

شبكة تعلم مع ناو

ووردبريس*جافاسكريبت*لينكس-برمجة عامة

تاريخ التحديث: نوفمبر 6, 2024

جافاسكريبت  تطوير الويب

Alpine.js الدليل الشامل

يوفر Alpine.js الدليل الشامل شرح واضح وسهل على كيفية تثبيت Alpine.js مع الأمثلة العملية التي تساعد على فهم أساسيات اطار العمل هذا وتطبيقه.

اسم الدليل: Alpine.js

رقم الدليل: 601

المجال: جافاسكريبت

المتطلبات: معرفة أساسية بتطوير الويب

تاريخ تحديث الدليل: نوفمبر 6, 2024

مقدمة عن Alpine.js

Alpine.js هو إطار عمل JavaScript خفيف يستخدم لإضافة تفاعلية وسلوك ديناميكي إلى صفحات الويب بسهولة. يركز على تبسيط كتابة الكود بتقنيات HTML و JavaScript، دون الحاجة إلى تحميل مكتبات ضخمة مثل Vue.js أو React. يعتبر خيارًا مثاليًا لتطبيقات بسيطة تحتاج إلى تحسين تجربة المستخدم.

تثبيت Alpine.js

1. التثبيت باستخدام CDN

أبسط طريقة لاستخدام Alpine.js هي عن طريق إضافته إلى مشروعك من خلال رابط CDN. كل ما عليك فعله هو إضافة هذا السطر داخل وسم <head> أو في نهاية ملف الـ HTML قبل وسم الإغلاق <body>:

```
1 <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x.x/di
2
```

بهذا، Alpine.js سيكون جاهز للاستخدام في أي مكان بصفتك دون الحاجة إلى إعدادات إضافية.

2. التثبيت باستخدام NPM

إذا كنت تستخدم مدير حزم مثل NPM وتفضل تضمين Alpine.js ضمن مشروعك، يمكنك تثبيته عبر هذا الأمر:

```
1 npm install alpinejs
```

وبعد التثبيت، قم باستيراد المكتبة في ملف JavaScript الخاص بك:

```
1 import Alpine from 'alpinejs';
2 window.Alpine = Alpine;
3 Alpine.start();
```

بهذه الطريقة، يمكنك التحكم بشكل أكبر في كيفية تحميل واستخدام Alpine.js ضمن مشروعك.

3. التأكد من التثبيت

قم بإنشاء ملف html فيه المعلومات التالية:

```
1 <html>
2 <head>
3   <script defer
4   src="https://cdn.jsdelivr.net/npm/alpinejs@3.x.x/dist/cdn.min.js
5 </script>
6 </head>
7 <body>
8   <h1 x-data="{ message: 'Hello, from Alpine' }" x-text="messa
9 </body>
10 </html>
```

إذا ظهر لك **Hello, from Alpine** في المتصفح، هذا يعني أنك قمت بالتثبيت بنجاح.

الأساسيات

x-data

المفهوم

تستخدم لإنشاء **نطاق بيانات تفاعلي** (reactive data) داخل عنصر (والعناصر الفرعية له) على الصفحة. هذا يعني أن البيانات الموجودة داخل x-data ستكون **تفاعلية**، أي أن أي تغييرات تُجرى على هذه البيانات ستؤدي إلى تحديث تلقائي للـ DOM (الواجهة)، مما يجعل التفاعل مع البيانات مرناً وسهلاً.

كيف تعمل

1. x-data هو حاوية لحفظ **حالة** العنصر، أي البيانات التي ترغب في تتبعها والتفاعل معها (مثل العدادات، قيم المدخلات، أو القيم الديناميكية الأخرى).
2. عند استخدام x-data، يصبح العنصر **تفاعلياً**، مما يعني أنه سيتم تحديث الـ DOM تلقائياً عند تغيير البيانات دون الحاجة للتلاعب اليدوي بالـ JavaScript.

```

1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4   </script>
5 </head>
6 <body>
7   <div x-data="{ name: 'Nawras' }">
8     <h1 x-text="name"></h1> <!-- Nawras -->
9   </div>
10 </body>
11 </html>

```

مثال على استخدام x-data

```

1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4   </script>
5 </head>
6 <body>
7   <div x-data="{ name: 'Nawras', age: 200, salary: 80000 }">
8     <h1 x-text="name"></h1> <!-- Nawras -->
9     <p x-text="age"></p> <!-- 200 -->

```

```
10 <p x-text="salary"></p> ←!— 80000 →
11 </div>
12 </body>
13 </html>
```

HTML

x-text

المفهوم

يستخدم لربط النص داخل عنصر مع قيمة من البيانات التفاعلية. يعمل x-text على تحديث النص داخل العنصر تلقائيًا عندما تتغير البيانات المرتبطة به. باستخدام x-text، يمكنك عرض أو تحديث النص داخل عنصر HTML بناءً على القيم المتغيرة في x-data أو البيانات التفاعلية الأخرى.

كيف يعمل

1. x-text يقوم بتغيير النص داخل العنصر الذي يتم تطبيقه عليه ليعكس قيمة البيانات التفاعلية أو التعبير الذي قمت بتمريره له.
2. عند تغيير قيمة البيانات المرتبطة بـ x-text، يتم تحديث النص داخل العنصر تلقائيًا دون الحاجة إلى إعادة تحميل الصفحة أو التعامل مع DOM باستخدام JavaScript.

مثال على استخدام x-text

```
1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4   </script>
5 </head>
6 <body>
7   <div x-data="{ username: 'Ali' }">
8     <h1 x-text="username"></h1> ←!— Ali →
9   </div>
10 </body>
11 </html>
```

HTML

x-show

المفهوم

يُستخدم لإظهار أو إخفاء عنصر بناءً على قيمة من البيانات التفاعلية. عندما تتغير القيمة المرتبطة بـ x-show، فإن العنصر سيظهر أو يختفي على الصفحة بشكل ديناميكي. بشكل عام، هو يستخدم للتحكم في الرؤية أو الظهور للعناصر على الصفحة.

كيف يعمل

1. x-show يقوم بتطبيق خاصية display على العنصر بناءً على القيمة المُعطاة له.

2. عندما تكون القيمة المرتبطة بـ `x-show` **صحيحة** (مثل `true` أو قيمة غير فارغة)، يتم إظهار العنصر

3. عندما تكون القيمة **خطأ** (مثل `false` أو `null`)، يتم إخفاء العنصر.

مثال على استخدام `x-show`

```
1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4   </script>
5 </head>
6 <body>
7   <div x-data="{ username: 'Ali', password: '123', showPassword
8     <h1 x-text="username"></h1> <!-- Ali -->
9     <p x-show='showPassword' x-text="password"></p> <!-- 123 --
10    </div>
11 </body>
12 </html>
```

`x-on:click`

المفهوم

تستخدم للانتباه إلى حدث **النقر (click)** على عنصر معين. عند النقر على العنصر، يتم تنفيذ الأكواد أو التغييرات التي تحددها في هذا الحدث `Event`.

كيف يعمل

1. عند استخدام `x-on:click`، يحدد Alpine.js أنه عندما يتم النقر على العنصر، يتم تنفيذ الكود المرتبط بهذا الحدث.
2. يمكنك من التفاعل مع المستخدم من خلال استجابة معينة للنقر على الأزرار أو العناصر الأخرى.

مثال على استخدام `x-on:click`

```
1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4   </script>
5 </head>
6 <body>
7   <div x-data="{ username: 'Ali', password: '123', showPassword
8     <h1 x-text="username"></h1> <!-- Ali -->
9     <button x-on:click="showPassword=true">Show Password</button
10    <p x-show='showPassword' x-text="password"></p> <!-- 123 --
11    </div>
```

```
12 </body>
13 </html>
```

HTML

يمكنك استخدام الاختصار `click@` بدلاً من `x-on:click`. كلاهما يؤدي نفس الوظيفة، ولكن الاختصار `click@` يعتبر أكثر بساطة وسهولة في الكتابة.

```
1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4   </script>
5 </head>
6 <body>
7   <div x-data="{ username: 'Ali', password: '123', showPassword
8     <h1 x-text="username"></h1> <!-- Ali -->
9     <button @click="showPassword=true">Show Password</button>
10    <p x-show='showPassword' x-text="password"></p> <!-- 123 --
11    </div>
12 </body>
13 </html>
```

HTML

المزيد من الأساسيات

x-init

المفهوم

يُستخدم لتنفيذ تعليمات **JavaScript** عند تحميل المكون (component) لأول مرة، أي عندما يتم تهيئة المكون أو العنصر الذي يحتوي عليه. يساعد في تنفيذ الأكواد أو إعداد القيم الأولية للبيانات عند بدء تشغيل العنصر.

كيف تعمل

1. يتم استخدام `x-init` لتشغيل كود عند تحميل الصفحة أو تهيئة الكائن التفاعلي.
2. يمكن استخدامه لإعداد القيم الافتراضية، أو لتشغيل دوال JavaScript أو أي عملية تحتاج إلى أن تحدث بمجرد أن يبدأ العنصر في الظهور على الصفحة

```
1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4   </script>
```

```

5 </head>
6 <body>
7   <div x-data="{ name: ' ' }" x-init="name = 'Ali'">
8     <h1 x-text="name"></h1> <!-- Ali -->
9   </div>
10 </body>
11 </html>
12

