num2words

v0.2.0 2026-05-07 LGPL-3.0-or-later

Convert numbers to their written word form.

Mario Vago Marzal
https://github.com/mariovagomarzal

A Typst package to convert numbers to their written word form. It provides a single function, `#num2words`, that automatically adapts to the document's language and supports multiple output forms.

Table of Contents

Part I

Usage

I.1 The `num2words` function

The package provides a single main function, `#num2words`, that converts numbers to their written word form. To use it, import the package and call the function with a number:

```
#import "@preview/num2words:0.2.0": num2words

#num2words(42)
```

forty-two

The `#num2words` function always takes a positional `⟨number⟩` argument and an optional `⟨lang⟩` keyword argument (explained in the next section). Any other arguments are forwarded to the language-specific converter function. This means that the available options depend on the selected language — see Section II for the full list of options for each language.

For example, the standard behavior for most of the languages is to produce cardinal forms by default, but if the language supports it, you can request a different `⟨form⟩`:

```
#num2words(3, lang: "en", form: "ordinal")
```

third

Depending on the language, you may also specify additional options. For instance, the English converter allows customizing the prefix for negative numbers:

```
#num2words(-10, lang: "en", negative: "minus")
```

minus ten

I.2 Language selection

The `#num2words` function selects the conversion language through the `⟨lang⟩` argument. It can be set to any supported language code (see Section II) or left as `auto` (the default), in which case it uses the document's current `text.lang` setting:

```
#set text(lang: "en")
The number 100 is written as #num2words(100).

#set text(lang: "es")
El número 100 se escribe #num2words(100).
```

The number 100 is written as one hundred.

El número 100 se escribe cien.

As shown in previous examples, you can also pass the language code explicitly — this overrides the document's `text.lang`:

```
#num2words(100, lang: "en") / #num2words(100, lang: "es")
```

---

one hundred / cien

I.3 Fallback

The optional ⟨`fallback`⟩ argument controls what `#num2words` does when the requested language is not supported. By default it is `none`, which produces an empty result silently:

```
[#num2words(100, lang: "xx")]
```

[]

As a string or array of strings, each code is tried in order:

```
#num2words(100, lang: "xx", fallback: ("es", "en"))
```

---

cien

A `none` entry inside the chain has the same meaning as the default — return an empty result if reached:

```
[#num2words(100, lang: "xx", fallback: ("yy", none))]
```

[]

To opt into strict behavior (panic on unsupported language), pass an empty chain:

```
#num2words(100, lang: "xx", fallback: ())
```

```
num2words: no supported language in fallback chain (tried: 'xx')
```

Part II

Supported languages

This chapter lists every supported language along with its available options, supported forms, and other language-specific features.

II.1 English (US)

Language code: `"en"`.

II.1.1 Options

`(form)` The output form. One of `"cardinal"` (default), `"ordinal"`, or `"year"`.

`(negative)` The prefix used for negative numbers. Defaults to `"negative"`.

II.1.2 Forms

Cardinal

The default form. Converts numbers to their cardinal word representation.

```
#num2words(42, lang: "en")
```

```
forty-two
```

```
#num2words(1000, lang: "en")
```

```
one thousand
```

Negative numbers are prefixed with the value of `(negative)`:

```
#num2words(-7, lang: "en")
```

```
negative seven
```

```
#num2words(-7, lang: "en", negative: "minus")
```

```
minus seven
```

Ordinal

Converts numbers to their ordinal word form.

```
#num2words(1, lang: "en", form: "ordinal")
```

```
first
```

```
#num2words(42, lang: "en", form: "ordinal")
```

```
forty-second
```

Year

Converts numbers using English year-reading conventions (e.g., splitting into two-digit groups).

```
#num2words(1999, lang: "en", form: "year")
```

```
nineteen ninety-nine
```

```
#num2words(2005, lang: "en", form: "year")
```

```
two thousand five
```

```
#num2words(2024, lang: "en", form: "year")
```

```
twenty twenty-four
```

II.2 Spanish

Language code: `"es"`.

II.2.1 Options

`form` The output form. One of `"cardinal"` (default) or `"ordinal"`.

