

Disciplina: Introdução a Programação

Aula 2: Conceitos básicos de programação

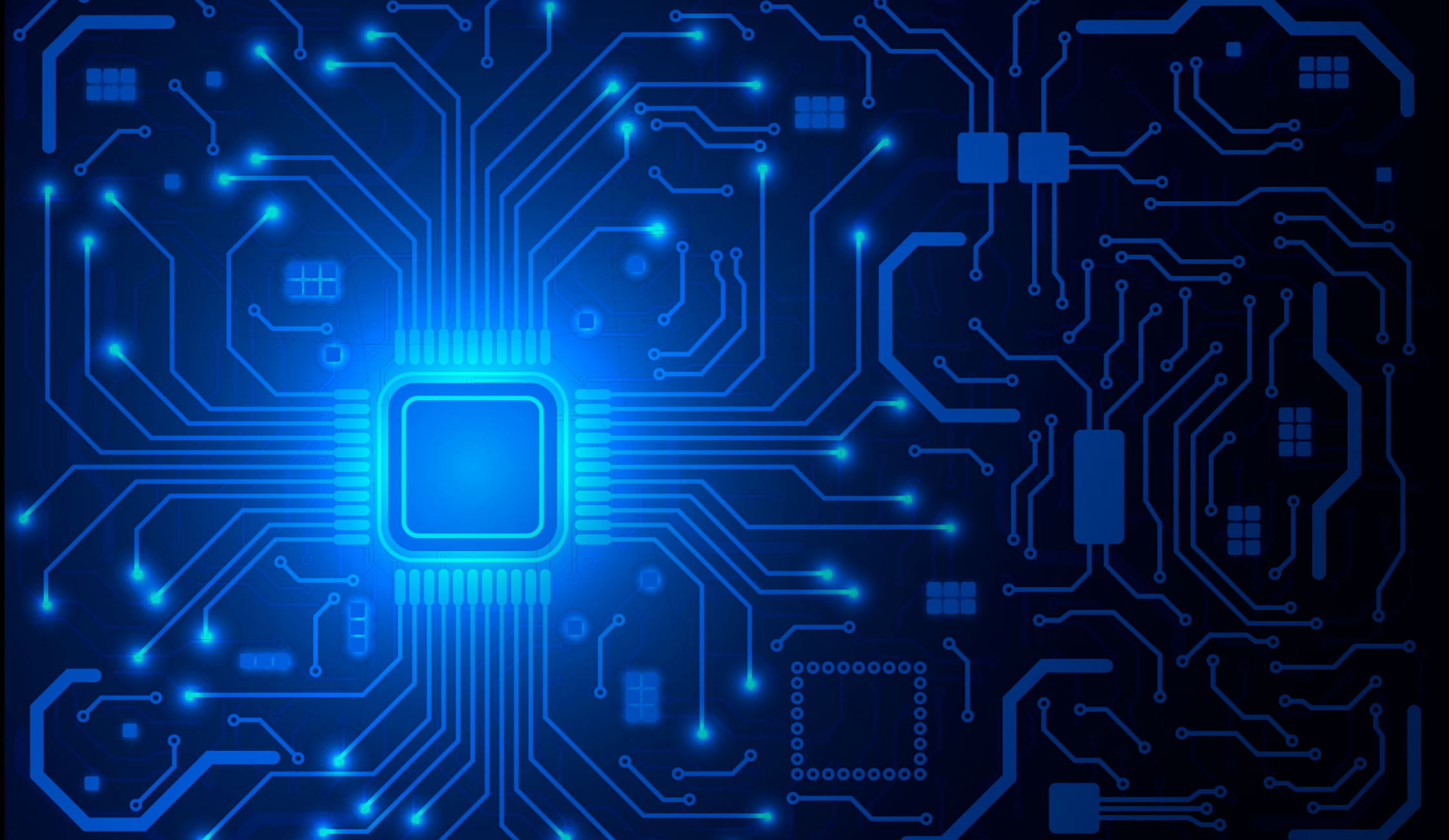
Apresentação

A codificação desenvolve habilidades para a resolução de problemas e permite que você crie uma solução que funcione exatamente da maneira que deseja. Para isso, é necessário definir as etapas para usar os recursos a sua disposição de forma a executar a tarefa com eficiência. Nesta aula, veremos alguns dos recursos disponíveis para os programadores: as variáveis e as constantes, seus tipos e os operadores.

As variáveis são essenciais e dificilmente você criará um programa ou algoritmo no qual não precise delas. Portanto, aprender a utilizá-las adequadamente é um requisito básico para os programadores. Os operadores também são muito importantes. Eles nos permitem criar expressões aritméticas, relacionais e lógicas e, assim como as variáveis, são muito utilizados nas soluções que desenvolveremos ao longo de nossos exercícios.

Objetivos

- Apontar o conceito de variáveis e constantes;
- Identificar os diferentes tipos de dados existentes;
- Examinar os diferentes tipos de operadores e sua aplicação.



Armazenamento de dados

Já estudamos que o princípio básico do funcionamento dos computadores envolve as etapas de

entrada > processamento > saída

Isso significa dizer que o computador recebe dados, faz o processamento necessário com eles e gera as informações necessárias para a saída. Na programação, trabalharemos diretamente com essas etapas.

Observe o exemplo a seguir:

```
Início

Escreva("Informe o 1o. número: ")
Leia(N1)
Escreva("Informe o 2o. número: ")
Leia(N2)
Subtra <- N1 - N2
Escreva("A subtração entre os números é ", Subtra)
Fim
```

Nesse exemplo, as cores destacam a etapa envolvida em cada momento. Veja que fica claro que ocorre a entrada de dois dados que servem para o cálculo da subtração, que é a etapa de processamento. Em seguida, o valor obtido com a realização do cálculo é mostrado para o usuário, o que configura a saída de informações.



Ilustração Desktop (Fonte: Shutterstock).

Os dados de entrada e as informações a serem exibidas como saídas ficam armazenados na memória principal do computador, conhecida como RAM¹ – *Random Access Memory* ou Memória de Acesso Aleatório.

Por isso, é preciso ter uma solução que permita armazenar os dados e recuperá-los sem dificuldade e sem que eles se misturem com todos os outros dados na memória. Essa solução existe: são as variáveis.



Compreendendo as variáveis

Uma variável é **um espaço da memória principal dedicado ao armazenamento temporário de valores**. Ela funciona como um recipiente em que colocamos os valores dos quais precisamos para o programa e retiramos de lá quando eles já não são mais necessários. É possível, também, alterar esses valores ao longo da execução do programa pois, como o nome diz, eles são variáveis

Quando criamos uma variável, precisamos dar a ela um identificador. Esse identificador funcionará como um rótulo para a área da memória onde a variável está armazenada. Toda as vezes que precisarmos recuperar o valor da variável ou armazenar um valor na mesma, faremos referência ao seu identificador.

As regras para a criação desses identificadores variam em função da linguagem de programação utilizada, por isso precisamos conhecer essas regras quando estivermos programando em uma nova linguagem.

