i18n | l10n

The principles for an accessible web

hello!

你好!

hola!

Olá!

привет!

I am **Romulo Cintra**

- Software Architect - Silk Caixabank
- TC39
- OSS Enthusiast
- Teacher at Barcelona Tech School

Hello!
hej!
привет!
Olá!
Ciao!
你好!
Mbote!
Hola!
Hallo!
こんにちは
Bawo!
Oi!
Hello

~195 | ~6K

Countries

страны

Languages

Языки

i18n

Internationalization

a. *Designing and developing in a way that removes barriers to localization or international deployment. This includes such things as enabling the use of Unicode, or ensuring the proper handling of legacy character encodings where appropriate, taking care over the concatenation of strings, avoiding dependency in code of user-interface string values, etc.*

b. *Enabling code to support local, regional, language, or culturally related preferences. Typically this involves incorporating predefined localization data and features derived from existing libraries or user preferences. Examples include date and time formats, local calendars, number formats and numeral systems, sorting and presentation of lists, handling of personal names and forms of address, etc.*

https://www.w3.org/International/questions/qa-i18n.en

a. ***Designing*** *and developing* ***in a way that removes barriers to localization*** *or international deployment. This includes such things as enabling the use of Unicode, or ensuring the proper handling of legacy character encodings where appropriate, taking care over the concatenation of strings, avoiding dependency in code of user-interface string values, etc.*

b. ***Enabling code to support local, regional, language, or culturally related preferences****. Typically this involves incorporating predefined localization data and features derived from existing libraries or user preferences. Examples include date and time formats, local calendars, number formats and numeral systems, sorting and presentation of lists, handling of personal names and forms of address, etc.*

https://www.w3.org/International/questions/qa-i18n.en

Prepare the **web** to be more **understandable** by *everybody*

l10n

Localization

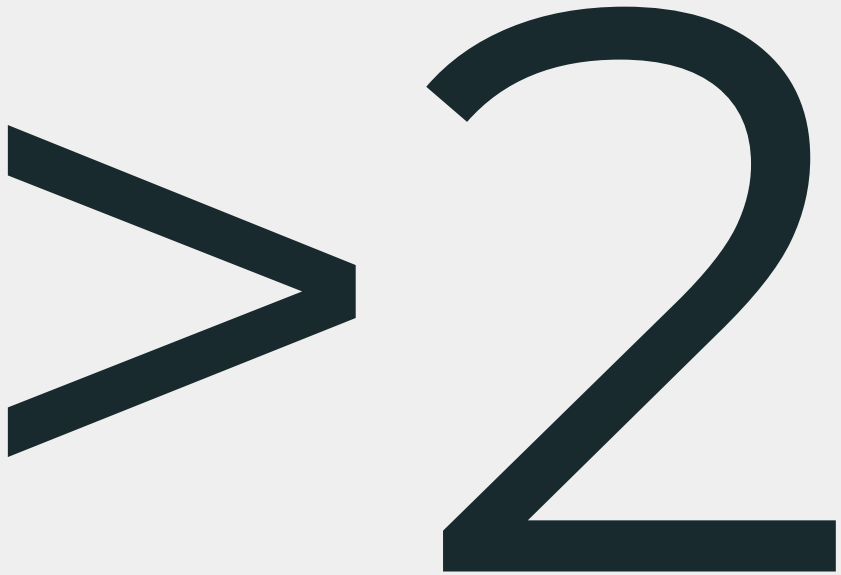
Localization refers to the adaptation of a product, application or document content to meet the language, cultural and other requirements of a specific target market (a locale).

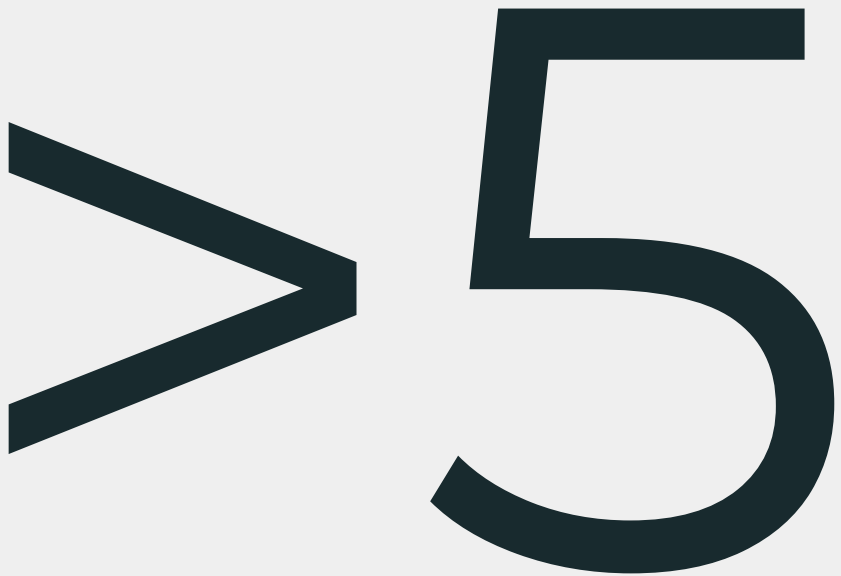
Localization refers to the ***adaptation of a product,*** *application or document content* ***to meet the language, cultural and*** *other requirements of a* ***specific target market*** *(a locale).*

Make the web more
familiar to everybody

Questions

How Many different
Languages do you support?


>2


>5

15

CaixaBank CaixaBankNow

Acceso a la banca digital CaixaBankNow

Seleccione cómo quiere conectarse a la banca digital CaixaBankNow

Castellano	Acceso a banca digital CaixaBankNow
Català	Accés a banca digital CaixaBankNow
Euskera	CaixaBankNow banka digitalerako sarbidea
Galego	Acceso a banca dixital CaixaBankNow
Valencià	Accés a banca digital CaixaBankNow
English	Acces to ditigal banking CaixaBankNow
Français	Accès à la banque en ligne CaixaBankNow
Deutsch	Zugang auf die Digitalbank CaixaBankNow
Italiano	Accesso a banca digitale CaixaBankNow
Português	Acesso à banca digital CaixaBankNow
Svenska	Åtkomst till digital bank CaixaBankNow
ελληνικά	Πρόσβαση στην ηλεκτρονική τραπεζική της CaixaBankNow
Dansk	Adgang til den elektroniske bank CaixaBankNow
Slovenčina	Vstup do digitálnej banky CaixaBankNow
Polski	Dostęp do banku cyfrowego CaixaBankNow
Nederlands	Toegang tot de digitale bank CaixaBankNow
Suomi	Pääsy CaixaBankNow verkkopankkiin
čeština	Přístup do digitální banky CaixaBankNow
Magyar	Belépés a CaixaBankNow digitális banki szolgáltatásokba
中文	访问数字银行CaixaBankNow
русский	Доступ к цифровому банку CaixaBankNow
Norsk	Tilgang til nettbanken CaixaBankNow
Türkçe	CaixaBankNow dijital bankacılığa erişim

