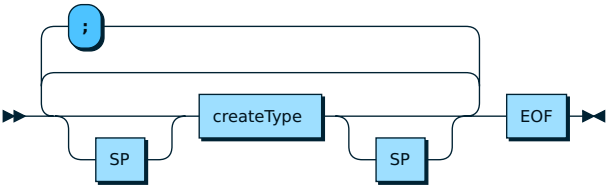


# PG-Schema Grammar

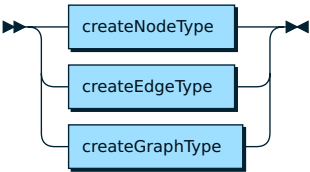
pgs:



`pgs ::= SP? createType SP? ( ';' SP? createType SP? )* EOF`

no references

createType:

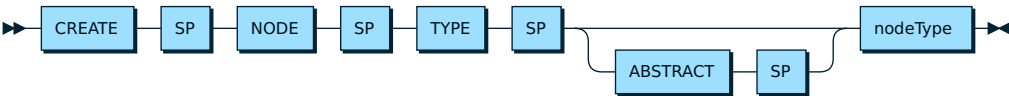


`createType ::= createNodeType | createEdgeType | createGraphType`

referenced by:

- `pgs`

createNodeType:

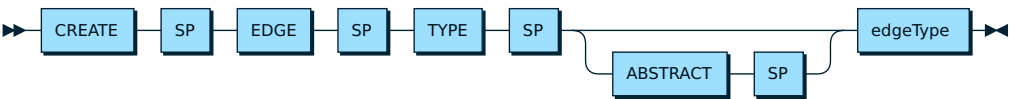


`createNodeType ::= CREATE SP NODE SP TYPE SP ( ABSTRACT SP )? nodeType`

referenced by:

- `createType`

createEdgeType:



`createEdgeType ::= CREATE SP EDGE SP TYPE SP ( ABSTRACT SP )? edgeType`

referenced by:

- `createType`

createGraphType:

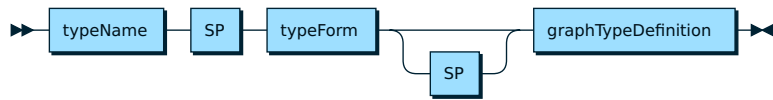


`createGraphType ::= CREATE SP GRAPH SP TYPE SP graphType`

referenced by:

- `createType`

graphType:

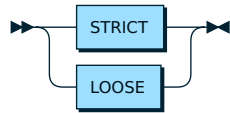


```
graphType
  ::= typeName SP typeForm SP? graphTypeDefinition
```

referenced by:

- [createGraphType](#)

#### typeForm:

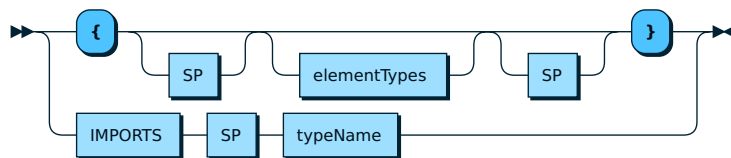


```
typeForm ::= STRICT
          | LOOSE
```

referenced by:

- [graphType](#)

#### graphTypeDefinition:

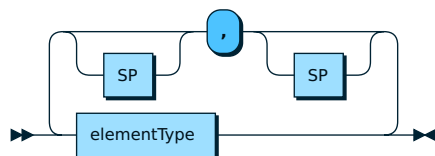


```
graphTypeDefinition
  ::= '{' SP? elementTypes? SP? '}'
  | IMPORTS SP typeName
```

referenced by:

- [graphType](#)

#### elementTypes:

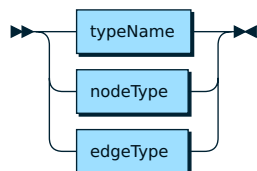


```
elementTypes
  ::= elementType ( SP? ',' SP? elementType )*
```

referenced by:

- [graphTypeDefinition](#)