Comentário

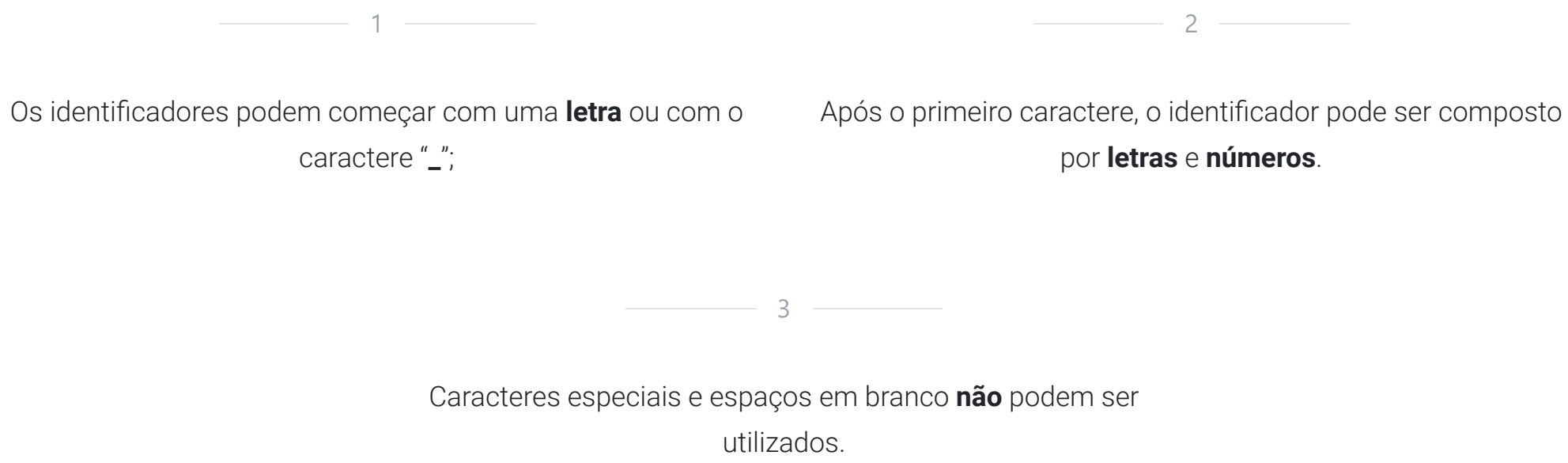
Em nossas aulas, utilizaremos o **Portugol Studio** para testar os algoritmos.

Para promover a familiarização com uma linguagem de programação, utilizaremos C++ e o ambiente de desenvolvimento Dev-C++. Este [software <https://sourceforge.net/projects/orwelldevcpp>](https://sourceforge.net/projects/orwelldevcpp) é gratuito e portátil.

Em geral, os identificadores de variáveis podem conter letras e números, não devem conter espaços em branco nem caracteres especiais, e não podem começar com números. O único caractere especial que costuma ser aceito em nomes de variáveis é o sublinhado (_).

É importante, também, que o identificador seja capaz de nos fazer lembrar o dado que está armazenado na variável. Se você precisa armazenar o nome de um cliente, por exemplo, o ideal é identificar sua variável como **nome** ou **nomeCliente**.

No C++, as regras para criação de identificadores de variáveis são as seguintes:

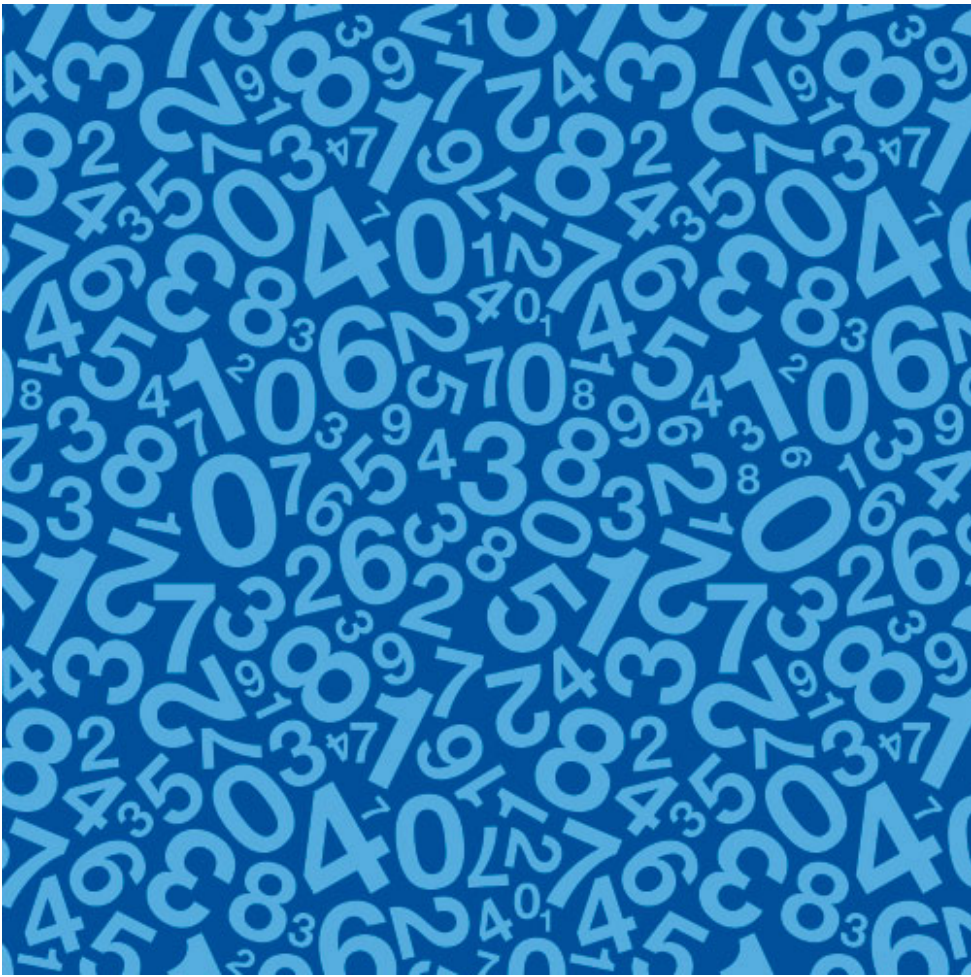


É importante ressaltar que algumas linguagens são *case sensitive* (ou sensíveis ao caractere). Isso significa que os caracteres maiúsculos são considerados diferentes dos seus equivalentes minúsculos. Assim, nesse tipo de linguagem, os identificadores **telefone**, **Telefone** e **TELEFONE** representariam três variáveis diferentes.

Atenção

Lembre-se: C++ é uma linguagem *case sensitive*.

Outro aspecto importante a respeito das variáveis é que elas podem ser de diferentes tipos. Em geral, esses tipos pertencem à categoria dos numerais, dos literais ou dos lógicos.



Numerais

Agrupam tipos que identificam variáveis cuja intenção é armazenar valores numéricos.



Literais

Permitem a criação de variáveis para armazenamento de caracteres.



Lógicos

Permitem a criação de variáveis para armazenamento de valores do tipo verdadeiro ou falso.

Observe a tabela a seguir e veja os tipos de variáveis existentes no Portugol Studio e seus tipos equivalentes no C++:

Tipos de Variáveis	
Portugol Studio	C++
Inteiro: número sem casa decimal significativa	int / long int
Real: número com casa decimal significativa	float / double
Caractere: somente um caractere	char
Cadeia: conjunto de caracteres	char
Logico: verdadeiro ou falso	bool

Atenção

No **Portugol Studio**, as palavras com o tipo de dado que define variáveis para armazenamento de valores lógicos são escritas sem os acentos.

Para que uma variável possa ser utilizada em um programa, ela precisa ser criada (ou **declarada**, que é o termo utilizado em programação).

A forma como a declaração de variáveis é feita depende da sintaxe da linguagem de programação utilizada.

A linguagem C++ é fortemente tipada, o que significa dizer que será necessário informar tipo e identificador da variável para que ela seja criada. O Portugol Studio segue o mesmo critério.

Veja exemplos de definições de variáveis a seguir.

Declaração de Variáveis	
Portugol Studio	C++
cadeianome = "Luciana" inteiroidade = 42 realsaldo = 550.00 cadeiasexo = 'F' logicotrabalha = verdadeiro	charnome[] = "Luciana"; intidade = 42; floatsaldo = 550.00; charsexo[] = "F"; booltrabalha = true;

Atenção

No C++, as variáveis lógicas são declaradas como do tipo **bool** e podem receber **true** (verdadeiro) ou **false** (falso).