@romulocintra

i18n

- *Facilitate localization*
- *Support Cultural specific features*

l10n

- *Linguistic adaptation*
- *Cultural, Marketing, Legal …*

Globalization (g11n)

“Internationalization is the **design** and **development** of a **product**, application or document content **that enables easy** localization for **target** audiences that vary in **culture**, **region**, or **language**.”

https://blog.mozilla.org/l10n/2011/12/14/i18n-vs-l10n-whats-the-diff

i18n !== l10n

...and sometimes @#@## happens! ...

They really meant this ?

Or This ?

РУССКАЯ
БАНЯ
Russian
crematorium
С легким
паром!
тел.: 8-982-45-200-74

Why this is so important ?

... to share the correct message.

The Impact of i18n as Accessibility

- Wider Audience
- Better conversion SEO/Marketing
- Improve the web - Help the people

Let's get Started with i18n - Basic Use Case

```
'es-ES': {
  HELLO_WORLD: '¡Hola mundo!'
},
'en-GB': {
  HELLO_WORLD: 'Hello world!'
```

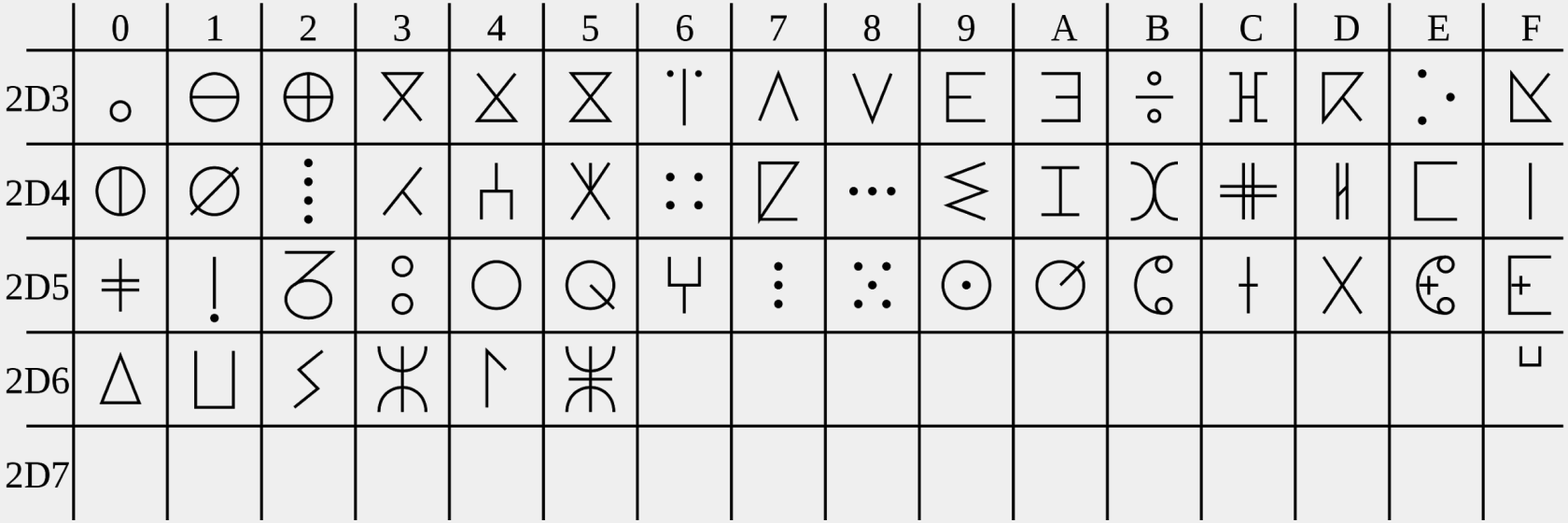
What is a

Message Format ?

Wait!!

some basic concepts first ...

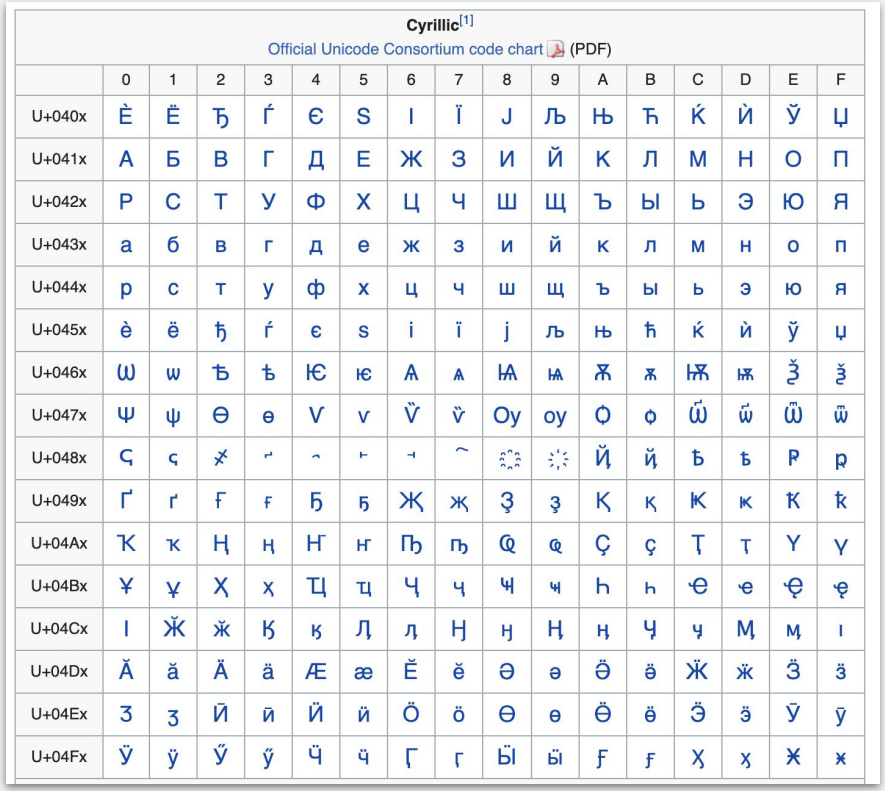
Unicode



Tifinagh Alphabet

Unicode

- Maps a unique number for every character, independent of platform or language
- Popular Unicode Transformation units - UTF-8, UTF-16 , UTF-32



