

# Грамматика языка Пфру

## Алфавит

```
<letter> := "A"..."Z" | "a"..."z" | "_"
<ruletter> := "A"..."Я" | "а"..."я" | "_"
<digit> := "0"..."9"
<any> := ?любойсимвол?
<identifier> := <letter>{<letter> | <digit>}
```

## Литералы

```
<integer_literal> := <digit>{<digit>}
<float_literal> := <digit>{<digit>} "." <digit>{<digit>}
<char_literal> := '"' [<letter>] "'"
<string_literal> := """ {<letter>} """
<bool_literal> := "true" | "false"
```

## Пробелы и переносы строк

```
<space> := " " | \t
<newline> := \n | \r | \r\n
```

## Типы

```
<primitive_type> := "i8" | "i16" | "i32" | "i64" |
                    "f32" | "f64" | "char" | "stringa" | "bool"
<array_type> := <primitive_type> "[" <integer_literal> "]"
<type> := <primitive_type> | <array_type>
```

## БЛОКИ

```
<block> := "{" {<statement>} "}"
```

# Операторы

```
<statement> :=
    <var_decl> ;"
| <assignment> ;"
| <expr> ;"
| <if_stmt>
| <while_stmt>
| <do_while_stmt>
| <for_stmt>
| <return_stmt> ;"

<var_decl> := <identifier> [":" <type>] [= " <expr>"]
```

## Ветвление и циклы

```
<if_stmt> := "if" <expr> <block>
            {"elif" <expr> <block>}

<while_stmt> := "while" <expr> <block>
<do_while_stmt> := "do" <block> "while" <expr> ;"

<for_stmt> := "for" <identifier> "in" <range> <block>

<range> := "[" <expr> ";" [<expr>] ";" <expr> "]"

<return_stmt> := "return" <expr>
```

## Выражения

```
<expr> := <comma_expr>
<assignment> := <identifier> "=" <comma_expr>
<comma_expr> := <logic_or>{", " <logic_or>}
<logic_or> := <logic_and>{"||" <logic_and>}
<logic_and> := <bit_or>{"&&" <bit_or>}
<bit_or> := <bit_xor>{"|" <bit_xor>}
<bit_xor> := <bit_and>{"^^" <bit_and>}
<bit_and> := <equality>{"&" <equality>}
<equality> := <rel>{"==" | "!=") <rel>}
<rel> := <shift>{("<" | ">" | "<=" | ">=") <shift>}
<shift> := <add>{"<<" | ">>") <add>}
```

```
<add> := <mul>{ ("+" | "-") <mul> }
<mul> := <unary>{ ("*" | "/" | "%") <unary> }
<unary> := [ ("+" | "-" | "!=") ] <primary>
```

## Первичные выражения

```
<primary> :=
    <literal>
  | <identifier>
  | <call_expr>
  | <array_literal>
  | "(" <expr> ")"

<call_expr> := <identifier> "(" [<arg_list>] ")"
<arg_list> := <expr>{ "," <expr> }

<array_literal> := "{" <expr>{ "," <expr>} "}"

<literal> := <integer_literal> | <float_literal> |
              <string_literal> | <char_literal> | <bool_literal>
```

## Программа

```
<program> := {<toplevel_decl>}
<toplevel_decl> := <repr_func> | <arrow_block>
```

## Функции repr(...)

```
<repr_func> := "repr" <identifier> "(" [<param_list>] ")"
               [ "-"> <return_type_list>] <block>

<param_list> := <param>{ "," <param> }
<param> := <identifier> ":" <type>

<return_type_list> := <type>{ "," <type> }
```

## Блок стрелок #strelki

```
<arrow_block> := "#" [<identifier>] "{" <arrow_line> "}"
<arrow_line> := <arrow_node> <arrow_op> <arrow_node> ";"
<arrow_node> := "start" | "end" | <identifier>
<arrow_op> := "->" | "-(" <literal_list> ")"
<literal_list> := <literal>", " <literal>
```