

# TELUGU GRAMMAR RULES DEFINITION

```
95 Grammar
96
97     0 $accept: program $end
98
99     1 program: input
100
101     2 eol: EOL
102
103     3 telugu_identifier: TELUGU_IDENTIFIER
104
105     4 telugu_function_name: TELUGU_IDENTIFIER
106
107     5 telugu_identifier_declaring: TELUGU_IDENTIFIER
108
109     6 telugu_imported_library: TELUGU_IDENTIFIER
110
111     7 telugu_print: TELUGU_PRINT
112
113     8 telugu_int: TELUGU_INT
114
115     9 telugu_input: TELUGU_INPUT
116
117    10 telugu_float: TELUGU_FLOAT
118
119    11 telugu_import: TELUGU_IMPORT
120
121    12 telugu_constant: TELUGU_INT
122    13                      | TELUGU_FLOAT
123    14                      | TELUGU_STRING
124    15                      | TELUGU_CHARACTER
125
126    16 telugu_arithmetic_operator: TELUGU_ARITHMETIC_OPERATOR
127
128    17 telugu_comparison_operator: TELUGU_COMPARISON_OPERATOR
129
130    18 telugu_assignment_operator: TELUGU_ASSIGNMENT_OPERATOR
131
132    19 telugu_logical_operator: TELUGU_LOGICAL_OPERATOR
133
134    20 telugu_datatype: TELUGU_DATATYPE
135
136    21 telugu_if: TELUGU_IF
137
```

```

138 22 telugu_elif: TELUGU_ELIF
139
140 23 telugu_else: TELUGU_ELSE
141
142 24 telugu_while: TELUGU_WHILE
143
144 25 telugu_string: TELUGU_STRING
145
146 26 telugu_open_curly_bracket: TELUGU_OPEN_CURLY_BRACKET
147
148 27 telugu_closed_curly_bracket: TELUGU_CLOSED_CURLY_BRACKET
149
150 28 telugu_open_square_bracket: TELUGU_OPEN_SQUARE_BRACKET
151
152 29 telugu_closed_square_bracket: TELUGU_CLOSED_SQUARE_BRACKET
153
154 30 telugu_open_floor_bracket: TELUGU_OPEN_FLOOR_BRACKET
155
156 31 telugu_closed_floor_bracket: TELUGU_CLOSED_FLOOR_BRACKET
157
158 32 telugu_punctuation_comma: TELUGU_PUNCTUATION_COMMA
159
160 33 telugu_finish: TELUGU_FINISH
161
162 34 telugu_function: TELUGU_FUNCTION
163
164 35 telugu_return: TELUGU_RETURN
165
166 36 telugu_character: TELUGU_CHARACTER
167
168 37 input: /* empty */
169 38     | input eol
170 39     | eol input
171 40     | input telugu_import telugu_imported_library telugu_finish
172 41     | telugu_import telugu_imported_library telugu_finish input
173 42     | input bunch_of_statements input
174 43     | input function_declaration input
175
176 44 exp: telugu_int
177 45     | telugu_float
178 46     | telugu_character
179 47     | telugu_string
180 48     | telugu_identifier
181 49     | function call

```

```

181 49 | function_call
182 50 | telugu_identifier telugu_open_square_bracket exp telugu_closed_square_bracket
183 51 | telugu_open_curly_bracket exp telugu_closed_curly_bracket
184 52 | exp telugu_arithmetic_operator exp
185 53 | exp telugu_logical_operator exp
186 54 | telugu_identifier telugu_open_square_bracket exp telugu_closed_square_bracket
187
188 55 bunch_of_statements: /* empty */
189 56 | eol bunch_of_statements
190 57 | bunch_of_statements eol
191 58 | bunch_of_statements if_else_ladder bunch_of_statements
192 59 | bunch_of_statements telugu_input telugu_open_curly_bracket telugu_identifier telugu_closed_curly_bracket telugu_finish bunch_of_statements
193 60 | bunch_of_statements while_loop bunch_of_statements
194 61 | bunch_of_statements print_statement telugu_finish bunch_of_statements
195 62 | bunch_of_statements variable_declaration telugu_finish bunch_of_statements
196 63 | bunch_of_statements telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_bracket bunch_of_statements
197 64 | bunch_of_statements function_call telugu_finish bunch_of_statements
198 65 | bunch_of_statements equation telugu_finish bunch_of_statements
199 66 | error telugu_finish
200
201 67 condition: exp
202 68 | exp telugu_comparison_operator exp
203 69 | exp telugu_logical_operator exp
204
205 70 if_statement: telugu_if telugu_open_curly_bracket condition telugu_closed_curly_bracket telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_bracket
206
207 71 elif_repeat: /* empty */
208 72 | eol elif_repeat
209 73 | elif_repeat telugu_elif telugu_open_curly_bracket condition telugu_closed_curly_bracket telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_bracket
210
211 74 else_statement: /* empty */
212 75 | eol else_statement
213 76 | telugu_else telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_bracket
214
215 77 if_else_ladder: if_statement elif_repeat else_statement
216 78 | if_statement elif_repeat
217
218 79 while_loop: telugu_while telugu_open_curly_bracket condition telugu_closed_curly_bracket telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_bracket
219
220 80 variable_declaration: telugu_datatype telugu_identifier_declaring telugu_assignment_operator exp
221 81 | telugu_datatype telugu_identifier_declaring
222 82 | telugu_datatype telugu_identifier_declaring telugu_open_square_bracket exp telugu_closed_square_bracket
223
224 83 parameters_repeat: /* empty */
225 84 | parameters_repeat telugu_datatype telugu_identifier_declaring telugu_punctuation_comma
226
227 85 parameters_line: /* empty */
228 86 | parameters_repeat telugu_datatype telugu_identifier_declaring
229
230 87 identifiers_repeat: /* empty */
231 88 | telugu_identifier
232 89 | telugu_constant
233 90 | telugu_identifier telugu_punctuation_comma identifiers_repeat
234 91 | telugu_constant telugu_punctuation_comma identifiers_repeat