Nas linguagens de **programação fortemente tipadas**, o tipo de dado a ser armazenado na variável deve ser **explicitamente definido**.

Outro aspecto importante é que a variável somente armazenará valores do tipo definido em sua declaração e a tentativa de armazenar valores de tipo distintos acarretará em erro.

Observe um **exemplo** de operação que gerará um erro:

```
soma = 2 + "2";
```

A operação tenta somar um tipo caractere com um tipo inteiro; o que não é permitido em linguagens fortemente tipadas. Alguns exemplos de linguagens fortemente tipadas são C e Java.

Existem também as linguagens de **programação fracamente tipadas**, nas quais o tipo de dado não é uma prioridade. Por conta disso, podemos especificar um tipo para um dado, mas usá-lo como outro tipo. Nesse caso, a conversão de tipos é realizado automaticamente.

Observe, por **exemplo**, o código a seguir:

```
soma = 2 + "2";
```

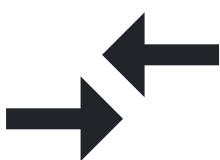
O resultado a ser armazenado na variável **soma** é **22**. Isso acontece porque o **número 2** é tratado como sendo do tipo caractere e é concatenado com o **caractere 2**". O mesmo código teria gerado um erro em uma linguagem fortemente tipada.

Comentário

Os programadores acham mais fácil escrever programas em linguagens de tipagem fraca. Alguns exemplos de linguagens fracamente tipadas são **Python** e **JavaScript**.

Nos dois exemplos seguimos basicamente o mesmo padrão: primeiro informamos o tipo da variável, seguido de seu identificador e de seu valor inicial. Perceba, entretanto, que, em C++, as declarações terminam com ";". Outro aspecto distinto diz respeito ao uso de aspas simples e duplas.

No Portugol Studio, as aspas simples são usadas para delimitar um único caractere (como em **sexo**), e as aspas duplas delimitam cadeias de caracteres (como em **nome**).



No C++ isso não acontece e as aspas duplas são utilizadas em ambos os casos.

Mas, se eu quiser usar um valor constante, como o de pi?

As constantes

Um valor constante é um valor que não se altera. Ao contrário das variáveis, que podem ter seu valor modificado ao longo do programa, as constantes têm um valor predefinido que não pode ser alterado.

Assim como as variáveis, entretanto, as constantes precisam ser declaradas e devem ter seu tipo e identificador definidos.

Observe como declarar uma constante no Portugol Studio e no C++:

Declaração de Variáveis	
Portugol Studio	C++
const real PI = 3.14 const inteiro MESES = 12	const float PI = 3.14; const int MESES = 12;

A declaração de constantes é muito parecida no Portugol Studio e no C++. Nos dois casos, a primeira palavra a ser escrita é **const**, seguida do tipo da constante, seu identificador e seu valor.

Por uma questão de convenção, os identificadores de constantes costumam ser escritos em caixa alta, mas essa não é uma obrigatoriedade.

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Operadores

Os operadores são símbolos que, combinados às variáveis e constantes, nos permitem construir expressões aritméticas, relacionais e lógicas.

QUANTO AO NÚMERO

Quanto ao número de operandos sobre os quais atuam, os operadores podem ser **unários** ou **binários**. Operadores unários atuam sobre um único operando, e operadores binários precisam de dois operandos para que possam ser utilizados.

QUANTO AO TIPO

Os operadores podem ser **aritméticos**, **relacionais**, **lógicos**, de **atribuição**, de **incremento** e de **decremento**.

Observe, a seguir, os operadores pertencentes a cada uma dessas categorias.

