# PSA grammar

<expr> -> ( <expr> )
<expr> -> <expr> + <expr>
<expr> -> <expr> _ <expr>
<expr> -> <expr> * <expr>
<expr> -> <expr> / <expr>
<expr> -> <expr> \ <expr>
<expr> -> <expr> = <expr>
<expr> -> <expr> <> <expr>
<expr> -> <expr> < <expr>
<expr> -> <expr> <= <expr>
<expr> -> <expr> > <expr>
<expr> -> <expr> >= <expr>
<expr> -> id <func_call>
<expr> -> id
<expr> -> int_literal
<expr> -> double_literal
<expr> -> string_literal
<expr> -> <built_in_func_call>
<func_call> -> ( <arg_list> )
<arg_list> -> <arg> <more_args_list>
<arg_list> -> ε
<more_args_list> -> , <arg> <more_args_list>
<more_args_list> -> ε
<arg> -> <expr>
<built_in_func_call> -> length <func_call>
<built_in_func_call> -> substr <func_call>
<built_in_func_call> -> asc <func_call>
<built_in_func_call> -> chr <func_call>