# Complete BNF Description of XXXXX Language

<program> ::= **start**<stmts>**end**

<stmts> ::= <stmt> | <stmt><stmts>

<stmt> ::= <if\_stmt> | <non-if\_stmt>

<if\_stmt> ::= <matched\_stmt> | <unmatched\_stmt>

<non\_if\_stmt> :: = <assign\_stmt><end\_stmt> | <input\_stmt><end\_stmt> | <output\_stmt><end\_stmt> | <func\_define> |<func\_call><end\_stmt> | <read\_from\_sensor><end\_stmt> | <time\_from\_timer><end\_stmt> | <connect\_stmt><end\_stmt> | <declare\_stmt><end\_stmt> | <break\_stmt><end\_stmt> | <continue\_stmt><end\_stmt> | <arithmetic\_op><end\_stmt> | <comment> | <end\_stmt> | <while\_stmt> | <for\_stmt>

## Function Definition and Function Call

<func\_define> ::= **func**<space><func\_var><LP><parameters><RP><LB><func\_body><RB>

<parameters> ::= <parameter> | <parameter><comma><parameters>

<parameter> ::= **int**<space><int\_var> | **float**<space><float\_var> | **string**<space><string\_var> | **bool**<space><bool\_var> | **char**<space><char\_var>

<func\_body> ::= <return\_stmt> | <stmts><return\_stmt>

<func\_call> ::= <func\_var><LP><variables><RP>

<variables> ::= <var> | <var><comma><variables>

## Continue-Break, Return, and End Statements

<continue\_stmt> ::= **continue**

<break\_stmt> ::= **break**

<return\_stmt> ::= **return**<space><values><end\_stmt>

<values> ::= <bool> | <string> | <char> | <int> | <float>

<end\_stmt> ::= <semicolon>

## Comment

<comment> ::= <hashtag><sentence><newline> |

<hashtag><asterisk><sentence><asterisk><hashtag>

## Arithmetic Operations

<arithmetic\_op>::= <numeric\_value> | <arithmetic\_op><plus\_op><numeric\_value> | <arithmetic\_op><minus\_op><numeric\_value>

<numeric\_value>::= <factor> | <numeric\_value><mult\_op> <factor> | <numeric\_value><divide\_op><factor>

<factor> ::= <arithmetic\_op> | <arithmetic\_op><expo\_op><factor>

<arithmetic\_op> ::= <int> | (<arithmetic\_op>)

## Data Types

<int> ::= <digits> | <minus\_op><digits>

<digits> ::= <digit> | <digit><digits>

<digit> ::= 0|1|2|3|4|5|6|7|8|9

<float> ::= <digits><dot><digits> | <minus\_op><digits><dot><digits> | <dot><digits> | <minus\_op><dot><digits>

<letter>::= <uppercase\_letter> | <lowercase\_letter>

<uppercase\_letter>::= 'A'|'B'|'C'|'D'|'E'|'F'|'G'|'H'|'I'|'J'|'K'|'L'|'M'|'N'|'O'|'P'|'Q'|'R'|'S'|'T'| 'U'|'V'|'W'|'X'|'Y'|'Z'

<lowercase\_letter>::= 'a'|'b|'c'|'d'|'e'|'f'|'g'|'h'|'i'|'j'| 'k'|'l'|'m'|'n'|'o'|'p'|'q'|'r'|'s'|'t'|'u'|'v'|'w'|'x'|'y'|'z'

## Input Output

<input\_stmt> ::= **input**<LP><inbody><RP>

<input\_stmt> ::= **input\_from\_connection**<LP><URL><comma><inbody><RP>

<output\_stmt> ::= **print**<LP><outbody><RP>

<output\_stmt> ::= **output\_to\_connection**<LP><URL><comma><outbody><RP>

<read\_from\_sensor> ::= **read\_from\_sensor**<LP><sensor\_name><comma><outbody><RP>

<time\_from\_timer> ::= **time\_from\_timer**<LP><timestamp><RP>

<inbody> ::= <bool\_exp> | <char> | <int> | <float> | <string>

<outbody> ::= <values><plus\_op><outbody> | <values> | <var><plus\_op><outbody> | <var>

<timestamp> ::= <int>

<URL> ::= <string\_id><string><string\_id>

## Connection to the Given URL

<connect\_stmt> ::= **connect**<LP><URL><RP>

## Conditional Statements: If-Else

<matched\_stmt> ::= **if**<LP><logic\_expr><RP><matched\_stmt>**else**<matched\_stmt> |

<non-if\_stmt>

<unmatched\_stmt> ::= **if**<LP><logic\_expr><RP><stmt> |

**if**<LP><logic\_expr><RP> <matched\_stmt> **else** <unmatched\_stmt>

## Loops: While and For

<while\_stmt>::= **while** <LP> <logic\_expr> <RP> <LB><stmts><RB>

<for\_stmt>::= **for** <LP> <declare\_stmt><end\_stmt> <bool\_expr> <end\_stmt><arithmetic\_operations> <LB><stmts><RB>

## Bool Expressions

<bool\_expr>::= <bool\_value> | <logic\_operations>| <comparison\_operations>

<logic\_operations>::= <bool\_expr><logic\_operands><bool\_expr>

<logic\_operands>::= <and\_op> | <or\_op>

<comparison\_operations>::= <int><comparison\_operands><int> | <float><comparison\_operands><float>

<comparison\_operands>::= <less\_op> | <greater\_op> | <equivalence\_op> | <greatEq\_op> | <lessEq\_op>

## Declaration Statements

<declare\_stmt>::= <int\_declare>

| <str\_declare>

| <char\_declare>

| <float\_declare>

| <bool\_declare>

<int\_declare>::= **int**<space><int\_var> | **int**<space><int\_assign>

<float\_declare>::= **float**<space><float\_var> | **float**<space><float\_assign>

<str\_declare>::= **string**<space><str\_var> | **string**<space><str\_assign>

<char\_declare>::= **char**<space><char\_var> | **char**<space><char\_assign>

<bool\_declare>::= **bool**<space><bool\_var> | **bool**<space><logic\_assign>

## Assignment Statements

<assign\_stmt>::= <int\_assign>

| <float\_assign>

| <str\_assign>

| <char\_assign>

| <bool\_assign>

<int\_assign>::= <int\_var>=<num\_var> | <int\_var>=<num\_value>

<float\_assign>::= <float\_var>=<num\_var> | <float\_var>=<num\_value>

<str\_assign>::= <str\_var>=<string\_with\_id>

<char\_assign>::= <char\_var>=<char\_with\_id>

<bool\_assign>::= <bool\_var>=<bool\_value> | <bool\_var>=<bool\_expr>

<num\_var>::= <int\_var> | <float\_var>

<num\_value>::= <int> | <float> | <arithmetic\_operation> | <char>

<bool\_value>::= **true** | **false** | 1 | 0 | **on** | **off**

<var>::= <string\_var>

| <char\_var>

| <int\_var>

| <float\_var>

| <bool\_var>

| <func\_var>

<string\_with\_id>::= <string\_id><string><string\_id>

<char\_with\_id>::= <char\_id><char><char\_id>

<string>::= <char\_list>

<char\_list>::= <char> | <char><char\_list>

<char>::= <letter> | <digit>

<char\_var>::= <letter> | <letter><string>

<string\_var>::= <letter> | <letter><string>

<int\_var>::= <letter> | <letter><string>

<float\_var>::= <letter> | <letter><string>

<bool\_var>::= <letter> | <letter><string>

<func\_var>::= <letter> | <letter><string>

# Reserved Words

**start** starts the program

**end** finishes the program

**if**

**if/else**

**while**

**for**

**break**

**return**

**continue**

**print**

**input**

**input\_from\_connection**

**output\_from\_connection**

**int**

**bool**

**float**

**string**

**char**

**func**

**true**

**false**

**on**

**off**