Cyrillic[1]

Official Unicode Consortium code chart (PDF)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+040x	Ѐ	Ё	Ђ	Ѓ	Є	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ	Ѝ	Ў	Џ
U+041x	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
U+042x	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
U+043x	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
U+044x	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
U+045x	ѐ	ё	ђ	ѓ	є	ѕ	і	ї	ј	љ	њ	ћ	ќ	ѝ	ў	џ
U+046x	Ѡ	ѡ	Ѣ	ѣ	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ	Ѭ	ѭ	Ѯ	ѯ
U+047x	Ѱ	ѱ	Ѳ	ѳ	Ѵ	ѵ	Ѷ	ѷ	Ѹ	ѹ	Ѻ	ѻ	Ѽ	ѽ	Ѿ	ѿ
U+048x	Ҁ	ҁ	҂	҃	◌҃	◌҄	◌҅	◌҆	◌҇	◌҈	Ҋ	ҋ	Ҍ	ҍ	Ҏ	ҏ
U+049x	Ґ	ґ	Ғ	ғ	Ҕ	ҕ	Җ	ж	Ҙ	ҙ	Қ	қ	Ҝ	ҝ	Ҟ	ҟ
U+04Ax	Ҡ	ҡ	Ң	ң	Ҥ	ҥ	Ҧ	ҧ	Ҩ	ҩ	Ҫ	ҫ	Ҭ	ҭ	Ү	ү
U+04Bx	Ұ	ұ	Ҳ	ҳ	Ҵ	ҵ	Ҷ	ҷ	Ҹ	ҹ	Һ	һ	Ҽ	ҽ	Ҿ	ҿ
U+04Cx	Ӏ	Ӂ	ӂ	Ӄ	ӄ	Ӆ	ӆ	Ӈ	ӈ	Ӊ	ӊ	Ӌ	ӌ	Ӎ	ӎ	ӏ
U+04Dx	Ӑ	ӑ	Ӓ	ӓ	Ӕ	ӕ	Ӗ	ӗ	Ә	ә	Ӛ	ӛ	Ӝ	ӝ	Ӟ	ӟ
U+04Ex	Ӡ	ӡ	Ӣ	ӣ	Ӥ	ӥ	Ӧ	ӧ	Ө	ө	Ӫ	ӫ	Ӭ	ӭ	Ӯ	ӯ
U+04Fx	Ӱ	ӱ	Ӳ	ӳ	Ӵ	ӵ	Ӷ	ӷ	Ӹ	ӹ	Ӻ	ӻ	Ӽ	ӽ	Ӿ	ӿ

https://bit.ly/3440CtU

Unicode

- Maps a unique number for every character, independent of platform or language
- Popular Unicode Transformation units - UTF-8, UTF-16 , UTF-32

- *Data repository that provides locales data*
- *Languages, currencies,countries , formatting , time zones ...*

- *OSS project that holds i18n for C++ Java libraries*
- *Collation , normalization etc …*
- *Used as a base for i18n in other languages*

Message Format

```
# Using YAML for simplicity of example.
hello: Hello, {user}!
# offset allows the categories to be subtracted by the specified amount before being processed. It does not, however, affect the exact-match system.
party: {user} has invited {player_count, plural, offset:1, =1 {nobody} one {a player} other {# players}} to {user_gender, select, male {his}, female {her}, other {their}} party.
```

```
// Using the simple form of https://messageformat.github.io/messageformat/page-build
import msg from './example.yaml'
function say(messageKey, options) { console.log(msg[messageKey](options)) }

say('hello', {user: 'Jimmy'})  // Hello, Jimmy!
say('party', {user: 'Whales', player_count: 5000, user_gender: 'male'})   // Whales has invited 4999 players to his game.
say('party', {user: 'Dolphin', player_count: 20, user_gender: 'other'})   // Dolphin has invited 19 players to their game.
say('party', {user: 'Elephant', player_count: 1, user_gender: 'female'})  // Elephant has invited nobody to her game.
```

https://bit.ly/2WgFttI

Solutions

How they solve it in other languages ?

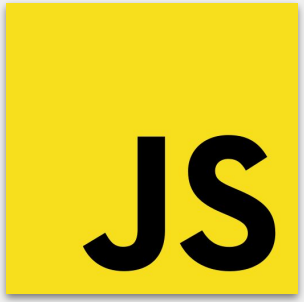
```
<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2">
   <!-- placeholder for a special unicode symbol -->
   <string name="star_rating">Check out our 5
       <xliff:g id="star">\u2605</xliff:g>
   </string>

   <!-- Placeholder for a name -->
   <string name="prod_name">Learn more at
      <xliff:g id="prod_gamegroup">Group</xliff:g>
   </string>
   ...
</resources>
```

https://bit.ly/31PN7N7

iOS

```
let alertTitle = NSLocalizedString("Welcome", comment: "")
let alertMessage = NSLocalizedString("Thank you for trying this app, you are a great person!", comment:
"")
let cancelButtonText = NSLocalizedString("Cancel", comment: "")
let signupButtonText = NSLocalizedString("Signup", comment: "")

let alert = UIAlertController(title: alertTitle, message: alertMessage, preferredStyle:
UIAlertControllerStyle.Alert)
let cancelAction = UIAlertAction(title: cancelButtonText, style: UIAlertActionStyle.Cancel, handler:
nil)
let signupAction = UIAlertAction(title: signupButtonText, style: UIAlertActionStyle.Default, handler:
nil)
```