```

HTML

ادارة التفاعلات في دالة

المفهوم

من الممكن إضافة البيانات والدوال مباشرة في الكود HTML باستخدام `x-data` و `x-init`، لكن في بعض الأحيان يكون من الأفضل إدارة هذه التفاعلات داخل دوال JavaScript خارج HTML، مما يسهل تنظيم الكود ويجعله أكثر قابلية للصيانة والقراءة.

```

1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x">
4 </head>
5 <body>
6   <div x-data="initData()">
7     <h1 x-text="username"></h1> <!-- Ali -->
8     <button x-on:click="showPassword=true">Show Password</button>
9     <p x-show='showPassword' x-text="password"></p> <!-- 123 -->
10  </div>
11
12  <script>
13    function initData(){
14      return {
15        username: "Ali",
16        password: "123",
17        showPassword: false
18      }
19    }
20  </script>
21 </body>
22 </html>
23

```

HTML

النماذج Forms

x-model

المفهوم

يُستخدم لربط القيم التفاعلية بين واجهة المستخدم (HTML) وبين البيانات (Data) الخاصة بك في JavaScript. يعمل هذا الموجه على **مزامنة** البيانات بين الحقول المدخلة (مثل input أو textarea) والقيم التي يتم تخزينها داخل المكون (داخل x-data).

باستخدام x-model، يمكن ربط القيم المدخلة في النماذج أو الحقول النصية بالبيانات في الجافا سكريبت، مما يجعل التفاعل مع النموذج أسهل وأسرع. عندما يُدخل المستخدم قيمة في الحقل المرتبط بـ x-model، يتم تحديث البيانات تلقائيًا.

كيف تعمل

1. **الربط التلقائي:** عندما يغير المستخدم قيمة في عنصر مُرتبط بـ x-model، يتم تحديث البيانات تلقائيًا.

2. **الانتقال من البيانات إلى الواجهة:** إذا تم تحديث القيمة في البيانات داخل x-data، سيتم تحديث العنصر في HTML الذي يرتبط بـ x-model تلقائيًا.

```
1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4 </head>
5 <body>
6   <div x-data="initData()">
7     <h1>
8       Hello, <span x-text="username"></span>
9       <span x-text="password"></span>
10    </h1>
11    <hr />
12
13    <form>
14      <input type="text" x-model="username" placeholder="P
15      <input type="password" x-model="password" placeholde
16    </form>
17  </div>
18
19  <script>
20    function initData() {
21      return {
22        username: "",
23        password: "",
24      }
25    }
26  </script>
27 </body>
28 </html>
29
```


@submit

يتم تنفيذ التعليمات الموجودة في submit event عند تقديم نموذج ما.

```
1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4 </head>
5 <body>
6   <div x-data="initData()">
7     <h1>
8       Hello, <span x-text="username"></span>
9       <span x-text="password"></span>
10    </h1>
11    <hr />
12
13    <form @submit.prevent="submitForm">
14      <input type="text" x-model="username" placeholder="P
15      <input type="password" x-model="password" placeholde
16      <button type="submit">Submit Form</button>
17    </form>
18  </div>
19
20  <script>
21    function initData() {
22      return {
23        username: "",
24        password: "",
25        submitForm() {
26          alert("Submitted");
27        }
28      }
29    }
30  </script>
31 </body>
32 </html>
33
```

التحقق من المدخلات

يمكن التحقق من المدخلات وعرض رسالة خطأ أو نجاح كما في المثال التالي:

```
1
2
3
```

```

1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4 </head>
5 <body>
6   <div x-data="initData()">
7     <h1>
8       Hello,
9       <span x-text="username"></span>
10      <span x-text="password"></span>
11    </h1>
12    <hr />
13
14    <!-- Message Section: Displays error or success message
15    <div class="message">
16      <p class="error" x-show="showError" x-text="errorMes
17      <p class="success" x-show="showSuccess" x-text="succ
18    </div>
19
20    <!-- Form Section -->
21    <form @submit.prevent="submitForm">
22      <input type="text" x-model="username" placeholder="P
23      <input type="password" x-model="password" placeholde
24      <button type="submit">Submit Form</button>
25    </form>
26  </div>
27
28  <script>
29    function initData() {
30      return {
31        username: "",
32        password: "",
33        showError: false,
34        showSuccess: false,
35        errorMessage: "",
36        successMessage: "",
37
38        // Handle form submission
39        submitForm() {
40          // Reset error and success states
41          this.showError = false;
42          this.showSuccess = false;
43
44          // Validate username
45          if (!this.username) {
46            this.errorMessage = "You must provide a
47            this.showError = true;
48            return;
49          }

```

```

50
51         // Validate password
52         if (!this.password) {
53             this.errorMessage = "You must provide a
54             this.showError = true;
55             return;
56         }
57
58         // If form is valid, show success message
59         this.showSuccess = true;
60         this.successMessage = "The form has been sub
61     }
62 }
63 }
64 </script>
65 </body>
66 </html>
67

```

HTML

آلية عمل الكود

1- تهيئة البيانات (x-data):

- يقوم الكود بتخزين البيانات داخل دالة initData التي تحتوي على المتغيرات التالية:
 - username: لتخزين اسم المستخدم.
 - password: لتخزين كلمة المرور.
 - showError: لتحديد ما إذا كان يجب عرض رسالة الخطأ.
 - showSuccess: لتحديد ما إذا كان يجب عرض رسالة النجاح.
 - errorMessage: لتخزين رسالة الخطأ.
 - successMessage: لتخزين رسالة النجاح.

