#### COMPILERS Language grammar – Raccoon Group 2: Paul Catusanu, Tom Ewbank, Élodie Van de Goor

We would like to have a language easy to understand and used as first language. To do that, we will use the python philosophy of “Simple is better than complex” and “Beautiful is better than ugly”. We will inspire ourselves of the python syntax for much of our language, particularly the indentation to structure the code instead of the punctuation in order to improve readability.

We will provide the following basic features:

* Integer, boolean, floating-point and array;
* Basic arithmetic and comparison operations;
* definition of functions and function calls;
* Literal constants (integer and floating-point);
* Mutable variables (definition/access/modification);
* Conditionals and loops;

Since our goal is to make a language easy to use for a child who has never programmed before, the *Raccoon* will have two key particularities:

* Implicit typing
* Use of meaningful keyword instead of some usual operator.

Here are some statements examples that should cover all the features of the language:

|  |  |  |
| --- | --- | --- |
| **C language** | **Raccoon language** | **Comments** |
| int a = 5; | a become 5 | There is no declaration, only assignment |
| const float pi = 3.14; | pi is 3.14 |  |
| bool b = (a == 3); | b become a =? 3 |  |
| a || b | a or b |  |
| a && b | a and b |  |
| int fct(int a, float b) {… return c;} | function fct(a,b):  …  return c | There is no function declaration, only definition. |
| int list[3] = {1 , 2, 3}; | list become [1,2,3] |  |
| for(i = 0; i < 4;++i) {…} | for i in range(0,3):  … |  |
| for(i = 0; i < list.size;++i)  {  element = liste[i];  ….  } | for element in list:  … |  |
| list[1] = 154;  a = list[1]; | list[2] become 154  a become list[2] | The index starts at 1 instead of 0 |
| While (n<19) {…} | While n<19:  … |  |
| //this is a comment | //this is a comment |  |
| a = 10%3; | a become 10 mod 3 |  |
| if(a>0)  printf(“a est positif.”);  elseif(a<0)  printf(“aest négatif.”);  else  printf(“a est nul.”); | if a > 0:  display("a est positif.")  else if a < 0:  display("a est négatif.")  else:  display("a est nul.") | The keyword “display” is not part of the grammar, it’s a function. |