| | | |
|---|---|---|
| *prg* | ⇒ | *[( module* $\mid$ *class )] [interface ]*$^*$ *[tfspec ]*$^*$ *[structdef ]*$^*$ *[typedef ]*$^*$ *[object ]*$^*$ *[function ]*$^*$ |
| *module* | ⇒ | **module** ***id*** *[* **deprecated** ***str*** *]* **;** |
| *class* | ⇒ | **class** ***id*** *[* **deprecated** ***str*** *]* **;** *classtype [pragmas ]* |
| *classtype* | ⇒ | **classtype** *ntype* **;** |
| | $\mid$ | **extern** **classtype** **;** |
| *typedef* | ⇒ | *( loctypedef* $\mid$ *exttypedef [pragmas ] )* |
| *object* | ⇒ | *( objdef* $\mid$ *extobjdef[pragmas ] )* |
| *function* | ⇒ | *(* **external** $\mid$ **specialize** *) fundec [pragmas ]* |
| | $\mid$ | *fundef [pragmas ]* |
| | $\mid$ | *main* |
| *interface* | ⇒ | *(* **import** $\mid$ **use** $\mid$ **export** $\mid$ **provide** *)* ***id*** **:** *(* **all** *[* **except** { *ext_id [* **,** *ext_id ]* } *]* $\mid$ { *ext_id [* **,** *ext_id ]* } *)* **;** |
| *structdef* | ⇒ | **struct** ***id*** { *[ntype* ***id*** *[* **,** ***id*** *]*$^*$ **;** *]*$^*$ } **;** |
| *loctypedef* | ⇒ | **typedef** *ntype* ***id*** **;** |
| *exttypedef* | ⇒ | **external** **typedef** ***id*** **;** |
| *objdef* | ⇒ | **objdef** *ntype* ***id*** = *expr_ap* **;** |
| *extobjdef* | ⇒ | **external** **objdef** *ntype* ***id*** **;** |
| *fundef* | ⇒ | *[* **inline** *] [* **thread** *] (* **void** $\mid$ *ntype [* **,** *ntype ]*$^*$ *)* *(* **(** *ext_id* **)** $\mid$ *ext_id )* **(** *vardec* **)** *body* |
| *vardec* | ⇒ | *[( arg [* **,** *arg ]*$^*$ $\mid$ *)]* |
| *varargs* | ⇒ | *arg [* **,** *( arg* $\mid$ *)] [* **...** *]* |
| *arg* | ⇒ | *ntype [* **&** *]* ***id*** |
| *main* | ⇒ | **int** **main** **(** *[* **void** *]* **)** *body* |

```
fundec          ⇒  ( void  | varntypes  |  . . .  ) ext_id
                      ( [( varargs  |  void  |  . . .  ) ] )   ;
hash_pragma     ⇒   #   pragma
pragmacachesim  ⇒  hash_pragma  cachesim  [[ str  ]+ ]
pragmas         ⇒  [pragma ]+
                |  hash_pragma  wlcomp  expr_ap
pragma          ⇒  hash_pragma (  linkname   |  cudalinkname
                      |  linkwith   |  linkobj   |  copyfun   |
                      freefun  ) [  str  ]+
                |  hash_pragma (  linksign  |  refcounting  )  [  nums  ]
                |  hash_pragma  effect  qual_ext_id [  ,   qual_ext_id ]*
                |  hash_pragma (  recountdots   |  mutcthreadfun   |
                      noinline  )
body            ⇒   {  [pragmacachesim ] [ntype  id  [  ,   id  ]*   ;   ]*
                    [statement ]* [ return  [  (  [exprs ]  )  ] ]   ;    }
statement       ⇒   ;
                |  let   ;
                |  cond
                |  doloop
                |  whileloop
                |  forloop
let             ⇒  id [  ,   id ]*   =   expr
                |   id   .   id   =   expr
                |   id   [  exprs  ]   =   expr
                |  expr_ap
                |  expr_with
                |   id  (  ++  |  --  )
                |  (  ++  |  --  )  id
                |   id  (  +=   |  -=   |  *=   |  /=   |  %=  ) expr
```