`gender` Grammatical gender. One of `"masculine"` (default) or `"feminine"`. Affects cardinals (`"una"`, `"veintiuna"`, `"doscientas"`, `"veintiuna mil personas"`) and ordinals (`"primera"`, `"vigésima tercera"`). Scale nouns (`mil`, `millón`, `billón`, …) are invariable and stay masculine.

`apocopated` Whether to return the apocopated ordinal form (`"primer"`, `"tercer"`, `"vigésimo primer"`, `"decimotercer"`). Defaults to `false`. Only available for ordinals in masculine; combining it with `gender: "feminine"` raises an error, since Spanish has no feminine apocopated ordinal.

`negative` The prefix used for negative numbers. Defaults to `"menos"`.

II.2.2 Forms

Cardinal

The default form. Converts numbers to their cardinal word representation, using the long-scale convention (10^9 is `mil millones`, 10^{12} is `billón`, etc.).

```
#num2words(42, lang: "es")
```

```
cuarenta y dos
```

```
#num2words(100, lang: "es") / #num2words(101, lang: "es")
```

```
cien / ciento uno
```

```
#num2words(21, lang: "es") / #num2words(21000, lang: "es")
```

```
veintiuno / veintiún mil
```

```
#num2words(1000000000000, lang: "es")
```

```
un billón
```

Negative numbers are prefixed with the value of `⟨negative⟩`:

```
#num2words(-7, lang: "es")
```

```
menos siete
```

```
#num2words(-7, lang: "es", negative: "bajo")
```

```
bajo siete
```

Use `⟨gender⟩` to obtain the feminine form. Hundreds and the digits “uno”/“veintiuno” inflect; multi-word numbers agree with the noun even before “mil”:

```
#num2words(1, lang: "es", gender: "feminine") / #num2words(200, lang: "es", gender: "feminine")
```

```
una / doscientas
```

```
#num2words(21000, lang: "es", gender: "feminine") personas
```

```
veintiuna mil personas
```

Scale words like `mil`, `millón`, `billón`, … are themselves nouns and impose their own (masculine) agreement, so the preceding cardinal stays masculine regardless of `(gender)`:

```
#num2words(1000000, lang: "es", gender: "feminine") de personas
```

```
un millón de personas
```

Ordinal

Converts numbers to their ordinal word form. Ordinals are supported in the closed range $[1, 999]$. Values outside the supported range raise an out-of-range error.

```
#num2words(1, lang: "es", form: "ordinal")
```

```
primero
```

```
#num2words(11, lang: "es", form: "ordinal")
```

```
undécimo
```

```
#num2words(21, lang: "es", form: "ordinal")
```

```
vigésimo primero
```

```
#num2words(100, lang: "es", form: "ordinal")
```

```
centésimo
```

The feminine form replaces the trailing `-o` of every word with `-a`:

```
#num2words(21, lang: "es", form: "ordinal", gender: "feminine")
```

```
vigésima primera
```

The apocopated form drops the final `-o` of `primero`/`tercero` (and their compounds) — used immediately before a noun. It is only available in masculine:

```
#num2words(1, lang: "es", form: "ordinal", apocopated: true) capítulo
```

```
primer capítulo
```

```
#num2words(23, lang: "es", form: "ordinal", apocopated: true) capítulo
```

```
vigésimo tercer capítulo
```

II.3 Catalan

Language code: `"ca"`.

II.3.1 Options

`(form)` The output form. One of `"cardinal"` (default) or `"ordinal"`.

`(gender)` Grammatical gender. One of `"masculine"` (default) or `"feminine"`. Affects cardinals (`"una"`, `"dues"`, `"vint-i-una"`, `"dues-centes"`, `"vint-i-una mil persones"`) and ordinals (`"primera"`, `"vint-i-unena"`). Scale nouns (`mil`, `milió`, `bilió`, …) are invariable and stay masculine.

`(apocopated)` Whether to return the apocopated cardinal form (`"un"`, `"vint-i-un"`, `"trenta-un"`, `"cent un"`). Defaults to `false`. Only available for cardinals in masculine; combining it with `form: "ordinal"` or `gender: "feminine"` raises an error.

`(negative)` The prefix used for negative numbers. Defaults to `"menys"`.

