

PROGRAMAÇÃO EM PYTHON



PROGRAMAÇÃO EM PYTHON



Por meio de uma parceria com o SENAI-SP, a Prefeitura de Santana de Parnaíba está disponibilizando um curso focado em Python, com o objetivo de preparar os alunos para o desenvolvimento de programas e aplicações.



PROGRAMAÇÃO EM PYTHON



Plano de Aula

- Conteúdo:**
- Expressões;
 - Operadores;
 - Aritméticos;
 - Relacionais;
 - Lógicos;
 - Atividades;

Início:

As informações deste conteúdo visam compreender o conteúdo do curso.



PROGRAMAÇÃO EM PYTHON



Objetivo

Capacitar profissionais para desenvolver aplicações em linguagem Python, por meio de técnicas de programação, seguindo boas práticas, procedimentos e normas.

PROGRAMAÇÃO EM PYTHON



Expressões

Operação	Símbolo
Maior	>
Menor	<
Maior ou igual	>=
Menor ou igual	<=
Igual	==
Diferente	!=

Operação	Resultado
$3 > 4$	Falso
$7 != 7$	Falso
$9 == 10 - 1$	Verdadeiro
$33 <= 100$	Verdadeiro
$6 >= 5 + 1$	Verdadeiro
$5 + 4 <= 11 - 2$	Verdadeiro

Definição

Uma expressão em uma linguagem de programação é uma combinação de valores explícitos, constantes, variáveis, operadores e funções que são interpretados de acordo com as regras específicas de precedência e de associação para uma linguagem de programação específica, que calcula e, em seguida, produz um outro valor.

Este processo, tal como para as expressões matemáticas, chama-se avaliação. O valor pode ser de vários tipos, tais como numérico, cadeia e lógico.



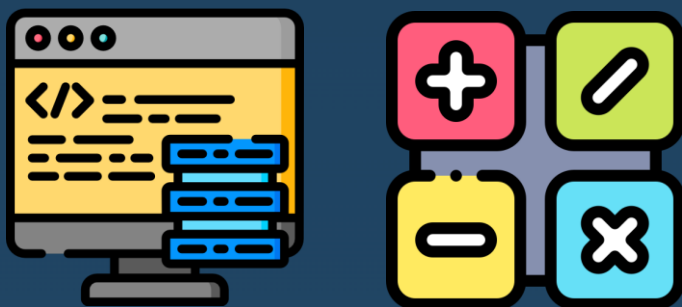
expressão aritmética

$$3 + 4 * 2$$

PROGRAMAÇÃO EM PYTHON



Operadores Atribuições



A = 10

Quando criamos uma variável, simplesmente separamos um espaço de memória para um conteúdo. Para especificar esse conteúdo, precisamos de alguma forma determinar um valor para essa variável. Para isso, usamos a operação de atribuição.

A instrução de atribuição serve para alterar o valor de uma variável. Ao fazer isso dizemos que estamos atribuindo um novo valor a esta variável. A atribuição de valores pode ser feita de variadas formas.

Exemplo de Sintaxe

- `valor = 6`
- `valor2 = valor + 10`
- `linguagem = "Python"`

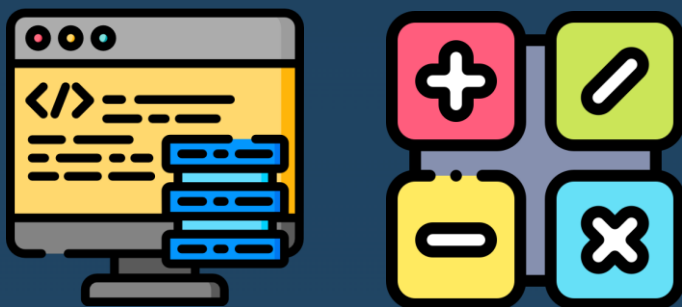
Explicação

O número 6 está sendo atribuído a variável `valor`. O sinal de atribuição é "`=`".

PROGRAMAÇÃO EM PYTHON



Operadores Aritméticos



As operações aritméticas são nossas velhas conhecidas da Matemática. Em algoritmos é muito comum usarmos operadores aritméticos para realizar cálculos.

Os símbolos que usamos para os operadores na Matemática mudam um pouquinho em algoritmos.

Operação	Símbolo	Prioridade
Adição	+	1
Subtração	-	1
Multiplicação	*	2
Divisão	/	2
Resto da divisão inteira	%	2

Segue alguns exemplos.

OBS: Caso professor desejar, pode ser utilizado a linguagem Python para demonstração dos exemplos e conhecimentos.

PROGRAMAÇÃO EM PYTHON



Ferramenta De Aprendizagem



Não é preciso a instalação:

<https://dgadelha.github.io/Portugol-Webstudio/>

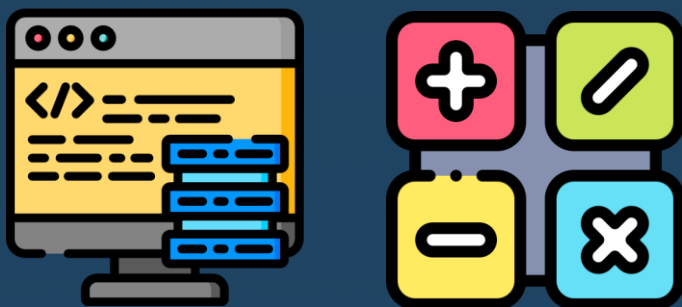


Ou a utilização do Visual Studio Code

PROGRAMAÇÃO EM PYTHON



Operadores Aritméticos



Será abordado explicação dos dois códigos:

```
1  programa {
2      funcao inicio() {
3          inteiro resultado
4          resultado = 3 + 4 * 2
5          escreva("O resultado da expressão (3 + 4 * 2) é: ", resultado)
6      }
7  }
```

0 resultado da expressão (3 + 4 * 2) é: 11

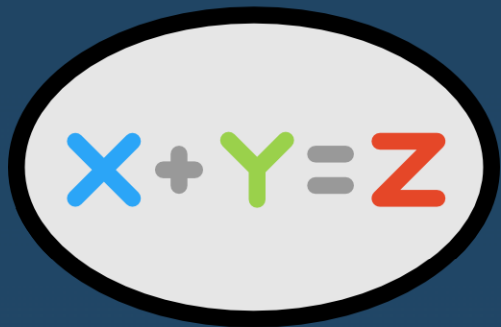
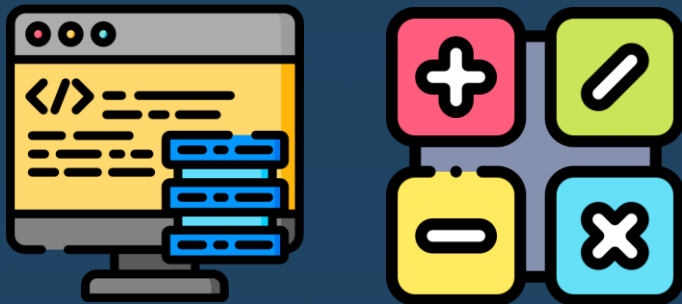
```
1  programa {
2      funcao inicio() {
3          inteiro resultado
4          resultado = (3 + 4) * 2
5          escreva("O resultado da expressão (3 + 4) * 2 é: ", resultado)
6      }
7  }
```

0 resultado da expressão (3 + 4) * 2 é: 14

PROGRAMAÇÃO EM PYTHON



Operadores Aritméticos



Primeiro código:

```
1  programa {  
2      funcao inicio() {  
3          inteiro resultado  
4          resultado = 3 + 4 * 2  
5          escreva("O resultado da expressão (3 + 4 * 2) é: ", resultado)  
6      }  
7  }
```

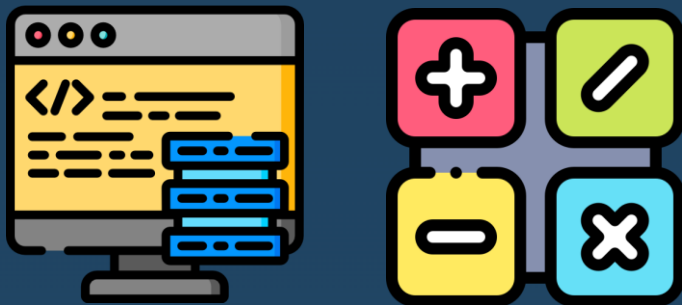
O resultado da expressão (3 + 4 * 2) é: 11

*3 + 4 * 2 segue a precedência, onde a multiplicação tem prioridade sobre a adição, resultando em 3 + (4 * 2) = 11.*

PROGRAMAÇÃO EM PYTHON



Operadores Aritméticos



Segundo código:

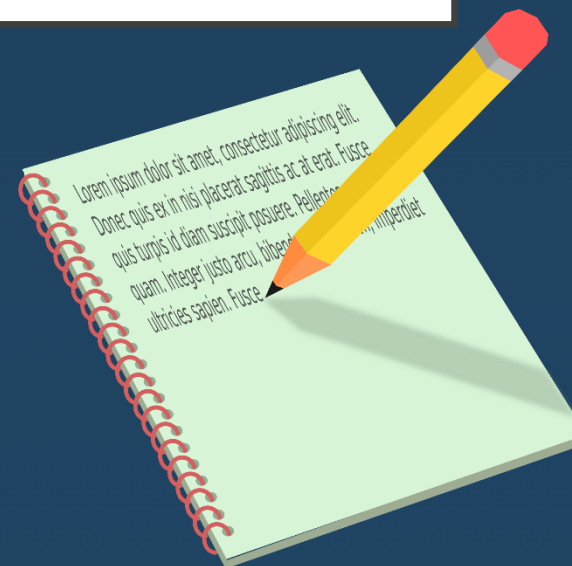
```
1  programa {  
2      funcao inicio() {  
3          inteiro resultado  
4          resultado = 3 + 4 * 2  
5          escreva("O resultado da expressão (3 + 4 * 2) é: ", resultado)  
6      }  
7  }
```

O resultado da expressão (3 + 4 * 2) é: 11

$(3 + 4) * 2$, com os parênteses, força a soma a ser executada primeiro, resultando em $(3 + 4) * 2 = 14$.

EXERCÍCIOS

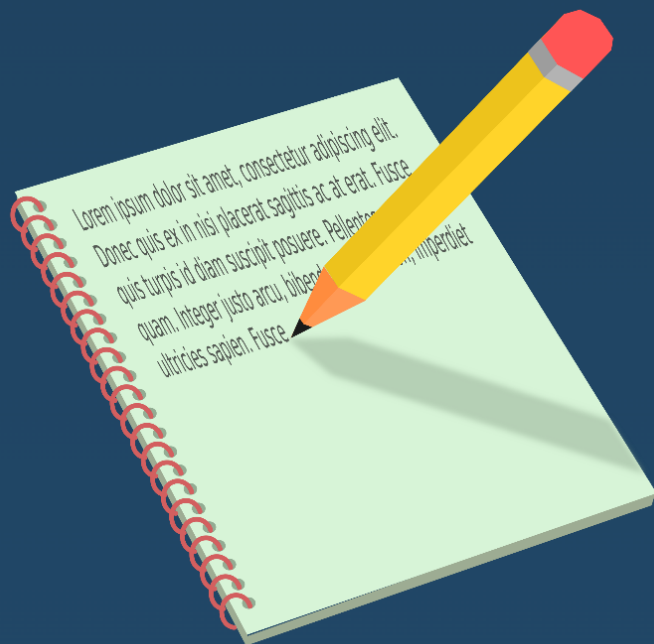
- Cálculo de Porcentagem de um Número.
 - O programa deve calcular e exibir o valor que corresponde a essa porcentagem do total. **Exemplo:** se o usuário digitar 200 como valor total e 15 como porcentagem, o programa deverá calcular que 15% de 200 é 30.
 - Exemplo de fórmula:
$$\text{valor_parte} = \text{valor_total} * (\text{porcentagem} / 100)$$



PROGRAMAÇÃO EM PYTHON



Cálculo de Porcentagem de um Número

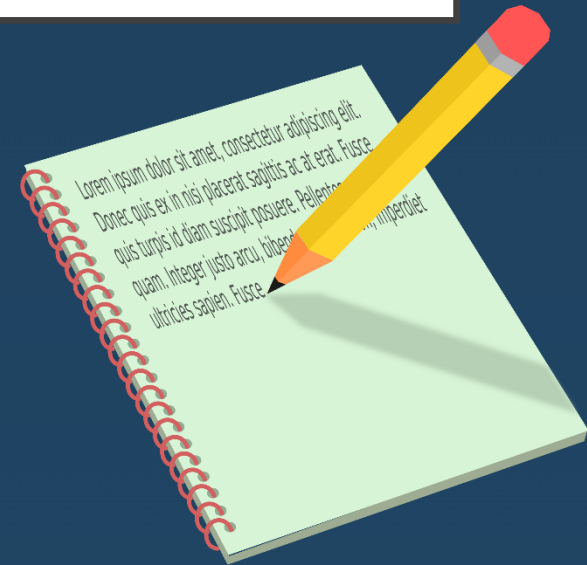


```
1  programa {
2      funcao inicio() {
3          // Declaração de variáveis do tipo real para permitir valores com casas decimais
4          real valor_total, porcentagem, valor_parte
5
6          // Solicita ao usuário que informe o valor total
7          escreva("Digite o valor total: ")
8          leia(valor_total)
9
10         // Solicita ao usuário que informe a porcentagem desejada
11         escreva("Digite a porcentagem: ")
12         leia(porcentagem)
13
14         // Calcula a parte correspondente à porcentagem do valor total
15         valor_parte = valor_total * (porcentagem / 100)
16
17         // Exibe o resultado
18         escreva("O valor correspondente a ", porcentagem, "% de ", valor_total, " é: ", valor_parte)
19     }
20 }
```

```
.....
Digite o valor total: 200
Digite a porcentagem: 15
O valor correspondente a 15% de 200 é: 30
```

EXERCÍCIOS

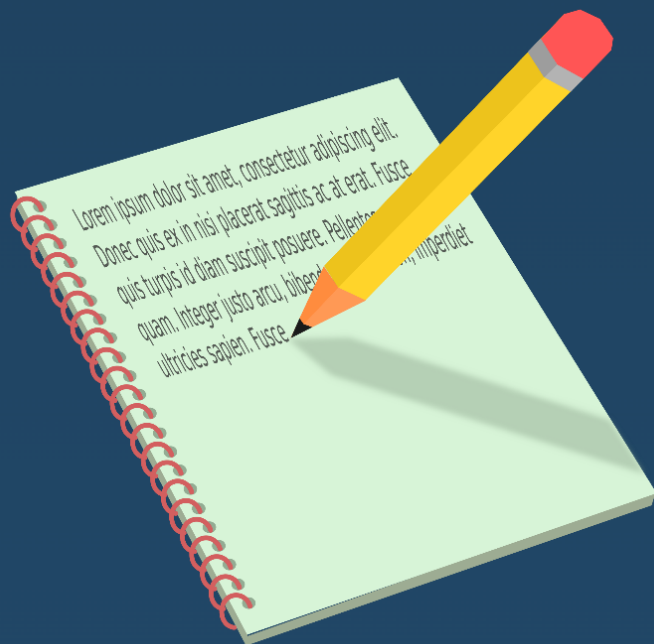
- Cálculo de Desconto em um Produto.
 - Se o usuário informar que o preço original é 100 e o desconto é de 20%, o programa deverá calcular que o valor do desconto é 20 e, conseqüentemente, o preço final será 80.
 - Exemplo de fórmula:
 $\text{valor_desconto} = \text{preco_original} * (\text{porcentagem_desconto} / 100)$
 $\text{preco_final} = \text{preco_original} - \text{valor_desconto}$



