|  |  |
| --- | --- |
| **Group Name: Team PLA** | **Section: T3L** |
| **Member 1: Antero, Elvis** | **Member 3: Ruz, Julianne** |
| **Member 2: Gutierrez, Reigner** | **Member 4:** |

**LOLCODE GRAMMAR**

Phrases enclosed by angle brackets (<,>) are abstractions. Words in small letters describe the lexemes that are already described by a regular expression (e.g. varident for variable identifiers, yarn for string literals, troof for boolean values, etc).

|  |  |  |
| --- | --- | --- |
| **LHS** | **::=** | **RHS** |
| <program> | ::= | HAI <linebreak> <statement> <linebreak> KTHXBYE |
| <statement> | ::= | <print> |
| <print> | ::= | VISIBLE varident | VISIBLE <expr> | VISIBLE <literal> |
| <literal> | ::= | numbr | numbar | yarn | troof |
| <expr> | ::= | <equals> | <both> | <not\_equals> | <greater> | <less> | <add> | <sub> | <mul> | <div> | <mod> | <cast> | <either> | <all> | <any> | <not> | <func> | <label> | <atom> |
| <comment> | ::= | BTW <statement>  | OBTW  <statement>  TLDR |
| <function declaration> | ::= | HOW DUZ I <funcname> [YR <arg1> [AN YR <arg2> …]]  <statement>  IF U SAY SO |
| <conditional> | ::= | <expr>  O RLY?  YA RLY  <statement>  NO WAI  <statement>  | <expr>  WTF?  OMG <value literal>  <statement>  [OMG <value literal>  <statement> …]  [OMGWTF  <statement>]  | <expr>  O RLY?  YA RLY  <statement>  [MEBBE <expr>  <statement>  [MEBBE <expr>  <statement>  …]]  [NO WAI  <statement>]  OIC |
| <assignment statement> | ::= | <var> <assignment operator> <expr> |
| <get> | ::= | GIMMEH <var> |
| <casting> | ::= | MAEK <expr> [A] <literal> |
| <comparison> | ::= | BOTH SAEM | DIFFRINT (numbr | number) [AN] (numbr | numbar) |
| <loop> | ::= | IM IN YR <label> WILE <expr> <linebreak> <statement> <linebreak> IM OUTTA YR <label> |
| <declaration> | ::= | I HAS A <label> | I HAS A <label> ITZ <value> |
| <switch statement> | ::= | WTF? [  OMG <value>  <statement>  ]  [  OMGWTF  <conditional>  ]  OIC |
| <equals> | ::= | BOTH SAEM <expr> AN <expr> |
| <not\_equals> | ::= | DIFFRINT <expr> AN <expr> |
| <both> | ::= | BOTH OF <expr> AN <expr> |
| <either> | ::= | EITHER OF <expr> AN <expr> |
| <greater> | ::= | BIGGR OF <expr> AN <expr> |
| <less> | ::= | SMALLR OF <expr> AN <expr> |
| <add> | ::= | SUM OF <expr> AN <expr> |
| <sub> | ::= | DIFF OF <expr> AN <expr> |
| <mul> | ::= | PRODUKT OF <expr> AN <expr> |
| <div> | ::= | QUOSHUNT OF <expr> AN <expr> |
| <mod> | ::= | MOD OF <expr> AN <expr> |
| <cast> | ::= | MAEK <expr> A <expr> |
| <all> | ::= | ALL OF <expr> [AN <expr> … ] MKAY? |
| <any> | ::= | ANY OF <expr> [AN <expr> … ] MKAY? |
| <not> | ::= | NOT <expr> |
| <func> | ::= | <label> <expr> [<expr>] MKAY? |
| <label> | ::= | (A-Za-z0-9)\* |
| <atom> | ::= | WIN | FAIL | NOOB | (0-9)\* | (0-9)\*.(0-9)\* | “<string>” |
| <string> | ::= | Any character string |
| <linebreak> | ::= | (newline character) | , |