```

```

234 91 | telugu_constant telugu_punctuation_comma identifiers_repeat
235
236 92 identifiers_line: identifiers_repeat
237
238 93 $@1: /* empty */
239
240 94 equation: telugu_identifier telugu_assignment_operator $@1 exp
241 95 | telugu_identifier telugu_open_square_bracket exp telugu_closed_square_bracket telugu_assignment_operator exp
242
243 96 function_content: function_content eol
244 97 | bunch_of_statements
245 98 | bunch_of_statements telugu_return telugu_finish bunch_of_statements
246 99 | bunch_of_statements telugu_return exp telugu_finish bunch_of_statements
247
248 100 print_content: /* empty */
249 101 | print_content eol
250 102 | eol print_content
251 103 | print_content telugu_string
252 104 | print_content exp
253 105 | print_content telugu_punctuation_comma telugu_string
254 106 | print_content telugu_punctuation_comma exp
255
256 107 print_statement: telugu_print telugu_open_curly_bracket print_content telugu_closed_curly_bracket
257
258 108 $@2: /* empty */
259
260 109 function_declaration: telugu_function telugu_function_name $@2 telugu_open_curly_bracket parameters_line
261 | telugu_closed_curly_bracket telugu_open_floor_bracket function_content telugu_closed_floor_bracket
262
263 110 $@3: /* empty */
264
265 111 function_call: telugu_identifier $@3 telugu_open_curly_bracket identifiers_line telugu_closed_curly_bracket
266

```

## Verbal Grammar Definition

### Variable Declaration:

sankhya a=3;

multiple variables declaration in the same line is not allowed like: sankhya a=3,b=4;

### Headers:

thechko library\_name;

multiple imports and renaming libraries like:

thechko lib1,lib2; // not allowed

thechko lib1 as my\_lib; // not allowed

### Function Declaration/Call:

```
function function_name(condition){  
    // bunch of statements  
    // function declarations not allowed in here  
    //can have return statement like 'return;' or 'return exp;'  
}
```

### Input/Output Commands:

```
chupi(var1_value);  
chupi("any_string");  
chupi("string1", var1_value, "string2", var2_value);  
theesko(var1);
```

### If-Else Ladder:

```
okavela(condition_1){ //flower braces are must  
    //bunch_of_statements  
}  
lekaokavela(condition_2){ // itself can repeat 0 or more number of times  
    //bunch_of_statements  
}  
lekapothe{ // need not be  
    //bunch_of_statements  
}
```

