# THE CATALA SYNTAX

english version

v0.8.0 · Revision #1 · © 2022

## Literate programming

Heading
```
# Title
### Sub-subtitle
```

Code block
````
```catala
```
````

Metadata block
````
```catala-metadata
```
````

File inclusion
```
> Include: foo.catala_en
```

## Types and literals

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

## Expressions

Local definition
```
let x equals 36 - 5 in ...
```

Pattern matching
```
match expr with pattern
-- Case1 of x: ...
-- Case2 : ...
```

Pattern test and optional binding
```
expr with pattern Case1
expr with pattern Case1 of x
          and x >= 2
```

Structure field access
```
struc1.fld2
```

Function call
```
f of $44.50
```

Subscope variable
```
sub1.var0
```

Direct scope call
```
outut of Scope1
  with { -- fld1: 9 -- fld2: true }
```

Conditional
```
if ... then ... else ...
```

## Metadata declaration

Structure declaration
```
declaration structure Struct1:
  data fld1 content integer
  data fld2 content boolean
```

Enumeration declaration
```
declaration enumeration Enum1:
  -- Case1 content integer
  -- Case2
```

Scope declaration
```
declaration scope Scope1:
  internal var1 content integer
  internal var2 condition
  sub1 scope Scope0
```

Input-output qualifiers
```
  internal var1 content ...
  output var3 content ...
  input var4 content ...
  input output var5 content ...
  context var6 content ...
  context output var7 content ...
```

State transitions declaration
```
internal var1 content ...
  state before
  state after
```

## Operators and built-ins

Logical operators
```
not a
a and b
a or b    # "or otherwise"
a xor b   # exclusive or
```

Arithmetic
```
- a
a + b        a - b
a * b        a / b
```

Comparison
```
a = b        a != b
a > b        a < b
a >= b       a <= b
```

Conversions
```
decimal of 44
money of 23.15
```

Rounding
```
round of $9.99
```

Date parts
```
get_day of ...
get_month of ...
get_year of ...
```

Explicitly typed operators
```
a +! b    # integer
a +. b    # decimal
a +$ b    # money
a +^ b    # duration
```

## Scope definition

Scope use
```
scope Scope1: ...
```

Use-wide condition
```
scope Scope1
  under condition var1 >= 2: ...
```

Unconditional definition
```
definition var1 equals ...
```

Conditional definition
```
definition var1
  under condition ...
  consequence equals ...
```

Rule (definition for conditions)
```
rule var2
  under condition var1 >= 2
  consequence fulfilled
```

Negative rule
```
  consequence not fulfilled
```

Function definition/rule
```
definition f of x equals ...
```

Labeled definition or rule
```
label lbl1 definition var1 ...
```

Exception to label
```
exception lbl1 definition var1 ...
```

Exception to implicit
```
exception definition var1 ...
```

State definition
```
definition var1
  state before
  equals ...
```

Assertion
```
assertion ...
```

## Collection operations

Presence test
```
coll contains 3
```

Cardinal
```
number of coll
```

Existence test
```
exists x among coll such that x >= 2
```

For all test
```
for all x among coll we have x >= 2
```

Mapping
```
(x + 2) for x among coll
```

Filter
```
x among coll such that x >= 2
```

Filter + map
```
(x - 2) for x among coll
        such that x >= 2
```

Merge
```
coll1 ++ coll2
```

Aggregation
```
sum integer coll
```

Count
```
number of coll
```

Extremum
```
maximum of coll
  or if collection empty then -1
```

Arg-extremum
```
x among coll
  such that (x * x) is minimum
  or if collection empty then -1
```

