# Open Data Description Language
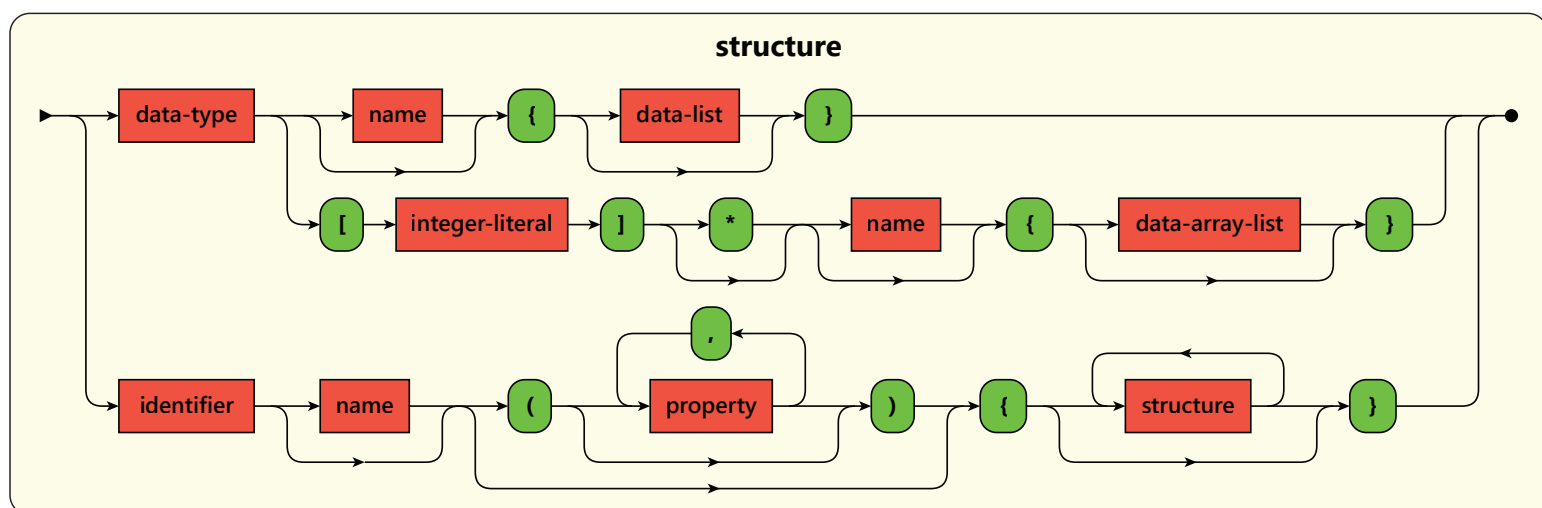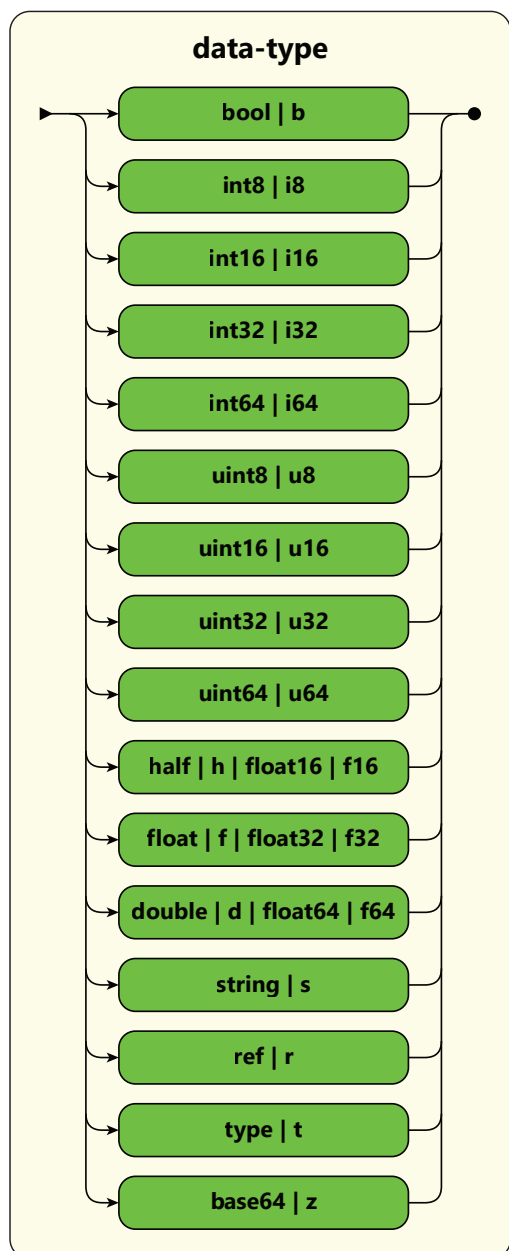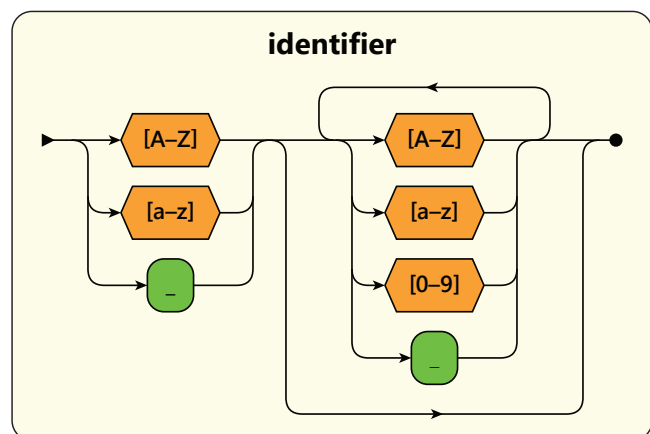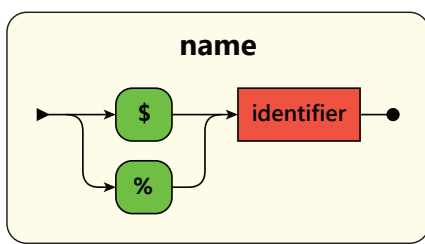
## openddl.org