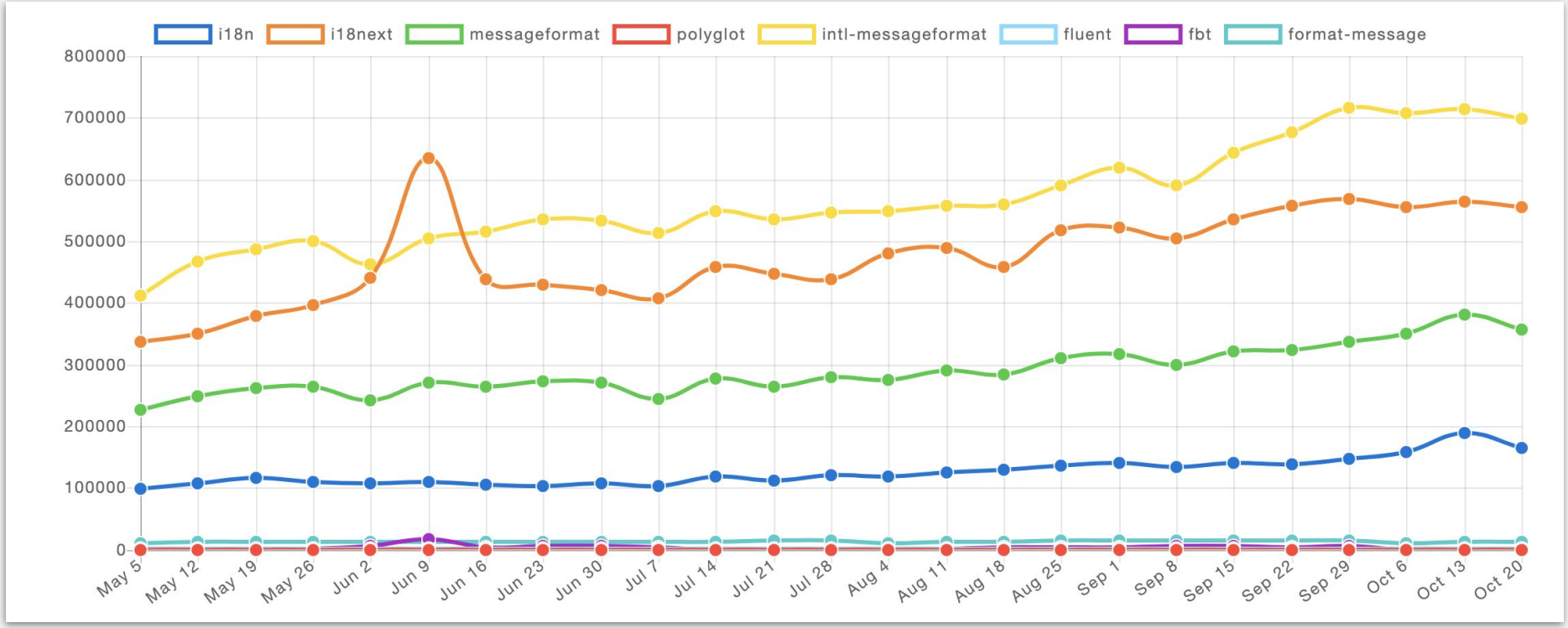
https://apple.co/344f3ho

& about 

How did they solved it ?

We didn't *not yet...*

But some popular libraries did...


i18n
i18next
messageformat
polyglot
intl-messageformat
fluent
fbt
format-message
800000
700000
600000
500000
400000
300000
200000
100000
0
May 5
May 12
May 19
May 26
Jun 2
Jun 9
Jun 16
Jun 23
Jun 30
Jul 7
Jul 14
Jul 21
Jul 28
Aug 4
Aug 11
Aug 18
Aug 25
Sep 1
Sep 8
Sep 15
Sep 22
Sep 29
Oct 6
Oct 13
Oct 20

Actual
Use Case for

i18n

Javascript

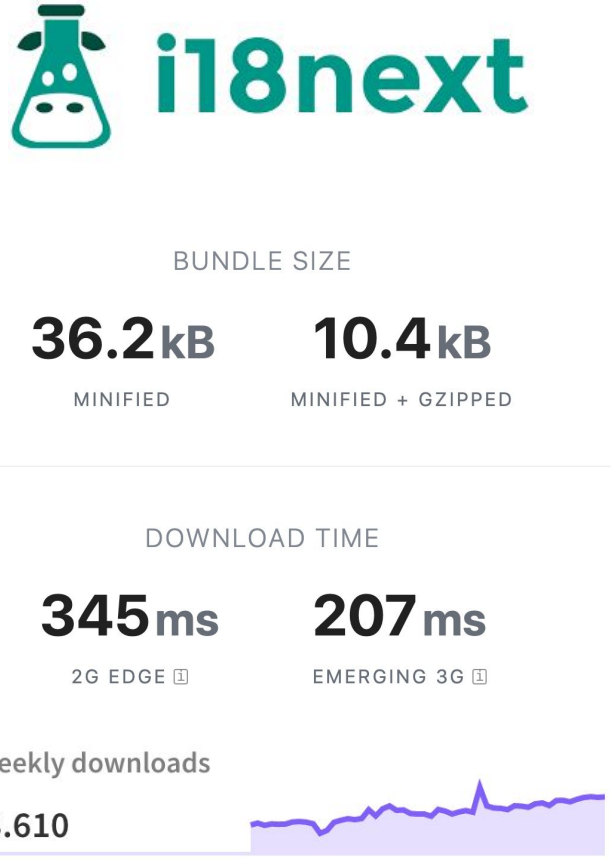
~2M

Weekly
downloads

i18next

- Detect the user language
- Load the translations
- Optionally cache the translations
- Extensible by their plugin system
- Translation functions like interpolation, formatting and plurals.

https://www.i18next.com/overview/api



Keys / Messages

i18next

```
{
 "button": {
     "save": "save {{count}} change",
     "save_plural": "save {{count}} changes"
 }
},

{
 "friend_male": "A boyfriend",
 "friend_female": "A girlfriend",
 "friend_male_plural": "{{count}} boyfriends",
 "friend_female_plural": "{{count}} girlfriends"
}
```

