



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA

DIM0320 Algoritmos e Programação de Computadores



#LISTAS

ELIEZIO SOARES

ELIEZIOSOARES@DIMAP.UFRN.BR

Contextualização

- Construa um programa que registre o nome dos alunos de uma turma. A turma tem 3 alunos. O programa deve ler o nome de cada aluno e registrar.

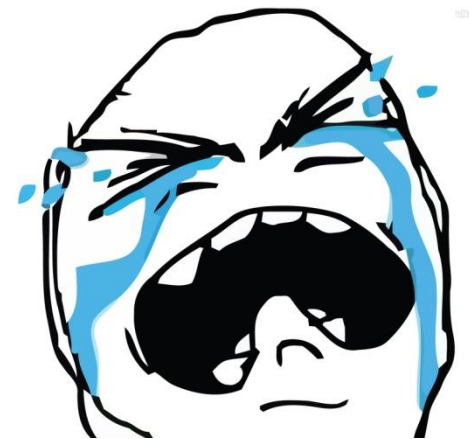
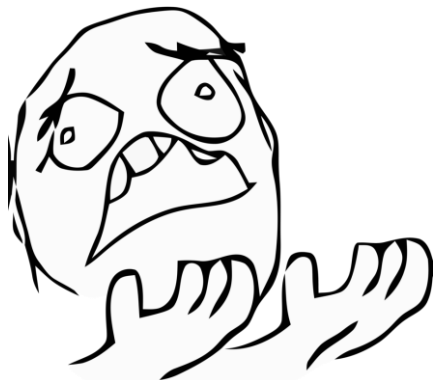
```
aluno1 = input("Informe seu nome: ")
aluno2 = input("Informe seu nome: ")
aluno3 = input("Informe seu nome: ")

print("Os alunos da turma são: \n %s \n %s \n %s" %(aluno1,aluno2,aluno3))
```



Contextualização

- Vamos alterar um pouco os requisitos do programa.
- Construa um programa que registre o nome dos alunos de uma turma. A turma pode variar a quantidade de alunos. O programa deve ler a quantidade de alunos da turma e ler o nome de cada aluno e registrar.