### structure

```
structure ::= data-type (name? "{" data-list? "}" | "[" integer-literal "]" "*"? name? "{" data-array-list? "}")
            | identifier name? ("(" (property ("," property)*)? ")")? "{" structure* "}"
```
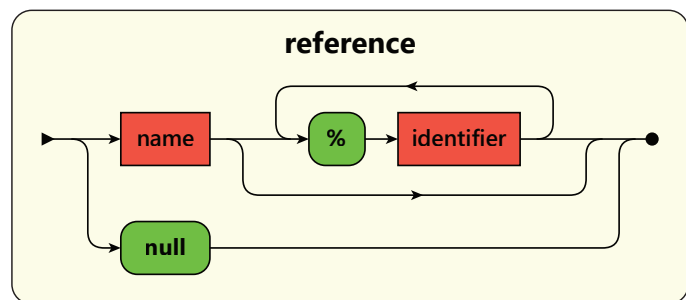
### data-type

```
data-type ::=
"bool"    | "b"   |
"int8"    | "i8"  |
"int16"   | "i16" |
"int32"   | "i32" |
"int64"   | "i64" |
"uint8"   | "u8"  |
"uint16"  | "u16" |
"uint32"  | "u32" |
"uint64"  | "u64" |
"half"    | "h"   |
"float16" | "f16" |
"float"   | "f"   |
"float32" | "f32" |
"double"  | "d"   |
"float64" | "f64" |
"string"  | "s"   |
"ref"     | "r"   |
"type"    | "t"   |
"base64"  | "z"
```
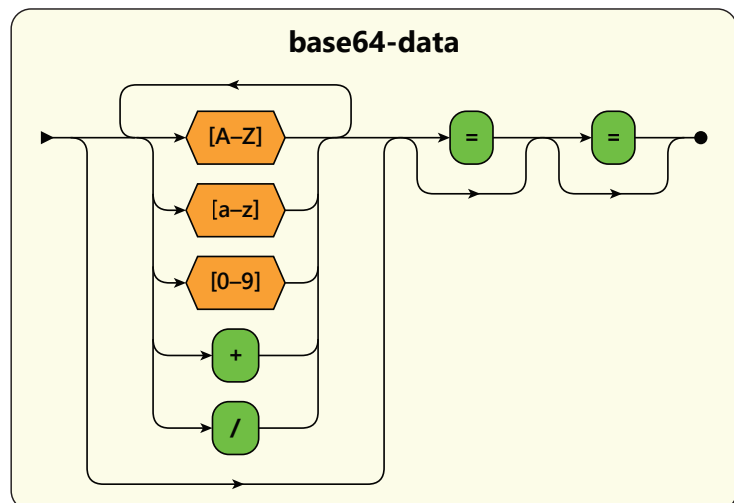
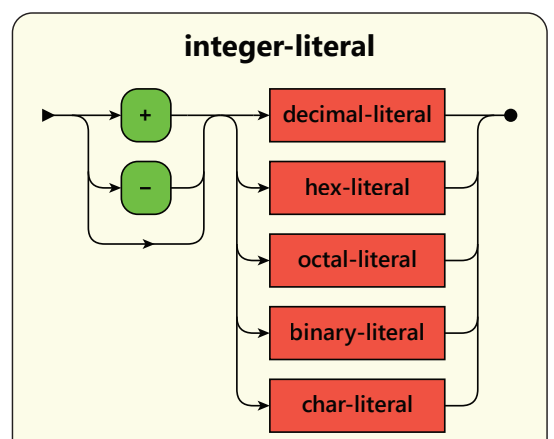### identifier

```
identifier ::= [A-Za-z_] [A-Za-z0-9_]*
```