2- التفاعل مع النموذج:

- عند تقديم النموذج عبر الزر إرسال:
 - يتم التحقق أولاً إذا كان حقل اسم المستخدم فارغاً أو كلمة المرور فارغة.
 - إذا كان أحد الحقول فارغاً، يتم عرض رسالة خطأ.
 - إذا كانت المدخلات صحيحة، يتم إخفاء رسائل الخطأ وعرض رسالة نجاح.

3- التحقق من المدخلات (دالة submitForm):

- إذا كان حقل اسم المستخدم فارغاً: تظهر رسالة خطأ تطلب إدخال اسم المستخدم.
- إذا كان حقل كلمة المرور فارغاً: تظهر رسالة خطأ تطلب إدخال كلمة المرور.
- إذا كانت الحقول كاملة: تظهر رسالة نجاح تؤكد أن النموذج تم إرساله بنجاح.

4- الرسائل التفاعلية:

- رسالة خطأ: تظهر فقط إذا كانت المدخلات غير كاملة (x-show="showError").
- رسالة نجاح: تظهر فقط إذا كانت المدخلات صحيحة (x-show="showSuccess").

x-bind يُستخدم لربط قيمة ديناميكية إلى سمة (attribute) معينة لعنصر HTML. يتيح لك تعديل الخصائص (مثل href, src, class, style وغيرها) بناءً على البيانات في **Alpine.js**. في المثال السابق، يمكن إضافة CSS Class بشكل ديناميكي لرسالة الخطأ أو النجاح.

```

1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4
5
6   <style>
7     .red{background: red;color: white;}
8     .green{background: green;color: white;}
9   </style>
10 </head>
11 <body>
12   <div x-data="initData()">
13     <h1>
14       Hello,
15       <span x-text="username"></span>
16       <span x-text="password"></span>
17     </h1>
18     <hr />
19
20     <!-- Message Section: Displays error or success message
21     <div class="message" x-bind:class="{ 'red': showError, 'g
22       <p class="error" x-show="showError" x-text="errorMes
23       <p class="success" x-show="showSuccess" x-text="succ
24     </div>
25
26     <!-- Form Section -->
27     <form @submit.prevent="submitForm">
28       <input type="text" x-model="username" placeholder="P
29       <input type="password" x-model="password" placeholde
30       <button type="submit">Submit Form</button>
31     </form>
32   </div>
33
34   <script>
35     function initData() {
36       return {
37         username: "",
38         password: "",
39         showError: false,
40         showSuccess: false,
41         errorMessage: "",
42         successMessage: "",

```

```

43
44         // Handle form submission
45         submitForm() {
46             // Reset error and success states
47             this.showError = false;
48             this.showSuccess = false;
49
50             // Validate username
51             if (!this.username) {
52                 this.errorMessage = "You must provide a
53                 this.showError = true;
54                 return;
55             }
56
57             // Validate password
58             if (!this.password) {
59                 this.errorMessage = "You must provide a
60                 this.showError = true;
61                 return;
62             }
63
64             // If form is valid, show success message
65             this.showSuccess = true;
66             this.successMessage = "The form has been sub
67         }
68     }
69 }
70 </script>
71 </body>
72 </html>
73

```

HTML

آلية عمل الكود

الكود يستخدم `x-bind:class` لإضافة أو إزالة CSS Classes بناءً على قيم البيانات:

- إذا كانت `showError` تساوي `true`، يتم إضافة `.red`.
- إذا كانت `showSuccess` تساوي `true`، يتم إضافة `.green`.

بمعنى آخر، `green` و `red` تتغير بشكل ديناميكي استنادًا إلى قيم المتغيرات في البيانات.

في `Alpine.js`، يمكنك استخدام اختصار `x-bind` عن طريق استخدام `:` بدلاً من كتابة `x-bind` بالكامل. هذا يجعل الكود أكثر نظافة وأسهل في القراءة.

```
1 Example:
2 <button x-bind:disabled="isDisabled">Submit</button>
3 <button :disabled="isDisabled">Submit</button>
```