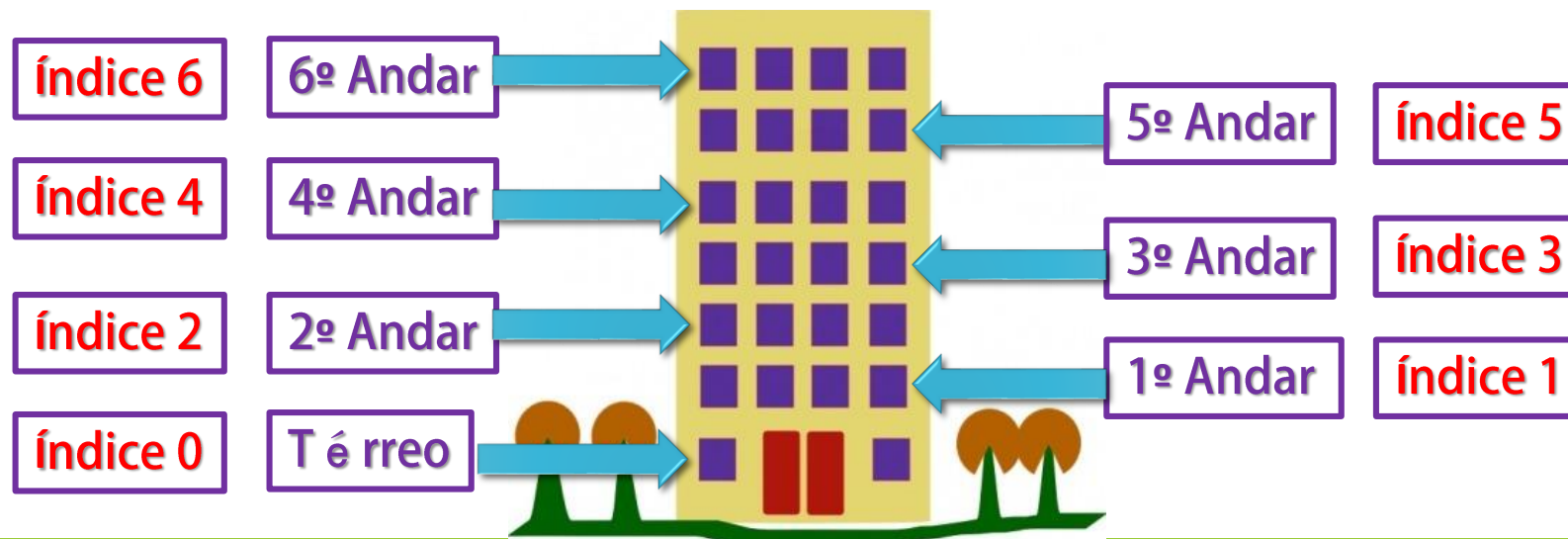


O que são listas?

- Uma lista é um conjunto de valores ordenados.
- Os valores são identificados por índices e chamamos de elementos.
- Uma lista pode conter zero ou mais elementos de qualquer tipo (em Python usa-se tipagem dinâmica)
- A medida que novos elementos são adicionados ou removidos o tamanho da lista cresce ou diminui.
- Exemplo:
 - [10,20,30,40,50,60,70,80,90,100]

O que são índices?

- Índice é uma referência a posição do elemento dentro da lista.
- Os índices se iniciam em zero.
- Imagine uma lista como um edifício de apartamentos, onde o térreo é o andar zero, o primeiro andar é o 1, conforme o exemplo abaixo:



Declaração de Listas

- Elementos envolvidos por colchetes [].
- Declaração e inicialização de uma **lista vazia**:
 - exemplo = []
- Declaração e inicialização de uma **lista com 5 elementos**:
 - exemplo = [10,50,5,2,100]
 - Exemplo = ["algoritmo", "é", "muito", "bom", "e fácil"]

Declaração de Listas

- Declaração e inicialização de uma **lista sequencial com o range**:
 - `minhaLista = range(10)`


- Declaração e inicialização de uma **lista ANINHADA**:
 - `suaLista = [10,50,5, [2, 100], "texto", 3.0]`

Acesso aos elementos de uma lista

- A forma de acesso a cada elemento de uma lista é através dos índices (endereços) de cada elemento:

```
suaLista = [10,50,5,2,100]
print(suaLista[0])
print(suaLista[1])
print(suaLista[2])
print(suaLista[3])
print(suaLista[4])
```

```
>>> ===== RESTART =====
>>>
10
50
5
2
100
>>> |
```



Acesso aos elementos de uma lista (Aprimorando o exemplo anterior!)

- A forma de acesso a cada elemento de uma lista é através dos índices (endereços) de cada elemento:

```
suaLista = [10,50,5,2,100]
i=0
while i<5:
    print(suaLista[i])
    i=i+1
```

```
>>> ===== RESTART =====
>>>
10
50
5
2
100
>>> |
```

Acesso aos elementos de uma lista (Aprimorando o exemplo anterior!)

- Assim como se acessa os elementos através do índice, é possível alterar os elementos na lista:
 - Exemplo:

```
suaLista = [10,50,5,2,100]
suaLista[3]=200
i=0
while i<5:
    print(suaLista[i])
    i=i+1
```

```
>>> ===== RESTART =====
>>>
10
50
5
200
100
>>> |
```

Adicionar elementos em uma lista

- Uma das maiores vantagens da utilização de listas é a possibilidade de adicionar elementos durante a execução do programa.
- Na prática o programador não precisa se preocupar com a quantidade de variáveis para os alunos da turma (lembra desse problema?)

- O método `append()`:

```
umaLista = []
umaLista.append("Joãozinho")
umaLista.append("Chico de José")
umaLista.append("Antônio")
umaLista.append("Zefinha")
i=0
while i<4:
    print(umaLista[i])
    i=i+1
```

Resolvendo o problema inicial

- Construa um programa que registre o nome dos alunos de uma turma. A turma pode variar a quantidade de alunos. O programa deve ler a quantidade de alunos da turma e ler o nome de cada aluno e registrar.

```
alunos = []
qtdAlunos = int(input("Informe a quantidade de alunos na turma: "))
i=0
while i < qtdAlunos:
    nome = input("Informe o nome do aluno %d: " %(i+1))
    alunos.append(nome)
    i = i+1

print("\n\n===== LISTA DE ALUNOS =====")
i=0
while i < qtdAlunos:
    print("Aluno %d: %s" %(i+1, alunos[i]))
    i = i+1
```

Remover elementos de uma lista (1/2)

- Para remover elementos da lista há dois métodos:
 - **del**: Apaga o elemento da lista.
 - Exemplo:

```
suaLista = [10,50,5,2,100]
del suaLista[2]

i=0
while i<4:
    print(suaLista[i])
    i=i+1
```

Remover elementos de uma lista (2/2)

- Para remover elementos da lista há dois métodos:
 - **pop**: Apaga o elemento da lista e retorna o elemento removido.
 - Exemplo:

```
suaLista = [10,50,5,2,100]
elementoRemovido = suaLista.pop(2)
print("Removido: %d" %elementoRemovido)

i=0
while i<4:
    print(suaLista[i])
    i=i+1
```

Tamanho de uma lista

- O tamanho de uma lista é igual à quantidade de elementos que ela contém.
- Para descobrir o tamanho de uma lista existe a função len.
- A função len retorna a quantidade de elementos na lista.

```
suaLista = [10,50,5,2,100]
tamanho = len(suaLista)
print("O tamanho dessa lista é: %d " %tamanho)
```

Aprimorando o while com a função len

- Utilizando a função len em vez de um inteiro literal na definição dos laços, o programa ganha em flexibilidade e manutenibilidade:

```
suaLista = [10,50,5,2,100]
i=0
while i<5:
    print(suaLista[i])
    i=i+1
```



```
suaLista = [10,50,5,2,100]
i=0
while i< len(suaLista):
    print(suaLista[i])
    i=i+1
```


Iterando listas com o for

- A estrutura de repetição for em Python é ideal para percorrer os elementos de uma lista.
 - Cada item da lista é armazenado (um por repetição) na variável elencada para essa finalidade.
 - A variável “elemento” receberá os itens da lista “suaLista”. Um de cada vez até que não hajam mais elementos.

```
suaLista = [10,50,5,2,100]
i=0
while i < len(suaLista):
    print(suaLista[i])
    i=i+1
```



```
suaLista = [10,50,5,2,100]
for elemento in suaLista:
    print(elemento)
```

Dúvidas



Exercícios

1. Escreva um programa que leia 10 inteiros e armazene-os em uma lista a.
2. Escreva um programa que escreva todos os dados da lista a.
3. Escreva um programa que leia 10 inteiros e armazene-os em uma lista a. O programa deve informar qual o maior e o menor número digitado.
4. Escreva um programa que leia as notas dos alunos de uma disciplina (A quantidade de alunos deve ser informada pelo usuário) e informe quantos alunos estão abaixo da média e quantos estão na média. (Considere a nota sendo um inteiro de 0 a 100 e a média 60)
5. Escreva um programa que leia 5 números inteiros. Em seguida, determine e imprima na tela o maior elemento par da lista(se houver), o menor elemento ímpar da lista(se houver), o somatório dos elementos da lista e a média.

Bibliografia

Aprenda Computação com Python. Disponível em
http://www3.ifrn.edu.br/~jurandy/fdp/doc/aprenda-python/capitulo_08.html