Subject: Grammar Parts
Posted by ckgalloway on Fri, 05 Feb 2016 15:35:43 GMT

Should comments, arrays, and printing have grammar rules?

Also, should function calls be included in the overarching program rule?

Lastly, how would a whileLoop or ifStatement rule look?
Would it be whileLoop : KEYWORD OPAREN expr CPAREN block
Or whileLoop : WHILE OPAREN expr CPAREN block

And should the expr part be replaced by a specific type of expr that evaluates to a boolean or is that a job for the parser?

Subject: Re: Grammar Parts
Posted by tscrompton on Fri, 05 Feb 2016 16:07:05 GMT

I can't answer for the first two questions, but in regard to the while loop question, I think the latter one would be correct similar to reason that you use OPAREN instead of PAREN. In regard to your last question, first, how you want to implement the truthiness of a non-Boolean value (if at all) is up to you. Second, I think that since the language has to be dynamically typed, it's not possible to specify that the expression evaluates to a Boolean value, so I think that that's handled during semantic analysis. If I'm wrong on either those, somebody correct me.

Subject: Re: Grammar Parts
Posted by tscrompton on Fri, 05 Feb 2016 16:12:35 GMT

In regard to your first question, arrays should probably have a grammar rule depending on your grammar. Dr. Lusth said in class yesterday that comments should be consumed in skipWhiteSpace(), but I suppose you could do it some other way if you so desire. And printing can have a grammar rule, but it depends on how you implement it. For example, Python 2 has a grammar rule for the print statement, but Python 3 uses a print function which would be encompassed in the functional call rule.

Subject: Re: Grammar Parts
Posted by ckgalloway on Fri, 05 Feb 2016 16:37:20 GMT

My thoughts on the arrays are that they should be identifiers just like any other variable. But have special semantics for using A[i] and also for being initialized as in A = {1, 2, 3}. Any comments?
A[i] should be parsed as a primary, as in ID OBRACKET expr CBRACKET.

The on-the-fly construction of an array, {1, 2, 3} would also be a primary, as in OBRACE exprList CBRACE.

should we need a 'new' keyword for memory allocation?

jmbeach1 wrote on Tue, 16 February 2016 10:51 should we need a 'new' keyword for memory allocation?

No, you can allocate an array with a built-in function, as in:

a = makeArray(1000); # allocate an array with 1000 slots

If you want to add objects to your language, I will show you (later) how to implement Scam-style objects (really simple).

Do we need to be able to recognize hexadecimal, binary integers etc or decimal integers only.

Decimal is sufficient.