

<b>Group Name:</b>	<b>Section:</b>
<b>Member 1:</b>	<b>Member 3:</b>
<b>Member 2:</b>	<b>Member 4:</b>

## 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
<varcall>	::=	I HAS A <variable>   I HAS A <variable> ITZ <literal>   I HAS A <variable> ITZ <expression>
<comments>	::=	<statement> BTW <comment>   OBTW <linebreak> <comments> <linebreak> TLDR
<arithmeticop>	::=	SUM OF <arithmeticop> <litvar> AN <litvar>   DIFF OF <arithmeticop> <litvar> AN <litvar>   PRODUKT OF <arithmeticop> <litvar> AN <litvar>   QUOSHUNT OF <arithmeticop> <litvar> AN <litvar>   MOD OF <arithmeticop> <litvar> AN <litvar>   BIGGR OF <arithmeticop> <litvar> AN <litvar>   SMALLR OF <arithmeticop> <litvar> AN <litvar>   SUM OF <litvar> AN <litvar>   DIFF OF <litvar> AN <litvar>   PRODUKT OF <litvar> AN <litvar>   QUOSHUNT OF <litvar> AN <litvar>   MOD OF <litvar> AN <litvar>   BIGGR OF <litvar> AN <litvar>   SMALLR OF <litvar> AN <litvar>
<booleanop>	::=	BOTH OF <litvar> AN <litvar>   EITHER OF <litvar> AN <litvar>   WON OF <litvar> AN <litvar>   NOT <litvar>   ALL OF <boolean> AN <boolean> ...MKAY   ALL OF <litvar> AN <litvar> ...MKAY   ANY OF <booleanclause> ...MKAY   ANY OF <litvar> AN <litvar> ...MKAY
<comparisonop>	::=	BOTH SAEM <litvar> AN <litvar>   DIFFRINT <litvar> AN <litvar>   BOTH SAEM <litvar> AN BIGGR OF <litvar> AN <litvar>   BOTH SAEM <litvar> AN SMALLR OF <litvar> AN <litvar>   DIFFRINT <litvar> AN BIGGR OF <litvar> AN <litvar>   DIFFRINT <litvar> AN SMALLR OF <litvar> AN <litvar>
<concatop>	::=	SMOOSH <anclause>
<input>	::=	GIMMEH <variable>
<expression>	::=	<comparison>   <arithmetic>   <boolean>   <concat>   <assignment>
<assignment>	::=	<variable> R <literal>   <variable> R <variable>   <variable> R <expression>
<ifthen>	::=	<expression><linebreak>O RLY?<linebreak> YA RLY<linebreak><statement><linebreak> NO WAI<linebreak><statement><linebreak> OIC
<switchcase>	::=	WTF? <linebreak><omgclause> OIC  WTF? <linebreak><omgclause> OMGWTF <linebreak><statement> OIC

<anclause>	::=	<literal> AN <literal>   <anclause><anclause>
<booleanclause>	::=	<boolean> AN <boolean>   <boolean><boolean> AN <boolean><boolean>
<loops>	::=	IM IN YR <label> <loopop> YR <variable> <testexpression><linebreak><statement><linebreak> IM OUTTA YR <label>
<omgclause>	::=	OMG <literal><linebreak><statement>   <omgclause><omgclause>
<testexpression>	::=	TIL <expression>   WILE <expression>
<loopop>	::=	UPPIN   NERFIN
<litvar>	::=	<literal>   <variable>
<troof>	::=	WIN   FAIL