Engineer

ローカリゼーション

一つの市場の言語と慣習にソフトウェアを合わせること

どのようなものか

ローカリゼーションは、製品を特定の市場に合わせる仕事です。言語だけでなく、日付、数値、価格、名前、住所、一覧の順序について、その読者が期待する慣習にも合わせます。W3Cは、製品、アプリケーション、文書の内容を、対象市場の言語と文化に合い、そのほかの要件も満たすように適応させることと定義し、その市場をロケールと呼びます。英語にはlocalizationとlocalisationの綴りがあり、最初のlと最後のnの間に10文字あることから、l10nと略します。

W3Cは、ローカリゼーションが、インターフェースと文書の翻訳と同じ意味に捉えられがちな一方、実際には、はるかに複雑な仕事になることが多いと付け加えます。適応が必要になり得るものとして、数値、日付、時刻の書式、通貨、キーボード配列、文字の比較と並べ替え、記号、アイコン、色、その文化では誤解されたり配慮に欠けると受け取られたりする文章や画像、法的要件、住所、郵便番号、電話番号の書式と名前の順序、各文字体系の読みやすさを挙げます。商慣行が違う場合には、W3Cが例に挙げる会計のように、製品のロジック、視覚的なデザイン、表示方法の再検討まで必要になることがあります。

ローカリゼーションは、国際化を前提に、その後に行います。国際化は、一つの市場に固有の前提が入り込まないよう、コードで一度行います。ローカリゼーションは、コードが読むデータで、ロケールごとに行います。翻訳したメッセージや、保存した日付や金額を読者の期待する文字列に変える慣習です。十分に国際化した製品なら、コードを変えずにロケールを追加できます。不十分なら、新しい市場に入るたびに変更が必要になります。

慣習のデータの大半は、チームが自作する必要はありません。Unicode ConsortiumのCLDRプロジェクトは、100を超える言語について、専門家と各言語の母語話者が確認したデータを集めます。プロジェクト自身が挙げる例は、携帯電話のファイル一覧です。日付、数値、ファイルサイズの単位、アルファベット順は言語によって異なり、通常はCLDRが提供します。同プロジェクトによれ

ば、そのデータは、すべての現代的なOSとブラウザー、Java、C#、Swift、JavaScriptなどの言語に組み込まれています。コードが書式を自作せず、プラットフォームに求めれば、製品の書式の大半はそこから得られます。

概念を判断に生かす

「ローカリゼーションとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 日付、通貨、名前の並び順まで含め、販売先のそれぞれの市場で自然に読めるソフトウェアにします。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 英語・デンマーク語のi18nシステムを構築 — 12,027 英語・デンマーク語の言語ごとのメッセージ

英語・デンマーク語のi18nシステムを構築 お問い合わせ →

なぜ重要なのか

翻訳されていても、ローカリゼーションされていない製品には、外国向けという印象が残ります。ラベルはすべてデンマーク語なのに、日付は月が先、価格はドル記号のまま、町の一覧ではAarhusがAの位置にあります。読者はそれぞれをすぐに見つけ、そのたびに、別の人に向けて作られた製品だと感じます。

不一致によって、読み手の理解が変わる場合もあります。10/11/2026は、米国の読者には10月11日、英国の読者には11月10日です。どちらも正しい日付なのでエラーにはならず、一方が間違った日に行動します。1,234という文字列は、英語では千二百三十四、デンマーク語では一点二三四であり、数量や価格に千倍の差が生じます。誤訳は、間違いに見えます。慣習の間違いは、正しく見えるまま、誤って読まれます。

W3Cの一覧は、画面の外にも及びます。市場によって法的要件も事業の進め方も異なるので、ロケールごとに違う取引条件、請求書の項目、購入の流れが必要になることがあります。これらは事業上の判断であり、最初のメッセージを訳す前に、そのロケールの計画に含めるべきものです。

