

FIAP GRADUAÇÃO

# ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Computational Thinking

PROF. EDUARDO GONDO

## necessidade

PROBLEMA 1: Dados uma sequência de 5 números inteiros.  
Calcule a soma de todos os números da sequência.

Vamos resolver esse nosso primeiro problema usando a linguagem Python.

```
1  soma = 0
2
3  aux = input("Digite um número: ")
4  num = int(aux)
5  soma = soma + num
6
7  aux = input("Digite um número: ")
8  num = int(aux)
9  soma = soma + num
10
11 aux = input("Digite um número: ")
12 num = int(aux)
13 soma = soma + num
```

## Solução do Problema 1 em Python

```
14
15 aux = input("Digite um número: ")
16 num = int(aux)
17 soma = soma + num
18
19 aux = input("Digite um número: ")
20 num = int(aux)
21 soma = soma + num
22
23 println("O resultado é ", soma)
```

## I Problema 2

Escreva um algoritmo que dados um número inteiro positivo  $n$ , imprime na tela todos os números de 1 a  $n$ .

### Considerações

- ▶ é impossível para este problema adotar a mesma estratégia de solução do problema 1
- ▶ note que o número  $n$  é uma informação fornecida pelo usuário e
- ▶ a quantidade de instruções depende diretamente desse valor

## | Solução do Problema 2

## Comando de repetição **while**

### Importante!

Os comandos de repetição são elementos fundamentais dentro de linguagens de programação pois permite a execução de um mesmo conjunto de instruções até que a condição não seja mais satisfeita.

Uma execução desse conjunto de instruções é chamado de **iteração**.  
Veja o comando de repetição **while** na linguagem Python:

- ▶ a sintaxe do comando **while** pode ser apresentada como:

```
1     while <expressão condicional>:  
2         //bloco contendo instruções que  
3         //serão executadas repetidamente
```

- ▶ só para lembrar, uma <expressão condicional> é uma expressão que retorna verdadeiro (True) ou falso (False)

## while — continuação

- ▶ interpretação: **enquanto a expressão condicional for verdadeira, as instruções do bloco são executadas**
- ▶ testa a expressão condicional antes de entrar no bloco, caso a expressão seja falsa, nenhuma instrução do bloco é executada
- ▶ ao término da execução das instruções do bloco, testa-se novamente a expressão condicional
- ▶ se a expressão for falsa a repetição termina
- ▶ podemos dizer que o **while** executa as instruções do bloco 0 ou mais vezes
- ▶ na maioria dos algoritmos, alguma hora a expressão condicional deve se tornar falsa para que o comando de repetição não entre em execução infinita



## Contador

- ▶ muitos problemas usam um contador associado ao comando `while`
- ▶ esses contadores são utilizados para controlar o número de iterações realizadas
- ▶ veja abaixo um exemplo:

```
1 contador = 0
2 limite = 10
3 while contador < limite:
4     #coloque aqui os comandos que
5     #serão repetidos
6     contador = contador + 1
7
8 print("Fim do while")
```

## Comando for

- ▶ na situação do exemplo anterior podemos também usar o comando `for` combinado com a função `range`
- ▶ veja o mesmo código anterior usando o `for` ao invés do `while`

```
1 limite = 10
2 for contador in range(limite):
3     #coloque aqui os comandos que
4     #serão repetidos
```

- ▶ a função `range(limite)` gera um intervalo de números inteiros de 0 até `limite - 1`
- ▶ no exemplo, ele irá gerar um intervalo de números de 0 até 9

## I Problema 3

Dada uma sequência de números inteiros onde o último elemento é o 0, escreva um algoritmo que calcula a soma dos números da sequência.

## I Problema 4

Dados uma quantidade  $n$  de alunos de uma turma de Algoritmos e suas notas da primeira prova, determinar a média das notas dessa turma.

## Problema 5

A conversão de graus Fahrenheit para centígrados é obtida pela fórmula  $C = \frac{5}{9}(F - 32)$ . Escreva um algoritmo que calcule e escreva uma tabela de graus centígrados em função da temperatura em graus Fahrenheit que variem de 50 a 150 Fahrenheit de 1 em 1.

## I Problema 6

Escreva um programa que dado um inteiro  $n$  positivo calcula e imprime a soma de todos os números inteiros entre 1 e  $n$ .

Por exemplo, se  $n = 10$  então deverá ser calculado:

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55$$

## Problema 7

Vamos escrever um programa que consiste em um Jogo de Adivinhação. O jogo funciona do seguinte modo: sorteia-se um número inteiro aleatório entre 1 e 1000. Sua tarefa é tentar adivinhar o número sorteado através de "chutes". A cada chute o programa deverá informar se o número "sorteado" é maior, menor ou igual ao número "chutado". Quando o usuário acertar o número deverá ser impresso uma mensagem dizendo que ele acertou e a quantidade de chutes dados. Para gerar números aleatórios entre 1 e 1000 use as seguintes instruções dentro do seu programa Python.

```
import random
```

```
sorteado = random.randint(1,1001)
```

## I Problema 8

Vamos deixar o problema anterior um pouco mais complicado, adicione um limite máximo de 10 tentativas para descobrir o número sorteado.



## I Problema 9

Validação de dados - alguns problemas necessitam de validação nos dados de entrada. Vamos pegar o Problema 4 e mostrar como garantir que as notas informadas estejam no intervalo entre 0 e 10.

## Referência Bibliográfica

- ▶ Puga e Rissetti - Lógica de Programação e Estrutura de Dados
- ▶ Ascêncio e Campos - Fundamentos da Programação de Computadores
- ▶ Forbelone e Eberspacher - Lógica de programação: a construção de algoritmos e estruturas de dados
- ▶ Documentação do Python - <https://docs.python.org/3.8/>
- ▶ Python Programming For Beginners: Learn The Basics Of Python Programming (Python Crash Course, Programming for Dummies) (English Edition). Kindle
- ▶ Python: 3 Manuscripts in 1 book: - Python Programming For Beginners - Python Programming For Intermediates - Python Programming for Advanced (English Edition). Kindle

# I Copyleft

Copyleft © 2021 Prof. Eduardo Gondo Todos direitos liberados.  
Reprodução ou divulgação total ou parcial deste documento é liberada.