

```

#-----
#   Topicos desta aula:
#   - Objectos compostos: listas e strings (cadeias de caracteres)
#   - Operacoes de: indexacao, 'len', '+', 'append', 'del'
#   - Funcao range
#   - Instrucao 'for'
#-----
#   objectos compostos - strings, listas, tuplos, ... (dics...)
#   listas [...,...]
#   Exemplos:
#       []      [2]      [2,5]  ["ana",5]  [["a",6],8]
#
#   indexacao:  lista[0] - primeiro elemento da lista
#   comprimento: len(lista) (indices de 0 a len(lista)-1
#   +:          concatenar listas
#
#-----
#   Experiencias com indexacao em listas e strings
#-----
#   FUNCAO  membro(a,s): a e' membro da lista s?
#           dominio: ...  contradominio: {False,True}
#   (ja' existe, a in s)

def existe(a,s):
    i=0
    while i<len(s): # 0...len(a)-1
        if s[i]==a:
            return True
        i=i+1
    return False

# print existe(5,[2,5,1,1])          --> True
# print existe("rui",["ana","Rui",66]) --> False
#-----
#
#   funcao range(n) -> [int]
#
# print range(3)
# print range(1,30)
# print range(1,30,5)
#-----
#
#   instrucao for
#
#       for VAR in LISTA:
#           INSTRUCAO
#           INSTRUCAO
#           ...
#           INSTRUCAO
#
#   Ja' conhecemos as instrucoes
#   atribuicao
#   def
#   while

```

```

#         for
#         if / if elif else
#         return
#
#-----
# Exercício: tabelar x, log(x) usando for; x=1,2,...10
#
# (nao resolvido - exercicio)
#-----
# Exercício: Escrever a funcao "existe" usando a instrucao for
#

def existe(a,s):
    for x in s:
        if x==a:
            return True
    return False

# print existe(2,[3,[2],5])      -> False
# print existe("rui",[3,"rui"])  -> True

#-----
# Exercício: inters(x,y): retorna a lista interseccao...
# Ex:         inters([5,1,2],[8,2,9,5]) --> [5,2]
#
# Em vez de usar a funcao 'existe' vamos usar 'in' (que e' equivalente)
# 2 in [5,2,1] -> True
# 3 in []      -> False

def inters(a,b):
    r=[]
    for x in a:
        if x in b:
            r = r+[x]
    return r

# inters([5,1,3],[8,3,9,5]) -> [5,3]
#-----
# Exercício: li: lista do tipo [[nome,numero],...,[nome,numero]]
#           x: nome
#           funcao procura(nome,li)
#           devolve o numero correspondente a "nome" ou "nao existe"
# Ex: procura("ana",[["quim","00"],[],[]])

#-----
# Strings
# Sao imutaveis, experimentar...
#
# Exercício: escreva maiusc(s) que, dado um string s, retorna
# o string obtido quando se troca cada minuscula por maiuscula
# Ex: maiusc("Ai, o gato!") -> "AI, O GATO!"

```

```

#

#Experiencias com 'char' e 'ord'
# for x in ["a","b","c"]:
#     print x, ord(x)      -> a 97      b 98      c 99

# for x in ["A","B","C"]:
#     print x, ord(x)      -> a 65      b 66      c 67
#
def maiusc(s):
    r=""
    delta=ord("a")-ord("A")
    n=len(s)
    for x in s:
        if x>="a" and x<="z":
            r = r + chr(ord(x)-delta)
        else:
            r = r+x
    return r

print maiusc("Ai, o gato!")

#-----
# Exercicio: li: lista do tipo [[nome,numero],..., [nome,numero]]
#             x: nome
#             funcao procura(nome,li)
#             devolve o numero correspondente a "nome" ou "nao existe"
# Ex: procura("ana", [ ["quim", "00"], [], [] ])

def procura(nome,li):
    for x in li:
        if nome==x[0]:
            return x[1]
    return "nao existe"

print procura("ana", [ ["quim", "00999"], ["ana", "936666"], ["Ze", "555"] ])

```