### name

```
name ::= ("$" | "%") identifier
```

### reference

```
reference ::= name ("%" identifier)* | "null"
```

### data-list

```
data-list ::=
bool-literal ("," bool-literal)* |
integer-literal ("," integer-literal)* |
float-literal ("," float-literal)* |
string-literal ("," string-literal)* |
reference ("," reference)* |
data-type ("," data-type)* |
base64-data ("," base64-data)*
```

### data-array-list

```
data-array-list ::=
identifier? "{" bool-literal ("," bool-literal)* "}"
    ("," "{" bool-literal ("," bool-literal)* "}")* |
identifier? "{" integer-literal ("," integer-literal)* "}"
    ("," "{" integer-literal ("," integer-literal)* "}")* |
identifier? "{" float-literal ("," float-literal)* "}"
    ("," "{" float-literal ("," float-literal)* "}")* |
identifier? "{" string-literal ("," string-literal)* "}"
    ("," "{" string-literal ("," string-literal)* "}")* |
identifier? "{" reference ("," reference)* "}"
    ("," "{" reference ("," reference)* "}")* |
identifier? "{" data-type ("," data-type)* "}"
    ("," "{" data-type ("," data-type)* "}")* |
identifier? "{" base64-data ("," base64-data)* "}"
    ("," "{" base64-data ("," base64-data)* "}")*
```

### bool-literal

```
bool-literal ::=
"false" | "0" | "true" | "1"
```

### property

```
property ::=
identifier ("="
(bool-literal |
integer-literal |
float-literal |
string-literal |
reference |
data-type |
base64-data))?
```