HTML

العرض الشرطي والتكراري

x-for

يُستخدم لتكرار عناصر معينة داخل قائمة أو مجموعة من البيانات، مثل المصفوفات. عند استخدامه، يُنشئ مكرراً للعناصر تلقائياً بناءً على بيانات معينة. يمكنك التفكير فيه كطريقة لإنشاء العناصر بشكل ديناميكي باستخدام بنية قائمة (مثل `` أو `<table>`)، وتسهيل عرض المحتوى من المصفوفات أو الكائنات داخل الكود دون الحاجة إلى كتابة العناصر بشكل يدوي.

```
1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4 </head>
5 <body>
6   <div x-data="listData()">
7     <h2>Product List</h2>
8
9     <ul>
10      <template x-for="item in items" :key="item.id">
11        <li>
12          <span x-text="item.title"></span>:
13          <span x-text="item.price"></span> USD
14        </li>
15      </template>
16    </ul>
17  </div>
18
19  <script>
20    function listData() {
21      return {
22        items: [
23          { id: 1, title: "Product 1", price: "50" },
24          { id: 2, title: "Product 2", price: "60" },
25          { id: 3, title: "Product 3", price: "70" },
26          { id: 4, title: "Product 4", price: "0" },
27          { id: 5, title: "Product 5", price: "0" }
28        ]
29      }
30    }
31  </script>
32 </body>
```

لماذا نستخدم template في Alpine.js؟

1. لا يظهر في DOM: عنصر <template> لا يتم عرضه في DOM، بل يُستخدم فقط كـ "قالب" (أو هيكل) يمكن تكراره. هذا يسمح بإنشاء عناصر متعددة دون إدخال عناصر HTML إضافية لا داعي لها.
2. تحسين الأداء: عند وضع x-for على <template>، يقوم Alpine بنسخ المحتوى داخل القالب وتكراره مباشرةً في الـ DOM. هذا الأسلوب يكون أسرع وأخف، ويسمح لـ Alpine بإدارة العناصر المتكررة بشكل أكثر كفاءة.
3. وضوح الكود: استخدام <template> يجعل الكود أوضح، حيث يدل المطورين على أن المحتوى بداخله هو للتكرار فقط، ويجعل الكود منظمًا وسهل القراءة.

يُستخدم العنصر <template> كعنصر حاوية لهيكل الكود الذي تريد تكراره عدة مرات، ولكن template لا يظهر فعليًا في صفحة HTML النهائية.

توضيح الكود

```

1 <ul>
2   <template x-for="item in items" :key="item.id">
3     <li>
4       <span x-text="item.title"></span>:
5       <span x-text="item.price"></span> USD
6     </li>
7   </template>
8 </ul>

```

HTML

هذا الكود يستخدم Alpine.js لإنشاء قائمة (ul) من المنتحات (items) حيث يتم عرض كل منتج (item) على سطر عنصر قائمه (li) يتضمن عنوان المنتج وسعره.

يمرّ على كل عنصر في مصفوفة items، ويعرض تفاصيل كل منتج مثل العنوان والسعر داخل عنصر .

- "x-for="item in items": يكرّر العناصر في مصفوفة items.
- "key="item.id": يستخدم id من كل عنصر كـ "مفتاح" لتسريع إعادة التحديث.
- داخل التكرار، يتم عرض العنوان والسعر باستخدام x-text.

بشكل عام، يستخدم x-for لتكرار HTML بناءً على مجموعة من البيانات في Alpine.js.

x-if

المفهوم

تستخدم لعرض أو إخفاء العناصر بناءً على شرط معين. عندما يكون الشرط صحيحًا، يتم إضافة العنصر إلى DOM (الشجرة الهيكلية للصفحة)؛ وعندما يكون الشرط غير صحيح، يتم إزالته.

كيف تعمل

1. `x-if="condition"`: إذا كانت القيمة في `condition` صحيحة (مثل `true` أو أي قيمة غير فارغة)، سيظهر العنصر في الصفحة.

2. إذا كانت القيمة غير صحيحة (مثل `false` أو `null`)، سيتم إزالة العنصر بالكامل من الصفحة.

```
1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4 </head>
5 <body>
6   <div x-data="listData()">
7     <h2>Product List</h2>
8
9     <ul>
10      <template x-for="item in items" :key="item.id">
11        <template x-if="item.price ≥ 60">
12          <li>
13            <span x-text="item.title"></span>:
14            <span x-text="item.price"></span> USD
15          </li>
16        </template>
17      </template>
18    </ul>
19  </div>
20
21  <script>
22    function listData() {
23      return {
24        items: [
25          { id: 1, title: "Product 1", price: "50" },
26          { id: 2, title: "Product 2", price: "60" },
27          { id: 3, title: "Product 3", price: "70" },
28          { id: 4, title: "Product 4", price: "0" },
29          { id: 5, title: "Product 5", price: "0" }
30        ]
31      }
32    }
33  </script>
34 </body>
```

توضيح الكود

- `<template x-if="item.price >= 60">`: يتحقق من إذا كان سعر العنصر (`item.price`) أكبر من أو يساوي 60.
 - إذا كانت القيمة `true`، سيتم عرض الـ `` داخل الـ `<template>`.
 - إذا كانت القيمة `false`، فلن يتم عرض العنصر.
 - ``: يعرض تفاصيل العنصر إذا كانت الشروط (`item.price >= 60`) صحيحة، بما في ذلك:
 - ``: يعرض عنوان المنتج.
 - ` USD`: يعرض سعر المنتج متبوعًا بـ "USD".
- ببساطة:** الكود يستخدم `x-if` لعرض العناصر التي يكون سعرها أكبر من أو يساوي 60.

