 5.Example

When a user clicks on a button, the elements with class="test" will be hidden:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js "></script>

<script>

$(document).ready(function(){

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

$(".test").hide();

});

});

</script>

</head>

<body>

<h2 class="test">This is a heading</h2>

<p class="test">This is a paragraph.</p>

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

<button>Click me</button>

</body>

</html>

# Syntax:

# 6.$(“\*”):

# It is used to selects the all elements.

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

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

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

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

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

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

<button>Click me</button>

</body>

</html>

# 7. $("p.intro"):

# Selects all <p> elements with class="intro"

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

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

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

});

});

</script>

</head>

<body>

<h2 class="intro">This is a heading</h2>

<p class="intro">This is a paragraph.</p>

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

<button>Click me</button>

</body>

</html>

# 8.$(“p:first”):

# Selects the first <p> element

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

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

$("p:first").hide();

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

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

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

<button>Click me</button>

</body>

</html>

# 9. $("ul li:first"):

# Selects the first <li> element of the first <ul>

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin. js"></script>

<script>

$(document).ready(function(){

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

$("ul li:first").hide();

});

});

</script>

</head>

<body>

<p>List 1:</p>

<ul>

<li>Coffee</li>

<li>Milk</li>

<li>Tea</li>

</ul>

<p>List 2:</p>

<ul>

<li>Coffee</li>

<li>Milk</li>

<li>Tea</li>

</ul>

<button>Click me</button>

</body>

</html>

# 10. $("ul li:first-child"):

# Selects the first <li> element of every <ul>

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

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

$("ul li:first-child").hide();

});

});

</script>

</head>

<body>

<p>List 1:</p>

<ul>

<li>Coffee</li>

<li>Milk</li>

<li>Tea</li>

</ul>

<p>List 2:</p>

<ul>

<li>Coffee</li>

<li>Milk</li>

<li>Tea</li>

</ul>

<button>Click me</button>

</body>

</html>

# 11. $("[href]"):

# Selects all elements with an href attribute

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin. js"></script>

<script>

$(document).ready(function(){

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

$("[href]").hide();

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

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

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

<p><a href="http://www.w3schools.com/html/">HTML Tutorial</a></p>

<p><a href="http://www.w3schools.com/css/">CSS Tutorial</a></p>

<button>Click me</button>

</body>

</html>

# 12. $("a[target='\_blank']"):

# Selects all <a> elements with a target attribute value equal to "\_blank"

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

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

$("a[target='\_blank']").hide();

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

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

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

<p><a href="http://www.w3schools.com/html/" target="\_blank">HTML Tutorial</a></p>

<p><a href="http://www.w3schools.com/css/">CSS Tutorial</a></p>

<button>Click me</button>

</body>

</html>

# 13. $("a[target!='\_blank']")

# Selects all <a> elements with a target attribute value NOT equal to "\_blank"

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

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

$("a[target!='\_blank']").hide();

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

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

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

<p><a href="http://www.w3schools.com/html/" target="\_blank">HTML Tutorial</a></p>

<p><a href="http://www.w3schools.com/css/">CSS Tutorial</a></p>

<button>Click me</button>

</body>

</html>

# 14. $(":button"):

# Selects all <button> elements and <input> elements of type="button"

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

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

$(":button").hide();

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

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

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

<button>Click me</button>

</body>

</html>

# 15. $("tr:even"):

# Selects all even <tr> elements

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

$("tr:even").css("background-color", "yellow");

});

</script>

</head>

<body>

<h1>Welcome to My Web Page</h1>

<table border="1">

<tr>

<th>Company</th>

<th>Country</th>

</tr>

<tr>

<td>Alfreds Futterkiste</td>

<td>Germany</td>

</tr>

<tr>

<td>Berglunds snabbköp</td>

<td>Sweden</td>

</tr>

<tr>

<td>Centro comercial Moctezuma</td>

<td>Mexico</td>

</tr>

<tr>

<td>Ernst Handel</td>

<td>Austria</td>

</tr>

<tr>

<td>Island Trading</td>

<td>UK</td>

</tr>

</table>

</body>

</html>

# 16. $("tr:odd"):

# Selects all odd <tr> elements

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

$("tr:odd").css("background-color", "yellow");

});

</script>

</head>

<body>

<h1>Welcome to My Web Page</h1>

<table border="1">

<tr>

<th>Company</th>

<th>Country</th>

</tr>

<tr>

<td>Alfreds Futterkiste</td>

<td>Germany</td>

</tr>

<tr>

<td>Berglunds snabbköp</td>

<td>Sweden</td>

</tr>

<tr>

<td>Centro comercial Moctezuma</td>

<td>Mexico</td>

</tr>

<tr>

<td>Ernst Handel</td>

<td>Austria</td>

</tr>

<tr>

<td>Island Trading</td>

<td>UK</td>

</tr>

</table>

</body>

</html>

# Functions In a Separate File

If your website contains a lot of pages, and you want your jQuery functions to be easy to maintain, you can put your jQuery functions in a separate .js file.

