



## Literate programming

<code># Title</code>	Heading
<code>### Sub-subtitle</code>	
<code>```catala</code>	Code block
<code>```catala-metadata</code>	Metadata block
<code>&gt; Include: foo.catala_en</code>	File inclusion

## Literals and types

<code>true</code>	<code>false</code>	boolean
<code>65536</code>		integer
<code>65536.262144</code>	<code>37%</code>	decimal
<code>\$1,234,567.89</code>		money
<code> 2021-01-31 </code>		date
<code>254 day 4 month 1 year</code>		duration
<code>[ 12; 24; 36 ]</code>		collection integer
<code>f of x, y equals x * y / \$12.0</code>		decimal depends on x content money, y content decimal
<code>Struct1 { -- fld1: 9 -- fld2: 7% }</code>		Struct1
<code>Case1 content 12 Case2</code>		Enum1

## Expressions

<code>let x equals 36 - 5 in ...</code>	Local definition
<code>match expr with pattern -- Case1 of x : ... -- Case2 : ... -- anything : ...</code>	Pattern matching
<code>expr with pattern Case1 expr with pattern Case1 of x and x &gt;= 2</code>	Pattern test and optional binding
<code>struc1.fld2</code>	Field access
<code>f of \$44.50, 1/3</code>	Function call
<code>sub1.var0</code>	Subscope variable
<code>output of Scope1 with { -- fld1: 9 -- fld2: 15% }</code>	Direct scope call
<code>if ... then ... else ...</code>	Conditional

## Metadata declaration

<code>declaration structure Struct1: data fld1 content integer data fld2 content decimal</code>	Structure declaration
<code>declaration enumeration Enum1: -- Case1 content integer -- Case2</code>	Enumeration declaration
<code>declaration scope Scope1: internal var1 content integer internal var2 condition sub1 scope Scope0</code>	Scope declaration
<code>internal var1 content ... output var3 content ... input var4 content ... input output var5 content ... context var6 content ... context output var7 content ...</code>	Input-output qualifiers
<code>internal var1 content ... state before state after</code>	State transitions declaration
<code>date round increasing # or decreasing</code>	Define date rounding mode
<code>declaration const content decimal equals 17.1</code>	Global definition
<code>declaration square content decimal depends on x content decimal equals x * x</code>	Global function definition

## Operators and built-ins

<code>not a</code>	<code>a and b</code>	Logical operators
<code>a or b</code>	<code># "or otherwise"</code>	
<code>a xor b</code>	<code># exclusive or</code>	
<code>- a</code>	<code>a + b</code>	Arithmetic
<code>a * b</code>	<code>a / b</code>	
<code>a = b</code>	<code>a != b</code>	Comparison
<code>a &gt; b</code>	<code>a &lt; b</code>	
<code>a &gt;= b</code>	<code>a &lt;= b</code>	
<code>decimal of 44</code>		Conversions
<code>money of 23.15</code>		
<code>round of \$9.99</code>		Rounding
<code>get_day of ...</code>		Date parts
<code>get_month of ...</code>		
<code>get_year of ...</code>		
<code>a +! b</code>	<code># integer</code>	Explicitly typed operators
<code>a +. b</code>	<code># decimal</code>	
<code>a +\$ b</code>	<code># money</code>	
<code>a +^ b</code>	<code># duration</code>	

## Scope definition

<code>scope Scope1: ...</code>	Scope use
<code>scope Scope1 under condition var1 &gt;= 2: ...</code>	Use-wide condition
<code>definition var1 equals ...</code>	Unconditional def.
<code>definition var1 under condition ... consequence equals ...</code>	Conditional definition
<code>rule var2 under condition var1 &gt;= 2 consequence fulfilled</code>	Rule (definition for conditions)
<code>consequence not fulfilled</code>	Negative rule
<code>definition f of x, y equals ...</code>	Function def. or rule
<code>label lbl1 definition var1 ...</code>	Labeled def. or rule
<code>exception lbl1 definition var1 ...</code>	Exception to label
<code>exception definition var1 ...</code>	Exception to implicit
<code>definition var1 state before equals ...</code>	State definition
<code>assertion ...</code>	Assertion

## Collection operations

<code>coll contains 3</code>	Presence test
<code>number of coll</code>	Cardinal
<code>exists x among coll such that x &gt;= 2</code>	Existence test
<code>for all x among coll we have x &gt;= 2</code>	For all test
<code>(x + 2) for x among coll</code>	Mapping
<code>x among coll such that x &gt;= 2</code>	Filter
<code>(x - 2) for x among coll such that x &gt;= 2</code>	Filter + map
<code>coll1 ++ coll2</code>	Merge
<code>sum integer coll</code>	Aggregation
<code>number of coll</code>	Count
<code>maximum of coll or if collection empty then -1</code>	Extremum
<code>x among coll such that (x * x) is minimum or if collection empty then -1</code>	Arg-extremum



