



## Literate programming

# Title	Heading
### Sub-subtitle	
```catala ```catala-metadata	Code / metadata block
> Module Mdl	Module declaration
> Using Mdl as M	Module import+alias
> Include: foo.catala_en	File inclusion

## Literals and types

true	false	boolean	
65,536		integer	
65536.262144	37%	decimal	
\$1,234,567.89		money	
2024-04-01		date	
254 day	4 month	1 year	duration
[ 12; 24; 36 ]		list of integer	
(  2024-04-01 , \$30, 1% )		(date,money,decimal)	
f of x, y equals x * y / \$12.0		decimal depends on x content money, y content decimal	
Struct1 { -- fld1: 9 -- fld2: 7% }		Struct1	
Case1 content 12	Case2	Enum1	

## Operators and built-ins

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

## Metadata declaration

declaration structure Struct1: data fld1 content integer data fld2 content decimal	Structure declaration
declaration enumeration Enum1: -- Case1 content integer -- Case2	Enumeration declaration
declaration scope Scope1: internal var1 content integer internal var2 condition sub1 scope Scope0	Scope declaration
internal var1 content ... output var3 content ... input var4 content ... input output var5 content ... context var6 content ... context output var7 content ... output sub2 scope Scope0	Input-output qualifiers
internal var1 content ... state before state after	State transitions declaration
declaration const content decimal equals 17.1	Global definition
declaration square content decimal depends on x content decimal equals x * x	Global function definition

## Expressions

let x equals 36 - 5 in ...	Local definition
match expr with pattern -- Case1 of x : ... -- Case2 : ... -- anything : ...	Pattern matching
expr with pattern Case1 expr with pattern Case1 of x and x >= 2	Pattern test and optional binding
struc1 but replace { -- fld2: 8% }	Field replacement
struc1.fld2 tuple1.2 sub1.var0	Field, tuple element, subscope variable
f of \$44.50, 1/3	Function call
output of Scope1 with { -- fld1: 9 -- fld2: 15% }	Direct scope call
if ... then ... else ...	Conditional
var1 state before	Variable state access

## Scope definition

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

## List operations

lst contains 3	Presence test
exists x among lst such that x > 2	Existence test
for all x among lst we have x > 2	For all test
x + 2 for x among lst	Mapping
list of x among lst such that x > 2	Filter
x - 2 for x among lst such that x > 2	Filter + map
x + y for (x, y) among (lst1, lst2)	Multiple mapping
lst1 ++ lst2	Merge
sum integer of lst	Aggregation
number of lst	Count
maximum of lst or if list empty then -1	Extremum
content of x among lst such that x * x is minimum or if list empty then -1	Arg-extremum



## Programmation littéraire

# Titre	En-têtes
### Sous-sous-titre	
# Article 1   JORFARTI000012345678	Référence au journal officiel
# Article 2   LEGIARTI000012345678	
# Décision 3   CETATEXT000012345678	
```catala      ```catala-metadata	Bloc de code / métadonnées
> Module Mdl	Déclaration de mod.
> Usage de Mdl en tant que M	Import de module
> Inclusion: foo.catala_en	Inclusion textuelle

## Littéraux et types

vrai	faux	booléen
65536		entier
65536,262144	37%	décimal
1 234 567,89€		argent
2024-04-01		date
254 jour 4 mois 1 an		durée
[ 12; 24; 36 ]		liste de entier
( 2024-04-01 , 30€, 1%)		(date, argent, décimal)
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

## Opérations

non a	a et b	Opérateurs logiques
a ou b	# "ou à défaut"	
a ou bien b	# ou exclusif	
- a	a + b	Arithmétique
a * b	a / b	
a = b	a < b	Comparaisons
a != b	a > b	
décimal de 44		Conversions
argent de 23,15		
arrondi de 9,99€		Arrondis
accès_jour de ... accès_mois de ...		Éléments de dates
accès_année de ...		
a +! b a +. b a +€ b a +^ b		Opérateurs à types explicites
# entier décimal argent durée		

## Déclaration des métadonnées

déclaration structure Struct1: donnée chp1 contenu entier donnée chp2 contenu décimal	Déclaration de structure
déclaration énumération Énum1: -- Cas1 contenu entier -- Cas2	Déclaration d'énumération
déclaration champ d'application Chp1: interne var1 contenu entier interne var2 condition sub1 champ d'application Chp0	Déclaration de champ d'application
interne var1 contenu ... résultat var3 contenu ... entrée var4 contenu ... entrée résultat var5 contenu ... contexte var6 contenu ... contexte résultat var7 contenu ... résultat sub2 scope Scope0	Qualificateurs d'entrée-sortie
interne var1 contenu ... état avant état après	Transitions d'état
déclaration const contenu décimal égal à 17,1	Définition globale
déclaration carré contenu décimal dépend de x contenu décimal égal à x * x	Définition de fonction globale

## Expressions

soit x égal à 36 - 5 dans ...	Définition locale
selon expr sous forme -- Cas1 de x : ... -- Cas2 : ... -- n'importe quel : ...	Filtrage par motif
expr sous forme Cas1 expr sous forme Cas1 de x et x >= 2	Test de filtrage avec variable optionnelle
struc1 mais en remplaçant { -- fld2: 8% }	Remplacement de champs
struc1.chp2 tuple1.2 sub1.var0	Champ, élément de n-uplet, var. de s/s-champ Appel de fonction
f de 44,50€, 1/3	
résultat de Chp1 avec { -- chp1: 9 -- chp2: 15% }	Appel direct de champ d'application
si ... alors ... sinon ...	Branchement
var1 état avant	Accès à un état

## 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, y 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
date arrondi décroissant	Mode arrondi dates

## Opérations sur les listes

lst contient 3	Test de présence
existe x parmi lst tel que x > 2	Test d'existence
pour tout x parmi lst on a x > 2	Test pour tout
x + 2 pour x parmi lst	Application un-à-un
liste de x parmi lst tel que x > 2	Filtrage
x - 2 pour x parmi lst tel que x > 2	Filtrage + application
x + y pour (x, y) parmi (lst1, lst2)	Multiple mapping
lst1 ++ lst2	Réunion
somme entier de lst	Agrégation
nombre de lst	Comptage
maximum de lst ou si liste vide alors -1	Extremums
contenu de x parmi lst tel que x * x est minimum ou si liste vide alors -1	Élément selon extremum