| | | |
|---|---|---|
| *cond* | ⇒ | **if** ( *expr* ) *statementblock* [ **else** *statementblock* ] |
| *doloop* | ⇒ | **do** *statementblock* **while** ( *expr* ) ; |
| *whileloop* | ⇒ | **while** ( *expr* ) *statementblock* |
| *forloop* | ⇒ | **for** ( *let* [ , *let* ]* ; *expr* ; *let* [ , *let* ]* ) *statementblock* |
| *statementblock* | ⇒ | { [ *pragmacachesim* ] [ *statement* ]* } |
| | \| | *statement* |
| *expr* | ⇒ | . |
| | \| | . . . |
| | \| | *subexpr* |
| *exprs* | ⇒ | *expr* [ , *expr* ]* |
| *subexpr* | ⇒ | *subexpr* . ***id*** |
| | \| | *nostrexpr* |

| | | |
|---|---|---|
| *nostrexpr* | ⇒ | *qual_ext_id* |
| | \| | **numbyte** |
| | \| | **numshort** |
| | \| | **numint** |
| | \| | **numlong** |
| | \| | **numlonglong** |
| | \| | **numubyte** |
| | \| | **numushort** |
| | \| | **numuint** |
| | \| | **numulong** |
| | \| | **numulonglong** |
| | \| | ***num*** |
| | \| | **float** |
| | \| | **double** |
| | \| | **char** |
| | \| | *[* ***str*** *]+* |
| | \| | **true** |
| | \| | **false** |
| | \| | *expr (* `&&` \| `\|\|` *) expr* |
| | \| | *expr* `?` *expr* `:` *expr* |
| | \| | `(` *expr* `)` |
| | \| | *expr qual_ext_id expr* |
| | \| | *(* `+` \| `-` \| `~` \| `!` *) expr* |
| | \| | *(* `+` \| `-` \| `~` \| `!` *)* `(` *expr* `,` *exprs* `)` |
| | \| | *expr* `[` *[exprs ]* `]` |
| | \| | *expr_ap* |
| | \| | *expr_with* |
| | \| | *expr_ar* |
| | \| | `(` `:` *ntype* `)` *expr* |
| | \| | { ***id*** `->` *expr* } |
| | \| | { `[` *exprs* `]` `->` *expr* } |
| *expr_ar* | ⇒ | `[` *[exprs ]* `]` |
| | \| | `[` `:` *ntype* `]` |
| | \| | `<` *exprs* `>` |

| | | |
|---|---|---|
| *expr_with* | ⇒ | *[* **local** *]* **with** *with* |
| *with* | ⇒ | *[* **{** *[ with_opt ] [ [ generators ]+ ]* **}** *]* |
| | | **:** *operators* |
| *with_opt* | ⇒ | *hash_pragma* **wlcomp** *expr_ap* |
| *expr_ap* | ⇒ | *qual_ext_id* **(** *[ exprs ]* **)** |
| | \| | *prf* **(** *[ exprs ]* **)** |
| | \| | **spawn** *[* **(** ***str*** **)** *] qual_ext_id* **(** *[ exprs ]* **)** |
| | \| | **rspawn** *[* **(** ***str*** **)** *] qual_ext_id* **(** *[ exprs ]* **)** |
| *generators* | ⇒ | **(** *generator* **)** *[* { *[ statement ]** } *]* |
| | | *[* **:** *(* **(** *expr* **,** *exprs* **)** \| *exprs ) ]* **;** |
| *generator* | ⇒ | *expr (* **<=** \| **<** *) (* ***id*** \| *[* ***id*** **=** *]* |
| | | **[** ***id*** *[* **,** ***id*** *]** **]** *) (* **<=** \| **<** *)* |
| | | *expr [* **step** *expr ] [* **width** *expr ]* |
| | \| | |
| *operators* | ⇒ | **(** *nwithop [* **,** *nwithop ]* **)** |
| | \| | *nwithop* |
| | \| | **void** |
| *genidx* | ⇒ | *[* ***id*** **=** *]* **[** ***id*** *[* **,** ***id*** *]** **]** |
| | \| | ***id*** |
| *nwithop* | ⇒ | **genarray** **(** *expr [* **,** *expr ]* **)** |
| | \| | **modarray** **(** *expr* **)** |
| | \| | **fold** **(** *qual_ext_id* **,** *expr* **)** |
| | \| | **foldfix** **(** *qual_ext_id* **,** *expr* **,** *expr* **)** |
| | \| | **propagate** **(** *expr* **)** |