Operadores Matemáticos	
+	Soma
-	Subtração
/	Divisão
*	Multiplicação
%	Resto da divisão

Operadores Relacionais	
>	Maior que
<	Menor que
>=	Maior ou igual
<=	Menor ou igual
==	Igual a
!=	Diferente de

Operadores Lógicos	
!	Não
&&	E
	Ou

Operadores Especiais de Atribuição	
+=	Soma e atribui. Exemplo: res+=2
-=	Subtrai e atribui. Exemplo: res-=2
/=	Divide e atribui. Exemplo: res/=5
=	Multiplica e atribui. Exemplo: res=3

Operadores de Atribuição

= Atribuição de valor a variáveis ou constantes.

Operadores de Incremento / Decremento

++	Incremento
--	Decremento

Na próxima aula, veremos de que maneira os operadores se combinam às variáveis e constantes para construção de expressões.

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Atividade

1. Correlacione os conceitos com suas definições:

Constante **1**

Variável **2**

Memória principal **3**

Entrada -> processamento -> saída **4**

a) Armazena, em caráter temporário, os dados a serem utilizados no programa

1 2 3 4

b) Esquema básico de funcionamento dos computadores.

1 2 3 4

c) Espaço da memória para armazenamento de um valor que não será modificado.

1 2 3 4

d) Espaço da memória para armazenamento de um valor que poderá ser modificado.

1 2 3 4

Gabarito comentado

2) Observe os identificadores a seguir. Seguindo as regras da linguagem de programação C++, informe se os mesmos são válidos (V) ou inválidos (I).

a) nomeAluno

b) data de nascimento

c) SalARioATuAL

d) _salario01

e) *Média*

f) cor_favorita

g) TotCompras

h) 1a.nota

i) tel.celular

Atividade

- 3) a) Você se lembra do exercício da aula 1, no qual escreveu um algoritmo para converter uma quantia em reais para uma quantia em dólares? Você precisou perguntar a quantia em reais e a cotação do dólar do dia e calculou quantos dólares seria possível comprar. Imagine que você vai escrever esse algoritmo no Portugol Studio. Como ficaria a declaração de variáveis para esta situação?
- b) Agora imagine que você vai escrever o algoritmo anterior na linguagem de programação C++. Como seria a declaração de variáveis?
- c) A aceleração da gravidade (ag) na Terra possui um valor constante aproximado de $9,8\text{ m/s}^2$. Imagine que um corpo é abandonado em queda livre de uma determinada altura e leva um número de segundos (seg) para chegar ao solo. Considere que a velocidade (vel) da queda pode ser calculada multiplicando-se a aceleração da gravidade pelo tempo. Se você precisasse criar um algoritmo para resolver esse problema no Portugol Studio, como seria a declaração de variáveis e constantes?
- d) Considerando o problema anterior, como ficaria a declaração de variáveis e constantes caso a solução fosse escrita em C++?

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Notas

RAM ¹

A RAM é a memória responsável por auxiliar o computador em qualquer processamento que seja realizado. Isso significa que, além de armazenar os valores manipulados pelo seu programa, a RAM armazena milhares de outras informações ao mesmo tempo.

Referências

MANZANO, J. A. N. G., OLIVEIRA, J. F. **Algoritmos**: lógica para desenvolvimento de programação de computadores. 28.ed. São Paulo: Érica, 2016.

PUGA, S.; RISSETTI, G. **Lógica de programação e estruturas de dados com aplicações em Java**. 2.ed. São Paulo: Prentice Hall, 2005.

Próxima aula

- Diferentes tipos de expressão;
- Forma como são escritas as expressões;
- Expressões lógicas para construção de tabelas da verdade.

Explore mais

Na aula anterior, você ajudou os personagens de Star Wars a construir uma galáxia. Agora, que tal praticar suas habilidades artísticas [ajudando um personagem a desenhar diferentes formas geométricas?](https://studio.code.org/s/course4/stage/3/puzzle/1)

[<https://studio.code.org/s/course4/stage/3/puzzle/1>](https://studio.code.org/s/course4/stage/3/puzzle/1)

Divirta-se enquanto pratica a estruturação de seu pensamento para chegar às soluções necessárias à finalização dos desafios.