Nest / interpolation

Plurals

Javascript API

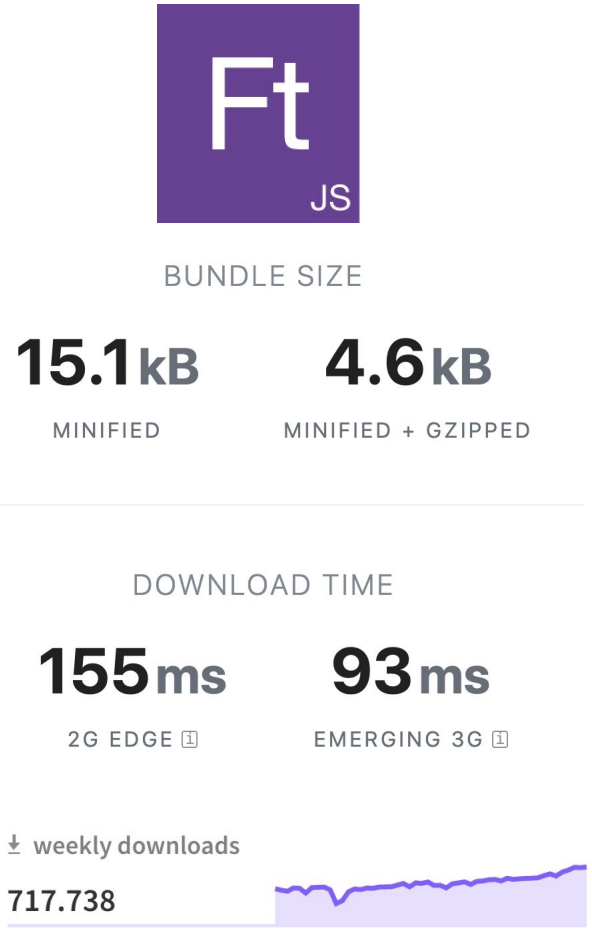
i18next

```
i18next.init({
   lng: 'de',
   resources: {
       de: { translation: {"hello world": "hallo Welt"}} }
});

// Nesting Object and Arrays
i18next.t('button', { returnObjects: true, count: 1 });

i18next.t('friend'); // -> "A friend"
i18next.t('friend', { context: 'male' }); // -> "A boyfriend"
i18next.t('friend', { context: 'female' }); // -> "A girlfriend"
i18next.t('friend', {context: 'male', count: 100}); // -> "100 boyfriends"
```

intl-messageformat

- FormatJS is aligned with: ECMAScript Internationalization API (ECMA-402), Unicode CLDR, and ICU Message syntax
- Offers interpolation and pluralization.

https://formatjs.io/



Keys / Messages

```
var MESSAGES = {
    'en-US': {
        NUM_PHOTOS: 'You have {numPhotos, plural, ' +
            '=0 {no photos.}' +
            '=1 {one photo.}' +
            'other {# photos.}}'
    },

    'es-MX': {
        NUM_PHOTOS: 'Usted {numPhotos, plural, ' +
            '=0 {no tiene fotos.}' +
            '=1 {tiene una foto.}' +
            'other {tiene # fotos.}}'
    }
};
```

Javascript API

```
var output;

var enNumPhotos = new IntlMessageFormat(MESSAGES['en-US'].NUM_PHOTOS, 'en-US');
output = enNumPhotos.format({numPhotos: 1000});
console.log(output); // => "You have 1,000 photos."

var esNumPhotos = new IntlMessageFormat(MESSAGES['es-MX'].NUM_PHOTOS, 'es-MX');
output = esNumPhotos.format({numPhotos: 1000});
console.log(output); // => "Usted tiene 1,000 fotos."
```

Finding more inspiration...

The new kids on the block

fluent

- Client- and server-side.
- Compatible with ICU and ECMA 402
- Date, time, and number formatting. Plural categories. Full bidirectionality support. Custom formatters. Easy-to-read syntax.
- Message Reference
- Multi-line blocks

```
import { FluentBundle, ftl } from 'fluent';

const bundle = new FluentBundle('en-US');

const errors = bundle.addMessages(ftl`
   -brand-name = Foo 3000
   welcome = Welcome, { $name }, to {-brand-name
}!`);

const welcome = bundle.getMessage('welcome');

bundle.format(welcome, { name: 'Anna' });
// → 'Welcome, Anna, to Foo 3000!'
```

- FBT works by transforming your <fbt> and fbt(...) constructs via Babel plugins.
- Translate payloads generated at build-time.
- FBT creates references of all possible variations for the given fbt phrase and accesses this at runtime.

HTML MARKUP

```
<button>
  <fbt desc="Canonical intro text">
    Hello, World!
  </fbt>
</button>
```

RESULTING KEY

```
{
  "hashToText":{
    "ni7kanCF2RfGZAS9mDOToQ==":
     "Hello, World!"
  },
  ...,
  "desc": "Canonical intro text"
}
```

What if the **browser** can do this ?

Intl.MessageFormat

Should be the future!!!!

We are *WIP* working on it...

- Finding Stakeholders and interested people
- Meeting 1-1 and getting requirements
- Defining roadmap and thinking ahead

https://qrgo.page.link/SRjMK

Wait!! *again*

We actually **have** lots of **i18n Capabilities** in the **browser** ...

