

# Proposta de Linguagem de Programação

**Discentes da Dupla:** Anderson Henrique da Silva  
Lurian Letícia dos Reis

**Nome da Linguagem Desenvolvida (Inventada):** 🌈 Rainbow

**Nome da linguagem de desenvolvimento:** Python


**IDE de desenvolvimento:** Pycharm

🌈 Rainbow é uma linguagem de programação simples e intuitiva, criada para facilitar o aprendizado de programação através de uma sintaxe amigável em português brasileiro. A linguagem foca em operações básicas como cálculos matemáticos, manipulação de texto, estruturas condicionais e de repetição, sendo ideal para iniciantes.

Inspirada na diversidade e inclusão (como as cores do arco-íris), 🌈 Rainbow utiliza palavras-chave coloridas e expressivas em português. As variáveis são declaradas com o símbolo "#" seguido do nome, mantendo a simplicidade visual. A linguagem não depende de indentação, usando "." para fim de linha e "{}" para blocos de código.

🌈 Rainbow prioriza a clareza e simplicidade, permitindo que qualquer pessoa, independente de sua experiência, possa criar programas básicos de forma natural e intuitiva. É uma linguagem feita com amor, para democratizar a programação.

## Tabela de Tokens

Nome do Token	Lexema	Expressão Regular	Descrição
Rainbow	Rainbow	Rainbow	Palavra reservada identificadora da linguagem
tipo_dado	cor_numero, cor_texto, cor_logico, cor_lista	cor_numero   cor_texto   cor_logico   cor_lista	Tipos de dados da linguagem  Rainbow
variavel	#nome, #idade, #contador, etc.	#[a-zA-Z][a-zA-Z0-9_]*	Variável identificada pelo símbolo #
oper_relacional	<, >, <=, >=, igual, diferente	<   >   <=   >=   igual   diferente	Operadores de comparação
oper_matematico	+, -, *, /	+   -   *   /	Operadores matemáticos básicos
oper_atribuicao	recebe	recebe	Operador de atribuição em português
oper_logico	E, OU, NAO	E   OU   NAO	Operadores lógicos em português
condicional_se	se	se	Estrutura condicional principal
condicional_senao	senao	senao	Estrutura condicional alternativa
condicional_senaose e	senaose	senaose	Estrutura condicional múltipla
repeticao_para	para	para	Laço de repetição determinado
repeticao_enquanto	enquanto	enquanto	Laço de repetição indeterminado
funcao_mostrar	mostrar	mostrar	Função para exibir saída

funcao_ler	ler	ler	Função para entrada de dados
valor_verdade	Verdadeiro, Falso	Verdadeiro   Falso	Valores lógicos
valor_texto	"Olá mundo", "Rainbow é lindo", etc.	"[^"]*"	Texto entre aspas duplas
valor_numero	42, -15, 3.14, 0, etc.	-?\d+(\.\d+)?	Números inteiros ou decimais
abre_parenteses	(	(	Abertura de parênteses
fecha_parenteses	)	)	Fechamento de parênteses
abre_chaves	{	{	Abertura de bloco
fecha_chaves	}	}	Fechamento de bloco
comentario	// qualquer texto	//[^\n]*	Comentário de linha
fim_linha	.	.	Terminador de comando
virgula	,	,	Separador de elementos
de	de	de	Preposição para laços
ate	ate	ate	Palavra para intervalos
passo	passo	passo	Incremento em laços

# Exemplos de códigos da linguagem

## Exemplo 1: Olá Mundo Colorido

Rainbow.

```
// O clássico Olá Mundo com as cores do arco-íris
mostrar("Olá Mundo 🌈 Rainbow!").
mostrar("Vermelho como a paixão pela programação").
mostrar("Laranja como a energia dos códigos").
mostrar("Amarelo como a luz das ideias brilhantes").
mostrar("Verde como a esperança de um futuro tech").
mostrar("Azul como a serenidade dos algoritmos").
mostrar("Anil como a profundidade do conhecimento").
mostrar("Violeta como a magia da inclusão").
mostrar("Bem-vindos ao mundo colorido da programação!").
```

## Exemplo 2: Programa simples de saudação

Rainbow.

```
// Programa que cumprimenta o usuário
#nome recebe ler("Qual é o seu nome? ").
mostrar("Olá, " + #nome + "! Bem-vindo ao 🌈 Rainbow!").
mostrar("Você está programando com as cores do arco-íris!").
```

## Exemplo 3: Calculadora básica

Rainbow.

```
// Calculadora simples
#primeiro_numero recebe ler("Digite o primeiro número: ").
#segundo_numero recebe ler("Digite o segundo número: ").

#soma recebe #primeiro_numero + #segundo_numero.
#subtracao recebe #primeiro_numero - #segundo_numero.
#multiplicacao recebe #primeiro_numero * #segundo_numero.
#divisao recebe #primeiro_numero / #segundo_numero.

mostrar("Soma: " + #soma).
mostrar("Subtração: " + #subtracao).
```

```
mostrar("Multiplicação: " + #multiplicacao).
mostrar("Divisão: " + #divisao).
```

## Exemplo 4: Estrutura condicional para classificar idade

Rainbow.

```
#idade recebe ler("Quantos anos você tem? ").
```

```
se (#idade < 13) {
    mostrar("Você é uma criança!").
} senaose (#idade < 18) {
    mostrar("Você é um adolescente!").
} senaose (#idade < 60) {
    mostrar("Você é um adulto!").
} senao {
    mostrar("Você é um idoso sábio!").
}

se (#idade >= 18) {
    mostrar("Você pode votar!").
} senao {
    mostrar("Ainda não pode votar, mas logo poderá!").
}
```

## Exemplo 5: Laços de repetição - tabuada colorida

Rainbow.

```
#numero recebe ler("Digite um número para a tabuada: ").
```

```
mostrar("Tabuada colorida do " + #numero).
```

```
para #i de 1 ate 10 passo 1 {
    #resultado recebe #numero * #i.
    mostrar(#numero + " x " + #i + " = " + #resultado).
}
```

```
mostrar("Tabuada finalizada com sucesso!").
```

## Exemplo 6: Contador com validação

Rainbow.

```
#contador recebe 0.
```

```
#limite recebe ler("Até que número quer contar? ").
```

```
se (#limite <= 0) {
```

```
    mostrar("Por favor, digite um número positivo!").
```

```
} senao {
```

```
    mostrar("Vamos contar juntos!").
```

```
enquanto (#contador < #limite) {
```

```
    #contador recebe #contador + 1.
```

```
    mostrar("Contando: " + #contador).
```

```
}
```

```
mostrar("Chegamos ao fim!").
```

```
mostrar("Contamos até " + #limite + " com muito amor!").
```

```
}
```