

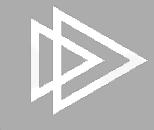
Translate the Application



Alisa Duncan

Developer | Google Developer Expert

linkedin.com/in/jalisaduncan/ | alisaduncan.dev



```
<element i18n>Translate me</element>
```

```
<element i18n attr="some text" i18n-attr>  
  Translate me and my attribute  
</element>
```

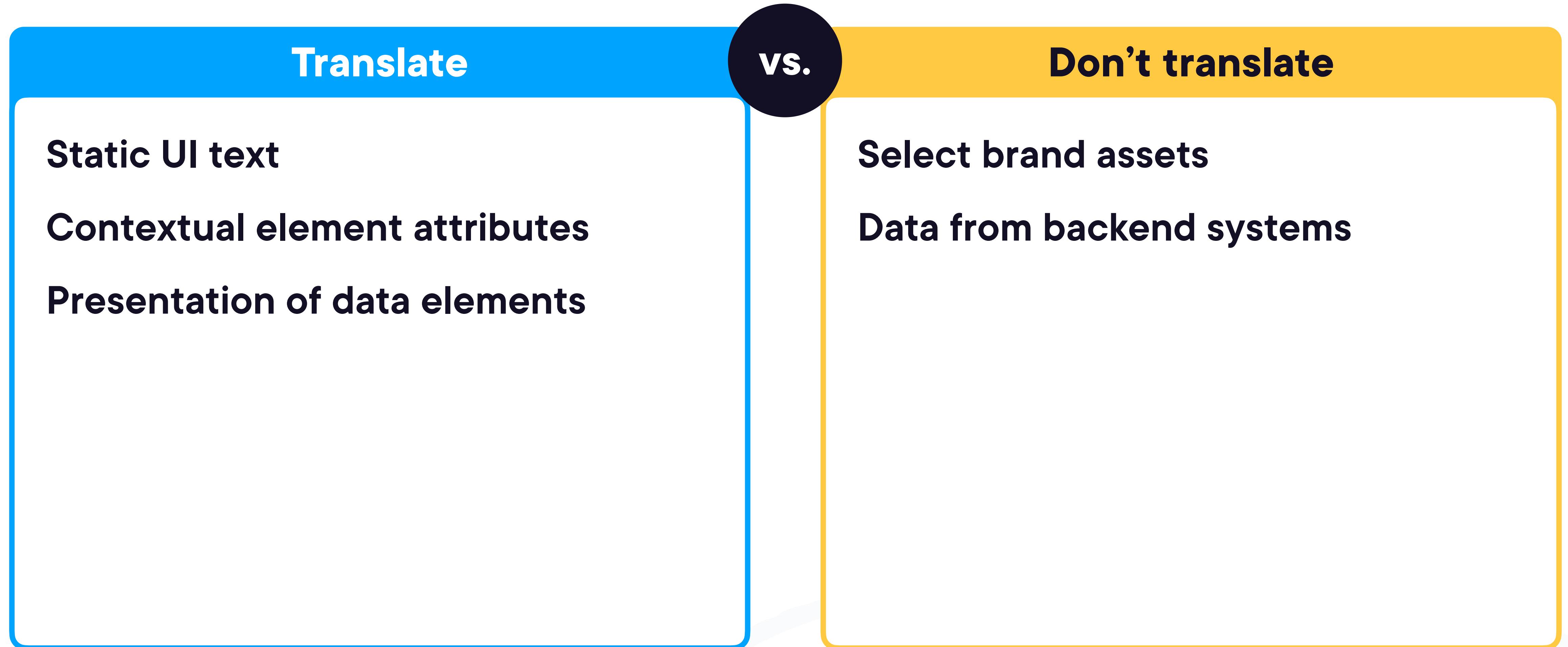
```
<p i18n title="friendly text" i18n-title>  
  {{todayDate | date:"short"}} is a great day to translate text  
</p>
```

Add i18n Attribute to HTML Elements and Attributes

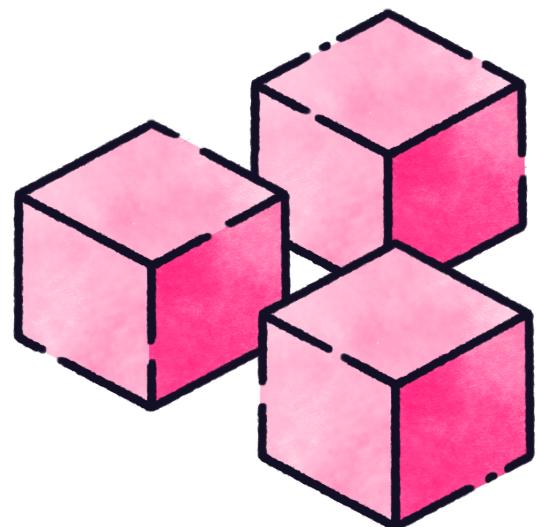
Translate static text in HTML elements and text defined in HTML attributes by marking the text with the i18n attribute.