II.3.2 Forms

Cardinal

The default form. Converts numbers to their cardinal word representation, using the long-scale convention (10^9 is `mil milions`, 10^{12} is `bilió`, etc.) and the IEC 2017 hyphenation rules within each hundreds–tens–units block.

```
#num2words(42, lang: "ca")
```

```
quaranta-dos
```

```
#num2words(100, lang: "ca") / #num2words(101, lang: "ca")
```

```
cent / cent u
```

```
#num2words(245, lang: "ca")
```

```
dos-cents quaranta-cinc
```

```
#num2words(1000000000000, lang: "ca")
```

```
un bilió
```

Negative numbers are prefixed with the value of ⟨negative⟩:

```
#num2words(-7, lang: "ca")
```

```
menys set
```

```
#num2words(-7, lang: "ca", negative: "negatiu")
```

```
negatiu set
```

The default form for `1` and its compounds is the counting form `"u"` (`"vint-i-u"`, `"trenta-u"`, `"cent u"`). Use ⟨apocopated⟩ to obtain the form used before a noun (`"un"`, `"vint-i-un"`, `"trenta-un"`):

```
#num2words(1, lang: "ca", apocopated: true) llibre
```

```
un llibre
```

```
#num2words(21, lang: "ca", apocopated: true) capítols
```

```
vint-i-un capítols
```

When a scale word (`mil`, `milió`, …) follows, the trailing unit is always apocopated regardless of ⟨apocopated⟩:

```
#num2words(21000, lang: "ca")
```

```
vint-i-un mil
```

Use ⟨gender⟩ to obtain the feminine form. Only `u` and `dos` (and the hundreds 200–900) inflect; multi-word numbers agree with the noun even before “mil”:

```
#num2words(1, lang: "ca", gender: "feminine") / #num2words(200, lang: "ca", gender: "feminine")
```

```
una / dues-centes
```

```
#num2words(21000, lang: "ca", gender: "feminine") persones
```

```
vint-i-una mil persones
```

Scale words like `mil`, `milió`, `bilió`, … are themselves nouns and impose their own (masculine) agreement, so the preceding cardinal stays masculine regardless of ⟨`gender`⟩:

```
#num2words(1000000, lang: "ca", gender: "feminine") de persones
```

```
un milió de persones
```

Ordinal

Converts numbers to their ordinal word form. Ordinals are supported in the closed range $[1, 999]$. Values outside the supported range raise an out-of-range error.

```
#num2words(1, lang: "ca", form: "ordinal")
```

```
primer
```

```
#num2words(11, lang: "ca", form: "ordinal")
```

```
onzè
```

```
#num2words(21, lang: "ca", form: "ordinal")
```

```
vint-i-unè
```

```
#num2words(100, lang: "ca", form: "ordinal")
```

```
centè
```

In compound ordinals, the suffix `-è/-ena` is applied only to the last word. Hyphenated blocks (21–99 and the fused hundreds 200–900) take the suffix on their final element (`vint-i-unè`, `trenta-dosè`, `dos-centè`); when the trailing unit is separated by a space (101–104, 201–204, …), the standalone forms `primer`, `segon`, `tercer`, `quart` are used.

```
#num2words(101, lang: "ca", form: "ordinal")
```

```
cent primer
```

```
#num2words(121, lang: "ca", form: "ordinal")
```

```
cent vint-i-unè
```

```
#num2words(999, lang: "ca", form: "ordinal")
```

```
nou-cents noranta-novè
```

The feminine form replaces the trailing `-è` with `-ena` (or appends `-a` to `primer`/`segon`/`tercer`/`quart`):

```
#num2words(21, lang: "ca", form: "ordinal", gender: "feminine")
```

```
vint-i-unena
```

```
#num2words(101, lang: "ca", form: "ordinal", gender: "feminine")
```

```
cent primera
```

Part III

API reference

III.1 Main module

#_normalize-fallback(⟨fallback⟩)

Validates the shape of a `fallback` argument and normalizes it to an array. Accepts a single string, a single `none`, or an array whose entries are strings or `none`.

#num2words(⟨number⟩, ⟨lang⟩: auto, ⟨fallback⟩: none, ..⟨args⟩) → content

Converts a number to its written word form.

Argument

⟨number⟩ — int

The number to convert.

Argument

⟨lang⟩: auto — str | auto

The language code (e.g., `"en"`). When `auto`, uses the current `text.lang`.

