Common Internationalization tasks

Use the following Angular tasks to internationalize your project.

- Use built-in pipes to display dates, numbers, percentages, and currencies in a local format.
- Mark text in component templates for translation.
- Mark plural forms of expressions for translation.
- Mark alternate text for translation.

After you prepare your project for an international audience, use the Angular CLI to localize your project. Complete the following tasks to localize your project.

- Use the CLI to extract marked text to a *source language* file.
- Make a copy of the *source language* file for each language, and send all of *translation* files to a translator or service.
- Use the CLI to merge the finished translation files when you build your project for one or more locales.

Create an adaptable user interface for all of your target locales that takes into consideration the differences in spacing for different languages. For more details, see How to approach internationalization.

Prerequisites

To prepare your project for translations, you should have a basic understanding of the following subjects.

- Templates
- Components
- Angular CLI command-line tool for managing the Angular development cycle
- Extensible Markup Language (XML) used for translation files

Learn about common Angular internationalization tasks

```
<a href="guide/i18n-common-add-package" class="docs-card" title="Add the localize package">
    <section>Add the localize package</section>
    <p>Learn how to add the Angular Localize package to your project</p>
    <p class="card-footer">Add the localize package</p>
</a>
<a href="guide/i18n-common-locale-id" class="docs-card" title="Refer to locales by ID">
    <section>Refer to locales by ID</section>
    <p>Learn how to identify and specify a locale identifier for your project</p>
    <p class="card-footer">Refer to locales by ID</p>
</a>
<a href="guide/i18n-common-format-data-locale" class="docs-card" title="Format data based on
    <section>Format data based on locale</section>
    <p>Learn how to implement localized data pipes and override the locale for your project
```

```html
        <p class="card-footer">Format data based on locale</p>
</a>
<a href="guide/i18n-common-prepare" class="docs-card" title="Prepare component for translati
        <section>Prepare component for translation</section>
        <p>Learn how to specify source text for translation</p>
        <p class="card-footer">Prepare component for translation</p>
</a>
<a href="guide/i18n-common-translation-files" class="docs-card" title="Work with translation
        <section>Work with translation files</section>
        <p>Learn how to review and process translation text</p>
        <p class="card-footer">Work with translation files</p>
</a>
<a href="guide/i18n-common-merge" class="docs-card" title="Merge translations into the appli
        <section>Merge translations into the application</section>
        <p>Learn how to merge translations and build your translated application</p>
        <p class="card-footer">Merge translations into the application</p>
</a>
<a href="guide/i18n-common-deploy" class="docs-card" title="Deploy multiple locales">
        <section>Deploy multiple locales</section>
        <p>Learn how to deploy multiple locales for your application</p>
        <p class="card-footer">Deploy multiple locales</p>
</a>
```

@reviewed 2021-10-07