START:

<start>🡪 def <function><start> | <Class><start> | <For><start> | <While><start> | <If><start> | <Try><start> | <Del> nl <start> |<initialize>nl <start> | €

INITIALIZATION STATEMENT:

<initialize>🡪ID<ID\_rel\_>|<const><init2>|\*<init3>| accessModifier init4>|<static\_final>ID<ID\_rel> = <init5> | this AcOp ID<ID\_rel><init1> | (<exp>)<OE> | Not <exp\_F> <OE> | <List2> | <Dictionary2>

<init1>🡪AsOP<exp>|=<init5> |<init2>

<init2> 🡪 <OE><In>

<init3>🡪accessModifier <init4>|<init4>

<init4>🡪<static\_final>ID<ID\_rel>=<init5>|ID<ID\_rel> = <init5>

<init5>🡪<const><OE>|<This>ID<ID\_rel><init6> | <List> | <Dictionary>| <Class\_call> | <lambda>

<init6>🡪=<init5>| <init2>

<ID\_rel\_>🡪<ID\_rel1\_>| (<arg\_list\_call>) <AcOP>

<ID\_rel1\_>🡪[<exp>] <ID\_rel1\_>| AcOp ID <ID\_rel\_> | <init1>

<AcOP>🡪 AcOp ID <ID\_rel\_>|Є

<AM>🡪 accessModifier | €

<static\_final>🡪 static <s\_final> | final

<s\_final>🡪 final | €

<id\_const>🡪<This> ID <ID\_rel> | <const>

<const>🡪 int\_const | float\_const | string\_const | char\_const | bool\_const

<Global>🡪 global <initialize>

<pointer>🡪 \* | €

EXPRESSION:

<OE>🡪<exp\_MDM><exp\_PM><exp\_RELOP><exp\_AND><exp\_OR>

<exp>🡪<expAND><exp\_OR>

<exp\_OR>🡪 Or <expAND><exp\_OR> | €

<expAND>🡪<expRELOP><exp\_AND>

<exp\_AND>🡪 And <expRELOP><exp\_AND> | €

<expRELOP>🡪<expPM><exp\_RELOP>

<exp\_RELOP>🡪 RelOp <expPM><exp\_RELOP> | €

<expPM>🡪<expMDM><exp\_PM>

<exp\_PM>🡪 PM <expMDM><exp\_PM> | €

<expMDM>🡪<exp\_F><exp\_MDM>

<exp\_MDM>🡪 DM <exp\_F><exp\_MDM> | \* <exp\_F><exp\_MDM> |€

<exp\_F>🡪<This>ID <ID\_rel><In> | <const><In> | (<exp>) | Not <exp\_F> | <lambda>

<ID\_rel> 🡪 [exp]<ID\_rel1> | (<arg\_list\_call>)<ID\_rel1> | AcOp ID <ID\_rel> | €

<ID\_rel1> 🡪 AcOp ID <ID\_rel> | <ID\_rel2>

<ID\_rel2> 🡪[exp]<ID\_rel1> | €

<This>🡪 this AcOp | €

<In>🡪 in <In\_>| €

<In\_>🡪 ID | <List>

BODY:

<body>🡪<S\_St> nl |nl IndentInit <mst> IndentOut | pass nl

<mst> 🡪 <M\_St> | pass nl

<M\_St>🡪<S\_St><NL><M\_St\_>

<M\_St\_>🡪<M\_St> | €

<S\_St>🡪 break | continue | return <sst1> | <For> | <While> | <If> | <Try> | <Del>| <Global> |<initialize>

<sst1> 🡪 <exp> | <List2> | Dictionary2>  
<NL>🡪 nl | €

<body>🡪<S\_St> |nl IndentInit <M\_St> IndentOut 1

<M\_St>🡪<S\_St><NL><M\_St\_>

<M\_St\_>🡪<M\_St> | €

<S\_St>🡪 <S\_St1> nl| <S\_St2>

<S\_St1>🡪 break | continue | return <sst1> | <Del>| <Global> |<initialize> | pass

<S\_St2>🡪<For> | <While> | <If> | <Try>

<sst1> 🡪 <exp> | <List2> | Dictionary2>  
<NL>🡪 nl | €

CLASS:

<Class>🡪 class ID (<inherit>) : nl IndentInit <class\_body><NL> IndentOut

<Inherit>🡪 ID <inherit\_> | €

<inherit\_>🡪 , ID | €

<class\_body>🡪<class\_body1> | pass

<class\_body1>🡪<initialize> nl <class\_body3> | def <class\_body2><class\_body3>

<class\_body2>🡪<constructor> | <function>

<class\_body3>🡪< class\_body1> | €

CLASS CALL:

<Class\_call>🡪 new ID ( <arg\_list\_call> )

CONSTRUCTOR:

<constructor>🡪<AM> ID (<arg\_list>) : <body>

<arg\_list>🡪<data\_type><pointer> ID <arg\_list1> | €

<arg\_list1>🡪 , <data\_type><pointer> ID <arg\_list1> | €

<arg\_list\_call>🡪<exp><arg\_list\_call1> | €

<arg\_list\_call1>🡪 , <exp><arg\_list\_call1> | €

FUNCTION:

<function>🡪<data\_type><AM><function\_>

<function\_> 🡪 <static\_final> ID (<arg\_list>) : <body> | ID (<arg\_list>) : <body>

<data\_type>🡪 DT | string | ID

LOOPS:

<For>🡪 for ID in <For\_> : <body>

<For\_>🡪 ID<ID\_rel> | <List2> | <Dictionary2>

<While>🡪 while (<exp>) : <body>

IF\_ELIF\_ELSE:

<If>🡪 if (<exp>) : <body><Elif>

<Elif>🡪 elif (<exp>) : <body><Elif> | <Else>

<Else>🡪 else : <body> | €

LIST/ARRAY:

<List>🡪<List2> | List ( <List1> )

<List1>🡪<List2> | ID

<List2>🡪 [ <List3> ]

<List3>🡪<id\_const><List4> | <List2><List4> | <Dictionary2> <List4>|Є

<List4>🡪 , <List5> | €

<List5> 🡪 <id\_const><List4> | <List2><List4> | <Dictionary2> <List4>

DICTIONARY:

<Dictionary>🡪<Dictionary2> | dict (<Dictionary1>)

<Dictionary1>🡪 ID | <Dictionary2> | €

<Dictionary2>🡪 { <Dictionary3> }

<Dictionary3>🡪<id\_const> : <Dictionary4>| €

<Dictionary4>🡪 <id\_const> <Dictionary6> | <Dictionary2> <Dictionary6> | <List2> <Dictionary6>

<Dictionary6>🡪 ,<Dictionary7> | €

<Dictionary7> 🡪<id\_const> : <Dictionary4>

DEL:

<Del>🡪 del ID <Del\_>

<Del\_>🡪 [<exp>] | €

TRY\_EXCEPT\_FINALLY:

<Try>🡪 try : <body><Except>

<Except>🡪<Finally> | except (<Exception>) : <body><Except>

<Finally>🡪 finally : <body>

<Exception>🡪 exception ID

LAMBDA:

<Lambda>🡪 lambda ID : <exp>

|  |  |  |  |
| --- | --- | --- | --- |
| Non-Terminals | First Set | Follow Set | Selection Set |
|  |  |  |  |
| <start> | def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, € | ~ | def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, ~ |
| <initialize> | ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final | nl |  |
| <init1> | AsOp, =, \*, DM, PM, RelOp, And, Or, in | nl |  |
| <init2> | \*, DM, PM, RelOp, And, Or, in | nl |  |
| <init2\_> | \*, DM, PM, RelOp, And, Or, in, € | nl | \*, DM, PM, RelOp, And, Or, in, nl |
| <init3> | accessModifier, static, final, ID | nl |  |
| <init4> | static, final, ID | nl |  |
| <init5> | int\_const, float\_const, string\_const, char\_const, bool\_const, ID, [, List, {, dict, new | nl |  |
| <init6> | =, DM, \*, PM, RelOp, And, Or, € | nl | =, DM, \*, PM, RelOp, And, Or, nl |
| <AM> | accessModifier, € | ID, static, final | accessModifier, ID, static, final |
| <static\_final> | static, final | ID |  |
| <s\_final> | final, € | ID | final, ID |
| <id\_const> | ID, int\_const, float\_const, string\_const, char\_const, bool\_const | DM, \*, PM, RelOp, And, Or, nl, ,, ], }, : |  |
| <const> | int\_const, float\_const, string\_const, char\_const, bool\_const | \*, DM, PM, RelOp, And, Or, in , nl, ,, ], }, :, in, ) |  |
| <Global> | global | nl |  |
| <pointer> | \*, € | ID | \*, ID |
| <OE> | DM, \*, PM, RelOp, And, Or, € | nl | DM, \*, PM, RelOp, And, Or, nl |
| <exp> | this, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, (, Not, lambda | ), ,, nl, ] |  |
| <exp\_OR> | Or, € | in, ), ,, nl, ] | Or, in, ), ,, nl, ] |
| <expAND> | this, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, (, Not, lambda | Or, in, ), ,, nl, ] |  |
| <exp\_AND> | And, € | Or, in, ), ,, nl, ] | And, Or, in, ), ,, nl, ] |
| <expRELOP> | this, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, (, Not, lambda | And, Or, in, ), ,, nl, ] |  |
| <exp\_RELOP> | RelOp, € | And, Or, in, ), ,, nl, ] | RelOp, And, Or, in, ), ,, nl, ] |
| <expPM> | this, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, (, Not, lambda | RelOp, And, Or, in, ), ,, nl, ] |  |
| <exp\_PM> | PM, € | RelOp, And, Or, in, ), ,, nl, ] | PM, RelOp, And, Or, in, ), ,, nl, ] |
| <expMDM> | this, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, (, Not, lambda | PM, RelOp, And, Or, in, ), ,, nl, ] |  |
| <exp\_MDM> | DM, \*, € | PM, RelOp, And, Or, in, ), ,, nl, ] | DM, \*, PM, RelOp, And, Or, in, ), ,, nl, ] |
| <exp\_F> | this, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, (, Not, lambda | DM, \*, PM, RelOp, And, Or, in, ), ,, nl, ] |  |
| <ID\_rel> | [, (, AcOp, € | AsOp, =, \*, DM, PM, RelOp, And, Or, in, nl, ), ,, ] | [, (, AcOp, AsOp, =, \*, DM, PM, RelOp, And, Or, in, nl, ), ,, ] |
| <ID\_rel1> | AcOp, [, € | AsOp, =, \*, DM, PM, RelOp, And, Or, in, nl, ), ,, ], :, } | AcOp, [, AsOp, =, \*, DM, PM, RelOp, And, Or, in, nl, ), ,, ] , :, } |
| <ID\_rel2> | [, € | AsOp, =, \*, DM, PM, RelOp, And, Or, in, nl, ), ,, ], :, } | [, AsOp, =, \*, DM, PM, RelOp, And, Or, in, nl, ), ,, ] |
| <This> | this, € | ID | this, ID |
|  |  |  |  |
| <In> | in, € | DM, \*, PM, RelOp, And, Or, in, ), ,, nl, ] | In, DM, \*, PM, RelOp, And, Or, in, ), ,, nl, ] |
| <In\_> | ID, [, List | DM, \*, PM, RelOp, And, Or, in, ), ,, nl, ] |  |
| <body> | break, continue, return, for, while, if, try, del, global, this, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, nl, pass | def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, ~, IndentOut, nl, elif, else, finally, except |  |
| <M\_St> | break, continue, return, for, while, if, try, del, global, this, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final | IndentOut |  |
| <M\_St\_> | break, continue, return, for, while, if, try, del, global, this, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, € | IndentOut | break, continue, return, for, while, if, try, del, global, this, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, IndentOut |
| <S\_St> | break, continue, return, for, while, if, try, del, global, this, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final | nl |  |
| <NL> | nl, € | IndentOut | nl, IndentOut |
| <Class> | class | def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, ~ |  |
| <Inherit> | ID, € | ) | ID, ) |
| <Inherit\_> | ,, € | ) | ,, ) |
| <class\_body> | ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, def, pass | IndentOut |  |
| <class\_body1> | ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, def | IndentOut |  |
| <class\_body2> | accessModifier, ID, DT, String | IndentOut |  |
| <class\_body3> | ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, def, € | IndentOut | ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, def, IndentOut |
| <Class\_call> | new | nl |  |
| <constructor> | accessModifier, ID | IndentOut |  |
| <arg\_list> | DT, string, € | ) | DT, string, ) |
| <arg\_list1> | ,, € | ) | ,, ) |
| <arg\_list\_call> | this, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, (, Not, lambda, € | ) | this, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, (, Not, lambda, ) |
| <arg\_list\_call1> | ,, € | ) | ,, ) |
| <function> | DT, string | def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, ~, IndentOut |  |
| <function\_> | static, final, ID | def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, ~, IndentOut |  |
| <data\_type> | DT, string | \*, accessModifier, ID, static, final |  |
| <For> | for | def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, ~, nl |  |
| <For\_> | (, € | : | (, : |
| <While> | while | def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, ~, nl |  |
| <If> | If | def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, ~, nl |  |
| <Elif> | elif, else, € | def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, ~, nl | elif, else, def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, ~, nl |
| <Else> | else, € | def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, ~, nl | Else, def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, ~, nl |
| <List> | [, List | DM, \*, PM, RelOp, And, Or, in, ), ,, nl, ] |  |
| <List1> | [, ID | ) |  |
| <List2> | [ | DM, \*, PM, RelOp, And, Or, in, ), ,, nl, ] |  |
| <List3> | ID, int\_const, float\_const, string\_const, char\_const, bool\_const, [, € | ] | ID, int\_const, float\_const, string\_const, char\_const, bool\_const, [, ] |
| <List4> | ,, € | ] | ,, ] |
| <List5> | ID, int\_const, float\_const, string\_const, char\_const, bool\_const, [ | ] |  |
| <Dictionary> | {, dict | nl |  |
| <Dictionary1> | ID, { | ) |  |
| <Dictionary2> | { | nl, ), } |  |
| <Dictionary3> | ID, int\_const, float\_const, string\_const, char\_const, bool\_const | } |  |
| <Dictionary4> | ID, int\_const, float\_const, string\_const, char\_const, bool\_const, { | } |  |
| <Dictionary6> | ,, € | } | ,, } |
| <Del> | del | nl |  |
| <Del\_> | [, € | nl | [, nl |
| <Try> | try | def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, ~, nl |  |
| <Except> | finally, except | def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, ~, nl |  |
| <Finally> | finally | def, class, for, while, if, try, del, ID, int\_const, float\_const, string\_const, char\_const, bool\_const, \*, accessModifier, static, final, ~, nl |  |
| <exception> | exception | ) |  |
| <Lambda> | lambda | DM, \*, PM, RelOp, And, Or, in, ), ,, nl, ] |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |