

Estruturas de Controle

[Instrução if](#)

[Exemplo 1](#)

[Exemplo 2](#)

[Exemplo 3](#)

[Instrução While](#)

[Exemplo 4](#)

[Operadores de atribuição especiais](#)

[Instrução FOR](#)

[Exercícios de Aula](#)

Nem sempre todas as linhas dos programas serão executadas. Muitas vezes, será mais interessante decidir que partes do programa devem ser executadas com base no resultado de uma condição. A base dessas decisões consistirá em expressões lógicas que permitam representar escolhas em programas.

Instrução if

Em Python, a estrutura de decisão é o `if`. Seu formato é apresentado a seguir:

```
if <condição>:  
    bloco verdadeiro
```

Exemplo 1

```
# Lê dois valores e imprime qual é o valor maior  
a = int(input("Primeiro valor: "))  
b = int(input("Segundo valor: "))
```

```
if a>b:
    print("O primeiro valor é o maior")
if b>a:
    print("O segundo valor é o maior")
```



Python é uma das poucas linguagens de programação que utiliza o deslocamento do texto à direita (**recuo**) para marcar o início e o fim de um bloco. Outras linguagens contam com palavras especiais para isso, como **BEGIN** e **END**, em Pascal; ou as famosas chaves (**{** e **}**), em C e Java.

Exemplo 2

Um problema comum é quando temos de pagar Imposto de Renda. Normalmente, pagamos o Imposto de Renda por faixa de salário. Imagine que, para salários menores que **R\$ 1.000,00**, não teríamos imposto a pagar, ou seja, alíquota 0%. Para salários entre **R\$ 1.000,00 e R\$ 3.000,00**, pagaríamos 20%. **Acima** desses valores, a alíquota seria de 35%. Quem ganha R\$ 4.000,00 tem os primeiros R\$ 1.000,00 isentos de imposto; com o montante entre R\$1.000,00 e R\$ 3.000,00 pagando 20%, e o restante pagando os 35%.

```
#Cálculo do Imposto de Renda
salario = float(input("Digite o salário para cálculo do impos
base = salario
imposto = 0
if base > 3000:
    imposto = imposto + ((base-3000)*0.35)
    base = 3000
if base>1000:
    imposto = imposto + ((base-1000)*0.20)
print(f"Salário: R${salario:6.2f} Imposto a pagar: R${imposto
```

```
print("Salário: R$%6.2f Imposto a pagar: R$%6.2f" %(salario, imposto))
print("Salário: R${} Imposto a pagar: R${}".format(salario, imposto))
```

Exemplo 3

```
#Categoriapreço
categoria = int(input("Digite a categoria do produto: "))
if categoria == 1:
    preco = 10
else:
    if categoria == 2:
        preco = 18
    else:
        if categoria == 3:
            preco = 23
        else:
            if categoria == 4:
                preco = 26
            else:
                if categoria == 5:
                    preco = 31
                else:
                    print("Categoria inválida, digite um valor entre 1 e 5")
                    preco=0
print(f"O preço do produto é: R${preco:6.2f}")
```

Usando a solução `elif` :

```
#Categoriapreço
categoria = int(input("Digite a categoria do produto: "))
if categoria == 1:
    preco = 10
elif categoria == 2:
    preco = 18
elif categoria == 3:
```

```
    preco = 23
elif categoria == 4:
    preco = 26
elif categoria == 5:
    preco = 31
else:
    print("Categoria inválida, digite um valor entre 1 e 5!")
    preco=0
print(f"O preço do produto é: R${preco:6.2f}")
```

Instrução While

Uma das estruturas de repetição do Python é o `while`, que repete um bloco enquanto a condição for verdadeira.

```
while<condição>:
    bloco
```

Exemplo 4

```
#contagem regressiva
int i=10
while(i>=1):
    print(i)
    i = i-1
print("Fogo!")
```



A instrução `break` é utilizada para interromper a execução de `while` independentemente do valor atual de sua condição.

