Banding

Displaying values with interpolation

التفاعل (Interpolation) هو طريقة في Angular لعرض القيم داخل صفحات الويب. في الواقع، إنها طريقة لوضع القيم داخل النصوص أو الصور على صفحات الويب باستخدام Angular.

لنفترض أن لديك برنامج حاسوبي يمكنه صنع صفحات ويب، وتريد عرض بعض الكلمات أو الصور الخاصة على صفحتك الويب. التفاعل يساعدك في القيام بذلك.

فلنقم بإيجاد صندوق اسمه "currentCustomer" وفي داخل هذا الصندوق، هناك اسم "Maria". تريد عرض اسم Maria على صفحتك الويب.

للقيام بذلك، تستخدم الأقواس المزدوجة المنحنية، مثل هذا: {{ currentCustomer }}. إنها كما لو كانت رمزًا سرياً يخبر الحاسوب بأن يأخذ أي شيء يكون داخل صندوق "currentCustomer" ويضعه على صفحة الويب.

إذاً، عندما يصنع برنامجك الصفحة الويب، ينظر إلى {{ currentCustomer }} ويرى "Maria" داخل الصندوق. ثم يأخذ "Maria" ويضعها على الصفحة الويب. لذا، سترى عنوانًا يقول "العميل الحالي: Maria".

نفس الأمر ينطبق على الصور. لديك صندوق اسمه "itemImageUrl" وفي داخله، هناك رابط لصورة. تريد عرض تلك الصورة على صفحتك الويب.

تستخدم {{itemImageUrl}} لتخبر الحاسوب بأن ينظر داخل الصندوق ويستخدم رابط الصورة. يقوم الحاسوب بعرض الصورة على صفحتك الويب.

هذا هو التفاعل! إنها طريقة لبرنامج الحاسوب الخاص بك لوضع الكلمات والصور على صفحة الويب عن طريق النظر داخل الصناديق واستخدام ما بداخلها. إنها كأنها سحر لصفحات الويب الخاصة بك!

Property binding

إعداد الخصائص (Property Binding) في Angular هو عملية تساعدك على تعيين قيم لخصائص عناصر HTML أو توجيهات (Directives). يُستخدم إعداد الخصائص لأغراض مثل تبديل ميزات الأزرار، وتعيين المسارات برمجيًا، ومشاركة القيم بين مكونات Angular.

فلنشرح هذا بطريقة أكثر بساطة:

عندما تريد تغيير خاصية (property) معينة لعنصر HTML بناءً على قيمة في مكون Angular، يمكنك استخدام إعداد الخصائص.

على سبيل المثال، إذا كنت تريد تعيين قيمة لخاصية src لعنصر الصورة <img>، يمكنك القيام بذلك بهذه الطريقة:

```html

<img alt="item" [src]="itemImageUrl">

```

هنا، نستخدم الأقواس المربعة [] لتحديد أن الخاصية التي نريد تعيينها هي src. وقيمة src تأتي من متغير component يسمى itemImageUrl.

بالإضافة إلى ذلك، إذا كنت تستخدم [] (الأقواس المربعة)، فإن Angular سيفهم القيمة اليمنى كتعبير دينامي، وهذا يتيح لك تغيير القيمة بناءً على قيمة متغير أو تعبير. إذا لم تستخدم []، ستكون القيمة اليمنى تمامًا كما هي وستكون قيمة ثابتة.

مثلا، إذا كنت تريد تعيين النص "parentItem" لخاصية childItem في عنصر معين، يمكنك القيام بذلك على النحو التالي:

```html

<app-item-detail childItem="parentItem"></app-item-detail>

```

في هذا المثال، إذا لم تكن تستخدم []، Angular سيعامل "parentItem" كقيمة نصية ثابتة، وليس كقيمة متغير.