Identify Elements to Translate

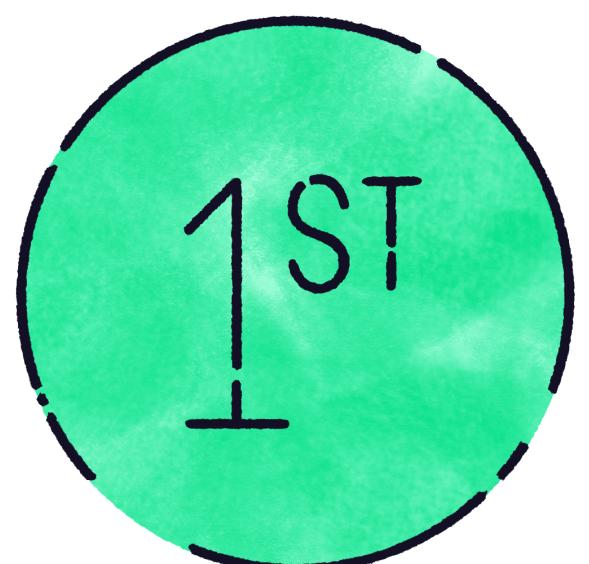


When Translations Aren't Straightforward



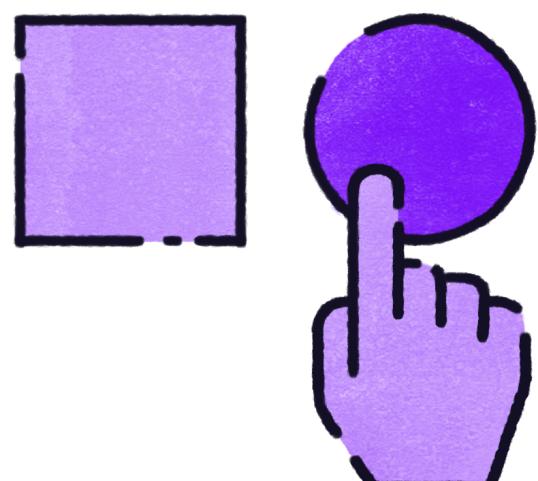
Plurals

One item and two items



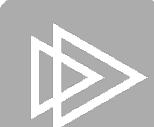
Ordinals

Fifth birthday, 2nd place, January 1st



Selects

Gendered references; she read a book, they're out at lunch



International Components for Unicode (ICU) Expressions in Angular

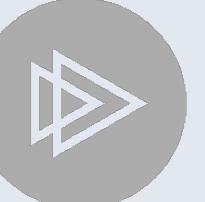
Format text with options using the plural or select ICU expressions

plural.html

```
<p i18n>You have {itemCount,  
    plural,  
    =0 {no items}  
    =1 {one item}  
    other {many items}  
    in the cart.  
</p>
```

select.html

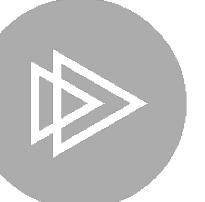
```
<p i18n>{gender,  
    select,  
    female {She is}  
    male {He is}  
    other {They are}}  
    shopping.  
</p>
```



```
<p i18n>The cart contains {itemCount, plural,  
  =0 {your healthiest option}  
  =1 {one {itemType, select, pie {pie}, cake {cake}} }  
  other {{itemCount}} {itemType, select, pie {pies}, cake {cakes}} }  
}</p>
```

Nest Expressions for Complex Translations

ICU expressions can combine, nest, and include interpolation.



ICU Expression Pipes

```
@Component({
  selector: 'app-cart-message',
  imports: [I18nPluralPipe],
  template: `<p i18n>
    The cart contains {{ itemCount() | i18nPlural: itemMapping }}
  </p>`,
})
export class CartMessageComponent {
  itemCount = input<number>(0);
  itemMapping: {[k: string]: string} = {
    '=0': 'nothing',
    '=1': 'one item',
    'other': 'many items',
  };
}
```



Localize in Code Use Cases

Dynamic menu
options

Table row headers

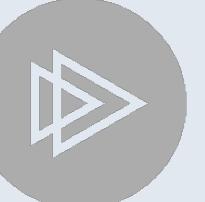
Overloading
complexity in
templates



Mark Text Using the \$localize API

cart-message.ts

```
test = $localize `Hello!`;  
  
@Component({  
  selector: 'app-cart-message',  
  template: `<p>{{ message }}</p>`,  
})  
export class CartMessageComponent {  
  itemCount = input<number>(2);  
  
  test = $localize `Hello!`;  
  message = $localize `${itemCount} cakes`;  
}
```



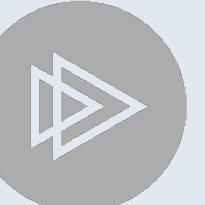
Add Context, Meaning, and Custom IDs

```
i18n="{meaning} | {description}@@{id}"
```

```
<h1 i18n="site header|Greeting for the course@@introGreeting">  
  Welcome to this course!  
</h1>
```

```

```



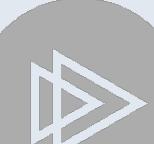
Adding Custom IDs Improves Maintainability

app.html

```
<h1 i18n="site header|Greeting for  
the course@@introGreeting">  
  Welcome to this course!  
</h1>  
<p i18n="@@remaining">  
  {{time}} remaining  
</p>
```

messages.json

```
{  
  "locale": "en-US",  
  "translations": {  
    "introGreeting": "Welcome to this  
course!",  
    "remaining": "{$INTERPOLATION}  
remaining",  
  }  
}
```



Name Interpolation Tokens

app.html

```
<p i18n="@@remaining">  
  {{time // i18n(ph ='minutes')}} remaining  
</p>
```

```
{  
  "translations": {  
    "remaining": "{$minutes} remaining",  
  }  
}
```



Add Custom IDs to Text in Code

cart-message.ts

```
test = $localize `:meaning|description@@id:Hi!`;

@Component({
  selector: 'app-cart-message',
  template: `<p>{{ message }}</p>`,
})
export class CartMessageComponent {
  itemCount = input<number>(2);

  message = $localize `:@@cartMessage:${itemCount}
cakes`;
}
```

