

Learn about how jQuery works

In the article below, we will guide and introduce you to some basic concepts about jQuery - designed to change the way you write JavaScript code - Write short, do a lot! That is the slogan of jQuery ...

TipsMake.com - In the article below, we will guide and introduce you to some basic concepts about jQuery - designed to change the way you write JavaScript code - Write short, do it a lot of! That is the slogan of jQuery.

In essence, **jQuery** is a **JavaScript** library, also known as **JavaScript Framework** , that supports you in handling **HTML** , handling events in **HTML** pages, creating beautiful effects, faster and shorter **Ajax** processing. for user website application.

1. Basic:

The following test is built to help you begin using **jQuery** . If you don't have a dedicated test page, start by creating a new **HTML** page with the source code as shown below:

Demo

jQuery

Change the src attribute in the script tag to point to your **jquery.js** file. For example, if the file **jquery.js** is in the same directory as the HTML file, we can use:

The latest jQuery version is downloaded here.

2. Match code on the available page:

One of the first things most Javascript programmers want to do is to pair code to their main application or program. For example:

```
window.onload = function () {alert ("welcome"); }
```

Inside that code is what we want the browser to work when the website is loaded. But the problem here is that the **Javascript** code will not run until all the image files have been downloaded to the device (including banner ads). The reason for using `window.onload` is because the HTML documents have not been fully loaded the first time.

To solve this problem, **jQuery** already has a clause that can check document variables and wait until it is fixed, which is the event ready:

```
$ (document) .ready (function () {  
  // Your code here  
});
```

Within that event ready, we insert the **handler** event to **click** on the path:

```
$ (document) .ready (function () {  
  $ ("a"). click (function (event) {  
    alert ("Thanks for visiting!");  
  });  
});
```

Save the HTML file and reload this test page, clicking on the link on that page will display a warning window as a pop up before moving to the main jQuery page. For clicks as well as many other events, we can block the default actions here, as in the example above, then click on the link **jquery.com** by calling **event.preventDefault ()** in the **handler event**:

```
$ (document) .ready (function () {  
  $ ("a"). click (function (event) {  
    alert ("As you can see, the link no longer takes you to jquery.com");  
    event.preventDefault ();  
  });  
});
```

3. Complete example:

Here is a complete example of the HTML file based on the above code. Note that the link here will connect to Google's CDN to download the original jQuery file. Besides, our custom script is included in the tag, it is more appropriate to place it in a separate file, then pass the parameter to that file via the element's corresponding `src` attribute:

jQuery demo

jQuery

4. Delete or add HTML class:

Note that some of the example **jQuery** code below needs to be placed inside the **event ready** to be executed when the website is ready to work. One of the common tasks is to add or remove classes.

First, try assigning some style information to the page's tag, such as:

Next, add `addClass` to the code:

```
$("#a").addClass("test");
```

All components will be bolded. If you want to remove the class, use `removeClass`:

```
$("#a").removeClass("test");
```

HTML allows assigning multiple classes to an element.

5. Some special effects:

In jQuery there are already quite a few special effects to make your site special. To insert our test page, change the click event that was previously assigned:

```
$("#a").click(function(event){
event.preventDefault();
$(this).hide("slow");
});
```

Now, when clicking on any link, the site will slowly disappear.

6. Callback and Functions:

In fact, the **callback** is a function that converts arguments to another function and is executed after the **function parent** completes. A special feature of the callback is that the function will be displayed immediately after the **function** at the **parent** level can be executed before the callback action occurs. Another important point to know is how the **callback** is 'called' at the right time.

The non- **argument** callback has the following general syntax:

```
$.get('myhtmlpage.html', myCallback);
```

Note that the second parameter here is simply the name of the function. **Javascript functions** are **First class citizens** and can be used as custom variables, executed at later times.

Callback with argument: the question here is what do you have to do if there are arguments that need to be ignored? If using the usual syntax as below:

```
$.get ('myhtmlpage.html', myCallBack (param1, param2));
```

will not work, because the system will call:

```
myCallBack (param1, param2)
```

and then ignore the value returned as the second parameter to `$.get ()`

However, the problem with the above example is that **myCallBack (param1, param2)** is evaluated before being moved as a function. **Javascript** and **jQuery** extension often have to use pointer functions in cases like this, such as `setTimeout` function .

For example, below, an **anonymous** function is initialized (simply a statement block) and registered as a callback **function** , noting the use of **'function () {}'** . The **anonymous** function performs exactly one function, which is calling **myCallBack** with the value of param1 param2.

```
$.get ('myhtmlpage.html', function () {  
myCallBack (param1, param2);  
});
```

In which **param1** and **param2** are used as a **callback** when the **'\$.get'** function completes the web page loading process.

You finished reading the article "**Learn about how jQuery works**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.