### hex-digit

```
hex-digit ::= [0-9A-Fa-f]
```

### decimal-literal

```
decimal-literal ::=
[0-9] ("_"? [0-9])*
```

### hex-literal

```
hex-literal ::=
("0x" | "0X")
hex-digit
("_"? hex-digit)*
```

### octal-literal

```
octal-literal ::=
("0o" | "0O")
[0-7] ("_"? [0-7])*
```

### binary-literal

```
binary-literal ::=
("0b" | "0B")
("0" | "1")
("_"? ("0" | "1"))*
```

### base64-data
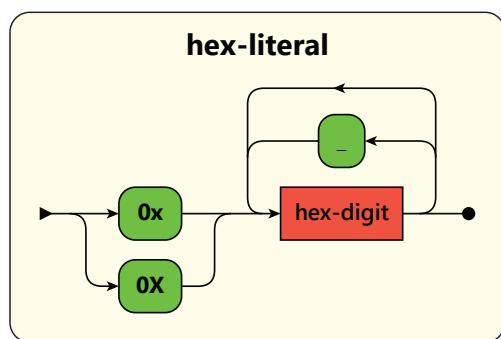
```
base64-data ::= ([A-Za-z0-9] | "+" | "/")* "="? "="?
```

### integer-literal

```
integer-literal ::= ("+" | "-")?
(decimal-literal | hex-literal | octal-literal