## Programmation littéraire

```
# Titre                               En-têtes
### Sous-sous-titre

# Article 1 | JORFARTI000012345678    Référence au journal
# Article 2 | LEGIARTI000012345678    officiel
# Décision 3 | CETATEXT000012345678

```catala      ```catala-metadata    Bloc de code /
```                                     métadonnées

> Inclusion: foo.catala_en             Inclusion de fichier
```

## Littéraires et types

vrai	faux	booléen
65536		entier
65536,262144	37%	décimal
1 234 567,89€		argent
2021-01-31		date
254 jour 4 mois 1 an		durée
[ 12; 24; 36 ]		collection entier
f de x, y égal à x * y / 12,0€		décimal dépend de x contenu argent, y contenu décimal
Struct1 { -- chp1: 9 -- chp2: 7% }		Struct1
Cas1 contenu 12      Cas2		Énum1

## Expressions

```
soit x égal à 36 - 5 dans ...          Définition locale
selon expr sous forme                 Filtrage par motif
-- Cas1 de x : ...
-- Cas2 : ...
-- n'importe quel : ...

expr sous forme Cas1                 Test de filtrage
expr sous forme Cas1 de x            avec variable
    et x >= 2                         optionnelle

struc1.chp2                           Champ de structure
f de 44,50€, 1/3                       Appel de fonction
ss_ch1.var0                             Var. de s/s-ch. d'app.

résultat de Chp1                       Appel direct de
avec { -- chp1: 9 -- chp2: 15% }       champ d'application

si ... alors ... sinon ...            Branchement
```

## Déclaration des métadonnées

```
déclaration structure Struct1:        Déclaration de
donnée chp1 contenu entier            structure
donnée chp2 contenu décimal

déclaration énumération Énum1:        Déclaration
-- Cas1 contenu entier                d'énumération
-- Cas2

déclaration champ d'application Chp1: Déclaration de
interne var1 contenu entier            champ d'application
interne var2 condition
ss_ch1 champ d'application Chp0

interne var1 contenu ...               Qualificateurs
résultat var3 contenu ...              d'entrée-sortie
entrée var4 contenu ...
entrée résultat var5 contenu ...
contexte var6 contenu ...

interne var1 contenu ...               Transitions d'état
état avant
état après

date arrondi croissant                Mode arrondi dates
# ou décroissant

déclaration const contenu décimal     Définition globale
égal à 17.1

déclaration carré contenu décimal     Définition de
dépend de x contenu décimal           fonction globale
égal à x * x
```

## Opérations

```
non a                    a et b        Opérateurs logiques
a ou b                    # "ou à défaut"
a ou bien b                # ou exclusif

- a                    a + b            a - b            Arithmétique
a * b                    a / b

a = b                    a != b           Comparaisons
a > b                    a < b
a >= b                   a <= b

décimal de 44            Conversions
argent de 23,15

arrondi de 9,99€        Arrondis

accès_jour de ...       Éléments de dates
accès_mois de ...
accès_année de ...

a +! b                    # entier        Opérateurs à types
a +. b                    # décimal       explicites
a +€ b                    # argent
a +^ b                    # durée
```

## Définition de champ d'application

champ d'application Chp1: ...	Utilisation
champ d'application Chp1 sous condition var1 >= 2: ...	Avec condition générale
définition var1 égal à ...	Déf. inconditionnelle
définition var1 sous condition ... conséquence égal à ...	Définition conditionnelle
règle var2 sous condition var1 >= 2 conséquence rempli	Règle (définition de condition)
conséquence non rempli	Règle négative
définition f de x égal à ...	Déf./règle fonction
étiquette étq1 définition var1 ...	Déf./règle étiquetée
exception étq1 définition var1 ...	Exc. à déf. étiquetée
exception définition var1 ...	Exception à implicite
définition var1 état avant égal à ...	Définition d'états
assertion ...	Assertion

## Opérations sur les collections

coll contient 3	Test de présence
nombre de coll	Cardinal
existe x parmi coll tel que x >= 2	Test d'existence
pour tout x parmi coll on a x >= 2	Test pour tout
(x + 2) pour x parmi coll	Application un-à-un
x parmi coll tel que x >= 2	Filtrage
(x - 2) pour x parmi coll tel que x >= 2	Filtrage + application
coll1 ++ coll2	Réunion
somme entier coll	Aggrégation
nombre de coll	Comptage
maximum de coll ou si collection vide alors -1	Extremums
x parmi coll tel que (x * x) est minimum ou si collection vide alors -1	Élément selon extremum