`x-if` يقوم بإضافة أو إزالة العنصر من DOM بشكل كامل، بينما `x-show` يختبئ العنصر باستخدام CSS (يبقى في DOM).

التوجيهات المتقدمة في Alpine.js

x-html

في `Alpine.js`، يُستخدم `x-html` لتحديد محتوى HTML ديناميكي لعناصر HTML مباشرةً من خلال ربطه بمتغير أو نص يحتوي على HTML. يشبه هذا الأمر `x-text`، ولكنه يسمح بإدراج HTML بدلًا من النص العادي. يعتبر `x-html` مفيدًا لعرض محتوى HTML معقد يتم إنشاؤه ديناميكيًا أو جلبه من مصدر خارجي.

```

1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4 </head>
5 <body>
6   <div x-data="initData()">
7     <h1 x-text="title"></h1>
8     <div x-html="content"></div>
9   </div>
10
11   <script>
12     function initData() {
13       return {
14         title : "Post Title",
15         content: "<strong>hello</strong> <small>small tex
16
17       }
18     }
19   </script>
20 </body>
21 </html>
22
23

```

عند استخدام x-html في Alpine.js، يجب الحذر من الثغرات الأمنية المتعلقة بـ XSS (البرمجة النصية عبر المواقع) إذا كانت البيانات التي يتم عرضها من مصدر خارجي أو غير موثوق. نظرًا لأن x-html يعرض HTML مباشرة، قد يؤدي إدخال محتوى غير موثوق إلى حقن تعليمات برمجية خبيثة داخل الصفحة، مما يعرض المستخدمين للخطر.

x-transition

x-transition في Alpine.js هي خاصية تُستخدم لتطبيق تأثيرات الانتقال (animations) على العناصر عند ظهورها أو اختفائها، مما يجعل التفاعل أكثر سلاسة وجمالاً. يمكن استخدام x-transition مع عناصر مثل x-show، بحيث تتحرك العناصر تدريجيًا عند ظهورها أو اختفائها بدلاً من أن تظهر أو تختفي فجأة.

```

1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4 </head>
5 <body>
6   <div x-data="initData()">
7
8     <button @click="showDiv = !showDiv">
9       Show/Hide
10    </button>
11
12    <div x-show="showDiv" x-html="content" x-transition></d
13    <div x-show="showDiv" x-html="content" x-transition.dur
14    <div x-show="showDiv" x-html="content" x-transition.sca
15    <div x-show="showDiv" x-html="content" x-transition.opa
16  </div>
17
18  <script>
19    function initData() {
20      return {
21        showDiv: false,
22        content: ` <header><h1>Welcome to My Website</h1><
23      }
24    }
25  </script>
26 </body>
27 </html>

```

في هذا الكود، x-transition هو خاصية في Alpine.js تُستخدم لإضافة تأثيرات انتقالية عند إظهار أو إخفاء العناصر. تُستخدم لإضافة تأثيرات مثل التلاشي، التحجيم، أو التلاشي التدريجي عند تغيير حالة

العنصر.

كيفية عمل x-transition:

1. الانتقال البسيط: x-transition: يُضيف تأثير انتقالي افتراضي عند تغيير حالة العنصر، مثل التلاشي التدريجي.

2. تحديد مدة الانتقال: x-transition.duration.5000ms: يُحدد مدة الانتقال بـ 5 ثواني.

3. تأثيرات متعددة:

◦ x-transition.scale.20: يُضيف تأثير مقياس حيث يتغير حجم العنصر.

◦ x-transition.opacity: يُضيف تأثير تلاشي حيث يصبح العنصر أكثر شفافية تدريجيًا.

◦ x-transition.duration.10000ms: يُحدد مدة الانتقال بـ 10 ثواني.

باختصار، x-transition يضيف تأثيرات بصرية لجعل تغيير حالة العنصر (ظهوره أو اختفائه) أكثر سلاسة وجاذبية.

x-effect

x-effect في Alpine.js هو خاصية تُستخدم لتنفيذ تأثيرات أو تغييرات على العناصر بناءً على تغيير القيم في الـ x-data. يعمل مثل الـ "watchers" في React أو Vue، حيث يراقب المتغيرات ويُنفذ التعليمات البرمجية عندما تتغير هذه القيم.

```
1 <html>
2 <head>
3   <script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x
4 </head>
5 <body>
6   <div x-data="initData()">
7
8     <button @click="count++" >
9       Show/Hide
10    </button>
11
12    <p x-text="count" x-effect="console.log(`changed the co
13  <script>
14    function initData() {
15      return {
16        count: 0,
17      }
18    }
19  </script>
20 </body>
21 </html>
22
```