When we demonstrate jQuery in this tutorial, the functions are added directly into the <head> section. However, sometimes it is preferable to place them in a separate file, like this (use the src attribute to refer to the .js file):

# Example

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js">

</script>

<script src="my\_jquery\_functions.js"></script>

</head>

# 17.Example:

# Use the correct selector to hide all <p> elements.

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

$("selector").hide();

});

</script>

</head>

<body>

<h2>This is a heading</h2>

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

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

</body>

</html>

# And

<!DOCTYPE html>

<html>

<head>

<script src="Plugin.js"></script>

<script>

$(document).ready(function(){

$("p").hide();

});

</script>

</head>

<body>

<h2>This is a heading</h2>

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

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

</body>

</html>

# 18.Example

# Use the correct selector to hide the element with id="test".

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

$("selector").hide();

});

</script>

</head>

<body>

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

<p id="test">This is a paragraph with id="test".</p>

</body>

</html>

# And

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

$("#test").hide();

});

</script>

</head>

<body>

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

<p id="test">This is a paragraph with id="test".</p>

</body>

</html>

# jQuery Event Methods

jQuery is tailor-made to respond to events in an HTML page.

# What are Events?

All the different visitor's actions that a web page can respond to are called events.

An event represents the precise moment when something happens.

# Examples:

1.moving a mouse over an element

2.selecting a radio button

3.clicking on an element

The term "fires/fired" is often used with events. Example: "The keypress event is fired, the moment you press a key".

Here are some common DOM events:

Mouse Events Keyboard Events Form Events Document/Window Events

click keypress submit load

dblclick keydown change resize

mouseenter keyup focus scroll

mouseleave blur unload

# jQuery Syntax For Event Methods

In jQuery, most DOM events have an equivalent jQuery method.

To assign a click event to all paragraphs on a page, you can do this:

$("p").click();

The next step is to define what should happen when the event fires. You must pass a function to the event:

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

// action goes here!!

});

# Commonly Used jQuery Event Methods

$(document).ready()

The $(document).ready() method allows us to execute a function when the document is fully loaded. This event is already explained in the jQuery Syntax chapter.

# click()

The click() method attaches an event handler function to an HTML element.

The function is executed when the user clicks on the HTML element.

The following example says: When a click event fires on a <p> element; hide the current <p> element:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src=”./Plugin.js”></script>

<script>

$(document).ready(function(){

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

$(this).hide();

});

});

</script>

</head>

<body>

<p>If you click on me, I will disappear.</p>

<p>Click me away!</p>

<p>Click me too!</p>

</body>

</html>

# dblclick()

The dblclick() method attaches an event handler function to an HTML element.

The function is executed when the user double-clicks on the HTML element:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

$("p").dblclick(function(){

$(this).hide();

});

});

</script>

</head>

<body>

<p>If you double-click on me, I will disappear.</p>

<p>Click me away!</p>

<p>Click me too!</p>

</body>

</html>

# mouseenter()

The mouseenter() method attaches an event handler function to an HTML element.

The function is executed when the mouse pointer enters the HTML element:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

$("#p1").mouseenter(function(){

alert("You entered p1!");

});

});

</script>

</head>

<body>

<p id="p1">Enter this paragraph.</p>

</body>

</html>

# mouseleave()

The mouseleave() method attaches an event handler function to an HTML element.

The function is executed when the mouse pointer leaves the HTML element:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

$("#p1").mouseleave(function(){

alert("Bye! You now leave p1!");

});

});

</script>

</head>

<body>

<p id="p1">This is a paragraph.</p>

</body>

</html>

# mousedown()

The mousedown() method attaches an event handler function to an HTML element.

The function is executed, when the left, middle or right mouse button is pressed down, while the mouse is over the HTML element:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

$("#p1").mousedown(function(){

alert("Mouse down over p1!");

});

});

</script>

</head>

<body>

<p id="p1">This is a paragraph.</p>

</body>

</html>

# mouseup()

The mouseup() method attaches an event handler function to an HTML element.