Argument

⟨fallback⟩: none — str | none | array

The fallback chain, where `none` can be used to return an empty result instead of panicking.

- ..args: Additional arguments forwarded to the language-specific converter.

III.2 Error helpers

#_prefix(⟨lang⟩) → str

Formats the `num2words` prefix, optionally scoped to a language.

Argument

⟨lang⟩ — str | none

The language code, or `none` for the top-level function.

#assert-lang(⟨lang⟩, ⟨supported⟩)

Asserts that a language code is supported.

Argument

`⟨lang⟩` `str`

The language code to check.

Argument

`⟨supported⟩` `array` | `dictionary`

The supported languages (array of strings or dictionary with language keys).

`#assert-option(⟨param⟩, ⟨value⟩, ⟨supported⟩, ⟨lang⟩: none)`

Asserts that a parameter value is among a set of supported values. Used for any option with a finite set of valid choices (e.g. `form`, `gender`).

Argument

`⟨param⟩` `str`

The parameter name.

Argument

`⟨value⟩` `any`

The value to check.

Argument

`⟨supported⟩` `array` | `dictionary`

The supported values (array, or dictionary whose keys are the supported values).

Argument

`⟨lang⟩: none` `str` | `none`

The language code, or `none` for the top-level function.

`#assert-type(⟨param⟩, ⟨expected-type⟩, ⟨value⟩, ⟨lang⟩: none)`

Asserts that a value has the expected type. Panics with a consistent message if not.

Argument

`⟨param⟩` `str`

The parameter name.

Argument

`⟨expected-type⟩` `type`

The expected type (e.g., `int`, `str`).

Argument

`⟨value⟩` `any`

The actual value received.

Argument

`(lang): none` `str` | `none`

The language code, or `none` for the top-level function.

`#out-of-range((number), (min): none, (max): none, (lang): none)`

Asserts that a number is within the supported range. Panics if not.

Argument

`(number)` `int`

The number to check.

Argument

`(min): none` `int` | `none`

The minimum supported value, or `none` if unbounded below.

Argument

`(max): none` `int` | `none`

The maximum supported value, or `none` if unbounded above.

Argument

`(lang): none` `str` | `none`

The language code, or `none` for the top-level function.

III.3 Languages

III.3.1 English (US)

`#en._cardinal-to-ordinal`	`#en._convert-below-1000`	`#en._convert-year`
`#en._chunk-and-convert`	`#en._convert-cardinal`	`#en._ordinalize`
`#en._convert-below-100`	`#en._convert-ordinal`	`#en.convert`

`#en._cardinal-to-ordinal((cardinal))` → `str`

Transforms a full cardinal string into its ordinal form by ordinalizing only the last word (handling hyphenated compounds like “forty-two”).

Argument

`(cardinal)` `str`

The cardinal string to transform.

`#en._chunk-and-convert((number), (scale-index))` → `array`

Recursively splits a number into 3-digit chunks and converts each chunk, appending the appropriate scale word.

Argument
⟨number⟩ `int`
The remaining number to convert.

Argument
⟨scale-index⟩ `int`
The current scale index (0 = units, 1 = thousands, etc.).

#en._convert-below-100(⟨number⟩) → `str`
Converts a number in the range 1–99 to its cardinal word form.

Argument
⟨number⟩ `int`
The number to convert (1–99).

#en._convert-below-1000(⟨number⟩) → `str`
Converts a number in the range 1–999 to its cardinal word form.

Argument
⟨number⟩ `int`
The number to convert (1–999).

#en._convert-cardinal(⟨number⟩) → `str`
Converts a positive integer to its cardinal word form.

Argument
⟨number⟩ `int`
The number to convert (>= 1).

#en._convert-ordinal(⟨number⟩) → `str`
Converts a positive integer to its ordinal word form.

Argument
⟨number⟩ `int`
The number to convert (>= 1).

#en._convert-year(⟨number⟩) → `str`
Converts a positive integer to its year reading form.

Argument
⟨number⟩ `int`
The number to convert (>= 1).

#en._ordinalize(⟨word⟩) → `str`
Converts a single cardinal word to its ordinal form.

Argument

`word` `str`

The cardinal word to ordinalize.

`#en.convert(number, form: "cardinal", negative: "negative") → str`

Converts a number to its English word form.