費用はロケールごとにかかり、一度払えば終わりではありません。新しいメッセージは毎回すべての言語に訳し、リリースのたびに、出荷前にすべてのロケールで読む必要があります。市場ごとに確認する慣習もあります。言語を増やすチームは、その後のすべてのリリースにも仕事を増やします。市場に入る判断が、リリース工程についての判断でもある理由です。

実際の活用例

コードが値とロケールを渡せば、仕事の大半はプラットフォームが行います。JavaScriptの `Intl` の書式機能は、両方を受け取って文字列を返します。

```
const when = new Date("2026-10-27T12:00:00Z");
const money = { style: "currency", currency: "DKK" };
for (const locale of ["en-US", "en-GB", "da", "nb", "sv", "uk"]) {
  const date = new Intl.DateTimeFormat(locale, { dateStyle: "short" });
  const price = new Intl.NumberFormat(locale, money);
  console.log(locale, date.format(when), price.format(800));
}
```

CLDR 47のデータを含むNode.js 25で実行すると、一つの日付と800デンマーククローネという一つの価格を、六通りで表示します。

```
en-US 10/27/26 DKK 800.00
en-GB 27/10/2026 DKK 800.00
da 27.10.2026 800,00 kr.
nb 27.10.2026 800,00 DKK
sv 2026-10-27 800,00 Dkr
uk 27.10.26 800,00 DKK
```

コードは、そのどれも選んでいません。デンマーク語の読者にはkr.、スウェーデン語の読者にはDkrと表示されます。スウェーデンでkrといえば、自国のクローナだからです。ノルウェー語とウクライナ語の読者には通貨コードが表示されます。日と月の順、区切り、小数点を表すコンマ、通貨の位置は、すべてロケールのデータで決まります。実際の出力で金額と通貨の間にあるのは、改行を禁止する空白U+00A0であり、両方を同じ行に保ちます。

並べ替えも同じです。デンマークの地名六つを、三つのロケール向けに並べます。

```
const towns = ["Aabenraa", "Ærø", "Åkirkeby", "Odense", "Østerbro", "Vejle"];
for (const locale of ["en", "da", "sv"]) {
  const order = new Intl.Collator(locale).compare;
  console.log(locale, towns.toSorted(order).join(", "));
}
```

```
en Aabenraa, Ærø, Åkirkeby, Odense, Østerbro, Vejle
da Odense, Vejle, Ærø, Østerbro, Aabenraa, Åkirkeby
sv Aabenraa, Odense, Vejle, Åkirkeby, Ærø, Østerbro
```

英語はæ、ø、åを、aとoの変種として扱います。デンマーク語はÆ、Ø、ÅをZの後に置き、古い綴りのAaをÅとして扱うため、Aabenraaは先頭から末尾の一つ前に移ります。スウェーデン語は、自国のÅ、Ä、ÖをZの後に置き、ÆをÄと、ØをÖとまとめます。どの順も、その言語の読者には正しいものです。英語順の一覧の終わり近くでAabenraaを探すデンマーク語の読者が、抜けていると思うのも無理はありません。

Hugoのサイトでも、テンプレートで同じ道具を使えます。Hugoの文書では、一つの金額を `lang.FormatCurrency` に渡します。

```
{{ 512.5032 | lang.FormatCurrency 2 "USD" }}
```

英語のページには$512.50、フランス語には512,50 $US、ドイツ語には512,50 $と表示されます。 `lang.FormatNumber` と `lang.FormatPercent` も、通常の数値とパーセントについて同じことを行います。

ローカリゼーションは、このサイトのサービスの一つにも含まれます。国際化とローカリゼーションのサービスは、顧客が販売するすべての市場で、日付、通貨、名前の並び順まで自然に読めるソフトウェアを目標とします。成果には、英語にある数値が翻訳から抜けたり、その言語で使わない表記が現れたりすると不合格にする、言語間の内容チェックがあります。指標が抜けたデンマーク語の説明二つと、英語式の引用符を使うウクライナ語の箇所十六個を見つけました。

