# Annex A
(informative)
# Language syntax summary

1 **NOTE 1** The notation is described in 6.1.

## A.1 Lexical grammar

### A.1.1 Lexical elements

(6.4) *token:*
> *keyword*
> *identifier*
> *constant*
> *string-literal*
> *punctuator*

(6.4) *preprocessing-token:*
> *header-name*
> *identifier*
> *pp-number*
> *character-constant*
> *string-literal*
> *punctuator*
> each universal-character-name that cannot be one of the above
> each non-white-space character that cannot be one of the above

### A.1.2 Keywords

(6.4.1) *keyword:* one of

| | | | |
|---|---|---|---|
| **`alignas`** | **`enum`** | **`short`** | **`void`** |
| **`alignof`** | **`extern`** | **`signed`** | **`volatile`** |
| **`auto`** | **`false`** | **`sizeof`** | **`while`** |
| **`bool`** | **`float`** | **`static`** | **`_Atomic`** |
| **`break`** | **`for`** | **`static_assert`** | **`_BitInt`** |
| **`case`** | **`goto`** | **`struct`** | **`_Complex`** |
| **`char`** | **`if`** | **`switch`** | **`_Decimal128`** |
| **`const`** | **`inline`** | **`thread_local`** | **`_Decimal32`** |
| **`constexpr`** | **`int`** | **`true`** | **`_Decimal64`** |
| **`continue`** | **`long`** | **`typedef`** | **`_Generic`** |
| **`default`** | **`nullptr`** | **`typeof`** | **`_Imaginary`** |
| **`do`** | **`register`** | **`typeof_unqual`** | **`_Noreturn`** |
| **`double`** | **`restrict`** | **`union`** | |
| **`else`** | **`return`** | **`unsigned`** | |

### A.1.3 Identifiers

(6.4.2.1) *identifier:*
> *identifier-start*
> *identifier identifier-continue*

(6.4.2.1) *identifier-start:*
> *nondigit*
> XID_Start character
> universal-character-name of class XID_Start

(6.4.2.1) *identifier-continue:*
> *digit*
> *nondigit*
> XID_Continue character
> universal-character-name of class XID_Continue

(6.4.2.1) *nondigit:* one of

**`_ a b c d e f g h i j k l m`**
**`n o p q r s t u v w x y z`**
**`A B C D E F G H I J K L M`**
**`N O P Q R S T U V W X Y Z`**

(6.4.2.1) *digit:* one of

**`0 1 2 3 4 5 6 7 8 9`**

## A.1.4 Universal character names

(6.4.3) *universal-character-name:*

`\u` *hex-quad*
`\U` *hex-quad hex-quad*

(6.4.3) *hex-quad:*

*hexadecimal-digit hexadecimal-digit hexadecimal-digit hexadecimal-digit*

## A.1.5 Constants

(6.4.4) *constant:*

*integer-constant*
*floating-constant*
*enumeration-constant*
*character-constant*
*predefined-constant*

(6.4.4.1) *integer-constant:*

*decimal-constant integer-suffix*$_{\text{opt}}$
*octal-constant integer-suffix*$_{\text{opt}}$
*hexadecimal-constant integer-suffix*$_{\text{opt}}$
*binary-constant integer-suffix*$_{\text{opt}}$

(6.4.4.1) *decimal-constant:*

*nonzero-digit*
*decimal-constant* **`'`**$_{\text{opt}}$ *digit*

(6.4.4.1) *octal-constant:*

**`0`**
*octal-constant* **`'`**$_{\text{opt}}$ *octal-digit*

(6.4.4.1) *hexadecimal-constant:*

*hexadecimal-prefix hexadecimal-digit-sequence*

(6.4.4.1) *binary-constant:*

*binary-prefix binary-digit*
*binary-constant* **`'`**$_{\text{opt}}$ *binary-digit*

(6.4.4.1) *hexadecimal-prefix:* one of

**`0x 0X`**

(6.4.4.1) *binary-prefix:* one of

**`0b 0B`**

(6.4.4.1) *nonzero-digit:* one of

**`1 2 3 4 5 6 7 8 9`**

(6.4.4.1) *octal-digit:* one of

**`0 1 2 3 4 5 6 7`**

*hexadecimal-digit-sequence:*

*hexadecimal-digit*
*hexadecimal-digit-sequence* **`'`**$_{\text{opt}}$ *hexadecimal-digit*

(6.4.4.1) *hexadecimal-digit:* one of

**`0 1 2 3 4 5 6 7 8 9`**
**`a b c d e f`**
**`A B C D E F`**

(6.4.4.1) *binary-digit:* one of

**`0 1`**

(6.4.4.1) *integer-suffix:*

*unsigned-suffix long-suffix*$_{\text{opt}}$
*unsigned-suffix long-long-suffix*
*unsigned-suffix bit-precise-int-suffix*
*long-suffix unsigned-suffix*$_{\text{opt}}$
*long-long-suffix unsigned-suffix*$_{\text{opt}}$
*bit-precise-int-suffix unsigned-suffix*$_{\text{opt}}$