The function is executed, when the left, middle or right mouse button is released, while the mouse is over the HTML element:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

$("#p1").mouseup(function(){

alert("Mouse up over p1!");

});

});

</script>

</head>

<body>

<p id="p1">This is a paragraph.</p>

</body>

</html>

# hover()

The hover() method takes two functions and is a combination of the mouseenter() and mouseleave() methods.

The first function is executed when the mouse enters the HTML element, and the second function is executed when the mouse leaves the HTML element:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

$("#p1").hover(function(){

alert("You entered p1!");

},

function(){

alert("Bye! You now leave p1!");

});

});

</script>

</head>

<body>

<p id="p1">This is a paragraph.</p>

</body>

</html>

# focus()

The focus() method attaches an event handler function to an HTML form field.

The function is executed when the form field gets focus:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

$("input").focus(function(){

$(this).css("background-color", "#cccccc");

});

$("input").blur(function(){

$(this).css("background-color", "#ffffff");

});

});

</script>

</head>

<body>

Name: <input type="text" name="fullname"><br>

Email: <input type="text" name="email">

</body>

</html>

# blur()

The blur() method attaches an event handler function to an HTML form field.

The function is executed when the form field loses focus:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

$("input").focus(function(){

$(this).css("background-color", "#cccccc");

});

$("input").blur(function(){

$(this).css("background-color", "#ffffff");

});

});

</script>

</head>

<body>

Name: <input type="text" name="fullname"><br>

Email: <input type="text" name="email">

</body>

</html>

# The on() Method

The on() method attaches one or more event handlers for the selected elements.

Attach a click event to a <p> element:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

$("p").on("click", function(){

$(this).hide();

});

});

</script>

</head>

<body>

<p>If you click on me, I will disappear.</p>

<p>Click me away!</p>

<p>Click me too!</p>

</body>

</html>

# Exmple:

Attach multiple event handlers to a <p> element:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

$("p").on({

mouseenter: function(){

$(this).css("background-color", "lightgray");

},

mouseleave: function(){

$(this).css("background-color", "lightblue");

},

click: function(){

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

}

});

});

</script>

</head>

<body>

<p>Click or move the mouse pointer over this paragraph.</p>

</body>

</html>

# jQuery Effects - Hide and Show

Hide, Show, Toggle, Slide, Fade, and Animate. WOW!

# Click to show/hide panel

Examples

# jQuery hide()

Demonstrates a simple jQuery hide() method.

# jQuery hide()

Another hide() demonstration. How to hide parts of text.

# jQuery hide() and show()

With jQuery, you can hide and show HTML elements with the hide() and show() methods:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="Plugin.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>

# Syntax:

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

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

The optional speed parameter specifies the speed of the hiding/showing, and can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the hide() or show() method completes (you will learn more about callback functions in a later chapter).

The following example demonstrates the speed parameter with hide():

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="Plugin.js"></script>

<script>

$(document).ready(function(){

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

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

});

});

</script>

</head>

<body>

<button>Hide</button>

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

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

</body>

</html>

# jQuery toggle()

With jQuery, you can toggle between the hide() and show() methods with the toggle() method.

Shown elements are hidden and hidden elements are shown:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="Plugin.js"></script>

<script>

$(document).ready(function(){

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

$("p").toggle();

});

});

</script>

</head>

<body>

<button>Toggle between hiding and showing the paragraphs</button>

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

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

</body>

</html>

# Syntax:

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

The optional speed parameter can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after toggle() completes.

# jQuery Effects - Fading

With jQuery you can fade elements in and out of visibility.

Click to fade in/out panel

# Examples

# jQuery fadeIn()

Demonstrates the jQuery fadeIn() method.

# jQuery fadeOut()

Demonstrates the jQuery fadeOut() method.

# jQuery fadeToggle()

Demonstrates the jQuery fadeToggle() method.

# jQuery fadeTo()

Demonstrates the jQuery fadeTo() method.

# jQuery Fading Methods

With jQuery you can fade an element in and out of visibility.

jQuery has the following fade methods:

1.fadeIn()

2.fadeOut()

3.fadeToggle()

4.fadeTo()

# jQuery fadeIn() Method

The jQuery fadeIn() method is used to fade in a hidden element.

Syntax:

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

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the fading completes.

The following example demonstrates the fadeIn() method with different parameters:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="Plugin.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>

# jQuery fadeOut() Method

The jQuery fadeOut() method is used to fade out a visible element.

# Syntax:

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

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the fading completes.