Argument

`number` `int`

The number to convert.

Argument

`form: "cardinal"` `str`

The form: `"cardinal"`, `"ordinal"`, or `"year"` (default: `"cardinal"`).

Argument

`negative: "negative"` `str`

The prefix for negative numbers (default: `"negative"`).

`#_lang-code`

The language code for this module.

`#_units`

Words for numbers 0–19.

`#_tens`

Words for multiples of ten from 20–90.

`#_scales`

Scale words for groups of three digits (short scale).

`#_ordinal-irregulars`

Cardinal words whose ordinal form is irregular.

`#_supported-forms`

Supported forms for this language module.

III.3.2 Spanish

`#es._apocopate-ordinal`	`#es._convert-below-million`	`#es._feminine-hundred`
`#es._chunk-and-convert`	`#es._convert-cardinal`	`#es._feminine-ordinal`
`#es._convert-below-100`	`#es._convert-ordinal`	`#es._feminine-unit`
`#es._convert-below-1000`	`#es._convert-ordinal-below-100`	`#es.convert`

#es._apocopate-ordinal((word))

Returns the apocopated form of a (possibly compound) ordinal: “primero” “tercero” suffixes drop their final “o”. Other ordinals are unchanged.

#es._chunk-and-convert((number), (scale-index), (feminine): false) → array

Recursively splits a number into 6-digit chunks (one long-scale group each) and converts each chunk, appending the appropriate scale word.

Argument

(number) `int`

The remaining number to convert.

Argument

(scale-index) `int`

The current scale index (0 = bottom group, 1 = millones, …).

Argument

(feminine): false `bool`

Whether the overall number modifies a feminine noun. Only the bottom chunk (scale-index 0) inherits the gender, since scale words (millón, billón…) are masculine and impose their own agreement.

#es._convert-below-100((number), (apocopate): false, (feminine): false) → str

Converts a number in the range 1–99 to its cardinal word form. The `apocopate` and `feminine` flags control the form of a trailing “uno”/“veintiuno”: apocopated (“un”/“veintiún”), feminine (“una”/“veintiuna”), or default masculine. `apocopate` takes precedence (used before “mil”).

Argument

(number) `int`

The number to convert (1–99).

Argument

(apocopate): false `bool`

Whether to apocopate a trailing “uno”.

Argument

(feminine): false `bool`

Whether to use the feminine form.

#es._convert-below-1000((number), (apocopate): false, (feminine): false) → str

Converts a number in the range 1–999 to its cardinal word form. The `apocopate` flag is forwarded to the trailing 1–99 part; it does not affect the 100 -> “cien” rule, which is intrinsic to this helper.

Argument
⟨number⟩ `int`
The number to convert (1–999).

Argument
⟨apocopate⟩: false `bool`
Whether to apocopate a trailing “uno”.

Argument
⟨feminine⟩: false `bool`
Whether to use feminine forms for “uno” and the hundreds 200–900. “cien”/“ciento” are invariable.

#es._convert-below-million(⟨number⟩, ⟨apocopate-units⟩: false, ⟨feminine⟩: false) → str

Converts a number in the range 1–999_999 (one long-scale group) to its cardinal word form. Splits the number into a thousands part (joined with “mil”) and a units part. The thousands part is always apocopated because “mil” follows; the units part is apocopated only if a scale noun follows the entire group (controlled by `apocopate-units`).

Argument
⟨number⟩ `int`
The number to convert (1–999_999).

Argument
⟨apocopate-units⟩: false `bool`
Whether the units part should apocopate (true when a scale noun like “millón” follows this 6-digit group).

Argument
⟨feminine⟩: false `bool`
Whether the chunk modifies a feminine noun. Affects the thousands part (e.g. “veintiuna mil personas”) and the units part when no scale word follows; ignored for units when `apocopate-units` is true, since scale nouns (millón, billón…) are masculine.

#es._convert-cardinal(⟨number⟩, ⟨feminine⟩: false) → str

Converts a positive integer to its cardinal word form.

Argument
⟨number⟩ `int`
The number to convert (>= 1).

Argument

`(feminine): false` `bool`

Whether the number modifies a feminine noun.