الخاصية (x-effect="console.log('changed the count', count) تعني أنه في كل مرة يتغير فيها المتغير count، سيتم تنفيذ الكود داخل x-effect، والذي في هذه الحالة هو طباعة رسالة في وحدة التحكم (console.log) تحتوي على القيمة الجديدة لـ count

x-ignore

x-ignore في Alpine.js هو خاصية تُستخدم لإخبار Alpine.js بتجاهل عنصر أو خاصية معينة أثناء عملية التفاعل مع البيانات. عند استخدام x-ignore، سيتم تعطيل التفاعل الخاص بـ Alpine.js على العنصر أو الخاصية المحددة.

```
1 <html>
2   <head>
3     <script
4       defer
5       src="https://cdn.jsdelivr.net/npm/alpinejs@3.x.x/dist/cdn.
6     ></script>
7   </head>
8   <body>
9     <div x-data="initData()">
10      <p x-text="label">lala</p>
11      <hr />
12      <div x-ignore>
13        <p x-text="label">lala</p>
14      </div>
15    </div>
16    <script>
17      function initData() {
18        return {
19          label: 'Hello world',
20        };
21      }
22    </script>
23  </body>
24 </html>
25
```

في هذا الكود، x-ignore تُستخدم في العنصر <div> لتوجيه Alpine.js بتجاهل أي تفاعل داخل هذا العنصر مع البيانات أو التفاعلات.

التوضيح:

- العنصر الأول <p x-text="label">lala</p> سيعرض القيمة Hello world لأن Alpine.js يتفاعل مع المتغير label داخل x-data.
- العنصر الثاني داخل <div x-ignore> يحتوي على نفس العنصر <p x-text="label">lala</p>، لكن بسبب استخدام Alpine.js x-ignore، يتجاهل التفاعل مع هذا العنصر، ولن يغير محتوى النص المعروض داخل هذا العنصر حتى لو تغيرت قيمة label.
- باختصار، x-ignore يجعل Alpine.js يتجاهل التفاعل مع البيانات داخل العنصر المحدد، لذا العنصر الثاني سيعرض النص الثابت "lala" ولن يتأثر بالقيمة المتغيرة لـ label.

x-cloak

x-cloak في Alpine.js هو خاصية تُستخدم لإخفاء العناصر مؤقتًا حتى يتم تحميل جميع البيانات والتفاعلات الخاصة بـ Alpine.js. يُستخدم بشكل أساسي لتجنب عرض العناصر بشكل مؤقت قبل أن يبدأ Alpine.js في تفعيل التفاعلات عليها.

```
1 <html>
2   <head>
3     <style>
4       .green,
5       .red {
6         color: white;
7         padding: 10px;
8         font-size: 18px;
9       }
10      .green {
11        background: green;
12      }
13      .red {
14        background: red;
15      }
16      [x-cloak] {
17        display: none;
18      }
19    </style>
20    <script
21      defer
22      src="https://cdn.jsdelivr.net/npm/alpinejs@3.x.x/dist/cdn.
23    ></script>
24  </head>
25  <body>
26    <div x-data="initData()">
27      <h1 class="red" x-show="showName" x-text="name1" x-cloak><
28      <h1 class="green" x-show="showName" x-text="name2"></h1>
29    </div>
30    <script>
31      function initData() {
32        return {
33          name1: 'Ali',
34          name2: 'Nawras',
35          showName: false,
36        };
37      }
38    </script>
39  </body>
40 </html>
41
42
```

في هذا الكود، `x-cloak` تُستخدم لإخفاء العنصر بشكل مبدئي حتى يتم تحميل وتفعيل `Alpine.js`.
التوضيح:

- العنصر الأول (`<h1 class="red" x-show="showName" x-text="name1" x-cloak>`) سيكون مخفياً عند تحميل الصفحة بفضل `x-cloak`.
 - العنصر الثاني (`<h1 class="green" x-show="showName" x-text="name2">`) سيظهر ويختفي بناءً على قيمة `showName` ولكن بدون استخدام `x-cloak` عند تحميل الصفحة:
- `x-cloak` يضمن أن العنصر الأول لن يظهر قبل أن يتم تفعيل `Alpine.js`. كلا العنصرين سيكونان مخفيين لأن `showName` يبدأ كـ `false`.

x-teleport

`x-teleport` في `Alpine.js` يُستخدم لنقل (`teleport`) عنصر من مكان إلى آخر في DOM (شجرة المستند) بدون تغيير حالته أو بياناته. يسمح لك هذا بنقل محتوى من جزء معين من الصفحة إلى مكان آخر، مثل داخل نافذة منبثقة أو مكون آخر.

```

1 <html>
2   <head>
3     <style>
4       .modal {
5         background: rgba(0, 0, 0, 0.7);
6         color: white;
7         padding: 20px;
8         position: fixed;
9         top: 50%;
10        left: 50%;
11        transform: translate(-50%, -50%);
12        border-radius: 10px;
13      }
14    </style>
15    <script
16      defer
17      src="https://cdn.jsdelivr.net/npm/alpinejs@3.x.x/dist/cdn.
18    ></script>
19  </head>
20  <body>
21    <div x-data="{ open: false }">
22      <!-- Button to toggle the modal -->
23      <button @click="open = !open">Toggle Modal</button>
24
25      <!-- The modal content will be teleported to the body -->
26      <template x-teleport="body">
27        <div x-show="open" x-transition class="modal">