# LA SYNTAXE DE CATALA

version française

v0.8.0 · Révision #1 · © 2022

## Programmation littéraire

En-têtes

```
# Titre
### Sous-sous-titre
```

Référence au journal officiel

```
# Article 1 | JORFARTI000012345678
```

Bloc de code

````
```catala
```
````

Bloc de métadonnées

````
```catala-metadata
```
````

Inclusion de fichier

```
> Inclusion: foo.catala_en
```

## Types et littéraux

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

## Expressions

Définition locale

```
soit x égale à 36 - 5 dans ...
```

Filtrage par motif

```
selon expr sous forme
-- Cas1 de x: ...
-- Cas2 : ...
```

Test de filtrage
avec variable optionnelle

```
expr sous forme Cas1
expr sous forme Cas1 de x
          et x >= 2
```

Champ de structure

```
struc1.chp2
```

Appel de fonction

```
f de 44,50€
```

Var. de sous-ch. d'app.

```
ss_ch1.var0
```

Appel direct
du champ d'application

```
résultat de ChApp1
  avec { -- chp1: 9 -- chp2: vrai }
```

Branchement conditionnel

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

## Déclaration des métadonnées

Déclaration de structure

```
déclaration structure Struct1:
  donnée chp1 contenu entier
  donnée chp2 contenu booléen
```

Déclaration d'énumération

```
déclaration énumeration Énum1:
  -- Cas1 contenu entier
  -- Cas2
```

Déclaration de champ
d'application

```
déclaration champ d'application ChApp1:
  interne var1 contenu entier
  interne var2 condition
  ss_ch1 champ d'application ChApp0
```

Qualificateurs d'entrée-sortie

```
  interne var1 contenu ...
  résultat var3 contenu ...
  entrée var4 contenu ...
  entrée résultat var5 contenu ...
  contexte var6 contenu ...
  contexte résultat var7 contenu ...
```

Transitions d'état

```
interne var1 contenu ...
  état avant
  état après
```

## Opérations

Opérateurs logiques

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

Arithmétique

```
- a
a + b        a - b
a * b        a / b
```

Comparaisons

```
a = b        a != b
a > b        a < b
a >= b       a <= b
```

Conversions

```
décimal de 44
argent de 23,15
```

Arrondis

```
arrondi de 9,99€
```

Éléments de dates

```
accès_jour de ...
accès_mois de ...
accès_année de ...
```

Opérateurs à types explicites

```
a +! b    # entier
a +. b    # décimal
a +€ b    # argent
a +^ b    # durée
```

## Définition de champ d'application

Utilisation

```
champ d'application ChApp1: ...
```

Avec condition générale

```
champ d'application ChApp1
  sous condition var1 >= 2: ...
```

Définition inconditionnelle

```
définition var1 égale à ...
```

Définition conditionnelle

```
définition var1
  sous condition ...
  conséquence égale à ...
```

Règle
(définition de condition)

```
règle var2
  sous condition var1 >= 2
  conséquence rempli
```

Règle négative

```
  conséquence non rempli
```

Définition de fonction/règle

```
définition f de x égale à ...
```

Définition/règle étiquetée

```
étiquette étq1 définition var1 ...
```

Exc. à définition étiquetée

```
exception étq1 définition var1 ...
```

Exception à implicite

```
exception définition var1 ...
```

Définition d'états

```
définition var1
  état avant
  égale à ...
```

Assertion

```
assertion ...
```

## Opérations sur les collections

Test de présence

```
coll contient 3
```

Cardinal

```
nombre de coll
```

Test d'existence

```
existe x parmi coll tel que x >= 2
```

Test pour tous

```
pour tout x parmi coll on a x >= 2
```

Application un-à-un

```
(x + 2) pour x parmi coll
```

Filtrage

```
x parmi coll tel que x >= 2
```

Filtrage + application

```
(x - 2) pour x parmi coll
        tel que x >= 2
```

Réunion

```
coll1 ++ coll2
```

Aggrégation

```
somme entier coll
```

Comptage

```
nombre de coll
```

Extremums

```
maximum de coll
  ou si collection vide alors -1
```

Élément selon extremum

```
x parmi coll
  tel que (x * x) est minimum
  ou si collection vide alors -1
```