#### elementType:

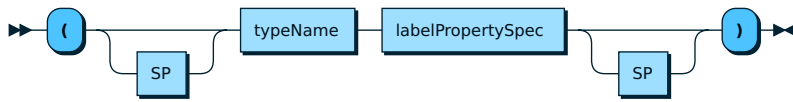


```
elementType
  ::= typeName
  | nodeType
  | edgeType
```

referenced by:

- [elementTypes](#)

#### nodeType:

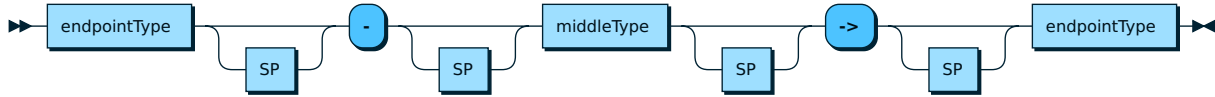


```
nodeType ::= '(' SP? typeName labelPropertySpec SP? ')'
```

referenced by:

- [createNodeType](#)
- [elementType](#)

**edgeType:**

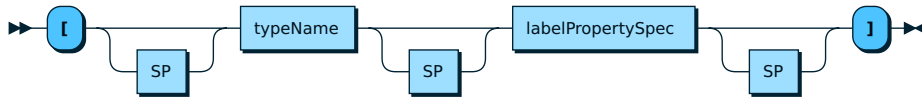


```
edgeType ::= endpointType SP? '-' SP? middleType SP? '->' SP? endpointType
```

referenced by:

- [createEdgeType](#)
- [elementType](#)

**middleType:**



```
middleType  
  ::= '[' SP? typeName SP? labelPropertySpec SP? ']'
```

referenced by:

- [edgeType](#)

**endpointType:**

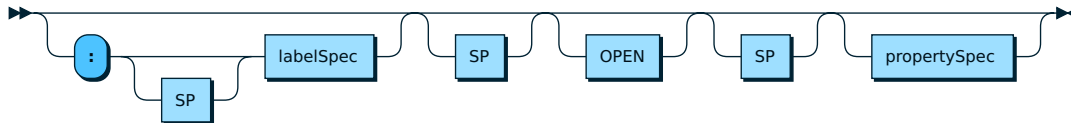


```
endpointType  
  ::= '(' SP? labelPropertySpec SP? ')'
```

referenced by:

- [edgeType](#)

**labelPropertySpec:**

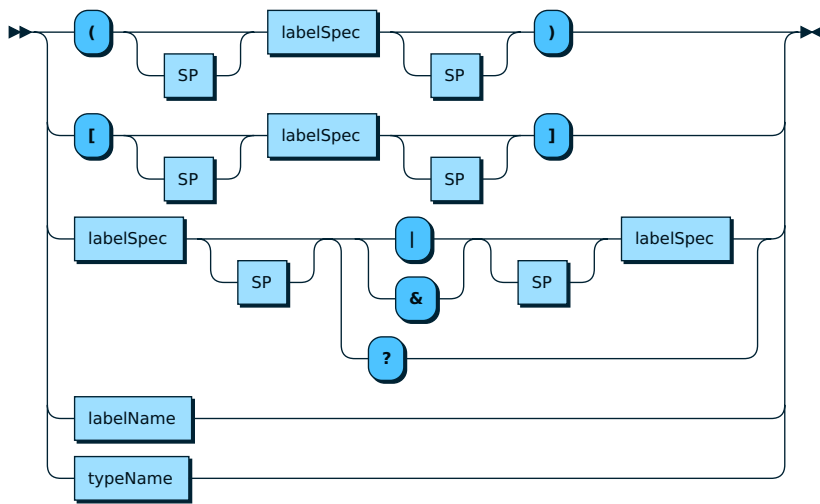


```
labelPropertySpec  
  ::= ( ':' SP? labelSpec )? SP? OPEN? SP? propertySpec?
```

referenced by:

- [elementType](#)
- [middleType](#)
- [nodeType](#)