```

```

28     <p>This is a modal!</p>
29     <button @click="open = false">Close</button>
30   </div>
31 </template>
32 </div>
33 </body>
34 </html>
35

```

HTML

ففي هذا الكود، `x-teleport` تُستخدم لنقل محتوى النافذة المنبثقة (المودال) إلى مكان آخر في الصفحة، تحديداً إلى عنصر `<body>`، بدلاً من إبقائها داخل نفس الحاوية الأصلية.

كيف يعمل الكود؟

- `"x-data="{ open: false }"` يقوم بتعريف حالة `open`، والتي تبدأ بالقيمة `false`، مما يعني أن النافذة المنبثقة مغلقة في البداية.
- `<button @click="open = !open">Toggle Modal</button>` زر يقوم بتبديل حالة `open` بين `true` و `false` عند الضغط عليه، مما يفتح أو يغلق النافذة المنبثقة.
- `<template x-teleport="body">` محتوى النافذة المنبثقة إلى عنصر `<body>` ليظهر في أعلى الصفحة، بدلاً من إبقائه داخل `<div x-data>`.
- `"x-show="open"` يجعل النافذة تظهر فقط إذا كانت حالة `open` تساوي `true`.
- `x-transition` يضيف تأثير انتقال (ظهور تدريجي) عندما يتم فتح أو إغلاق النافذة.

الهدف من x-teleport

`x-teleport` يساعد في تجنب أي مشاكل تتعلق بمكان النافذة المنبثقة داخل صفحة الويب، مثل تداخلها مع عناصر أخرى أو مشاكل التنسيق.

الخصائص السحرية

\$el

في `Alpine.js`، الخاصية `el` تُستخدم للإشارة إلى العنصر (HTML element) الحالي الذي يتم تنفيذ الكود فيه.

```

1 <html>
2   <head>
3     <style>
4       .text {
5         background: rgba(0, 0, 0, 0.7);
6         color: white;
7       }
8     </style>
9     <script
10      defer
11      src="https://cdn.jsdelivr.net/npm/alpinejs@3.x.x/dist/cdn.

```

```

12     ></script>
13 </head>
14 <body>
15     <div x-data="{ }">
16         <p
17             class="text"
18             @mouseenter="$el.innerHTML = 'you are changed'"
19             @mouseleave="$el.innerHTML = 'Change me'"
20         >
21             Change me
22         </p>
23     </div>
24 </body>
25 </html>
26

```

HTML

\$refs

\$refs في Alpine.js هي طريقة للوصول إلى عناصر HTML داخل الكود الجافاسكريبت المرتبط بـ Alpine، بحيث تُمكنك من التعامل مع هذه العناصر مباشرةً.

```

1 <html>
2   <head>
3     <script
4       defer
5       src="https://cdn.jsdelivr.net/npm/alpinejs@3.x.x/dist/cdn.
6     ></script>
7   </head>
8   <body>
9     <div x-data="{ state: false }">
10      <button
11        @click="state = !state; state ? $refs.text.textContent =
12      >
13        Change the text
14      </button>
15      <span x-ref="text">Old Text</span>
16    </div>
17  </body>
18 </html>
19

```

HTML

في هذا الكود، يتم استخدام x-ref لتعيين مرجع (ref) لعنصر HTML، بحيث يمكن الوصول إليه مباشرة من خلال Alpine.js باستخدام \$refs.

كيف يعمل x-ref هنا:

- في العنصر `<x-ref="text">Old Text</x-ref>`, `x-ref="text"` يعين مرجعًا لهذا العنصر باسم "text".
- باستخدام `$.text.ref`, يمكنك الوصول إلى هذا العنصر مباشرة في الكود.

في الكود:

- عند النقر على الزر، يتم تعديل النص داخل العنصر `` عن طريق `$.text.textContent.ref`.
- إذا كانت حالة `state` صحيحة، يتم تغيير النص إلى "changed"، وإذا كانت الحالة خاطئة، يبقى النص "Old Text".

بعد دراستك لكل هذا الدليل، ستكون على دراية بأساسيات Alpine وسيكون لديك القدرة على إنشاء مشروع متكامل باستخدام Alpine.

تواصل معي

يسرني التواصل معكم عبر البريد الإلكتروني

mail at LearnWithNaw dot net

كيف أستطيع تسهيل أعمالك؟