

## Literate programming

Feature	English syntax	French syntax
Heading	<b>## Foo</b> <b>### Bar</b>	<b>## Foo</b> <b>### Bar</b>
Code block	<code>```catala</code> <code>```</code>	<code>```catala</code> <code>```</code>
Metadata block	<code>```catala-metadata</code> <code>```</code>	<code>```catala-metadata</code> <code>```</code>
File inclusion	<code>&gt; Include: foo.catala_en</code>	<code>&gt; Inclusion: foo.catala_fr</code>

## Metadata declaration

Feature	English syntax	French syntax
Structure declaration	<b>declaration structure Foo:</b> <b>data</b> bar <b>content integer</b> <b>data</b> baz <b>content boolean</b>	<b>déclaration structure Foo:</b> <b>donnée</b> bar <b>contenu entier</b> <b>donnée</b> baz <b>contenu booléen</b>
Enumeration declaration	<b>declaration enumeration Foo:</b> -- <b>Bar</b> <b>content integer</b> -- <b>Baz</b>	<b>déclaration énumération Foo:</b> -- <b>Bar</b> <b>contenu entier</b> -- <b>Baz</b>
Scope declaration	<b>declaration scope Foo:</b> <b>internal</b> bar <b>content integer</b> <b>internal</b> baz <b>condition</b> fizz <b>scope Buzz</b>	<b>déclaration champ d'application Foo:</b> <b>interne</b> bar <b>contenu entier</b> <b>interne</b> baz <b>condition</b> fizz <b>champ d'application Buzz</b>
Input-output qualifiers	<b>internal</b> bar <b>content</b> ... <b>output</b> baz <b>content</b> ... <b>input</b> fizz <b>content</b> ... <b>input output</b> buzz <b>content</b> ... <b>context</b> biz <b>content</b> ... <b>context output</b> boz <b>content</b> ...	<b>interne</b> bar <b>contenu</b> ... <b>sortie</b> baz <b>contenu</b> ... <b>entrée</b> fizz <b>contenu</b> ... <b>entrée sortie</b> buzz <b>contenu</b> ... <b>contexte</b> biz <b>contenu</b> ... <b>contexte sortie</b> boz <b>contenu</b> ...
State transitions declaration	<b>internal</b> foo <b>content</b> ... <b>state</b> bar <b>state</b> buzz	<b>interne</b> foo <b>contenu</b> ... <b>état</b> bar <b>état</b> buzz

## Types

Feature	English syntax	French syntax
Natural integers	<b>integer</b>	<b>entier</b>
Rational numbers	<b>decimal</b>	<b>décimal</b>
Booleans	<b>boolean</b>	<b>booléen</b>
Money	<b>money</b>	<b>argent</b>
Date	<b>date</b>	<b>date</b>
Duration	<b>duration</b>	<b>durée</b>
Function	<b>Foo depends on Bar</b>	<b>Foo dépend de Bar</b>
Collection	<b>collection Foo</b>	<b>collection Foo</b>

## Literals

Feature	English syntax	French syntax
Integers	65536	65536
Decimals	65536.262144	65536.262144
Money	\$1,234,567.89	1 234 567,89 €
Date	2021-01-31	2021-31-01
Durations	254 day 4 month 1 year	254 jour 4 mois 1 an
Boolean	<b>true</b> <b>false</b>	<b>vrai</b> <b>faux</b>

## Scope use and related items

Feature	English syntax	French syntax
Scope use	<b>scope Foo:</b> ...	<b>champ d'application Foo:</b> ...
Use-wide condition	<b>scope Foo</b> <b>under condition</b> bar: ...	<b>champ d'application Foo</b> <b>sous condition</b> bar: ...
Unconditional definition	<b>definition</b> foo <b>equals</b> ...	<b>définition</b> foo <b>égal à</b> ...
Conditional definition	<b>definition</b> foo <b>under condition</b> bar <b>consequence equals</b> ...	<b>définition</b> foo <b>sous condition</b> bar <b>conséquence égal à</b> ...
Rule (definition for conditions)	<b>rule</b> foo <b>under condition</b> bar <b>consequence fulfilled</b>	<b>règle</b> foo <b>sous condition</b> bar <b>conséquence rempli</b>
Negative rule	<b>rule</b> foo <b>under condition</b> bar <b>consequence not fulfilled</b>	<b>règle</b> foo <b>sous condition</b> bar <b>conséquence non rempli</b>
Function definition/rule	<b>definition</b> foo <b>of</b> bar ...	<b>définition</b> foo <b>de</b> bar ...
Labeled definition or rule	<b>label</b> foo <b>definition</b> bar ...	<b>étiquette</b> foo <b>définition</b> bar ...
Exception to label	<b>exception</b> foo <b>definition</b> bar ...	<b>exception</b> foo <b>définition</b> bar ...
Exception to implicit	<b>exception</b> <b>definition</b> bar ...	<b>exception</b> <b>définition</b> bar ...
State definition	<b>definition</b> foo <b>state</b> bar <b>equals</b> ...	<b>définition</b> foo <b>état</b> bar <b>égal à</b> ...
Assertion	<b>assertion</b> ...	<b>assertion</b> ...

