Web Design for Beginners

16. JQUERY UI/Plugins

17. Sass

CHAW SINT YIN

What is jQuery?

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

The purpose is **to make it much easier to use JavaScript** on your website.

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

The jQuery library contains the following features:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

jQuery Syntax

Basic syntax is: **$(*selector*).*action*()**

A **$** sign to define/access jQuery

A (***selector***) to "query (or find)" HTML elements

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

Examples:

- $(this).hide() - hides the current element.
- $("p").hide() - hides all <p> elements.
- $(".test").hide() - hides all elements with class="test".
- $("#test").hide() - hides the element with id="test".

jQuery UI

jQuery UI is a curated **set of user interface interactions, effects, widgets, and themes** built on top of the jQuery JavaScript Library.

Whether you're building highly interactive web applications or you just need to add a date picker to a form control, jQuery UI is the perfect choice.

Why jQueryUI?

- Cohesive and Consistent APIs.
- Comprehensive Browser Support.
- Open Source and Free to Use.
- Good Documentation.
- Powerful Theming Mechanism.
- Stable and Maintenance Friendly.

Ways to use jQueryUI

JqueryUI library can be used in two ways:

1. Downloading UI Library from its official website

   http://jqueryui.com

2. Downloading UI Library from CDNs

   https://cdnjs.com/libraries/jquery

jQuery vs. jQuery Slim

jQuery Slim is considered good when it comes to prototyping and creating animation projects. As a slimmer version of jQuery (estimated a further saving of 17Kb), apart from AJAX and effects modules, it also does not carry deprecated code. It is a customized build of jQuery specially designed to meet the requirements of having a smaller-sized version.

Popular jQuery Alternatives

- Cash
- Zepto
- Syncfusion Essential JS2
- UmbrellaJS
- jQuery Slim
- JavaScript
- ReactJS
- ExtJS
- Bootstrap
- AngularJS
- Vue.js
- Chibi JS
- Flux
- MooTools
- D3.js

jQuery plugins

https://plugins.jquery.com/

jQuery Combinations

Query UI offers a combination of

- interaction,
- effects,
- widgets,
- utilities, and themes designed to work well together or on their own.

Interactions

Interactions add basic **mouse-based behaviors** to any element. You can create sortable lists, resizable elements, drag & drop behaviors and more with just a few lines of code. Interactions also make great building blocks for more complex widgets and applications.

E.g.

- Draggable
- Droppable
- Resizable
- Selectable
- Sortable

Widgets

Widgets are full-featured UI controls that bring the richness of desktop applications to the Web. All widgets provide a solid core with plenty of extension points for customizing behavior, as well as full theming support.

Eg.

- Accordion
- Autocomplete
- Button
- Checkboxradio
- Controlgroup
- Datepicker

Effects

Effects add support for **animating colors and class transitions**, as well as providing several **additional easings**. In addition, a full suite of custom effects are available for use when showing and hiding elements or just to add some visual appeal.

- Effect
- Visibility
  - Show
  - Hide
  - Toggle

Utilities

Utilities used by jQuery UI **to build interactions and widgets**.

- Position
- Widget Factory

What is Sass?

Sass stands for **S**yntactically **A**wesome **S**tyle**s**heet

- ✓Sass is an extension to CSS
- ✓Sass is a CSS pre-processor
- ✓Sass is completely compatible with all versions of CSS
- ✓Sass reduces repetition of CSS and therefore saves time
- ✓Sass was designed by Hampton Catlin and developed by Natalie Weizenbaum in **2006**
- ✓Sass is free to download and use

Why Use Sass?

Stylesheets are getting larger, more complex, and harder to maintain. This is where a CSS pre-processor can help.

Sass lets you use features that do not exist in CSS, like variables, nested rules, mixins, imports, inheritance, built-in functions, and other stuff.

Sass Example

```
/* define variables for the primary colors */
$primary_1: #a2b9bc;
$primary_2: #b2ad7f;
$primary_3: #878f99;

/* use the variables */
.main-header {
  background-color: $primary_1;
}

.menu-left {
  background-color: $primary_2;
}

.menu-right {
  background-color: $primary_3;
}
```

Sass Compilation

Get NodeJS first.

https://github.com/nodejs/Release

Embed it in your web page as :

```
<script src="test.json"></script>
 <link rel="stylesheet/less" type="text/css" href="mystyle.scss" />
```

Compiling .sass

Open your .sass file , open Terminal and type >> `npx sass --watch mystyle.scss output.css`

Sass Variables

Variables are a way to store information that you can re-use later.

With Sass, you can store information in variables, like:

- strings
- numbers
- colors
- booleans
- lists
- nulls

Sass uses the **$** symbol, followed by a name, to declare variables.

Sass Variable

```
$myFont: Helvetica, sans-serif;
$myColor: red;
$myFontSize: 18px;
$myWidth: 680px;

body {
  font-family: $myFont;
  font-size: $myFontSize;
  color: $myColor;
}

#container {
  width: $myWidth;
}
```

Sass Nested Rules

```
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }
  li {
    display: inline-block;
  }
  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

Sass @mixin and @include

The **@mixin** directive lets you create CSS code that is to be reused throughout the website.

The **@include** directive is created to let you use (include) the mixin.

```
@mixin important-text {
  color: red;
  font-size: 25px;
  font-weight: bold;
  border: 1px solid blue;
}

.danger {
  @include important-text;
  background-color: green;
}
```

Sass @extend and Inheritance

The **@extend** directive lets you share a set of CSS properties from one selector to another.

The **@extend** directive is useful if you have almost identically styled elements that only differ in some small details.

```
.button-basic  {
  border: none;
  padding: 15px 30px;
  text-align: center;
  font-size: 16px;
  cursor: pointer;
}

.button-report  {
  @extend .button-basic;
  background-color: red;
}

.button-submit  {
  @extend .button-basic;
  background-color: green;
  color: white;
}
```

Learning Resources

https://www.tutorialspoint.com/internet_technologies/internet_overview.htm

https://www.w3schools.com/jquery/default.asp

https://cdnjs.com/libraries/jquery

https://jqueryui.com/