`#es._convert-ordinal((number), (feminine): false, (apocopated): false) → str`

Converts a positive integer in the range 1–999 to its ordinal word form. Panics if `number` is outside [1, 999]. The masculine form is built first and then transformed: `apocopated` drops the final “o” of a trailing “primero”/“tercero”; `feminine` swaps the final “o” of every word for “a”.

Argument

`(number)` `int`

The number to convert (1–999).

Argument

`(feminine): false` `bool`

Whether to return the feminine form.

Argument

`(apocopated): false` `bool`

Whether to return the apocopated form (masculine only; the public entry point rejects the feminine combination).

`#es._convert-ordinal-below-100((number)) → str`

Converts a number in the range 1–99 to its ordinal word form (masculine, non-apocopated).

Argument

`(number)` `int`

The number to convert (1–99).

`#es._feminine-hundred((word))`

Returns the feminine form of a hundreds word. “ciento” is invariable when used as a combiner; “doscientos”…“novecientos” become “doscientas”…“novecientas”.

`#es._feminine-ordinal((words))`

Feminizes every word in a (possibly multi-word) ordinal expression. Every supported ordinal ends in “o”, which is replaced by “a”.

`#es._feminine-unit((word))`

Returns the feminine form of a cardinal unit word (0–29). Only “uno” and “veintiuno” inflect; words like “cuatro” or “ocho” end in “o” but are invariable, so the match is on the “uno” suffix specifically.

```
#es.convert(
  (number),
  (form): "cardinal",
  (gender): "masculine",
  (apocopated): false,
  (negative): "menos"
) → str
```

Converts a number to its Spanish word form.

Cardinals are returned across the full long-scale range. Ordinals are supported within the closed range [1, 999]; values outside that range panic with an out-of-range error.

`gender` controls grammatical agreement: with `"feminine"`, cardinals produce forms like “una”, “veintiuna”, “doscientas” (and “veintiuna mil personas”); ordinals end in “-a”. Scale nouns (mil, millón, billón…) are invariable and stay masculine.

`apocopated` is only available for ordinals in masculine. It produces the short form used before a noun: “primer”, “tercer”, “vigésimo primer”, “decimotercer”. Combining `apocopated: true` with `gender: "feminine"` panics, since Spanish has no feminine apocopated ordinal.

Argument

`(number)` `int`

The number to convert.

Argument

`(form): "cardinal"` `str`

The form: `"cardinal"` (default) or `"ordinal"`.

Argument

`(gender): "masculine"` `str`

`"masculine"` (default) or `"feminine"`.

Argument

`(apocopated): false` `bool`

Use the apocopated ordinal form. Ordinal + masculine only.

Argument

`(negative): "menos"` `str`

The prefix for negative numbers (default: `"menos"`).

`#_lang-code`

The language code for this module.

#_units

Words for numbers 0–29. Numbers 16–29 are written as a single word per RAE.

#_units-apocopated

Apocopated unit words. Used when “uno”/“veintiuno” precedes a noun-like scale word (mil, millón, …): “uno” → “un”, “veintiuno” → “veintiún”. All other entries match `_units`.

#_tens

Words for multiples of ten from 30–90. Indexed by `tens-digit - 3`.

#_hundreds

Words for multiples of one hundred from 100–900. Indexed by hundreds digit; index 0 is unused. Note: the form for exactly 100 is “cien”, handled inline in `_convert-below-1000`; this table holds the combining form “ciento” for 1.

#_scales-singular

Singular long-scale words by 6-digit group index (RAE long scale: each step adds 6 zeros). Index 0 is empty (no scale word at the bottom group).

#_scales-plural

Plural long-scale words, paired with `_scales-singular`.

#_ord-units

Ordinal forms for 1–9. Index 0 is unused.

#_ord-tens

Ordinal forms for tens 10–90, indexed by tens digit. Index 0 is unused.

#_ord-hundreds

Ordinal forms for hundreds 100–900, indexed by hundreds digit. Index 0 is unused.

#_ord-teens

Ordinal forms for 13–19 (the contemporary fused single-word forms). Indexed by `n - 13`. 11 → “undécimo” and 12 → “duodécimo” are special-cased.

#_supported-forms

Supported forms for this language module.

#_supported-genders

Supported gender values.