Intl.DateTimeFormat

dateStyle | timeStyle | formatRange

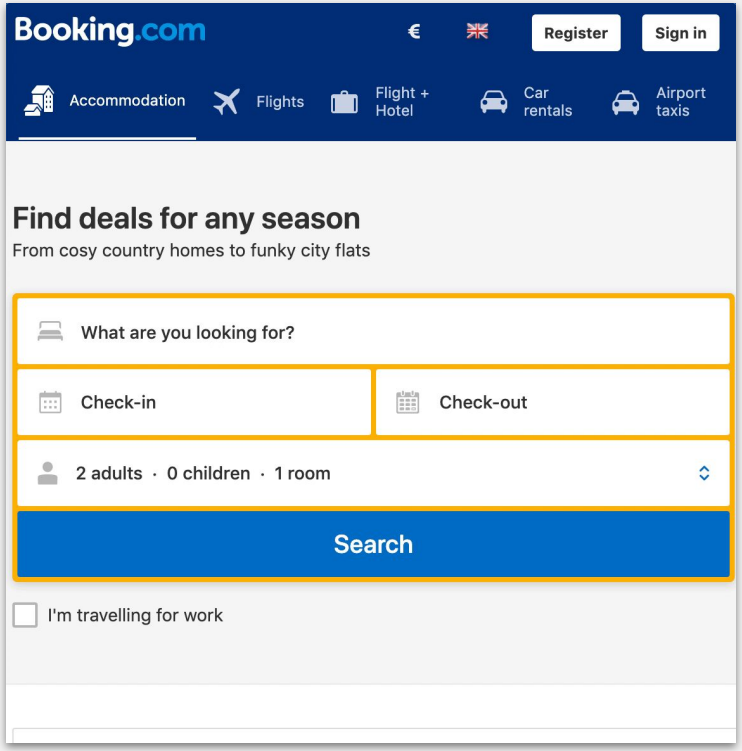
STAGE 3

Use Case | *Intl.*DateTimeFormat

You don't (may not) need...

```
const myDate = new Date();

new Intl.DateTimeFormat('ru', { timeStyle : 'short'}).format(myDate);

// short → 19:49
// medium → 19:49:17
// long → 19:49:17 GMT+2
// full → 19:49:17 Центральная Европа, летнее время
```

```
const myDate = new Date();

new Intl.DateTimeFormat('en', { dateStyle : 'full'}).format(myDate);

// → Saturday, 19 October 2019
// → Scyббота, 19 октября 2019 г.
// → sábado, 19 de octubre de 2019
```




```
const checkIn = new Date('10-20-2019 , 15:30');
const checkOut = new Date('11-03-2019 , 15:30');

const holiday = new Intl.DateTimeFormat("en", {
    year: '2-digit',
    month: 'long',
    day: 'numeric',
    hour: 'numeric',
    minute: 'numeric'
});

holiday.formatRange(checkIn , checkOut)
// October 20, 19,3:30 PM – November 3,19, 3:30 PM
// 10/20/19, 3:30 PM – 11/3/19, 3:30 PM
```

Intl.RelativeTimeFormat

Use Case | *Intl.*RelativeTimeFormat

The World Wide Web debuted **28 years**, **2 months**, **21 days**, **6 hours**, **21 minutes** and **30 seconds** ago!

```
const myTime = new Intl.RelativeTimeFormat('ru', { style: 'narrow' });

myTime.format(2 , 'quarter');

//Style Narrow :  +2 кв. → in 2 qtrs. → dentro de 2 trim.

//Style Long : через 2 квартала → in 2 quarters → dentro de 2 trimestres
```

```
const myTime = new Intl.RelativeTimeFormat('ru', { style: 'narrow' });

myTime.format(2 , 'day');

//Style Narrow :  +2 дн. → in 2 days → dentro de 2 días
//Style Long : через 2 дня  → dentro de 2 días

myTime.format(-1 , 'day');

//Style Narrow :  -1 дн. → 1 day ago → hace 1 día
//Style Long : 1 день назад  → 1 day ago → hace 1 día
//Numeric(auto) : вчера → yesterday → ayer
```

```
const myTime = new Intl.RelativeTimeFormat('ru', { style: 'narrow' });

myTime.format(20 , 'seconds');

//Style Narrow :  +20 с → in 20 sec. → dentro de 20 s

//Style Long : через 20 секунд  → in 20 seconds → dentro de 20 segundos
```

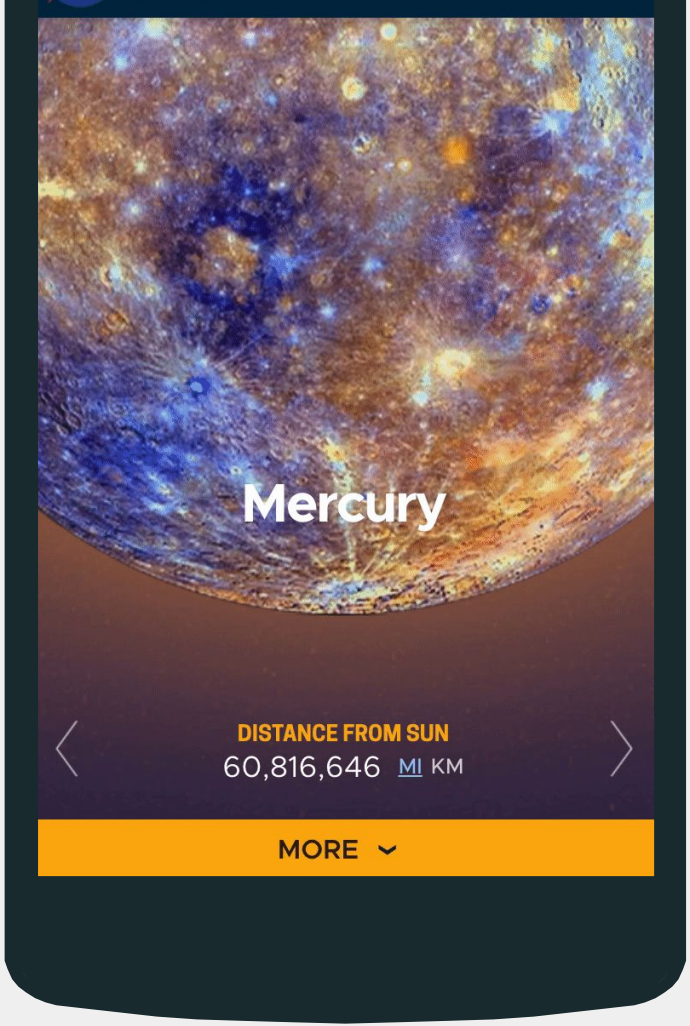
Intl.NumberFormat

Unit | Notation | Sign

@romulocintra

Use Case | *Intl.*NumberFormat

- Format Using Unit / Notation and Sign
- Multiples alternatives to format our daily use numbers in different formats
  - length-meter
  - mile-per-hour
  - 140+ available, listed in spec
- New ways to show formatted numbers signs and style according your locale



UNIT

```
new Intl.NumberFormat("ru", {
  style: "unit",
  unit: "liter",
  unitDisplay: "long"
}).format(16);
// → 16 литров → 16 liters → 16 litros
```

UNIT

```
new Intl.NumberFormat("ru", {
  style: "unit",
  unit: "kilometer-per-hour",
  unitDisplay: "long"
}).format(120);
// → 120 километров в час
// → 120 kilometers per hour
// → 120 kilómetros por hora
```

UNIT

```
new Intl.NumberFormat("ru", {
  style: "unit",
  unit: "mile-per-hour",
  unitDisplay: "narrow"
}).format(50);
// → 50 миль/ч → 50mph → 50mi/h
```