PROGRAMAÇÃO EM PYTHON



Cálculo de Desconto em um Produto



```
1 programa {
2     funcao inicio() {
3         // Declaração de variáveis do tipo real para permitir valores com casas decimais
4         real preco_original, porcentagem_desconto, valor_desconto, preco_final
5
6         // Solicita ao usuário o preço original do produto
7         escreva("Digite o preço original do produto: ")
8         leia(preco_original)
9
10        // Solicita ao usuário a porcentagem de desconto
11        escreva("Digite a porcentagem de desconto: ")
12        leia(porcentagem_desconto)
13
14        // Calcula o valor do desconto
15        valor_desconto = preco_original * (porcentagem_desconto / 100)
16
17        // Calcula o preço final após o desconto
18        preco_final = preco_original - valor_desconto
19
20        // Exibe os resultados
21        escreva("Valor do desconto: ", valor_desconto)
22        escreva("\nPreço final do produto: ", preco_final)
23    }
24 }
```

```
Digite o preço original do produto: 100
Digite a porcentagem de desconto: 20
Valor do desconto: 20
Preço final do produto: 80
```

PROGRAMAÇÃO EM PYTHON



Operadores Relacionais

$A > B$

$A == B$

$A < B$

$A != B$

Definição:

Vamos imaginar que você precise verificar se um número digitado pelo usuário é positivo ou negativo.

Como poderíamos verificar isto? **Através de uma operação relacional.**

As operações relacionais também são nossas conhecidas da **Matemática.**

Em algoritmos, os operadores relacionais são importantes, pois permitem realizar comparações que terão como resultado um valor lógico (verdadeiro ou falso).

OBS: Caso professor desejar, pode ser utilizado a linguagem Python para demonstração dos exemplos e conhecimentos.

PROGRAMAÇÃO EM PYTHON



Operadores Relacionais

A > B

A == B

A < B

A != B

Definição:

Os símbolos que usamos para os operadores também mudam um pouco em relação ao que usamos no papel. Os símbolos para diferente, maior ou igual e menor ou igual mudam pois não existem nos teclados convencionais. A tabela a seguir mostra todas as operações relacionais e os símbolos que o Portugol utiliza.

Operação	Símbolo
Maior	>
Menor	<
Maior ou igual	>=
Menor ou igual	<=
Igual	==
Diferente	!=

PROGRAMAÇÃO EM PYTHON



Operadores Relacionais

A > B

A == B

A < B

A != B

Definição:

A tabela a seguir apresenta a estrutura de algumas dessas operações. Nos dois últimos exemplos, temos operadores aritméticos e relacionais juntos. Nestes casos, realiza-se primeiro a operação aritmética e depois a relacional.

Operação	Resultado
$3 > 4$	Falso
$7 \neq 7$	Falso
$9 == 10 - 1$	Verdadeiro
$33 \leq 100$	Verdadeiro
$6 \geq 5 + 1$	Verdadeiro
$5 + 4 \leq 11 - 2$	Verdadeiro

PROGRAMAÇÃO EM PYTHON



Operadores Relacionais

Vamos
realizar
juntos!



```
1  programa {
2      funcao inicio() {
3          inteiro a, b
4
5          escreva("Digite o valor para variável A: ")
6          leia(a)
7
8          escreva("Digite o valor para variável B: ")
9          leia(b)
10
11         escreva("Comparações:\n")
12         escreva("a > b = ", (a > b), "\n")
13         escreva("a < b = ", (a < b), "\n")
14         escreva("a != b = ", (a != b), "\n")
15         escreva("a == b = ", (a == b), "\n")
16     }
17 }
```

