[ZANG] - [Chuanjie]

Student ID: 699382953

[chuanjiezang@gmail.com](mailto:chuanjiezang@gmail.com)

**1. When specifying the duration of an animation, what values are**

**allowed?**

specify the duration in several ways:

* **Milliseconds**: A specific time in milliseconds (e.g., 500 for half a second).
* **Strings**: "slow" (600ms) or "fast" (200ms) are predefined shorthand values.

**2.What does jQuery's slideDown() method do?**

The slideDown() method animates an element by sliding it down to make it visible. It reveals the element by adjusting its height from 0 to its full height.

**3.Which methods would you use to display or hide an element using an animation of that element's opacity?**

use the .fadeIn() and .fadeOut() methods to animate an element's opacity, making it fade in or out, respectively.

$("element").fadeIn();

$("element").fadeOut();

**4.What method would you use to create a custom animation?**

To create a custom animation, use the .animate() method. It allows you to define custom properties to animate.

$("element").animate({

opacity: 0.5,

height: "toggle"

}, 1000); // Animates over 1 second

**5. Which easings are provided with jQuery core?**

jQuery core provides two default easing options:

* **"swing"**: Starts slow, accelerates, then slows down again.
* **"linear"**: Maintains a constant speed throughout the animation.

Additional easing effects are available through jQuery UI.

1. **What method do you use to add a plugin to jQuery?**

To add a plugin to jQuery, extend it by attaching your plugin to the $.fn object.

$.fn.myPlugin = function() {

// Plugin code

};

1. **How do you list all the plugins in use within jQuery?**

There isn’t a built-in method to list all active plugins directly. However, you can check jQuery's $.fn object to see which plugins are added to it, as each plugin is registered here.

**8. How would you create a private API for your own use within your**

**jQuery plugin?**

create a private API by defining private functions or variables within a closure inside your plugin.

(function($) {

var privateFunction = function() {

// Private functionality

};

$.fn.myPlugin = function() {

privateFunction(); // Private API call

};

})(jQuery);

1. **How are items you've selected with jQuery accessed in your custom plugin?**

Items selected with jQuery are accessed through this within the plugin function.

$.fn.myPlugin = function() {

this.each(function() {

// `this` refers to each selected element

$(this).css("color", "red");

});

};

1. **What value should your plugin return?**

A jQuery plugin should return this to maintain chainability. This allows further jQuery methods to be called on the selected elements.

$.fn.myPlugin = function() {

return this.each(function() {

// Plugin functionality

});

};

1. **Why is namespacing your event names a good idea?**

Namespacing event names helps prevent event conflicts, especially when multiple event handlers are used. It allows you to target and remove specific event listeners without affecting others.

$("element").on("click.myNamespace", function() {

// Handler for the click event

});

**12.Customize the jQuery contextMenu plugin example to provide one or more options:**

**a. Have your contextMenu plugin execute a callback function**

**when the menu is opened and positioned.**

modify the plugin to accept a callback function when the menu opens.

$.contextMenu({

selector: 'element',

callback: function() {

// Custom callback when menu is opened

console.log("Menu opened");

}

});

**b. Have your contextMenu plugin provide options for the list of**

**items used in the context menu itself.**

customize the options for the context menu by providing a list of items.

$.contextMenu({

selector: 'element',

items: {

option1: { name: "Option 1", callback: function() { console.log("Option 1 selected"); } },

option2: { name: "Option 2", callback: function() { console.log("Option 2 selected"); } }

}

});