(6.4.4.1) *bit-precise-int-suffix:* one of

**`wb WB`**

(6.4.4.1) *unsigned-suffix:* one of

**`u U`**

(6.4.4.1) *long-suffix:* one of

**`l L`**

(6.4.4.1) *long-long-suffix:* one of

**`ll LL`**

(6.4.4.2) *floating-constant:*

*decimal-floating-constant*
*hexadecimal-floating-constant*

(6.4.4.2) *decimal-floating-constant:*

*fractional-constant exponent-part*$_{\text{opt}}$ *floating-suffix*$_{\text{opt}}$
*digit-sequence exponent-part floating-suffix*$_{\text{opt}}$

(6.4.4.2) *hexadecimal-floating-constant:*

*hexadecimal-prefix hexadecimal-fractional-constant*
*binary-exponent-part floating-suffix*$_{\text{opt}}$
*hexadecimal-prefix hexadecimal-digit-sequence*
*binary-exponent-part floating-suffix*$_{\text{opt}}$

(6.4.4.2) *fractional-constant:*

*digit-sequence*$_{\text{opt}}$ **`.`** *digit-sequence*
*digit-sequence* **`.`**

(6.4.4.2) *exponent-part:*

**`e`** *sign*$_{\text{opt}}$ *digit-sequence*
**`E`** *sign*$_{\text{opt}}$ *digit-sequence*

(6.4.4.2) *sign:* one of

**`+ -`**

(6.4.4.2) *digit-sequence:*

*digit*
*digit-sequence* **`'`**$_{\text{opt}}$ *digit*

(6.4.4.2) *hexadecimal-fractional-constant:*

*hexadecimal-digit-sequence*$_{\text{opt}}$ **`.`** *hexadecimal-digit-sequence*
*hexadecimal-digit-sequence* **`.`**

(6.4.4.2) *binary-exponent-part:*

**`p`** *sign*$_{\text{opt}}$ *digit-sequence*
**`P`** *sign*$_{\text{opt}}$ *digit-sequence*

(6.4.4.2) *floating-suffix:* one of

**`f l F L df dd dl DF DD DL`**

(6.4.4.3) *enumeration-constant:*

*identifier*

(6.4.4.4) *character-constant:*

*encoding-prefix*$_{\text{opt}}$ **`'`** *c-char-sequence* **`'`**

(6.4.4.4) *encoding-prefix:* one of

**`u8`** **`u`** **`U`** **`L`**

(6.4.4.4) *c-char-sequence:*

*c-char*
*c-char-sequence c-char*

(6.4.4.4) *c-char:*

any member of the source character set except
the single-quote `'`, backslash `\`, or new-line character
*escape-sequence*

(6.4.4.4) *escape-sequence:*

*simple-escape-sequence*
*octal-escape-sequence*
*hexadecimal-escape-sequence*
*universal-character-name*

(6.4.4.4) *simple-escape-sequence:* one of

`\' \" \? \\`
`\a \b \f \n \r \t \v`

(6.4.4.4) *octal-escape-sequence:*

`\` *octal-digit*
`\` *octal-digit octal-digit*
`\` *octal-digit octal-digit octal-digit*

(6.4.4.4) *hexadecimal-escape-sequence:*

`\x` *hexadecimal-digit*
*hexadecimal-escape-sequence hexadecimal-digit*

(6.4.4.5) *predefined-constant:*

**`false`**
**`true`**
**`nullptr`**

## A.1.6 String literals

(6.4.5) *string-literal:*

*encoding-prefix*$_{\text{opt}}$ **`"`** *s-char-sequence*$_{\text{opt}}$ **`"`**

(6.4.5) *s-char-sequence:*

*s-char*
*s-char-sequence s-char*

(6.4.5) *s-char:*

any member of the source character set except
the double-quote `"`, backslash `\`, or new-line character
*escape-sequence*

### A.1.7 Punctuators

(6.4.6) *punctuator:* one of

```
[  ]  (  )  {  }  .  ->
++  --  &  *  +  -  ~  !
/  %  <<  >>  <  >  <=  >=  ==  !=  ^  |  &&  ||
?  :  ::  ;  ...
=  *=  /=  %=  +=  -=  <<=  >>=  &=  ^=  |=
,  #  ##
<:  :>  <%  %>  %:  %:%:
```

### A.1.8 Header names

(6.4.7) *header-name:*
  **<** *h-char-sequence* **>**
  **"** *q-char-sequence* **"**

(6.4.7) *h-char-sequence:*
  *h-char*
  *h-char-sequence h-char*

(6.4.7) *h-char:*
  any member of the source character set except
    the new-line character and >

(6.4.7) *q-char-sequence:*
  *q-char*
  *q-char-sequence q-char*

(6.4.7) *q-char:*
  any member of the source character set except
    the new-line character and **"**

### A.1.9 Preprocessing numbers

(6.4.8) *pp-number:*
  *digit*
  **.** *digit*
  *pp-number identifier-continue*
  *pp-number* **'** *digit*
  *pp-number* **'** *nondigit*
  *pp-number* **e** *sign*
  *pp-number* **E** *sign*
  *pp-number* **p** *sign*
  *pp-number* **P** *sign*
  *pp-number* **.**

## A.2 Phrase structure grammar

### A.2.1 Expressions

(6.5.1) *primary-expression:*
  *identifier*
  *constant*
  *string-literal*
  **(** *expression* **)**
  *generic-selection*

(6.5.1.1) *generic-selection:*
  **_Generic (** *assignment-expression* **,** *generic-assoc-list* **)**

(6.5.1.1) *generic-assoc-list:*
  *generic-association*
  *generic-assoc-list* **,** *generic-association*