

Gramática Pyscal

[PROGRAMA]	[DECLARACOES] [BLOCO]
[DECLARACOES]	[DEF_CONST] [DEF_VAR] [DEF_FUNC]
[DEF_CONST]	(const) [CONSTANTE] (;) [LIST_CONST] ε
[LIST_CONST]	[CONSTANTE] (;) [LIST_CONST] ε
[CONSTANTE]	[ID] (=) [CONST_VALOR]
[CONST_VALOR]	Seqüência alfanumérica iniciada por aspas e terminada em aspas (tratado no léxico) [EXP_MAT]
[TIPO_DADO]	(integer) (real) (string) (array) (l) [NUMERO] (l) (of) [TIPO_DADO] (record) [CAMPOS] (end)
[CAMPOS]	[ID] (;) [TIPO_DADO] [LISTA_CAMPOS]
[LISTA_CAMPOS]	(;) [CAMPOS] [LISTA_CAMPOS] ε
[DEF_VAR]	(var) [VARIABEL] (;) [LIST_VAR] ε
[LIST_VAR]	[VARIABEL] (;) [LIST_VAR] ε
[VARIABEL]	[ID] [LISTA_ID] (;) [TIPO_DADO]
[LISTA_ID]	(,) [ID] [LISTA_ID] ε
[DEF_FUNC]	[FUNCAO] [DEF_FUNC] ε
[FUNCAO]	(function) [NOME_FUNCAO] [BLOCO_FUNCAO]
[NOME_FUNCAO]	[ID] [PARAM_FUNC] (;) [TIPO_DADO]
[PARAM_FUNC]	(l) [CAMPOS] (l) ε
[BLOCO_FUNCAO]	[DEF_VAR] [BLOCO]
[BLOCO]	(begin) [COMANDO] (;) [LISTA_COM] (end)
[LISTA_COM]	[COMANDO] (;) [LISTA_COM] ε
[COMANDO]	[ID] [NOME] (:=) [EXP_MAT] (while) [EXP_LOGICA] [BLOCO] (if) [EXP_LOGICA] [BLOCO] [ELSE] (write) [PARAMETRO] (read) [ID] [NOME]
[ELSE]	(else) [BLOCO] ε
[LISTA_PARAM]	[PARAMETRO] (,) [LISTA_PARAM] [PARAMETRO] ε
[EXP_LOGICA]	[EXP_COMP] [OP_LOGICO] [EXP_LOGICA] [EXP_COMP]
[OP_LOGICO]	(or) (and)
[EXP_COMP]	[EXP_MAT] [OP_COM] [EXP_COMP] [EXP_MAT]
[OP_COMP]	(>) (<) (=) (!)
[EXP_MAT]	[PARAMETRO] [OP_MAT] [EXP_MAT] [PARAMETRO]
[OP_MAT]	(+) (-) (*) (/)

[PARAMETRO]	[ID] [NOME]
	[NUMERO]
[NOME]	(.) [ID] [NOME]
	(() [PARAMETRO] ())
	(() [LISTA_PARAM] ())
	ε
[ID]	Seqüência alfanumérica iniciada por char (tratado no léxico)
[NUMERO]	Seqüência numérica com a ocorrência de no máximo um ponto (tratado no léxico)

```

const    TAM = 10;
        MSG = "digite a nota do aluno";
var      A, B, C, D : integer;
        E : array [15] of integer;
        F : record
            nota1 : real;
            nota2 : real
        end;

```

```

function fatorial(a:integer) : integer
var i : integer;
begin
    i := 1;
    result:=1;
    while i < a and result < 100
    begin
        result:=result*i;
        i:=i+1;
    end;
end

```

```

function exp(a: real; b: real) : real
var i : integer;
begin
    i := 1;
    result := a;
    if b = 0 or b > 10
    begin
        result := 1;
    end
    else
    begin
        while i < b
        begin
            result := a * a;
            i := i + 1;
        end;
    end;
end

```

```

function maior(a : array [15] of integer) : integer
var i : integer;
begin
    i := 0;
    result := a[0];
    while i < 15
    begin
        if a[i] > result
        begin
            result := a[i]
        end;
    end;
end

```

```

function lerDados : real
begin
    write MSG;
    read result;
end

```

```

function menor(a : array [15] of integer) : integer
var i : integer;
begin
    i := 0;
    result := a[0];
    while i < 15
    begin
        if a[i] < result
        begin
            result := a[i];
        end;
    end;
end

```

```

function media(a : array [15] of integer) : integer
var m : integer;
begin
    m := maior(a) + menor(a);
    result := m / 2;
end

begin
    A:=TAM + 20;
    B := fatorial(A);
    C := exp(A,B);
    D := media(E);
    F.nota1 := lerDados();
    F.nota2 := lerDados();
end

```