| binary-literal | char-literal)
```
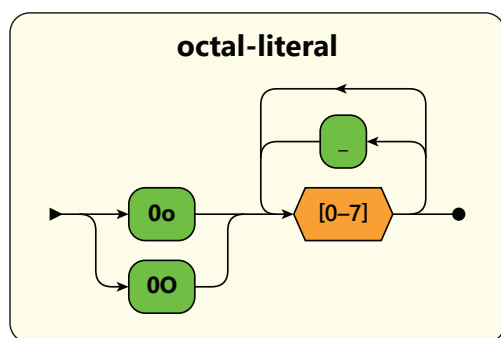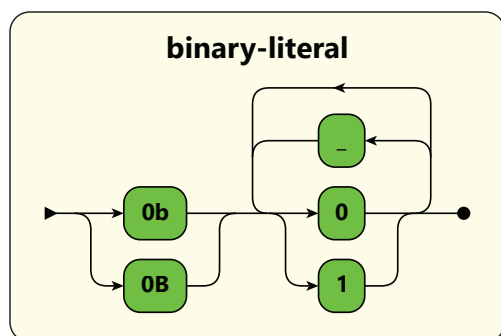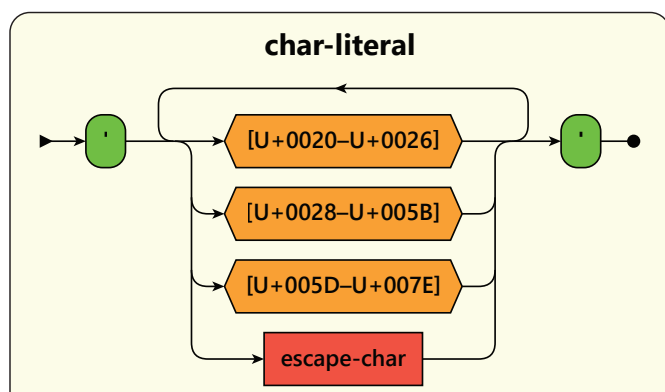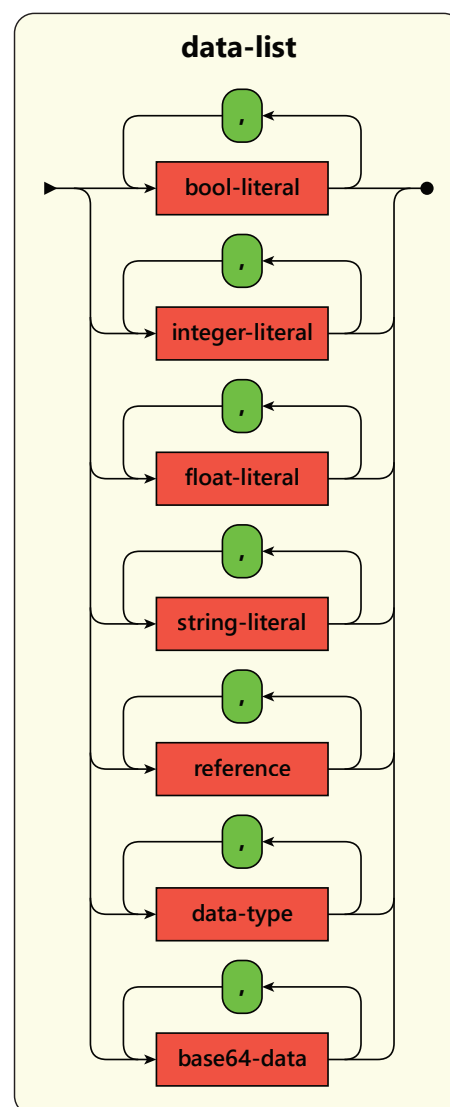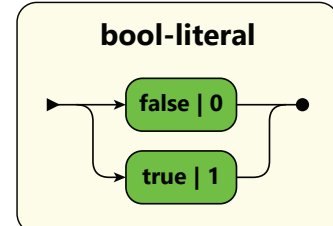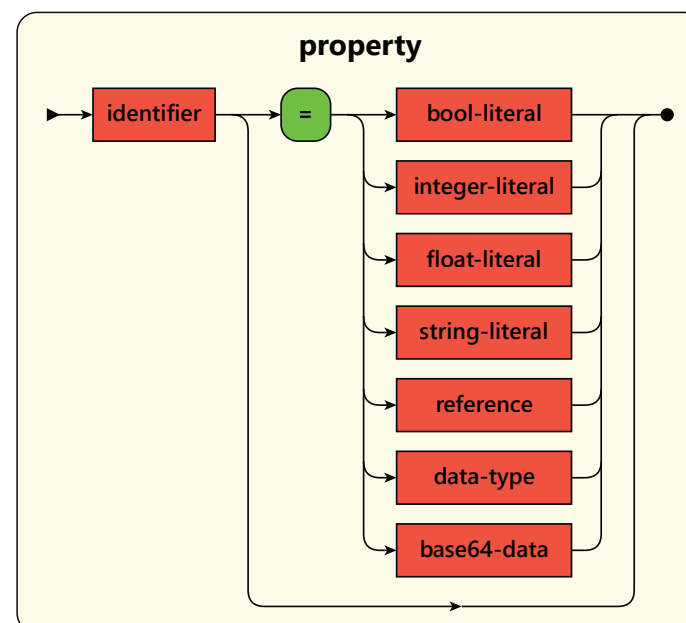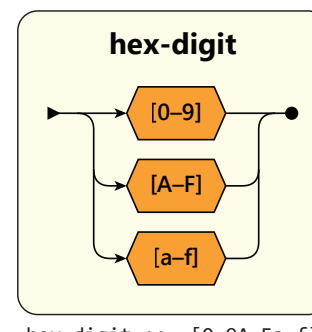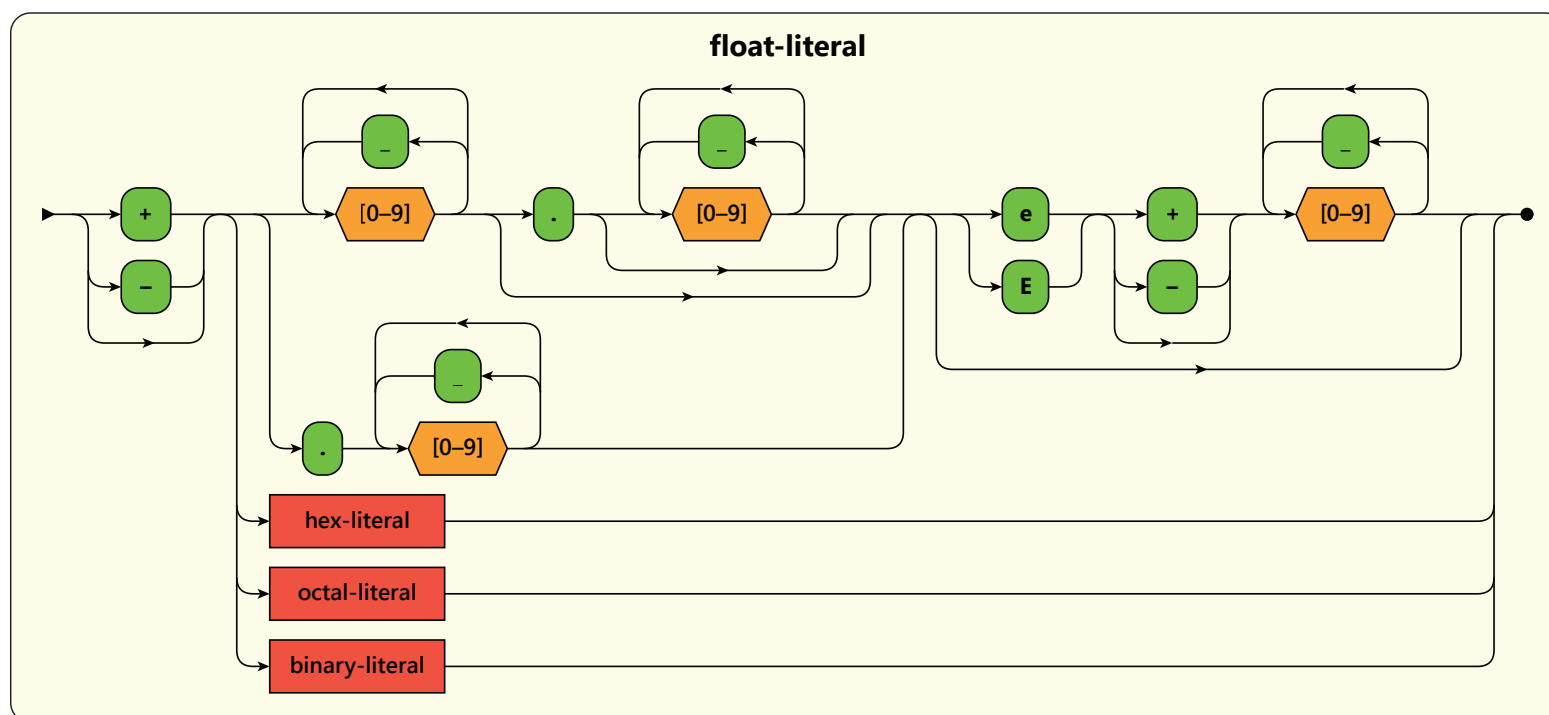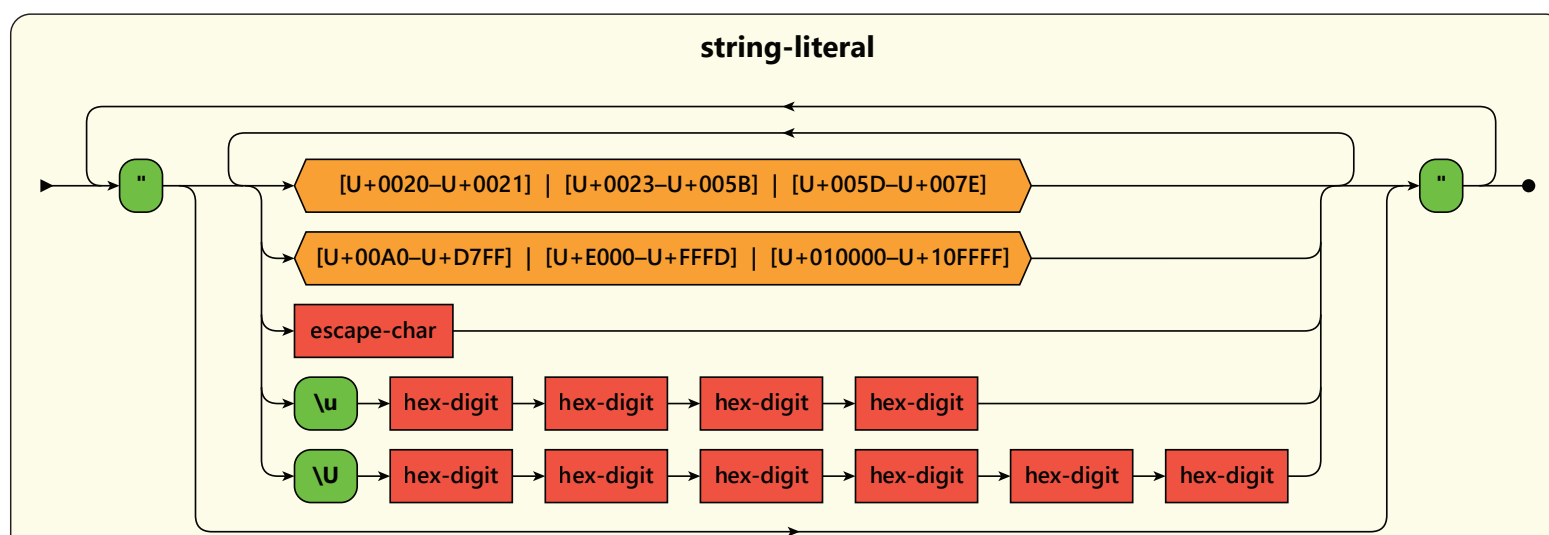
### float-literal

```
float-literal ::= ("+" | "-")? (([0-9] ("_"? [0-9])* ("." ([0-9] ("_"? [0-9])*)?)? | "." [0-9] ("_"? [0-9])*)
(("e" | "E") ("+" | "-")? [0-9] ("_"? [0-9])*)*) | hex-literal | octal-literal | binary-literal)
```

### char-literal

```
char-literal ::=
"'" ([#x20-#x26#x28-#x5B#x5D-#x7E] | escape-char)+ "'"
```

### escape-char

```
escape-char ::= '\"' | "\'" | "\?" | "\\"
| "\a" | "\b" | "\f" | "\n" | "\r" | "\t" | "\v"
| "\x" hex-digit hex-digit
```

### string-literal

```
string-literal ::= ('"' ([#x20-#x21#x23-#x5B#x5D-#x7E#xA0-#xD7FF#xE000-#xFFFD#x010000-#x10FFFF] | escape-char |
"\u" hex-digit hex-digit hex-digit hex-digit | "\U" hex-digit hex-digit hex-digit hex-digit hex-digit hex-digit)* '"')+
```