# Language Builder

We’re going to use the following website to help us build up a whole language defined in BNF.

<https://bnfplayground.pauliankline.com>

Using it we’re going to try and define a simple programming language.

Copy the following into the editor:

<letter>

<expression> ::= <factor> | <factor> "\*" <factor> | <factor> "/" <factor>

<factor> ::= <term> | <term> "+" <term> | <term> "-" <term>

<term> ::= "-" <number> | <number>

<number> ::= <digit> | <digit> <number>

<digit> ::= [0-9]

<letter> ::= [a-z] | [A-Z]

1. Now you should be able to select to generate <expression> and then generate a random expression. You can also type one in that you think should be correct to get it checked.
2. Identifier. An identifier is a like a variable name in program or a function name. It’s made up of a lot of letters. How can we define this using <letter> HINT: Check <number> and <digit> above.

Try generating some random identifiers to test them.

1. Assignment. Now we want be able to assign a value to an identifier e.g. i = 50. Write a rule for this called <assignment>. The left hand side can be any identifier and the right hand side can be any expression.
2. Modify Term. We also want to be able to use variables like numbers in our program. e.g. i = i + 1

Modify the <term> rule to allow this behaviour.

1. Program. Now we want to define a program as a list of assignments separated by “;” e.g. x = 5 ; x = 10 ; z = x + 5 ;

Created a rule to support this behaviour. HINT: We’ve done this before with identifier and numbers.

EXTENSION

If you’ve managed all that congratulations!!! Here’s a few ideas for what you can add to your language.

1. Boolean Support. Add support for Booleans similar to python e.g. the expressions: true, false, true and false, false or false etc.
2. If Statements. Can you add another kind of instruction e.g.

x = 5 ; if x == 5 then x = 10 else x = 20 ;