

Gramática - Compiladores

```
func_type : ' (' [type_spec] ')' -> ' expression NOUVELLELIGNE * ENDMARKER

type_spec :
| 'int'

declarations: declaration+

declaration: structure_decl | simple_decl t

declaration_newline :
| structure_decl NOUVELLELIGNE | simple_decl t NOUVELLELIGNE | ENDMARKER

simple_decl t :
| simple_decl ! ';' NOUVELLELIGNE | ';' . simple_decl + [ ';' ] NOUVELLELIGNE

simple_decl :
| assignment | star_expressions | return_decl | 'passe' | del_decl | 'rupture' | 'continuez'

structure_decl :
| function_def | if_decl | with_decl | for_decl | while_decl

Operators_keys :
| ' + + = '
| ' - - = '
| ' * * = '
| ' / / = '
| ' mod mod = '
| ' = '
| ' > '
| ' < '
| ' < = '
| ' > = '
| ' ( '
| ' ) '
| ' [ '
| ' ] '
| ' | '
| ' , '
| ' ; '
| ' . '

del_decl :
| ' effacer ' del_targets ( ' ; ' | NOUVELLELIGNE )

if_decl :
| ' oui ' name_structure ' : ' block | if_decl | ' oui ' name_structure ' : ' block [ else_block ]
```

```
elif_decl :
|'changer' name_structure' :' block elif_decl |'changer' name_structure' :' block [else_block]

else_block :
|'autre'' :' block

while_decl :
|'tandis_que' name_structure' :' block [else_block]

for_decl :
|'pour' star_targets' dans' star_expressions' :' [TYPE_COMMENT] block [else_block] | ASYNC' pour' star_targets' dans'
[TYPE_COMMENT] block [else_block]

literal_pattern :
| signed_number! ('+' | '-' ) | strings | 'Vrai' | 'Faux'

literal_expr :
| signed_number! ('+' | '-' ) | strings | 'Vrai' | 'Faux'

signed_number :
| NÚMERO | '-' NÚMERO

return_decl :
| 'revenir' [star_expressions]

function_def :
| decorators function_def_raw | function_def_raw

function_def_raw :
| 'définir' NOM' ('[params]') ['-' '>' expression]' :' [func_type_comment] block | ASYNC' définir' NOM' ('[params]') ['-'
expression]' :' [func_type_comment] block

func_type_comment :
| NOUVELLELIGNE TYPE_COMMENTAIRE (NOUVELLELIGNE INDENT) | TYPE_COMMENTAIRE

params:
— parameters

parameters:
— slash_no_default param_no_default * param_with_default * [star_etc] | slash_with_default param_with_default *
[star_etc] | param_no_default + param_with_default * [star_etc] | param_with_default + [star_etc] | star_etc

slash_no_default :
| param_no_default + '/' '/' , | param_no_default + '/' '/' )'

slash_with_default :
| param_no_default * param_with_default + '/' '/' , | param_no_default * param_with_default + '/'
/' '/' )'

star_etc :
| '**' param_no_default param_maybe_default * [kwds] | '**' , ' param_maybe_default + [kwds] | kwds
```

kwds: *''' param_no_default*

param_no_default :
|param', 'TYPE_COMMENTAIRE'?|paramTYPE_COMMENTAIRE?')'

param_with_default :
|paramdefault', 'TYPE_COMMENTAIRE'?|paramdefaultTYPE_COMMENTAIRE?')'

param_maybe_default :
|paramdefault?', 'TYPE_COMMENTAIRE'?|paramdefault?TYPE_COMMENTAIRE?')'

param: NOM annotation?

annotation: ':' expression
default: '=' expression

decorators: ('-' comment COMMENTAIRE)+

block:
— COMMENTAIRE INDENT declarations DEDENT — *simple_declt*

star_expressions :
|star_expression('', star_expression) + ['']|star_expression', '|star_expression

star_expression :
*|' * '*'comparer|expression*

star_name_expressions :', '.star_name_expression + ['']

star_name_expression :
*|' * '*'comparer|name_expression*

assignment_expression :
|NOM' :=' expression

name_expression :
|assignment_expression|expression!' :='

annotated_rhs : star_expressions

expressions:
— expression ('', expression)+ ['','] — expression ',', — expression

expression:
— disjunction 'oui' disjunction 'autre' expression — disjunction

disjunction:
— conjunction ('ou' conjunction)+ — conjunction

conjunction:
— inversion ('et' inversion)+ — inversion

inversion:
— 'ne_ — pas'inversion|comparison

comparison:
— comparer compare_op_comparer_pair + |comparer

compare_op_comparer_pair :
|eq_comparer|noteq_comparer|lte_comparer|lt_comparer|gte_comparer|gt_comparer|notin_comparer|in_comparer|i

eq_comparer : '=' comparer

noteq_comparer :
|('!' =')comparer

lte_comparer : '<=' comparer

lt_comparer : '<' comparer

gte_comparer : '>=' comparer

gt_comparer : '>' comparer

notin_comparer : ' ne_pas''dans'comparer

in_comparer : ' dans'comparer

isnot_comparer : ' est''ne_pas'comparer

is_comparer : ' est'comparer

comparer:
— comparer '—' comparer_xor|comparer_xor

comparer_xor :
|comparer_xor'comparer_and|comparer_and

comparer_and :
|comparer_and''shift_expr|shift_expr

shift_expr :
|shift_expr' <<' sum|shift_expr' >>' sum|sum

sum:
— sum '++' term — sum '-' term — term

term:
— term '**' factor — term '/' factor — term 'modmod' factor — factor

factor:
— '++' factor — '-' factor

