

Professor Isidro

# Guia de Referência - Portugol

---



## Objetivo deste material

Vamos explorar um resumo das principais instruções do Portugol. Use este documento como um “guia de referência rápida” para ser utilizado no seu dia-a-dia enquanto estuda.

---


## Tipos de Dados

<code>inteiro</code>	números positivos e negativos sem precisão decimal. Ex: 1, 2, -1, -50
<code>real</code>	números positivos e negativos com precisão decimal. Ex: 52.12, -35.14
<code>caractere</code>	1 única letra. Ex: 'a', 'b', 'c'
<code>cadeia</code>	Textos e sequencias de símbolos delimitados por aspas duplas. Ex. "isidro"
<code>logico</code>	Apenas valores "verdadeiro" ou "falso"

## Declarando variáveis

Estrutura	Exemplo
<code>tipo nome</code>	<code>inteiro a</code>
<code>tipo nome1, nome2, nome3</code>	<code>real x, y, z</code>

### Nomenclatura das variáveis

- sempre começar com uma letra
  - não pode ter espaços em branco
  - nunca use símbolos especiais (cedilha, acentos, etc)
  - nunca use hífen nos nomes (o hífen é a operação de subtração)
- 

---

## Entrada e Saída

<code>leia (variavel)</code>	O Programa fica "bloqueado" esperando o usuário digitar e teclar ENTER. O valor é armazenado na variável
<code>escreva ("texto")</code>	Exibe na tela o texto definido entre as aspas
<code>escreva (variavel)</code>	Exibe na tela o conteúdo armazenado na variável
<code>escreva ("texto", variavel)</code>	Exibe na tela o texto seguido do conteúdo da variável
<code>escreva ("\n")</code>	Pula uma linha
<code>escreva ("\t")</code>	Mostra um espaçamento de tabulação

## Operações Aritméticas

+	soma
-	subtração
*	multiplicação
/	divisão
%	resto (esta operação é restrita apenas a números inteiros)

---

## Tipos de Atribuições

=	atribuição simples. $a = 3$
+=	atribuição com soma: $a += 3$ → Equivalente $a = a + 3$
-=	atribuição com subtração: $a -= 3$ → Equivalente $a = a - 3$
*=	atribuição com multiplicação: $a *= 3$ → Equivalente $a = a * 3$
/=	atribuição com divisão: $a /= 3$ → Equivalente $a = a / 3$
++	incremento: $a++$ → Equivalente $a = a + 1$
--	decremento: $a--$ → Equivalente $a = a - 1$

---

## Operadores Relacionais (a resposta é somente verdadeiro ou falso)

>	maior que
>=	maior ou igual
<	menor que
<=	menor ou igual
==	igual a
!=	diferente de

## Operações Lógicas

e	conjunção (AND)
ou	disjunção (OR)
nao	negação (NOT)

---

## Tabelas Verdade (para operadores lógicos)

Tabela verdade "E"

condição 1	condição 2	resultado (condição 1 E condição 2)
verdadeiro	verdadeiro	verdadeiro
verdadeiro	falso	falso
falso	verdadeiro	falso
falso	falso	falso

Tabela verdade "OU"

condição 1	condição 2	resultado (condição 1 OU condição 2)
verdadeiro	verdadeiro	verdadeiro
verdadeiro	falso	verdadeiro
falso	verdadeiro	verdadeiro
falso	falso	falso

Tabela verdade "NAO"

condição	resultado
verdadeiro	falso
falso	verdadeiro

## Decisões

Tipo	Instrução	Exemplo
se	<pre>se (condicao) {     comandos }</pre>	<pre>se (media &gt;= 6.0){     escreva("Aprovado") }</pre>
se/senão	<pre>se (condicao) {     comandos-veradeiros } senao{     comandos-falso }</pre>	<pre>se (media &gt;= 6.0){     escreva("Aprovado") } senao {     escreva("Reprovado") }</pre>
se/senão se	<pre>se (condicao1){     comandos-alternativa1 } senao se (condicao2){     comandos-alternativa2 } senao se (condicao3){     comandos-alternativa3 } ... senao{     comandos-senao }</pre>	<pre>se (media &gt;=8.5){     escreva("Conceito A") } senao se (media &gt;= 7.0){     escreva("Conceito B") } senao se (media &gt;= 5.0){     escreva("Conceito C") } senao{     escreva("Reprovado") }</pre>
escolha/caso	<pre>escolha (variavel) {     caso "valor":         comandos     pare     ...     caso contrario:         comandos }</pre>	<pre>escolha (opcao) {     caso 1:         escreva("Deposito")     pare     caso 2:         escreva("Saque")     pare     caso contrario:         escreva("Invalido") }</pre>

## Repetições

Tipo	Instrução	Exemplo
enquanto	<pre>enquanto (condicao) {     comandos }</pre>	<pre>enquanto (contador &lt;=10) {     tabuada = numero *     contador     escreva (tabuada, "\n")     contador++ }</pre>
faça	<pre>faça {     comandos } enquanto (condicao)</pre>	<pre>faça {     tabuada = numero *     contador     escreva (tabuada, "\n")     contador++ } enquanto (contador &lt;= 10)</pre>
para	<pre>para (inicio;cond;incr) {     comandos }</pre>	<pre>para (inteiro contador = 1;     contador &lt;= 10;     contador++) {     tabuada = numero *     contador     escreva (tabuada, "\n") }</pre>