إذا كنت تريد تمكين أو تعطيل خاصية على عنصر معين (مثل تمكين أو تعطيل زر) بناءً على قيمة متغير في المكون الخاص بك، يمكنك القيام بذلك باستخدام [] كما في المثال التالي:

```html

<button type="button" [disabled]="isUnchanged">Disabled Button</button>

```

هنا، نستخدم [] للربط بين خاصية الـ `disabled` لزر الأنقرة وقيمة متغير الـ `isUnchanged`. إذا كانت قيمة `isUnchanged` تساوي `true`، سيتم تعطيل الزر.

إن إعداد الخصائص يُساعدك في تفعيل وتعطيل الخصائص أو تغييرها بناءً على البيانات والمتغيرات في تطبيقك

Angular.

Attribute binding

إعداد السمات (Attribute Binding) في Angular هو عملية تساعدك على تعيين قيم للسمات (الخصائص) مباشرة. باستخدام إعداد السمات، يمكنك تحسين إمكانية الوصول (accessibility)، تنسيق تطبيقك بشكل ديناميكي، وإدارة العديد من الفئات (classes) أو الأنماط (styles) في نفس الوقت.

لنشرح هذا ببساطة:

باستخدام إعداد السمات، يمكنك تعيين قيم للسمات (الخصائص) مباشرة على العناصر HTML. يكون القالب الأساسي للإعداد كالتالي:

```html

<p [attr.السمة-التي-تستهدفها]="التعبير"></p>

```

عندما يتحقق التعبير من قيمة `null` أو غير معرفة (`undefined`)، يقوم Angular بإزالة السمة بشكل كامل.

أحد استخدامات إعداد السمات الرئيسية هو تعيين سمات ARIA (Accessible Rich Internet Applications)، والتي تساهم في تحسين إمكانية الوصول للمستخدمين.

على سبيل المثال، يمكنك استخدام إعداد السمات لتعيين سمة ARIA لعنصر زر كالتالي:

```html

<button type="button" [attr.aria-label]="actionName">{{actionName}} with Aria</button>

```

وهنا نستخدم إعداد السمات لتعيين قيمة لسمة ARIA بناءً على قيمة `actionName`.

بالإضافة إلى ذلك، يمكن استخدام إعداد السمات بشكل شائع للتحكم في سمة `colspan` في الجداول. ذلك يُمكنك من جعل جداولك ديناميكية برمجيًا. على سبيل المثال، يمكن أن يتغير عدد الأعمدة التي يغطيها صف معين اعتمادًا على كمية البيانات التي يملأها تطبيقك في الجدول.

يمكنك استخدام إعداد السمات مع السمة colspan في الجداول بالشكل التالي:

```html

<tr><td [attr.colspan]="1 + 1">One-Two</td></tr>

```

هذا الربط يجعل الصف يمتد عبر عمودين. يجب ملاحظة أن هناك اختلاف بين اسم الخاصية والسمة في هذا السياق. في هذا المثال، نستخدم السمة colspan بأحرف صغيرة "s"، عكس الخاصية colSpan التي تأتي مع "S" كبير. بالتالي، يجب استخدام colspan بأحرف صغيرة عند استخدام إعداد السمات.

Class and style binding

إعداد الصفات والأنماط في Angular يساعدك في جعل صفحات الويب تبدو أفضل وتعمل بشكل مختلف. دعني أشرح لك ببساطة:

1. \*\*الصفات (Classes):\*\* الصفات تستخدم لتغيير مظهر العناصر على صفحة الويب. على سبيل المثال، إذا كنت تريد تغيير لون زر عندما تضغط عليه، يمكنك استخدام صفات. إذا كان زر الضغط عليه يعني أن الصفحة ستستخدم صفة معينة لجعل الزر يظهر بلون مختلف.

2. \*\*الأنماط (Styles):\*\* الأنماط تساعدك على تغيير الحجم أو اللون أو أي خاصية أخرى لعنصر على صفحة الويب. مثلما يمكنك تغيير لون خط النص أو حجم خلفية الزر.

الآن، في Angular، يمكنك استخدام البرمجة لتغيير هذه الصفات والأنماط ديناميكيًا. مثلاً، إذا كنت تريد تغيير لون زر عندما يحدث شيء ما في تطبيقك، يمكنك استخدام Angular للقيام بذلك.

