

Table 1: Tokens in Go (As specified by the "Lexical Elements" section in Go spec)

Token	Token Name	Token Class	Informal Description (regex)	Formal Description	Example(s)
Identifier	tok_id	identifier	at least one char, alphanumeric with -	<code>[_a-zA-Z][_a-zA-Z0-9]*</code>	<code>i, myId, _abc, var1</code>
(tok_lparen	punctuation	-	<code>[()</code>	-
)	tok_rparen	punctuation	-	<code>[)]</code>	-
{	tok_lcurly	punctuation	-	<code>[{]</code>	-
}	tok_rcurly	punctuation	-	<code>[}]</code>	-
;	tok_semicolon	punctuation	-	<code>[;]</code>	-
Main	tok_main	reserved keyword	letters "m" "a" "i" "n"	<code>^main\$</code>	-
Package Declaration	tok_package	reserved keyword	letters "p" "a" "c" "k" "a" "g" "e"	<code>^package\$</code>	-
Function Declaration	tok_func	reserved keyword	letters "f" "u" "n" "c"	<code>^func\$</code>	-
EOF	tok_eof	reserved	-	<code>[\Z]</code>	-