

1.How to hide html element using JavaScript ? jQuery ?

(a) JavaScript (Vanilla):

- We can hide an HTML element using the style property of the element to modify its display property. example:
- // To hide an element by ID
`document.getElementById("elementId").style.display = "none";`
- // To hide an element by class name
`document.getElementsByClassName("className")[0].style.display = "none";`
- This will set the display property to "none," making the element invisible, and it won't occupy space in the layout.

(b) jQuery:

- If we are using jQuery, we can achieve the same result more concisely. First, make sure we have included the jQuery library in your HTML file. Then, we can hide an element like this:
- // To hide an element by ID
`$("#elementId").hide();`
- // To hide an element by class name
`$(".className").hide();`
- jQuery's `hide()` method essentially sets the display property to "none" as well, achieving the same effect as the vanilla JavaScript method.

2.How to select all textboxes using jQuery selector ?

- To select all textboxes in a webpage using a jQuery selector without relying on class names or IDs, we can use the `:input` selector to target all input elements, and then filter for the ones with the `'type'` attribute set to `"text."` Here's the code:

```
$(document).ready(function() {  
  var allTextboxes = $(':input[type="text"]');  
  // Now allTextboxes contains all textboxes on the page  
  // You can perform actions on these elements as needed  
});
```
- This code uses the `:input` selector to select all input elements and then narrows it down to only those with the `'type'` attribute set to `"text."`

3.What is the difference between “this” and \$(this) in jQuery programming ?

- "this": It refers to the current DOM element that is being operated on within the context of an event handler or a callback function. When we use "this" in a jQuery event handler, it points to the specific DOM element that triggered the event. For example, if we have a click event handler on a button, "this" inside the handler will refer to that button element.
- "\$(this)": This is a jQuery object that wraps the DOM element referred to by "this." It allows to apply jQuery methods and functions to the selected element. "\$(this)" is often used when you want to perform operations on the DOM element using jQuery functions.
- Here's an example to illustrate the difference:

```
$("#button").click(function() {  
  // Using "this" to refer to the DOM element  
  alert("Button text: " + this.innerText);  
  
  // Using "$(this)" to create a jQuery object  
  $(this).css("background-color", "red");  
});
```
- “this” is used to access the DOM element directly to get its text, while “\$(this)” is used to apply a jQuery method to change the background color of the button.