The following example demonstrates the fadeOut() method with different parameters:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="Plugin.js"></script>

<script>

$(document).ready(function(){

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

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

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

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

});

});

</script>

</head>

<body>

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

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

<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>

<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>

<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>

</body>

</html>

# jQuery fadeToggle() Method

The jQuery fadeToggle() method toggles between the fadeIn() and fadeOut() methods.

If the elements are faded out, fadeToggle() will fade them in.

If the elements are faded in, fadeToggle() will fade them out.

# Syntax:

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

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the fading completes.

The following example demonstrates the fadeToggle() method with different parameters:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="Plugin.js"></script>

<script>

$(document).ready(function(){

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

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

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

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

});

});

</script>

</head>

<body>

<p>Demonstrate fadeToggle() with different speed parameters.</p>

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

<div id="div1" style="width:80px;height:80px;background-color:red;"></div>

<br>

<div id="div2" style="width:80px;height:80px;background-color:green;"></div>

<br>

<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>

</body>

</html>

# jQuery fadeTo() Method

The jQuery fadeTo() method allows fading to a given opacity (value between 0 and 1).

# Syntax:

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

The required speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The required opacity parameter in the fadeTo() method specifies fading to a given opacity (value between 0 and 1).

The optional callback parameter is a function to be executed after the function completes.

The following example demonstrates the fadeTo() method with different parameters:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

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

$("#div1").fadeTo("slow", 0.15);

$("#div2").fadeTo("slow", 0.4);

$("#div3").fadeTo("slow", 0.7);

});

});

</script>

</head>

<body>

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

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

<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>

<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>

<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>

</body>

</html>

# jQuery Effects - Sliding

The jQuery slide methods slide elements up and down.

# Examples:

# jQuery slideDown()

Demonstrates the jQuery slideDown() method.

# jQuery slideUp()

Demonstrates the jQuery slideUp() method.

# jQuery slideToggle()

Demonstrates the jQuery SlideToggle() method.

# jQuery Sliding Methods

With jQuery you can create a sliding effect on elements.

# jQuery has the following slide methods:

1.slideDown()

2.slideUp()

3.slideToggle()

# jQuery slideDown() Method

The jQuery slideDown() method is used to slide down an element.

# Syntax:

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

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the sliding completes.

The following example demonstrates the slideDown() method:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.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() Method

The jQuery slideUp() method is used to slide up an element.

# Syntax:

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

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the sliding completes.

The following example demonstrates the slideUp() method:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.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>

# jQuery slideToggle() Method

The jQuery slideToggle() method toggles between the slideDown() and slideUp() methods.

If the elements have been slid down, slideToggle() will slide them up.

If the elements have been slid up, slideToggle() will slide them down.

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

The optional speed parameter can take the following values: "slow", "fast", milliseconds.

The optional callback parameter is a function to be executed after the sliding completes.

The following example demonstrates the slideToggle() method:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.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>

# jQuery Effects - Animation

The jQuery animate() method lets you create custom animations.

# jQuery Animations - The animate() Method

The jQuery animate() method is used to create custom animations.

# Syntax:

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

The required params parameter defines the CSS properties to be animated.

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the animation completes.

The following example demonstrates a simple use of the animate() method; it moves a <div> element to the right, until it has reached a left property of 250px:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.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() - Manipulate Multiple Properties

Notice that multiple properties can be animated at the same time:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

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

$("div").animate({

left: '250px',

opacity: '0.5',

height: '150px',

width: '150px'

});

});

});

</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() - Using Relative Values

It is also possible to define relative values (the value is then relative to the element's current value). This is done by putting += or -= in front of the value:

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

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

$("div").animate({

left: '250px',

height: '+=150px',

width: '+=150px'

});

});

});

</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() - Using Pre-defined Values

You can even specify a property's animation value as "show", "hide", or "toggle":

# Example:

<!DOCTYPE html>

<html>

<head>

<script src="/.Plugin.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() - Uses Queue Functionality

By default, jQuery comes with queue functionality for animations.

This means that if you write multiple animate() calls after each other, jQuery creates an "internal" queue with these method calls. Then it runs the animate calls ONE by ONE.

So, if you want to perform different animations after each other, we take advantage of the queue functionality:

# Example 1:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.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>

# The example below first moves the <div> element to the right, and then increases the font size of the text:

# Example 2:

<!DOCTYPE html>

<html>

<head>

<script src="./Plugin.js"></script>

<script>

$(document).ready(function(){

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

var div = $("div");

div.animate({left: '100px'}, "slow");

div.animate({fontSize: '3em'}, "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:200px;position:absolute;">HELLO</div>

</body>

</html>