## Expressions

Feature	English syntax	French syntax
Pattern matching	<b>match</b> ... <b>with pattern</b> -- <b>Foo of</b> foo: ... -- <b>Bar</b> : ...	<b>selon</b> ... <b>sous forme</b> -- <b>Foo de</b> foo: ... -- <b>Bar:</b> ...
Pattern test and optional binding	... <b>with pattern Foo</b> ... <b>with pattern Bar of</b> bar and ...	... <b>sous forme Foo</b> ... <b>sous forme Bar de</b> bar et
Constructor injection	<b>Foo content</b> ... <b>Bar</b>	<b>Foo contenu</b> ... <b>Bar</b>
Structure literal	<b>Foo</b> { -- bar: ... -- baz: ... }	<b>Foo</b> { -- bar: ... -- baz: ... }
Structure field access	(...).foo	(...).foo
Function call	... <b>of</b> ...	... <b>de</b> ...
Subscope variable	foo.bar	foo.bar
Direct scope call	<b>Foo of</b> { -- bar: ... -- baz: ... }	<b>Foo de</b> { -- bar: ... -- baz: ... }
Conditional	<b>if</b> ... <b>then</b> ... <b>else</b> ...	<b>si</b> ... <b>alors</b> ... <b>sinon</b>

Collections

Operators

Feature	English syntax	French syntax
Collection literal	[ ...; ...; ...]	[ ...; ...; ...]
Presence test	... in ...	... dans ...
Cardinal	number of ...	nombre de ...
Existence test	exists foo in ... such that ...	existe foo dans ... tel que ...
For all test	for all foo in ... we have ...	pour tout foo dans ... on a ...
For all test	for all foo in ... we have ...	pour tout foo dans ... on a ...
Map/filter	map for foo in ... of ... filter for foo in ... of ...	application pour foo dans ... de ... filtre pour foo dans ... de ...
Aggregation	sum money for foo in ... of ...	somme argent pour foo dans ... de ...
Conditional count	number for foo in ... of ...	nombre pour foo dans ... de ...
Extremum	maximum integer initial ... for ... in ...	maximum entier initial ... pour ... dans ...
Arg-extremum	content minimum decimal initial ... for ... in ...	contenu minimum décimal initial ... pour ... dans ...

Feature	English syntax	French syntax
Integer to decimal	integer_to_decimal of ...	entier_vers_décimal de ...
Money rounding	round_money of ...	arrondi_argent de ...
Date parts	get_day of ... get_month of ... get_year of ...	accès_jour de ... accès_mois de ... accès_année de ...
Logical inclusive or	... or ...	... ou ...
Logical exclusive or	... xor ...	... ou bien ...
Logical and	... and ...	... et ...
Polymorphic structural equality	... = ... ... != ...	... = ... ... != ...
Integer sum	(integer) + (integer)	(entier) + (entier)
Integer substraction	(integer) - (integer)	(entier) - (entier)
Integer multiplication	(integer) * (integer)	(entier) * (entier)
Integer division	(integer) / (integer)	(entier) / (entier)
Integer comparison	<    <=    >    >=	<    <=    >    >=
Decimal sum	(decimal) +. (decimal)	(décimal) +. (décimal)
Decimal substraction	(decimal) -. (decimal)	(décimal) -. (décimal)
Decimal multiplication	(decimal) *. (decimal)	(décimal) *. (décimal)
Decimal division	(decimal) /. (decimal)	(décimal) /. (décimal)
Decimal comparison	<.    <=.    >.    >=.	<.    <=.    >.    >=.
Money sum	(money) +\$ (money)	(argent) +€ (argent)
Money substraction	(money) -\$ (money)	(argent) -€ (argent)
Money multiplication	(money) *\$ (decimal)	(argent) *€ (décimal)
Money division	(money) /\$ (money)	(argent) /€ (argent)
Money comparison	<\$    <=\$    >\$    >=\$	<€    <=€    >€    >=€
Date sum	(date) +@ (duration)	(date) +@ (durée)
Date substraction	(date) -@ (date)	(date) -@ (date)
Date comparison	<@    <=@    >@    >=@	<@    <=@    >@    >=@
Duration sum	(duration) +^ (duration)	(durée) +^ (durée)
Duration substraction	(duration) -^ (duration)	(durée) -^ (durée)
Duration multiplication	(duration) *^ (integer)	(durée) *^ (entier)
Duration comparison	<^    <=^    >^    >=^	<^    <=^    >^    >=^
Collection merging	(collection) ++ (collection)	(collection) ++ (collection)