| | | |
|---|---|---|
| *prf* | ⇒ | **_dim_A_** |
| | \| | **_shape_A_** |
| | \| | **_reshape_VxA_** |
| | \| | **_sel_VxA_** |
| | \| | **_modarray_AxVxS_** |
| | \| | **_sel_VxIA_** |
| | \| | **_hideValue_SxA_** |
| | \| | **_hideShape_SxA_** |
| | \| | **_hideDim_SxA_** |
| | \| | **_add_SxS_** |
| | \| | **_add_SxV_** |
| | \| | **_add_VxS_** |
| | \| | **_add_VxV_** |
| | \| | **_sub_SxS_** |
| | \| | **_sub_SxV_** |
| | \| | **_sub_VxS_** |
| | \| | **_sub_VxV_** |
| | \| | **_mul_SxS_** |
| | \| | **_mul_SxV_** |
| | \| | **_mul_VxS_** |
| | \| | **_mul_VxV_** |
| | \| | **_div_SxS_** |
| | \| | **_div_SxV_** |
| | \| | **_div_VxS_** |
| | \| | **_div_VxV_** |
| | \| | **_mod_SxS_** |
| | \| | **_mod_SxV_** |
| | \| | **_mod_VxS_** |
| | \| | **_mod_VxV_** |
| | \| | **_abs_S_** |
| | \| | **_abs_V_** |
| | \| | **_neg_S_** |
| | \| | **_neg_V_** |
| | \| | **_reciproc_S_** |
| | \| | **_reciproc_V_** |
| | \| | **_min_SxS_** |
| | \| | **_min_SxV_** |
| | \| | **_min_VxS_** |
| | \| | **_min_VxV_** |
| | \| | **_max_SxS_** |
| | \| | **_max_SxV_** |
| | \| | **_max_VxS_** |
| | \| | **_max_VxV_** |
| | \| | **_eq_SxS_** |
| | \| | **_eq_SxV_** |
| | \| | **_eq_VxS_** |
| | \| | **_eq_VxV_** |
| | \| | **_neq_SxS_** |

```
qual_ext_id     ⇒   [ id  ::  ] ext_id
ext_id          ⇒   ( id | reservedid )

reservedid      ⇒    genarray
                |    modarray
                |    all
                |    &
                |    !
                |    ++
                |    --
                |    +
                |    -
                |    *
                |    <=
                |    <
                |    >
tfspec          ⇒   [ tfdef ]+ [ tfrel ]+
tfdef           ⇒    abstract-typedef  id  [ tfarg ]  ;
                |    user-typedef  id  tfarg  ;
                |    builtin-typedef  ( simplentype | id
                      [ (||  tfarg  ||)  ] )  ;
tfarg           ⇒    id  [ ::  id  ] [  ,  tfarg ]
tfrel           ⇒    typerel  simplentype  <:  ( simplentype | id  )  ;
                |    typerel  id  <:  id  [ iff  tfexprs ]  ;
tfexprs         ⇒   tfexprs (  .  (  <  |  <=  )  |  .>  |  .>=  |  .*  |
                      ./  |  .+  |  .-  )
                |    (  tfexprs  )
                |   (  id  |  num  )
varntypes       ⇒   ntype [  ,  ( ntype [  ,  ntype ]* |  ...  ) ]
ntype           ⇒   basentype
                |   basentype  [  [ exprs ]  ]
basentype       ⇒   simpletype
                |   userntype
                |   polyntype
```

| | | |
|---|---|---|
| *simplentype* | ⇒ | **byte** |
| | \| | **short** |
| | \| | **int** |
| | \| | **long** |
| | \| | **longlong** |
| | \| | **ubyte** |
| | \| | **ushort** |
| | \| | **uint** |
| | \| | **ulong** |
| | \| | **ulonglong** |
| | \| | **float** |
| | \| | **bool** |
| | \| | **char** |
| | \| | **double** |
| *userntype* | ⇒ | ( [ **struct** ] \| ***id*** `::` ) ***id*** |
| *polyntype* | ⇒ | `<` ***id*** [ `=` ***id*** `[` ***id*** `]` ] `>` |
| | \| | `<` ***id*** ( `->` \| `<-` ) ***id*** `[` ***id*** `]` `>` |
| *targets* | ⇒ | [ **target** ***id*** [ `::` ***id*** ]* `:` *resources* ]* |
| *resources* | ⇒ | [ ***id*** ( `:` \| `+=` ) `=` ( ***num*** \| [ ***str*** ]* ) ]* |