



INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 02 – LÓGICA DA PROGRAMAÇÃO

O QUE IREMOS APRENDER

- 01** EXERCÍCIOS DE REVISÃO
- 02** INTRODUÇÃO
- 03** OPERADORES
- 04** INDENTAÇÃO
- 05** ESTRUTURA CONDICIONAL
- 06** TIPOS DE ESTRUTURAS CONDICIONAIS
- 07** EXERCÍCIOS PRÁTICOS
- 08** INÍCIO DO PROJETO DO MÓDULO
- 09** INTRODUÇÃO AULA 03



Exercício de revisão

Faça um Programa que pergunte quanto você ganha por hora e o número de horas trabalhadas no mês. Calcule e mostre o total do seu salário no referido mês, sabendo-se que são descontados 11% para o Imposto de Renda, 8% para o INSS e 5% para o sindicato, faça um programa que nos dê:

Salário bruto

Quanto pagou ao INSS.

Quanto pagou ao sindicato.

O salário líquido.

Calcule os descontos e o salário líquido, conforme a tabela abaixo:

Salário Bruto : R\$

IR (11%) : R\$

INSS (8%) : R\$

Sindicato (5%) : R\$

Salário Líquido : R\$



Introdução

É muito comum na programação, precisarmos que determinado bloco de código seja executado apenas se determinada condição for satisfeita.

Para esses casos, o Python disponibiliza formas de controlar o fluxo para execução de programas:

Estruturas Condicionais IF/ELIF/ELSE.
Um conceito muito importante para se trabalhar com condições em Python é entender o conceito dos operadores lógicos.

Operadores em python

Os Operadores em Python possibilitam que o desenvolvedor consiga transcrever a lógica para código.

Python disponibiliza uma série desses operadores para os desenvolvedores e é muito importante dominá-los.



Operadores aritméticos

São utilizados para criarmos expressões matemáticas comuns, como soma, subtração, multiplicação e divisão.

Operador	Nome	Função
+	Adição	Realiza a soma de ambos operandos.
-	Subtração	Realiza a subtração de ambos operandos.
*	Multiplicação	Realiza a multiplicação de ambos operandos.
/	Divisão	Realiza a Divisão de ambos operandos.
//	Divisão inteira	Realiza a divisão entre operandos e a parte decimal de ambos operandos.
%	Módulo	Retorna o resto da divisão de ambos operandos.
**	Exponenciação	Retorna o resultado da elevação da potência pelo outro.



Operadores de comparação

São usados para comparar dois valores:

Operador	Nome	Função
==	Igual a	Verifica se um valor é igual ao outro
!=	Diferente de	Verifica se um valor é diferente ao outro
>	Maior que	Verifica se um valor é maior que outro
>=	Maior ou igual	Verifica se um valor é maior ou igual ao outro
<	Menor que	Verifica se um valor é menor que outro
<=	Menor ou igual	Verifica se um valor é menor ou igual ao outro

Operadores de atribuição

São utilizados no momento da atribuição de valores à variáveis e controlam como a atribuição será realizada.

Operador	Equivalente a
=	$x = 1$
+=	$x = x + 1$
-=	$x = x - 1$
*=	$x = x * 1$
/=	$x = x / 1$
%=	$x = x \% 1$



Operadores lógicos

Operadores que possibilitam construir um tipo de teste muito útil e muito utilizado em qualquer programa Python: os testes lógicos.

Python nos disponibiliza três tipos de Operadores Lógicos: o and, o or e o not.

Operador	Definição
and	Retorna True se ambas as afirmações forem verdadeiras
or	Retorna True se uma das afirmações for verdadeira
not	retorna Falso se o resultado for verdadeiro



Operadores de associação

Servem para verificar se determinado objeto está associado ou pertence a determinada estrutura de dados.

Operador	Função
in	Retorna True caso o valor seja encontrado na sequência
not in	Retorna True caso o valor não seja encontrado na sequência

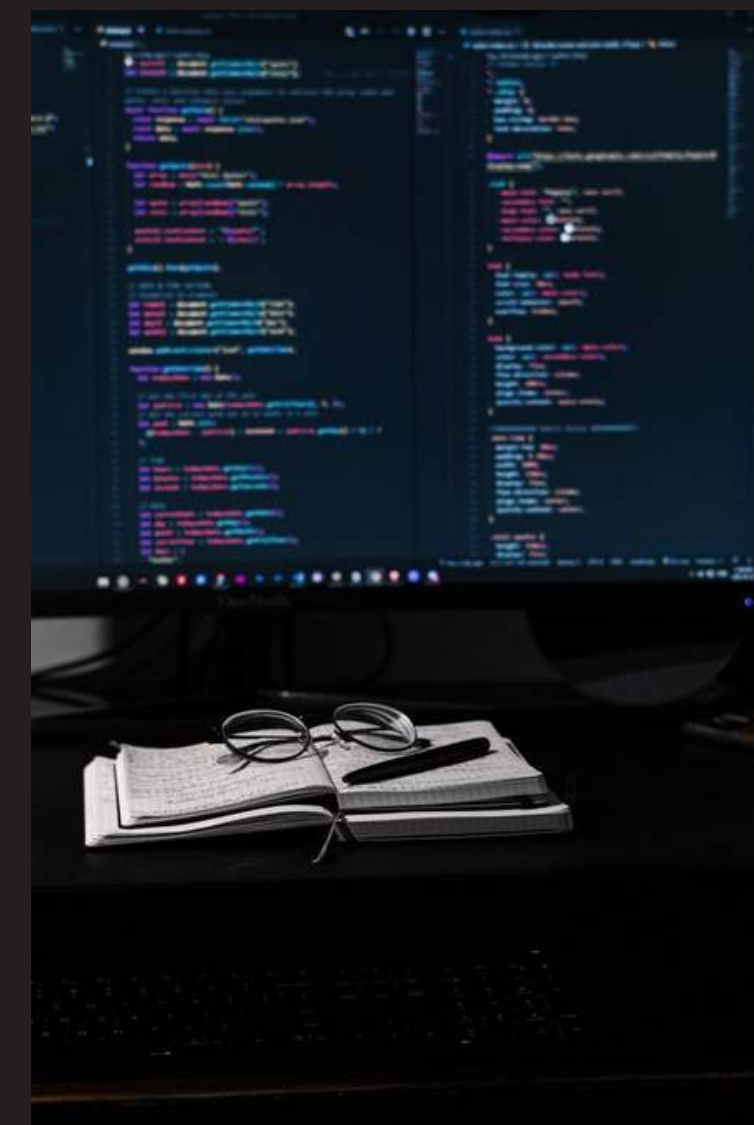


Indentação

Indentação é uma forma de arrumar o código em python, fazendo com que algumas linhas fiquem mais à direita que outras, à medida que adicionamos espaços em seu início. Para a maioria das linguagens a indentação não é obrigatória, mas no caso Python isso é diferente.

A indentação é importante no Python, pois é essencial para o bom funcionamento do código.

Se a indentação não estiver adequada o programa pode se comportar de forma inesperada ou até mesmo não compilar (rodar).



Estrutura condicional (if, elif, else)

O que é a estrutura condicional?

A estrutura condicional é uma seção que ajuda a definir condições para a execução de determinados blocos de comandos.

Ao invés de executar tudo de vez, sem nenhuma interrupção, o programa deve parar para executar um teste e decidir qual caminho seguir.

É como uma bifurcação, um momento importante para tomada de decisão. Nesse caso, a decisão é o resultado de uma análise, de uma comparação.

Funciona assim: Se essa condição X é satisfeita, então execute esse bloco de comandos; senão, execute esse outro bloco de comandos.

Estrutura condicional (if, elif, else)

If = "SE"

Elif = "Senão Se"


Else = senão



Estrutura condicional simples

Podemos dizer que quando aplicamos uma condição à programação, entendemos a grosso modo, validar se a condição é verdadeira ou falsa, com isso, dependendo do retorno, verdadeiro ou falso, é direcionado ao bloco de instruções correspondente.

Indentação.



```
idade = 18
if idade < 20:
    print('Você é jovem!')
```


ATIVIDADE PRÁTICA

1. Faça um Programa que peça dois números e imprima o maior deles.

Estrutura condicional simples

```
var = 5

if var == 5:
    print('Os valores são iguais')

if var != 7:
    print('O valor não é igual a 7')

if var > 2:
    print('O valor da variável é maior de 2')

if var >= 5:
    print('O valor da variável é maior ou igual a 5')

if var < 7:
    print('O valor da variável é menor que 7')

if var <= 5:
    print('O valor da variável é menor ou igual a 5')
```

Estrutura condicional simples

```
num1 = 7
num2 = 4

# Exemplo and
if num1 > 3 and num2 < 8:
    print("As Duas condições são verdadeiras")

# Exemplo or
if num1 > 4 or num2 <= 8:
    print("Uma ou duas das condições são verdadeiras")

# Exemplo not
if not (num1 < 30 and num2 < 8):
    print('Inverte o resultado da condição entre os parênteses')
```

Estrutura condicional composta

Um outro método de aplicar uma condição é com a aplicação da estrutura condicional composta, que é representada na linguagem de programação Python, com a aplicação de if e else no mesmo código.

```
1 | idade = 18
2 | if idade >= 18:
3 |     print('maior de idade')
4 | else:
5 |     print('menor de idade')
```

ATIVIDADE PRÁTICA

1. Faça um Programa que peça um valor e mostre na tela se o valor é positivo ou negativo.

Estrutura condicional alinhada

Estrutura condicional, que tem a aplicação do if, elif e else. Interessante nessa estrutura que o elif pode ser utilizando quantas vezes for necessário ao seu código e, a composição da sintaxe é parecida do if, ou seja, a frente do elif vem a condição e ao final ':' (dois pontos) para fechar.

```
linguagem = "Python"

if linguagem == "C++":
    print('C++ é uma linguagem de programação compilada.')
elif linguagem == "Python":
    print("Python é uma linguagem de programação de alto nível")
elif linguagem == "Java":
    print("Java é uma linguagem de programação amplamente utilizada no mercado")
else:
    print('Não é nenhuma das duas opções')
```


Estrutura condicional ternária

No Python existe uma forma concisa de se testar valores com apenas uma linha de código.

São os chamados if-ternários, ou condições ternárias, ou operadores ternários.

```
velocidade = 75

resultado = 'Multado' if velocidade > 60 else 'Dentro do limite'

print(resultado)
```

ATIVIDADE PRÁTICA

1. Faça um Programa que verifique se uma letra digitada é "F" ou "M". Conforme a letra escrever:
F - Feminino,
M - Masculino,
Outra letra qualquer - Sexo Inválido.

Fixando

IF em Python

O IF deve propor alguma coisa. É preciso escrever o IF e logo depois colocar a condição analisada. Então, em seguida, o bloco de comandos.

Else em Python

O ELSE surge depois do IF, em complemento lógico a ele. Então, não existe hipótese de escrever um ELSE sem um IF antes.

Geralmente, o ELSE não requer um teste, uma comparação, pois ele executa algo caso a comparação do IF não passe.

Portanto, você só precisa declarar o ELSE e adicionar o bloco de comandos.

Elif em Python

O elif é uma estrutura intermediária dentro da seção if-else no python e deve vir como um complemento a ambos. Quando você já tem um IF e um ELSE, mas precisa de uma condição para especificar outra regra, pode usar o elif.



ATIVIDADE PRÁTICA

- Faça um Programa que verifique se uma letra digitada é vogal ou consoante.
- Faça um programa para a leitura de duas notas parciais de um aluno. O programa deve calcular a média alcançada por aluno e apresentar:

A mensagem "Aprovado", se a média alcançada for maior ou igual a sete;

A mensagem "Reprovado", se a média for menor do que sete;

A mensagem "Aprovado com Distinção", se a média for igual a dez.

Match Case

- O match case em Python oferece uma maneira mais expressiva e concisa de fazer correspondência de padrões em valores.
- Ele promove a legibilidade do código, reduzindo erros de indentação e garantindo a exaustividade no tratamento de todos os casos possíveis. Além disso, o match case é mais seguro do que if-elif-else, pois obriga o tratamento explícito de todos os casos, minimizando a possibilidade de erros.

```
1  # Definindo um dado
2  dado = "2"
3
4  # Verificando o tipo do dado com match case
5  match dado:
6      case str():
7          print("O dado é uma string.")
8      case int():
9          print("O dado é um número inteiro.")
10     case float():
11         print("O dado é um número decimal.")
12     #equilave ao else
13     case _:
14         print("Tipo de dado não reconhecido.")
```

ATIVIDADE PRÁTICA

- Faça um Programa que leia três números e mostre o maior deles.
- Faça um Programa que leia três números e mostre o maior e o menor deles.
- Faça um programa que pergunte o preço de três produtos e informe qual produto você deve comprar, sabendo que a decisão é sempre pelo mais barato.
- Faça um Programa que leia três números e mostre-os em ordem decrescente.

INICIANDO PROJETO

- Neste módulo você construirá um simulador de cadastro de senha e inicialização de celular.

Vamos supor que o usuário acabou de comprar um celular na loja e após a primeira carga, ele liga o celular pela primeira vez.

Ao iniciar o celular, a primeira coisa que aparece na tela é o pedido para cadastrar a senha e, logo após, confirmar a senha.

Essa é a sua primeira tarefa.

Construa o pedido de senha e confirmação.

Se a senha confirmada estiver correta, exiba a mensagem: Senha correta. Bem-vindo.

Senão, exiba a mensagem: Senha incorreta. Tente novamente.

SE LIGA NO CONTEÚDO DA PRÓXIMA AULA!

AULA 03 DE LÓGICA DA PROGRAMAÇÃO.
ESTRUTURA DE REPETIÇÃO LOOPING (WHILE).

IN

INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

Estrutura de repetição

O laço while executa um ou mais comandos até que uma condição seja verdadeira. Na prática, a instrução contém uma expressão logo no início, que é avaliada a cada loop, e um bloco de códigos que é executado enquanto a condição for verdadeira.

Geralmente, o laço while é utilizado para repetir uma ou várias instruções por determinado número de vezes. Para isso, usamos uma variável com a função de contador para controlar o número de repetições dos comandos, ou uma expressão para determinar quando a condição for verdadeira ou falsa.



Estrutura de repetição

```
contador = 0

while contador < 5:
    print(f'Valor do contador é {contador}')
    contador += 1
```



INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 02 - LÓGICA DE PROGRAMAÇÃO