III.3.3 Catalan

#ca._chunk-and-convert(⟨number⟩, ⟨scale-index⟩, ⟨feminine⟩: false, ⟨apocopated⟩: false) → array

Recursively splits a number into 6-digit chunks (one long-scale group each) and converts each chunk, appending the appropriate scale word.

Argument
⟨number⟩ — int
The remaining number to convert.

Argument
⟨scale-index⟩ — int
The current scale index (0 = bottom group, 1 = milions, …).

Argument
⟨feminine⟩: false — bool
Whether the overall number modifies a feminine noun. Only the bottom chunk (scale-index 0) inherits the gender, since scale words (milió, bilió…) are masculine and impose their own agreement.

Argument
⟨apocopated⟩: false — bool
Whether the user requested the apocopated form. Only affects the bottom chunk when no scale word follows.

#ca._convert-below-100(⟨number⟩, ⟨apocopate⟩: false, ⟨feminine⟩: false) → str

Converts a number in the range 0–99 to its cardinal word form. The `apocopate` and `feminine` flags control the form of a trailing “u”/“dos”: apocopated (“un”), feminine (“una”/“dues”), or default (“u”/“dos”). `apocopate` takes precedence over `feminine`.

Argument
⟨number⟩ — int
The number to convert (0–99).

Argument
⟨apocopate⟩: false — bool
Whether to apocopate a trailing “u”.

Argument

`(feminine): false` `bool`

Whether to use the feminine form.

`#ca._convert-below-1000((number), (apocopate): false, (feminine): false) → str`

Converts a number in the range 1–999 to its cardinal word form. The `apocopate` and `feminine` flags are forwarded to the trailing 1–99 part and used to pick the feminine variant of a hundreds word (e.g. 200 → “dues-centes”).

Argument

`(number)` `int`

The number to convert (1–999).

Argument

`(apocopate): false` `bool`

Whether to apocopate a trailing “u”.

Argument

`(feminine): false` `bool`

Whether to use feminine forms.

`#ca._convert-below-million((number), (apocopate-units): false, (feminine): false) → str`

Converts a number in the range 1–999_999 (one long-scale group) to its cardinal word form. Splits the number into a thousands part (joined with “mil”) and a units part. The thousands part is always apocopated because “mil” follows; the units part is apocopated only if a scale noun follows the entire group OR the caller requested apocopation for the final unit (controlled by `apocopate-units`).

Argument

`(number)` `int`

The number to convert (1–999_999).

Argument

`(apocopate-units): false` `bool`

Whether the units part should apocopate (true when a scale noun like “milió” follows this 6-digit group, or when the user requested `apocopated` for the bottom chunk).

Argument

`(feminine): false` `bool`

Whether the chunk modifies a feminine noun. Affects the thousands part (e.g. “vint-i-una mil persones”) and the units part when no scale word follows;

ignored for units when `apocopate-units` is true, since scale nouns (milió, bilió…) are masculine.

`#ca._convert-cardinal(⟨number⟩, ⟨feminine⟩: false, ⟨apocopated⟩: false) → str`

Converts a positive integer to its cardinal word form.

Argument
`⟨number⟩` `int`

The number to convert (>= 1).

Argument
`⟨feminine⟩: false` `bool`

Whether the number modifies a feminine noun.

Argument
`⟨apocopated⟩: false` `bool`

Whether to apocopate a trailing “u” → “un”.

`#ca._convert-ordinal(⟨number⟩, ⟨feminine⟩: false) → str`

Converts a positive integer in the range 1–999 to its ordinal word form. Panics if `number` is outside [1, 999]. The masculine form is built first and `feminine` swaps the trailing suffix on the last word for `-ena` (or appends `-a` to `primer`/`segon`/`tercer`/`quart`).

Argument
`⟨number⟩` `int`

The number to convert (1–999).

Argument
`⟨feminine⟩: false` `bool`

Whether to return the feminine form.

`#ca._convert-ordinal-below-100(⟨number⟩) → str`

Converts a number in the range 1–99 to its ordinal word form (masculine, non-feminine). Compound forms 21–29 and 31–99 use the fused suffix variants (`vint-i-unè`, `trenta-dosè`, `quaranta-cinquè`).

Argument
`⟨number⟩` `int`

The number to convert (1–99).

`#ca._feminine-hundred(⟨word⟩)`

Returns the feminine form of a hundreds word. “cent” is invariable; “dos-cents” → “dues-centes” (both prefix and suffix inflect); the rest “tres-cents”…“nou-cents” only change the suffix to “-centes”.

`#ca._feminine-ordinal((words))`

Feminizes the last token of a (possibly multi-word) ordinal expression: trailing “-è” becomes “-ena”; the standalone forms `primer`/`segon` `tercer`/`quart` get a final “a”. All preceding tokens stay unchanged.

`#ca._feminine-unit((word))`

Returns the feminine form of a cardinal unit word. Only “u” and “dos” inflect: “u” → “una”, “dos” → “dues”. Compound forms ending in “-u” or “-dos” (e.g. “vint-i-u”, “trenta-dos”) inflect at the suffix.

`#ca.convert(⟨number⟩, ⟨form⟩: "cardinal", ⟨gender⟩: "masculine", ⟨apocopated⟩: false, ⟨negative⟩: "menys") → str`

Converts a number to its Catalan word form.

Cardinals are returned across the full long-scale range. Ordinals are supported within the closed range [1, 999]; values outside that range panic with an out-of-range error.

`gender` controls grammatical agreement: with `"feminine"`, cardinals inflect “u”/“dos” and the hundreds 200–900 (“una”, “vint-i-una”, “dues”, “vint-i-dues”, “dues-centes”, “vint-i-una mil persones”); ordinals end in `-ena` (or `-a` for `primer`/`segon`/`tercer`/`quart`). Scale nouns (mil, milió, bilió…) are invariable and stay masculine.

`apocopated` is only available for cardinals in masculine. It produces the short form “un” instead of “u” for the trailing unit (“vint-i-un”, “trenta-un”, “cent un”). Combining `apocopated: true` with `form: "ordinal"` or `gender: "feminine"` panics.

Argument

`⟨number⟩` — `int`

The number to convert.

Argument

`⟨form⟩: "cardinal"` — `str`

The form: `"cardinal"` (default) or `"ordinal"`.

Argument

`⟨gender⟩: "masculine"` — `str`

`"masculine"` (default) or `"feminine"`.

Argument

`(apocopated): false` `bool`

Use the apocopated cardinal form. Cardinal + masculine only.

Argument

`(negative): "menys"` `str`

The prefix for negative numbers (default: `"menys"`).

`#_lang-code`

The language code for this module.

`#_units`

Words for numbers 0–19. Numbers 16–19 are single irregular words in Catalan (`setze`, `disset`, `divuit`, `dinou`).

`#_units-apocopated`

Apocopated unit words. Used when the trailing "u" precedes a noun-like element (a scale word `mil`/`milió`/… or a noun in the document): "u" → "un". All other entries match `_units`.

`#_tens`

Words for multiples of ten from 30–90. Indexed by `tens-digit - 3`.

`#_hundreds`

Words for multiples of one hundred from 100–900. Indexed by hundreds digit; index 0 is unused. The same form is used both alone (100 → "cent") and as the leading element of a 1XX number (101 → "cent u"); there is no separate combining form in Catalan.

`#_scales-singular`

Singular long-scale words by 6-digit group index (long scale: each step adds 6 zeros). Index 0 is empty (no scale word at the bottom group).

`#_scales-plural`

Plural long-scale words, paired with `_scales-singular`.

`#_ord-units`

Standalone ordinal forms for 1–9 (masculine). Index 0 is unused.

`#_ord-units-compound`

Compound-ordinal forms for the trailing unit 1–9 used inside hyphenated blocks like “vint-i-unè” or “trenta-dosè”. Differs from `_ord-units` only for 1–4 (`unè`, `dosè`, `tresè`, `quatrè` instead of `primer`…`quart`).

`#_ord-teens-and-ten`

Ordinal forms for 10–19, indexed by `n - 10`.

`#_ord-tens`

Ordinal forms for tens 10–90, indexed by tens digit. Index 0 is unused.

`#_ord-hundreds`

Fused ordinal forms for hundreds 100–900, indexed by hundreds digit. Used when the number is an exact multiple of 100. Index 0 is unused.

`#_supported-forms`

Supported forms for this language module.

`#_supported-genders`

Supported gender values.

Part IV

Index