Operadores de atribuição especiais

Expressão	Forma Compacta
<code>x = x + y</code>	<code>x += y</code>
<code>x = x - y</code>	<code>x -= y</code>
<code>x = x * y</code>	<code>x *= y</code>
<code>x = x / y</code>	<code>x /= y</code>
<code>x = x % y</code>	<code>x %= y</code>

Instrução FOR

Existe um outro *loop* que simplifica essa ideia de começar com um valor e incrementá-lo até chegar em um valor final: o *loop* **for**.

```
for rodada in range(1,10):  
    print(rodada)
```

```
for rodada in range(1,10,2):  
    print(rodada)
```

```
for rodada in [1,2,3,4,5]:  
    print(rodada)
```



A função `range(m, n, p)`, é muito útil em laços, pois retorna uma lista de inteiros, começando em m e menores que n , em passos de comprimento p , que podem ser usados como sequência para o laço. `range` não é exatamente uma lista.

Exercícios de Aula

1. Escreva um programa que exiba uma lista de opções (menu): adição, subtração, divisão, multiplicação e sair. Imprima a tabuada da operação escolhida. Repita até que a opção saída seja escolhida.

```
num1 = int(input("Entre com um número: "))
num2 = int(input("Entre com outro numero: "))
ope = 1

while ope!=0:
    ope = int(input("Digite a operação desejada\n \
        1- Soma\n \
        2- Subtração\n \
        3- Multiplicação\n \
        4 - Divisão \n \
        0 - Para sair \n"))
    if ope==1:
        print(f"A soma de {num1}+{num2}={num1+num2} \n")
    elif ope==2:
        print(f"A subtração de {num1}-{num2}={num1-num2} \n")
    elif ope==3:
        print(f"A soma de {num1}x{num2}={num1*num2} \n")
    elif ope==4:
        print(f"A soma de {num1}/{num2}={num1/num2} \n")
    else:
        print("Opção inválida")
```

2. Escreva um programa que leia um número e verifique se é ou não um número primo.

```

num = int(input("Entre com um número: "))

counter=0
base = num

while base!=0:
    if num%base==0:
        counter = counter +1
        base = base -1

    if counter == 2:
        print(f"O número {num} é primo!")
    else:
        print(f"O número {num} não é primo!")

```

3. Modifique o programa anterior de forma a ler um número **n**. Imprima os **n** primeiros números primos.
4. O fatorial de um inteiro não negativo n é escrito como $n!$ (pronuncia-se " n fatorial") e é definido como segue:

$$n! = n \cdot (n - 1) \cdot (n - 2) \cdot \dots \cdot 1 \text{ (para valores de } n \text{ maiores ou iguais a 1) e } n! = 1 \text{ (para } n = 0)$$

Por exemplo, $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$, o que dá 120.

Escreva um aplicativo que lê um inteiro não negativo, calcula e imprime seu fatorial.

5. O quadrado de um número natural n é dado pela soma dos n primeiros números ímpares consecutivos. Por exemplo, $1^2 = 1$, $2^2 = 1+3$ etc. Dado um número n , calcule seu quadrado usando a soma de ímpares ao invés de produto.
6. A série de **Fibonacci** é 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... Os dois primeiros termos são iguais a 1, e a partir do terceiro, o termo é dado pela soma dos dois termos anteriores. Dado um número $n \geq 3$, exiba o n -ésimo termo da série de **Fibonacci**.

7. Numa certa agência bancária, as contas são identificadas por números de até seis dígitos seguidos de um dígito verificador, calculado conforme exemplificado a seguir. Dado um número de conta n , exiba o número de conta completo correspondente. *Seja $n = 7314$ o número da conta.*
Adicionamos os dígitos de n e obtemos a soma $s = 4+1+3+7 = 15$;
Calculamos o resto da divisão de s por 10 e obtemos o dígito $d = 5$. Número de conta completo: 007314-5