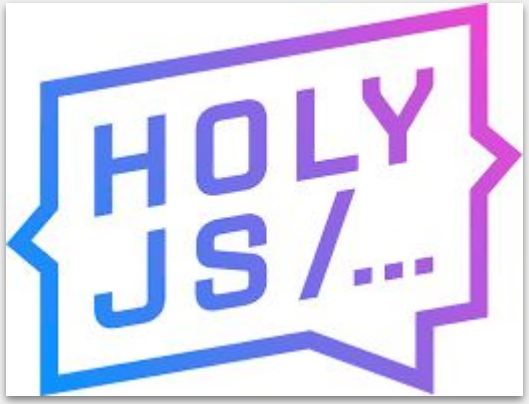
Notation

```
const nbr = 987654321;

new Intl.NumberFormat('ru', { notation: 'scientific' }).format(nbr);
// →  9,877E8  → 9.877E8 (en-US)


new Intl.NumberFormat('ru', { notation: 'engineering' }).format(nbr);
// →  987,654E6  → 987.654E6 (en-US)


new Intl.NumberFormat('ru', { notation: 'compact' }).format(nbr);
// →  988 млн  → 988M (en-US) → 9.9亿 (zn-CN)


new Intl.NumberFormat('ru', { notation: 'compact', compactDisplay: 'long'
}).format(nbr);
// → 988 миллионов → 988 millions (fr)
```

Intl.ListFormat

Use Case | *Intl.*ListFormat

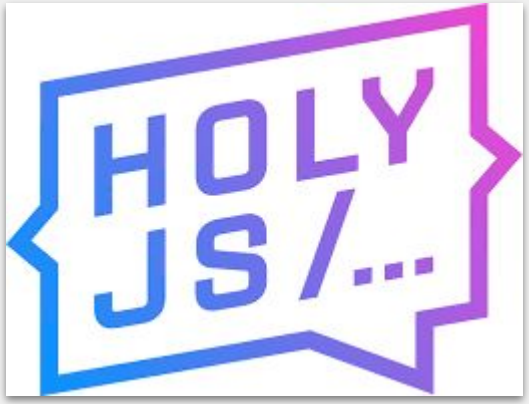
l'am attending at ...




```
var holyJS = ['St. Petersburg', 'Moscow'];

new Intl.ListFormat('ru', { style: 'long', type:
'conjunction' }).format(holyJS);
// → Moscow и St. Petersburg
// → Moscow and St. Petersburg
```

Use Case | *Intl.*ListFormat

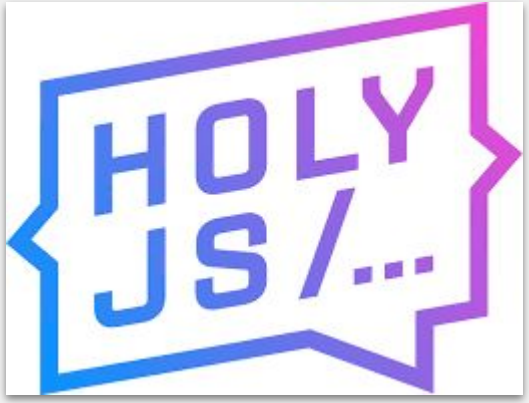
l’am attending at ...




```
var holyJS = ['St. Petersburg', 'Moscow'];

new Intl.ListFormat('ru', { style: 'short', type:
'disjunction' }).format(holyJS);
// → Moscow или St. Petersburg
// → Moscow or St. Petersburg
```

Use Case | *Intl.*ListFormat

I'am attending at ...




```
var holyJS = ['St. Petersburg', 'Moscow'];

new Intl.ListFormat('ru', { style: 'narrow', type:
'unit' }).format(holyJS);
// Moscow St. Petersburg
```

Intl.PluralRules

STAGE 4

Use Case | *Intl.*PluralRules

2019 DRIVERS CHAMPIONSHIP

1	Lewis **HAMILTON**	MERCEDES		338
2	Valtteri **BOTTAS**	MERCEDES		274
3	Charles **LECLERC**	FERRARI		221
4	Max **VERSTAPPEN**	RED BULL RACING		212
5	Sebastian **VETTEL**	FERRARI		212
6	Carlos **SAINZ**	MCLAREN		76
7	Pierre **GASLY**	TORO ROSSO		73

```
const pr = new Intl.PluralRules('en-US', {
  type: 'ordinal'
});

const suffixes = new Map([
  ['one',   'st'],
  ['two',   'nd'],
  ['few',   'rd'],
  ['other', 'th'],
]);

const formatOrdinals = (n) => {
  const rule = pr.select(n);
  const suffix = suffixes.get(rule);
  return `${n}${suffix}`;
};

formatOrdinals(1);   // '1st'
formatOrdinals(2);   // '2nd'
formatOrdinals(3);   // '3rd'
formatOrdinals(4);   // '4th'
```

Use Case | *Intl.*PluralRules

```
const suffixes = new Map([
  ['one',   'cat'],
  ['other', 'cats'],
]);


const pr = new Intl.PluralRules('en-US');


const formatCats = (n) => {
  const rule = pr.select(n);
  const suffix = suffixes.get(rule);
  return `${n} ${suffix}`;
};

formatCats(1);   // '1 cat'
```

Use Case | *Intl.*PluralRules

```
const suffixes = new Map([
  ['one',   'cat'],
  ['other', 'cats'],
]);


const pr = new Intl.PluralRules('en-US');


const formatCats = (n) => {
  const rule = pr.select(n);
  const suffix = suffixes.get(rule);
  return `${n} ${suffix}`;
};

formatCats(1);   // '1 cat'
formatCats(0);   // '0 cats'
```

Use Case | *Intl.*PluralRules

```
const suffixes = new Map([
  ['one',   'cat'],
  ['other', 'cats'],
]);


const pr = new Intl.PluralRules('en-US');


const formatCats = (n) => {
  const rule = pr.select(n);
  const suffix = suffixes.get(rule);
  return `${n} ${suffix}`;
};

formatCats(1);   // '1 cat'
formatCats(0);   // '0 cats'
formatCats(0.5); // '0.5 cats'
```