Digite o valor para variável A: 7

Digite o valor para variável B: 5

Comparações:

a > b = verdadeiro

a < b = falso

a != b = verdadeiro

a == b = falso

EXEMPLO

PROGRAMAÇÃO EM PYTHON



Operadores Lógicos

E

OU

NAO

Definição:

Os operadores lógicos são usados para combinar ou inverter condições em expressões que retornam valores lógicos (verdadeiro ou falso). Em termos simples:

Por exemplo:

Operações	Resultado
$5 > 3$ e $2 < 1$	falso
nao ($8 < 4$)	verdadeiro
$1 > 3$ ou $1 \leq 1$	verdadeiro

Eles existem para permitir que tomemos **decisões mais complexas nos programas**. Em vez de verificar apenas uma condição isolada, **podemos verificar várias condições ao mesmo tempo** (com E, OU) ou negar alguma condição (com NÃO). Isso torna a lógica dos programas **mais rica e versátil**.

PROGRAMAÇÃO EM PYTHON



Operadores Lógicos

Vamos
realizar
juntos!



```
1  programa {
2      funcao inicio() {
3          inteiro a, b, c
4
5          escreva("Digite o valor para variável A: ")
6          leia(a)
7
8          escreva("Digite o valor para variável B: ")
9          leia(b)
10
11         escreva("Digite o valor para variável C: ")
12         leia(c)
13
14         escreva("Comparações:\n")
15         escreva("(a > b) E (c == b) = ", (a > b e c == b) , "\n")
16         escreva("(a < b) OU (c > b)= ", (a < b ou c > b), "\n")
17         escreva("NAO (a != b)= ", nao (a != b), "\n")
18     }
19 }
```

```
Digite o valor para variável A: 7
Digite o valor para variável B: 5
Digite o valor para variável C: 6
Comparações:
(a > b) E (c == b) = falso
(a < b) OU (c > b)= verdadeiro
NAO (a != b)= falso
```

PROGRAMAÇÃO EM PYTHON



Explicação sobre o conteúdo:

Operadores:

1. Aritméticos
2. Relacionais
3. Lógicos



PROGRAMAÇÃO EM PYTHON



Questões Mediadoras



- De que forma os operadores aritméticos influenciam o resultado de uma expressão, considerando a ordem de precedência?
- Como funciona a atribuição de valores às variáveis e por que ela é fundamental?
- Como funciona a atribuição de valores às variáveis e por que ela é fundamental?
- De que maneira os exercícios de cálculo de porcentagem e desconto reforçam o entendimento dos conceitos de operadores e atribuição?
- **Observação:** Pode ser utilizado o site <https://sorteio.com/sorteio-de-nomes> para sortear alunos a responderem, se achar necessário.

REFERÊNCIAS

ALVES, William Pereira. Lógica de programação de computadores. São Paulo: Érica, 2012.

ASCENCIO, Ana F. G.; CAMPOS, Edilene A. V. Fundamentos da programação de computadores. 3. ed. São Paulo: Pearson, 2012.

FARRELL, Joyce. Lógica e design de programação. São Paulo: Cengage Learning, 2010.

MANZANO, J. A. N. G.; OLIVEIRA, J. F. Algoritmos: Lógica para desenvolvimento de programação de computadores. 27. ed. rev. São Paulo: Érica, 2014.

MEDINA, Marco; FERTIG, Cristina. Algoritmos e programação. Teoria e prática. São Paulo: Novatec, 2005.

PEREIRA, Silvio do Lago. Algoritmos e lógica de programação em C: Uma abordagem didática. São Paulo: Érica, 2010.

Importante:

Os conteúdos disponibilizados são específicos para este curso/turma, a divulgação ou reprodução do material para outras pessoas/organização não é autorizada.



daniel.petinice@sp.senai.br



Daniel Tadeu Petinice



Instrutor de Formação Profissional III