問題が起きるところ

翻訳だけを予算に含めます。 新市場の計画にメッセージファイルしか含めなければ、新しい言語で出荷しても、日付、価格、住所、並び順は元の市場のままで、顧客が不足を見つけます。W3Cの一覧全体を使ってロケールを計画し、その市場に該当する項目を選び、母語の読み手にはメッセージファイルではなく、ビルドした製品を確認してもらいます。

言語をロケールと同一視します。 `en` だけでは、10月27日が `10/27/26` か `27/10/2026` かは分かりません。上の出力では、二つの英語ロケールは順序も年の表記も違います。言語しか保存しない製品は残りを推測し、一つの市場の慣習を、ほかの市場すべてにも当てはめます。利用者ごとに `en-GB` や `da-DK` のような完全なタグを保持し、明示的な設定があればそこから取得して、すべての書式機能に渡します。

ロケールから通貨を推測します。 CLDRの仕様は、既知の通貨コードなしに、金額をシステム間で受け渡してはならないと明言します。そうしなければ、同じ3.5が「一方の利用者には$3.50、別の利用者には€3.50と解釈される」ことがあるからです。金額の通貨は読者の言語とは独立しており、コードを省略してロケールから推測するのは、設計の悪いソフトウェアだけだとも付け加えます。北欧では、デンマーク語、ノルウェー語、スウェーデン語が自国の通貨をすべてkrと略すため、推測が間違っても気づきにくくなります。800と保存された価格を、一人は800デンマーククローネ、次の人は800ノルウェークローネと読みます。すべての金額をISO 4217コードと一緒に保存し、読者向けに書式を整えます。そうすれば、デンマーク語の読者にも、ノルウェーの価格は800,00 NOKと表示されます。

コードポイントで並べ替えます。 JavaScriptの既定の `sort` はUTF-16のコード単位を比較するため、ЄとIをウクライナ語のほかの文字より前に、ҐをЯの後に置きます。それらのコードポイントが、Aからяまでの範囲外にあるためです。その順では、 `["Ялта", "Ґанок", "Гора", "Єнот", "Іван", "Дуб"]` はЄнот、Іван、Гора、Дуб、Ялта、Ґанокになります。 `new Intl.Collator("uk").compare` なら、ウクライナ語のアルファベット順であるГора、Ґанок、Дуб、Єнот、Іван、Ялтаになります。表示する文章は、コード内のロケール対応の比較機能か、データベースの照合順序を使い、読者のロケールで並べます。コードポイント順は、読者に見えないキーのために残します。

書式済みの期待値を手で書いて比較します。 通常の空白を使った `800,00 kr.` を期待するテストは、U+00A0を入れる書式機能に対して失敗します。同じ改行禁止空白は、ノルウェー語、スウェーデン語、ウクライナ語で、数値を三桁ごとに区切るときにも使われます。ロケールのデータはCLDRのリリース間でも改訂されるため、プラットフォームの更新前に一致した文字列が、更新後は一致しない場合があります。期待値もコードと同じ書式機能とロケールで作るか、描画した文字列ではなく、保存した値と渡したロケールを検証します。

英語に合わせて配置の寸法を決めます。 Saveが入るボタンには、その倍の文字数のЗберегтиも入らなければなりません。Settingsが入るメニューには、デンマーク語のIndstillingerも必要です。固定幅では長い語が切れたり二行目に送られ

たりし、最初に見るのはその市場の読者であることがよくあります。文章に合わせて領域を広げ、翻訳する語の周りを固定幅にせず、製品が対応する最も狭い画面で、すべてのロケールを確認します。

参考資料

- ローカリゼーションと国際化 — W3Cの実務上の定義、ロケールごとに適応が必要になり得る項目、国際化との関係です。
- Hugoの多言語モード — テンプレートでの日付、通貨、数値、パーセントのローカライズと、翻訳表、言語別URLです。
- UTS #35、第3部：数値 — CLDRの数値と通貨の書式、金額には常に通貨コードを添えるという規則です。
- Unicode CLDRプロジェクト — 100を超える言語のロケールデータを集め、OS、ブラウザー、プログラミング言語に提供するプロジェクトです。

用語集

実際の制約に合わせて選ぶ

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「ローカリゼーションとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/localization/