*expression*

**true**

**false**

**nil**

**dot**

**unknown**

*string*

*number*

*identifier*

*funject*  
*array*

*invocation*

*strict-assignment*

*lazy-assignment*

*reset-strict-assignment*

*reset-lazy-assignment*

*strict-funject-assignment*

*lazy-funject-assignment*

*inheritance*

*inverse-definition*

*expression expression (at equal lines of indentation)*

*string*

**''**

**""**

**'***single-quoted-chars'*  
**"***double-quoted-chars***"**

*single-quoted-chars*

*single-quoted-char*

*single-quoted-char single-quoted-chars*

*double-quoted-chars*

*double-quoted-char*

*double-quoted-char double-quoted-chars*

*single-quoted-char*

*char*

**"**

*double-quote-char*

*char*

**'**

*char*

*any-character-except-***'***-***"***-or-***\**   
**\'**

**\"**  
**\\**  
**\n**  
**\r**  
**\t**

*number[[1]](#footnote-1)*

*int*  
*int frac*  
*int exp*  
*int frac exp*

*int*

*digit*  
*digit1-9 digits*   
**-** *digit*  
**-** *digit1-9 digits*

*frac*

**.** *digits*

*exp*

*e* *digits*

*digits*

*digit*  
*digit* *digits*

*e*

**e**  
**e+**  
**e-**

*funject*

**{}**  
**{** *members* **}**

*members*

*pair*  
*pair* *newline members*

*pair*

*pattern* **:** *lazy-expression*

*pattern*

*parameter*

*expression-with-a-parameter-substituted-for-any-of-its-components*

*parameter*

**@** *identifier*

*lazy-expression*

*expression*

*lazy-expression-lines(x > 0)*

*lazy-expression-lines(x)*

*lazy-expression-line(x)*

*lazy-expression-line(x) lazy-expression-lines(x)*

*lazy-expression-line*

*spaces(x) expression newline*

*array*

**[]**  
**[** *elements* **]**

*elements*

*expression*  
*expression***,** *elements*

*identifier*

*any-alphanumeric-underscore-plus-dash-or-equals-sign identifier*

*invocation*

*expression expression*

*strict-assignment*

*identifier* **=** *expression*

*lazy-assignment*

*identifier* **:=** *lazy-expressions*

*reset-strict-assignment*

*identifier* **|=** *expression*

*reset-lazy-assignment*

*identifier* **|:=** *expression*

*strict-funject-assignment*

*invocation* **=** *expression*

*lazy-funject-assignment*

*invocation* **:=** *lazy-expressions*

*inheritance*

*expression* **<<** *expression*

*inverse-definition*

*expression* **<-** *expression*

*module*

**module** *expression*

**module** *identifier expression*

**module** *identifier < identifier expression*

*class*

**class** *expression*

**class** *identifier expression*

**class** *identifier < identifier expression*

*conditional*

**if** *condition newline-character indented-expression*

**if** *condition newline-character indented-expression* **else** *expression*

**if** *condition* **then** *expression*

**if** *condition* **then** *expression* **else** *expression*

1. I adapted the definition of “number” from json.org [↑](#footnote-ref-1)