Use Case | *Intl.*PluralRules

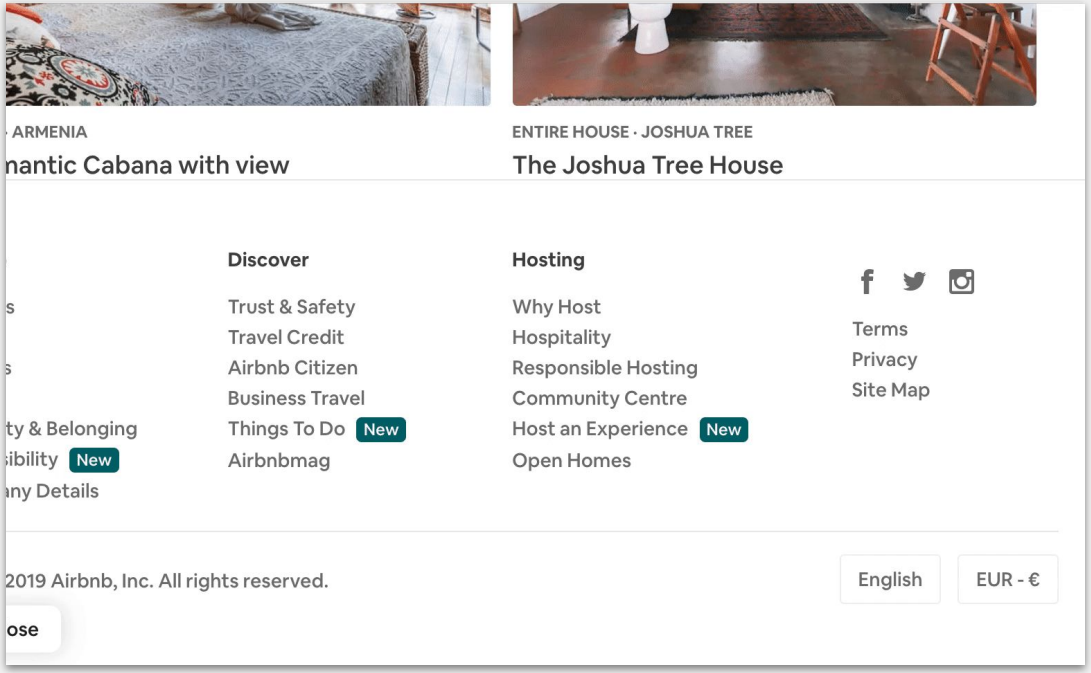
+

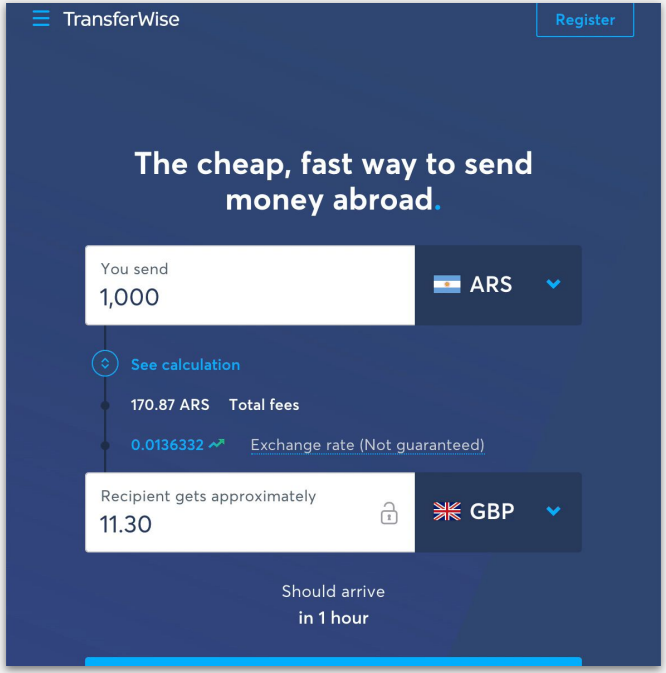
```
const suffixes = new Map([
  ['one',   'cat'],
  ['other', 'cats']
]);


const pr = new Intl.PluralRules('en-US');


const formatCats = (n) => {
  const rule = pr.select(n);
  const suffix = suffixes.get(rule);
  return `${n} ${suffix}`;
};

formatCats(1);   // '1 cat'
formatCats(0);   // '0 cats'
formatCats(0.5); // '0.5 cats'
formatCats(1.5); // '1.5 cats'
```

Intl.DisplayNames

STAGE 2

Use Case | *Intl.*DisplayNames





```
const currencyNames = new Intl.DisplayNames(['en'], {type: 'currency'});

currencyNames.of('USD'); // "US Dollar"
currencyNames.of('EUR'); // "Euro"
currencyNames.of('TWD'); // "New Taiwan Dollar"
currencyNames.of('CNY'); // "Chinese Yuan"
```

```
const languageNames = new Intl.DisplayNames(['en'], {type: 'language'});

languageNames.of('fr'); // "French"
languageNames.of('de'); // "German"
languageNames.of('fr-CA'); // "Canadian French"
languageNames.of('zh-Hant'); // "Traditional Chinese"
```

Intl. The Future

WIP

Intl.**MessageFormat**

- Is really needed and we hope to have it soon...
- Is a greenfield in JS
- Is time to review the actual State of message format since the original implementations

Impact

- All web universe , Companies , Projects , Libraries
- A new Chapter in i18n
- A better and more international web for everybody

Unique

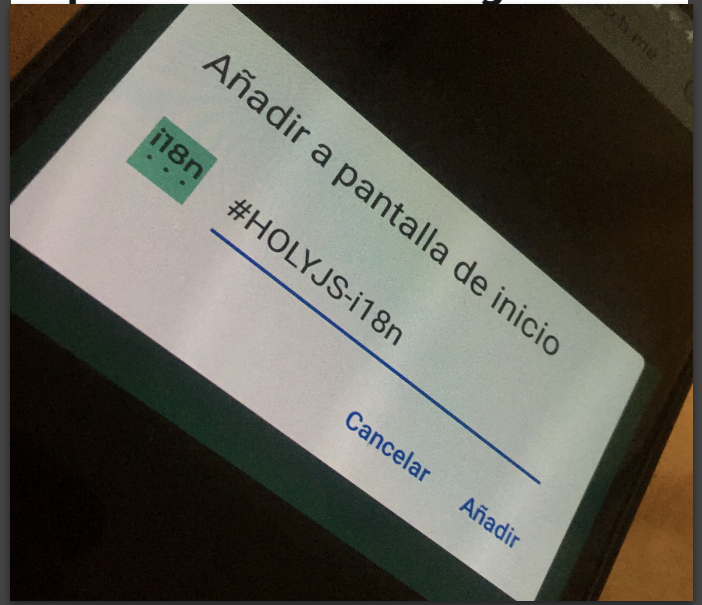
- Make it cross-platform - Android / iOS / Web
- Make it easy to integrate with tools
- Modular and plugable to add your own formatters and libraries
- Make it native and not System Dependent

Experience

- Better integrations
- Developers and Translators serving the same purpose
- Should be easy and accessible , no mather your expertise

Slides here !!!!!!!!

https://i18-l10n-slides.glitch.me/



Спасибо большое

Muchas Gracias

Thanks

谢谢啦