هناك بعض الأمثلة:

- إذا كان شيء ما في التطبيق على البيع، فيمكنك جعل الزر يظهر بلون معين باستخدام `[class.sale]="true"` حيث يمكن أن يجعل الزر يظهر بلون البيع.

- إذا كنت تريد تغيير حجم عنصر على الصفحة، يمكنك استخدام `[style.width]="عرض"` لتغيير عرض العنصر بناءً على القيمة التي تقوم بتعيينها.

باختصار، إعداد الصفات والأنماط يساعد في جعل صفحات الويب تبدو وتعمل بالشكل الذي تريده

.

Event binding

إعداد الأحداث (Event Binding) في Angular يسمح لك بالاستماع لأحداث المستخدم، مثل النقرات بالفأرة، الضغط على الأزرار، وحركات الماوس.

لشرح ذلك ببساطة للأطفال، إليك مثال:

عندما تنقر على زر "حفظ" على موقع الويب، يمكننا استخدام إعداد الأحداث لسماع لهذا النقر والقيام بشيء ما عند حدوثه.

لذلك، إذا كان لدينا زر "حفظ" في تطبيق Angular، يمكننا أن نقول للتطبيق أن يستمع إلى النقرات على هذا الزر. للقيام بذلك، نستخدم البرمجة التالية:

```html

<button (click)="onSave()">حفظ</button>

```

هنا، نحدد أننا نرغب في الاستماع لحدث النقر (`click`) على الزر، وعندما يتم النقر، سيتم تشغيل الوظيفة `onSave()`.

بهذه الطريقة، يمكن للأطفال أن يتعلموا كيفية جعل التطبيق يتفاعل معهم عندما يقوموا بأشياء على الصفحة، مثل النقر على الأزرار أو الضغط على الأزرار على لوحة المفاتيح.

إن إعداد الأحداث هو طريقة رائعة للتفاعل مع التطبيق وإضافة المزيد من التفاعل إلى صفحة الويب.

Two-way binding

الربط ذو الاتجاهين (Two-way binding) في Angular هو طريقة تتيح لمكونات التطبيق مشاركة البيانات بينها. يُستخدم الربط ذو الاتجاهين للاستماع للأحداث وتحديث القيم بين مكونات الوالدين والأطفال بشكل متزامن.

في الأساس، الربط ذو الاتجاهين يجمع بين ربط الخصائص (Property binding) وربط الأحداث (Event binding)، وذلك باستخدام الصيغة `[()]`.

لشرح هذا ببساطة للأطفال:

عندما نريد تحديث قيمة على الصفحة وفي الوقت نفسه نريد أن نستمع للأحداث، نستخدم الربط ذو الاتجاهين. على سبيل المثال، إذا كان لدينا زر لزيادة ونقصان حجم الخط على الصفحة، يمكننا استخدام الربط ذو الاتجاهين.

لنشرح ذلك ببساطة:

1. لدينا مكون يُسمى `SizerComponent` يحتوي على خاصية تسمى `size` وأحداث تسمى `sizeChange`.

2. الخاصية `size` تُستخدم لتخزين حجم الخط على الصفحة.

3. الأحداث `sizeChange` تُستخدم للإبلاغ عندما يتم تغيير حجم الخط.

ثم، لدينا زرين في صفحتنا، أحدهما يقوم بزيادة حجم الخط والآخر يقوم بتقليل حجم الخط.

عندما ننقر على أحد الأزرار، تتغير قيمة الحجم، وتقوم الأحداث بإبلاغ الصفحة الرئيسية بالتغيير. وبالتالي، الصفحة الرئيسية تعرض النص بحجم جديد ويتم تحديث الخصية `fontSizePx` في الصفحة الرئيسية.

هذه هي طريقة تمكين الربط ذو الاتجاهين في Angular لجعل التفاعل بين الصفحة والأزرار يكون سهلاً ومتزامنًا.