### Loop:

```
aithaunte(condition){ // only while loop allowed  
    //bunch_of_statements  
}
```

### Array Indexing:

```
sankhya arr[15];  
arr[3];  
arr[2+1*(x)+4]; // arr[any valid expression] is allowed
```

## PARSE TABLE:

(Only few lines of the parse table are shown below)

```

424  ✓ state 0
425
426      0 $accept: . program $end
427
428      EOL          shift, and go to state 1
429      TELUGU_IMPORT shift, and go to state 2
430
431      EOL          [reduce using rule 37 (input)]
432      TELUGU_IMPORT [reduce using rule 37 (input)]
433      $default      reduce using rule 37 (input)
434
435      program      go to state 3
436      eol          go to state 4
437      telugu_import go to state 5
438      input        go to state 6
439
440
441  ✓ state 1
442
443      2 eol: EOL .
444
445      $default      reduce using rule 2 (eol)
446
447
448  ✓ state 2
449
450  ✓ 11 telugu_import: TELUGU_IMPORT .
451
452      $default      reduce using rule 11 (telugu_import)
453
454
455  ✓ state 3
456
457      0 $accept: program . $end
458
459      $end          shift, and go to state 7
460
461
462  ✓ state 4
463
464  ✓ 39 input: eol . input
465
466      EOL          shift, and go to state 1
467      TELUGU_IMPORT shift, and go to state 2
468
469      EOL          [reduce using rule 37 (input)]
470      TELUGU_IMPORT [reduce using rule 37 (input)]
471      $default      reduce using rule 37 (input)
472
473      eol          go to state 4
474      telugu_import go to state 5
475      input        go to state 8
476

```

```

947  state 34
948
949      94 equation: telugu_identifier . telugu_assignment_operator $@1 exp
950      95      | telugu_identifier . telugu_open_square_bracket exp telugu_closed_square_bracket telugu_assignment_operator exp
951  111 function_call: telugu_identifier . $@3 telugu_open_curly_bracket identifiers_line telugu_closed_curly_bracket
952
953      TELUGU_ASSIGNMENT_OPERATOR  shift, and go to state 54
954      TELUGU_OPEN_SQUARE_BRACKET  shift, and go to state 55
955
956      $default  reduce using rule 110 ($@3)
957
958      telugu_assignment_operator  go to state 56
959      telugu_open_square_bracket  go to state 57
960      $@3                        go to state 58
961
962
963  state 35
964
965  107 print_statement: telugu_print . telugu_open_curly_bracket print_content telugu_closed_curly_bracket
966
967      TELUGU_OPEN_CURLY_BRACKET  shift, and go to state 59
968
969      telugu_open_curly_bracket  go to state 60
970
971

```

```

3768  state 197
3769
3770      73 elif_repeat: elif_repeat . telugu_elif telugu_open_curly_bracket condition telugu_closed_curly_bracket telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_bracket elif_repeat
3771      73      | elif_repeat telugu_elif telugu_open_curly_bracket condition telugu_closed_curly_bracket telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_bracket elif_repeat .
3772
3773      TELUGU_ELIF  shift, and go to state 101
3774
3775      TELUGU_ELIF [reduce using rule 73 (elif_repeat)]
3776      $default  reduce using rule 73 (elif_repeat)
3777
3778      telugu_elif  go to state 104
3779
3780
3781  state 198
3782
3783      57 bunch_of_statements: bunch_of_statements . eol
3784      58      | bunch_of_statements . if_else_ladder bunch_of_statements
3785      59      | bunch_of_statements . telugu_input telugu_open_curly_bracket telugu_identifer telugu_closed_curly_bracket telugu_finish bunch_of_statements
3786      60      | bunch_of_statements . while_loop bunch_of_statements
3787      61      | bunch_of_statements . print_statement telugu_finish bunch_of_statements
3788      62      | bunch_of_statements . variable_declaration telugu_finish bunch_of_statements
3789      63      | bunch_of_statements . telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_bracket bunch_of_statements
3790      64      | bunch_of_statements . function_call telugu_finish bunch_of_statements
3791      65      | bunch_of_statements . equation telugu_finish bunch_of_statements
3792      99 function_content: bunch_of_statements telugu_return exp telugu_finish bunch_of_statements .
3793
3794      EOL  shift, and go to state 1
3795      TELUGU_DATATYPE  shift, and go to state 26
3796      TELUGU_IF  shift, and go to state 27
3797      TELUGU_WHILE  shift, and go to state 28
3798      TELUGU_OPEN_FLOOR_BRACKET  shift, and go to state 29
3799      TELUGU_IDENTIFIER  shift, and go to state 30
3800      TELUGU_PRINT  shift, and go to state 31
3801      TELUGU_INPUT  shift, and go to state 32
3802
3803      EOL  [reduce using rule 99 (function_content)]
3804      $default  reduce using rule 99 (function_content)
3805
3806      eol  go to state 51
3807      telugu_identifer  go to state 34
3808      telugu_print  go to state 35
3809      telugu_input  go to state 36
3810      telugu_datatype  go to state 37
3811      telugu_if  go to state 38
3812      telugu_while  go to state 39
3813      telugu_open_floor_bracket  go to state 40
3814      if_statement  go to state 42
3815      if_else_ladder  go to state 43
3816      while_loop  go to state 44
3817      variable_declaration  go to state 45
3818      equation  go to state 46
3819      print_statement  go to state 47
3820      function_call  go to state 48
3821

```