**labelSpec:**



```

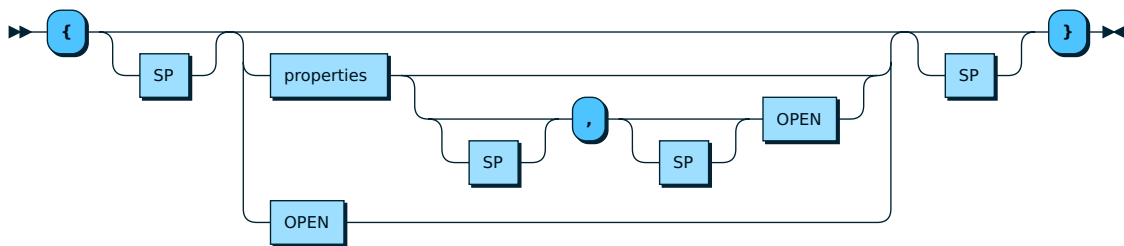
labelSpec
  ::= '(' SP? labelSpec SP? ')'
  | '[' SP? labelSpec SP? ']'
  | labelSpec SP? ( ( '|' | '&' ) SP? labelSpec | '?' )
  | labelName
  | typeName

```

referenced by:

- [labelPropertySpec](#)
- [labelSpec](#)

#### propertySpec:



```

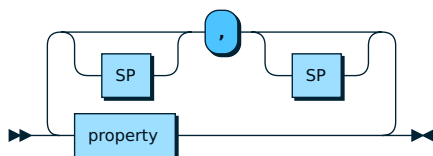
propertySpec
  ::= '{' SP? ( properties ( SP? ',' SP? OPEN )? | OPEN )? SP? '}'

```

referenced by:

- [labelPropertySpec](#)

#### properties:



```

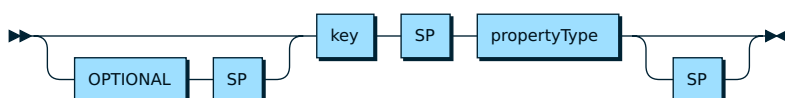
properties
  ::= property ( SP? ',' SP? property )*

```

referenced by:

- [propertySpec](#)

#### property:



```

property ::= ( OPTIONAL SP )? key SP propertyType SP?

```

referenced by:

- [properties](#)

propertyType:



```
propertyType ::= StringLiteral
```

referenced by:

- [property](#).

key:



```
key ::= StringLiteral
```

referenced by:

- [property](#).

labelName:



```
labelName ::= StringLiteral
```

referenced by:

- [labelSpec](#)

typeName:

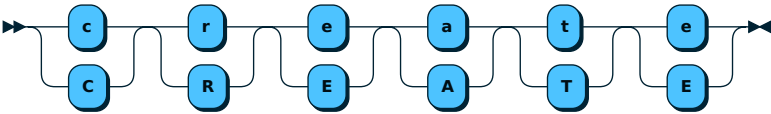


```
typeName ::= StringLiteral
```

referenced by:

- [elementType](#)
- [graphType](#)
- [graphTypeDefinition](#)
- [labelSpec](#)
- [middleType](#)
- [nodeType](#)

CREATE:

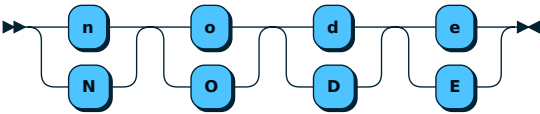


```
CREATE ::= ( 'c' | 'C' ) ( 'r' | 'R' ) ( 'e' | 'E' ) ( 'a' | 'A' ) ( 't' | 'T' ) ( 'e' | 'E' )
```

referenced by:

- [createEdgeType](#)
- [createGraphType](#)
- [createNodeType](#)

NODE:

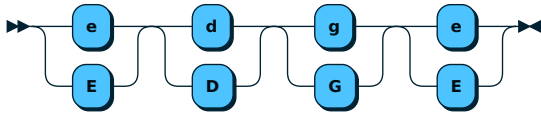


```
NODE ::= ( 'n' | 'N' ) ( 'o' | 'O' ) ( 'd' | 'D' ) ( 'e' | 'E' )
```

referenced by:

- [createNodeType](#)

EDGE:

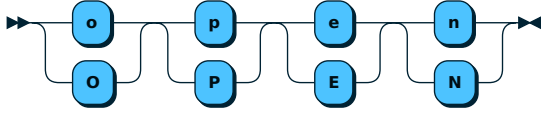


EDGE ::= ( 'e' | 'E' ) ( 'd' | 'D' ) ( 'g' | 'G' ) ( 'e' | 'E' )

referenced by:

- [createEdgeType](#)

#### OPEN:

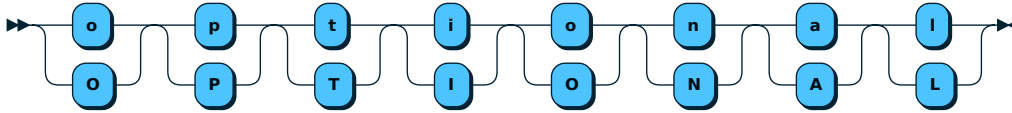


OPEN ::= ( 'o' | 'O' ) ( 'p' | 'P' ) ( 'e' | 'E' ) ( 'n' | 'N' )

referenced by:

- [labelPropertySpec](#)
- [propertySpec](#)

#### OPTIONAL:

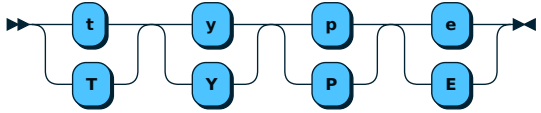


OPTIONAL ::= ( 'o' | 'O' ) ( 'p' | 'P' ) ( 't' | 'T' ) ( 'i' | 'I' ) ( 'o' | 'O' ) ( 'n' | 'N' ) ( 'a' | 'A' ) ( 'l' | 'L' )

referenced by:

- [property](#)

#### TYPE:

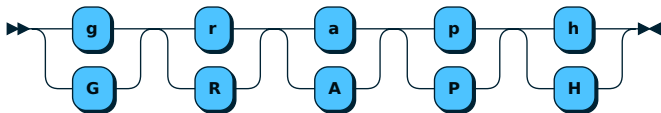


TYPE ::= ( 't' | 'T' ) ( 'y' | 'Y' ) ( 'p' | 'P' ) ( 'e' | 'E' )

referenced by:

- [createEdgeType](#)
- [createGraphType](#)
- [createNodeType](#)

#### GRAPH:

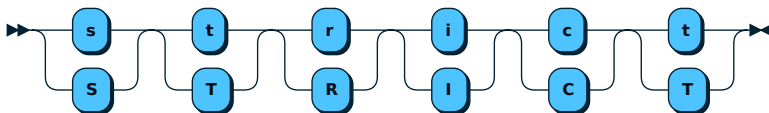


GRAPH ::= ( 'g' | 'G' ) ( 'r' | 'R' ) ( 'a' | 'A' ) ( 'p' | 'P' ) ( 'h' | 'H' )

referenced by:

- [createGraphType](#)

#### STRICT:

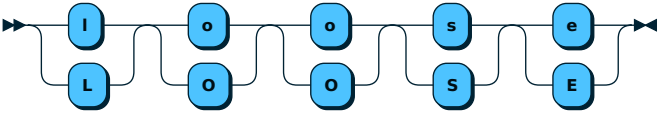


STRICT ::= ( 's' | 'S' ) ( 't' | 'T' ) ( 'r' | 'R' ) ( 'i' | 'I' ) ( 'c' | 'C' ) ( 't' | 'T' )

referenced by:

- [typeForm](#)

LOOSE:

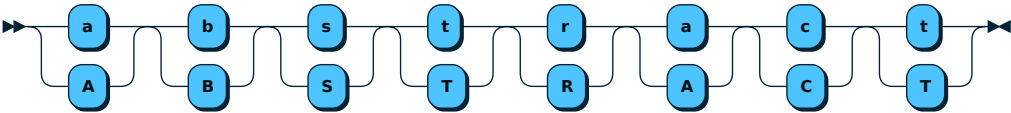


LOOSE ::= ( 'l' | 'L' ) ( 'o' | 'O' ) ( 'o' | 'O' ) ( 's' | 'S' ) ( 'e' | 'E' )

referenced by:

- [typeForm](#)

ABSTRACT:

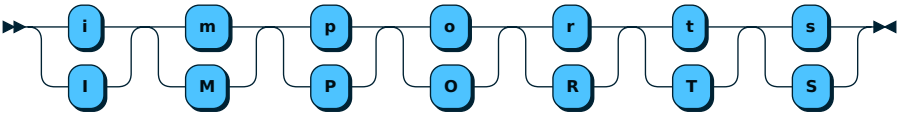


ABSTRACT ::= ( 'a' | 'A' ) ( 'b' | 'B' ) ( 's' | 'S' ) ( 't' | 'T' ) ( 'r' | 'R' ) ( 'a' | 'A' ) ( 'c' | 'C' ) ( 't' | 'T' )

referenced by:

- [createEdgeType](#)
- [createNodeType](#)

IMPORTS:

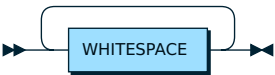


IMPORTS ::= ( 'i' | 'I' ) ( 'm' | 'M' ) ( 'p' | 'P' ) ( 'o' | 'O' ) ( 'r' | 'R' ) ( 't' | 'T' ) ( 's' | 'S' )

referenced by:

- [graphTypeDefinition](#)

SP:

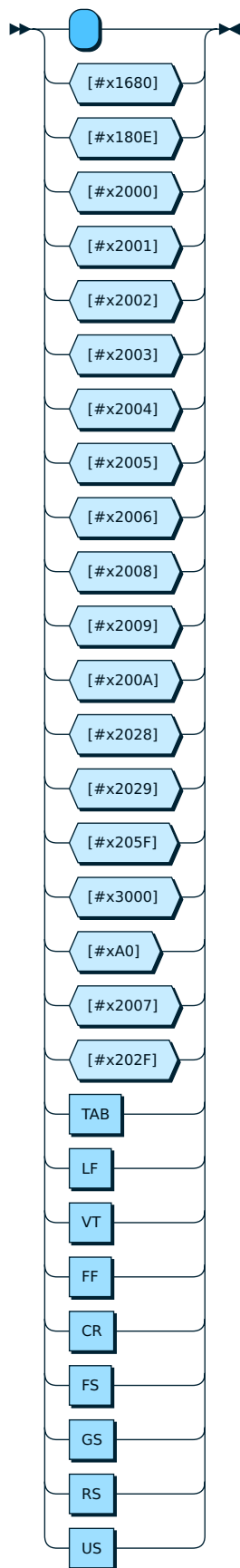


SP ::= WHITESPACE+

referenced by:

- [createEdgeType](#)
- [createGraphType](#)
- [createNodeType](#)
- [edgeType](#)
- [elementTypes](#)
- [endpointType](#)
- [graphType](#)
- [graphTypeDefinition](#)
- [labelPropertySpec](#)
- [labelSpec](#)
- [middleType](#)
- [nodeType](#)
- [pgs](#)
- [properties](#)
- [property](#)
- [propertySpec](#)

WHITESPACE:



```

WHITESPACE
::= [
    | TAB
    | LF
    | VT
    | FF
    | CR
    | FS
    | GS
    | RS
    | US
    | [#x1680#x180E#x2000#x2001#x2002#x2003#x2004#x2005#x2006#x2008#x2009#x200A#x2028#x2029#x205F#x3000#xA0#x2007#x202F]

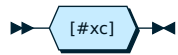
```

referenced by:



- [SP](#)

#### FF:



```
FF      ::= #xc
```

referenced by:

- [WHITESPACE](#)

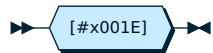
#### EscapedSymbolicName\_0:



```
EscapedSymbolicName_0  
      ::= [^`]
```

no references

#### RS:



```
RS      ::= #x001E
```

referenced by:

- [WHITESPACE](#)

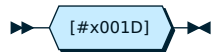
#### StringLiteral\_1:



```
StringLiteral_1  
      ::= [^'\\]
```

no references

#### GS:

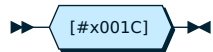


```
GS      ::= #x001D
```

referenced by:

- [WHITESPACE](#)

#### FS:

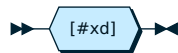


```
FS      ::= #x001C
```

referenced by:

- [WHITESPACE](#)

#### CR:

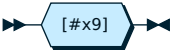


```
CR      ::= #xd
```

referenced by:

- [WHITESPACE](#)

TAB:

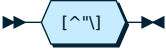


TAB ::= #x9

referenced by:

- [WHITESPACE](#)

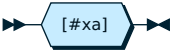
StringLiteral\_0:



StringLiteral\_0 ::= [^"]

no references

LF:

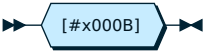


LF ::= #xa

referenced by:

- [WHITESPACE](#)

VT:

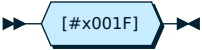


VT ::= #x000B

referenced by:

- [WHITESPACE](#)

US:

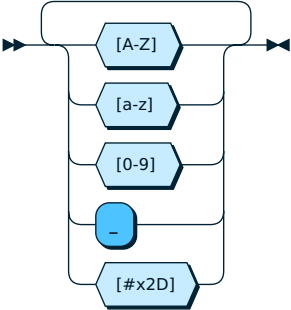


US ::= #x001F

referenced by:

- [WHITESPACE](#)

StringLiteral:

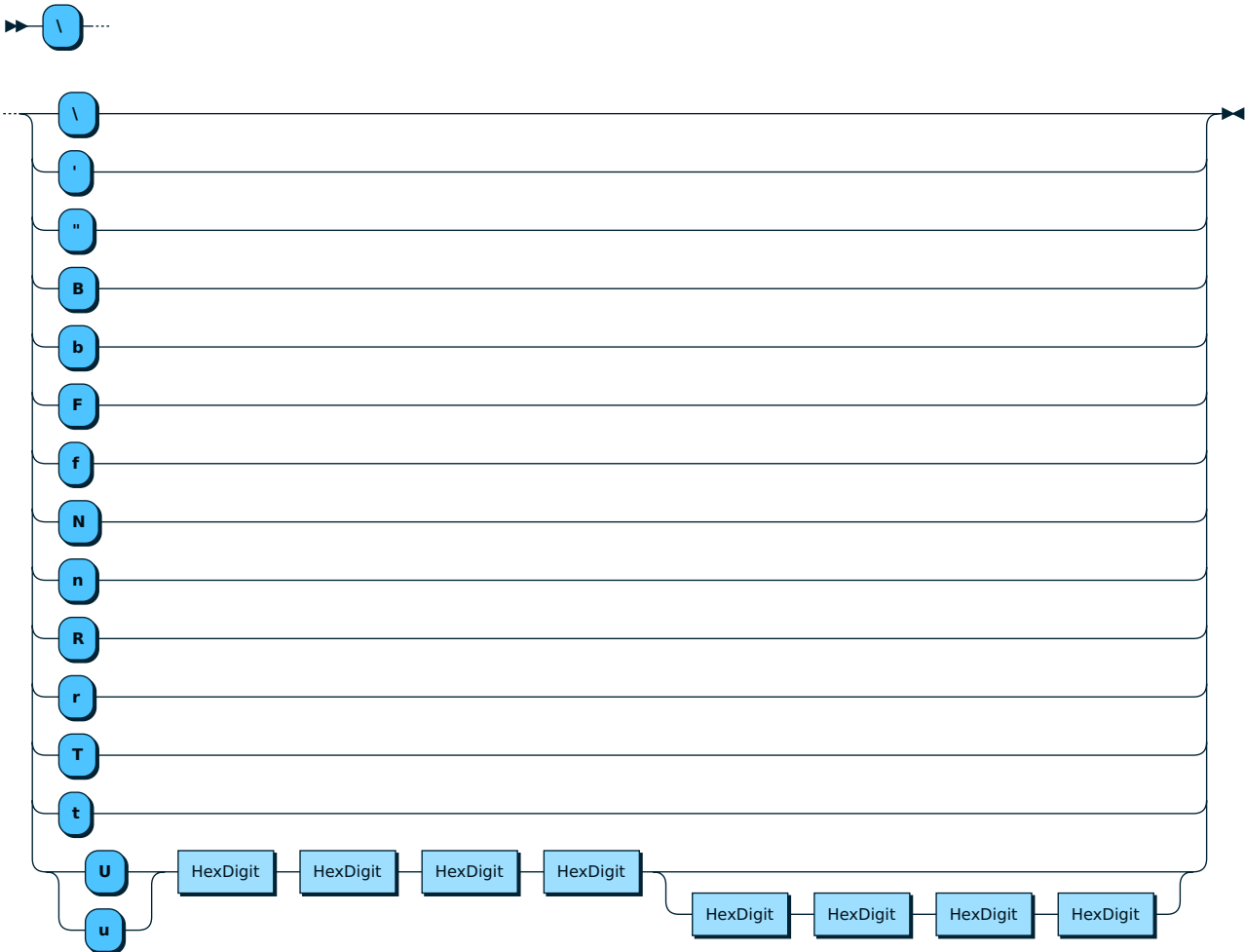


StringLiteral ::= [A-Za-z0-9\_#x2D]+

referenced by:

- [key](#)
- [labelName](#)
- [propertyType](#)
- [typeName](#)

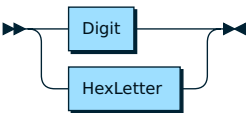
EscapedChar:



EscapedChar ::= '\ ( ' | '"' | "'" | 'B' | 'b' | 'F' | 'f' | 'N' | 'n' | 'R' | 'r' | 'T' | 't' | ( 'U' | 'u' ) HexDigit HexDigit HexDigit HexDigit ( HexDigit HexDigit HexDigit HexDigit )? )

no references

HexDigit:

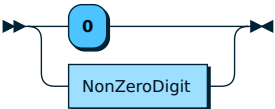


HexDigit ::= Digit | HexLetter

referenced by:

- EscapedChar

Digit:

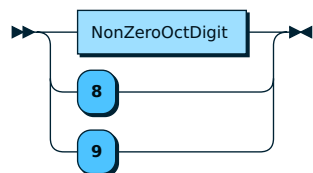


Digit ::= '0' | NonZeroDigit

referenced by:

- HexDigit

NonZeroDigit:

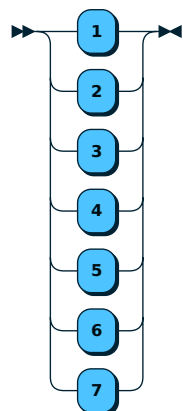


```
NonZeroDigit
  ::= NonZeroOctDigit
     | '8'
     | '9'
```

referenced by:

- [Digit](#)

#### NonZeroOctDigit:

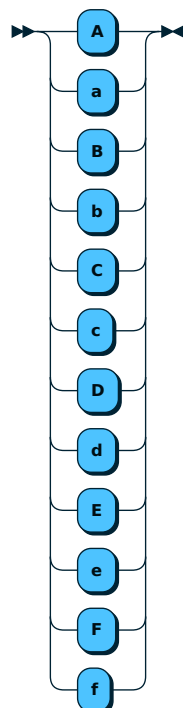


```
NonZeroOctDigit
  ::= '1'
     | '2'
     | '3'
     | '4'
     | '5'
     | '6'
     | '7'
```

referenced by:

- [NonZeroDigit](#)

#### HexLetter:



```
HexLetter
  ::= 'A'
     | 'a'
```

	'B'
	'b'
	'C'
	'c'
	'D'
	'd'
	'E'
	'e'
	'F'
	'f'

referenced by:

- [HexDigit](#)

**EOF:**



EOF	::=	\$
-----	-----	----

referenced by:

- [pgs](#)