```python
# -*- coding: utf-8 -*-
"""Resumo_Python.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1q-KABSPPqfTepcEFHeTE6qp_aBKU
"""

print ('Olá Mundo')

# print ()-> Saída de dados

print ('Olá Mundo')

print ('Curso de \n Python') #\n - Quebra de linha

print  ('Curso de \t Python') # \t - Tabulação

print ('Curso ', end='') # end='' - Retira a quebra de linha
print ('Python')

# cpf
print ('556.789.587',end='-')
print ('70')

# Variáveis

nome = 'Lucas'
idade = 14

#print ('Nome: Cosmo - Idade:35')
#print ('Nome: Cosmo - Idade:35')
#print ('Nome: Cosmo - Idade:35')

print (f'Nome: {nome} - Idade:{idade}')
print (f'Nome: {nome} - Idade:{idade}')

# Tipos de dados
# str -> strings - textos
# int -> Números inteiros
# float -> Números decimais
```

```python
# bool -> Expressão booleana - True ou False

texto = 'Python'
inteiro = 10 # 5 , 8, -5
decimal = 10.5 # 5.5 , -8.5
booleano = True # ou False

# type() -> Retorna o tipo de dado
print (texto ,end=' ')
print (type(texto))

print (inteiro,end=' ')
print (type(inteiro))

print (decimal,end=' ')
print (type(decimal))

print (booleano,end=' ')
print (type(booleano))

# Operadores Aritiméticos
# Adição -> +
# Subtração -> -
# Divisão -> /
# Multiplicação -> *
# Módulo -> %  - ou resto da divisão
# Divisão inteira -> //
# Potenciação -> **

x = 10
y = 3

print (x + y)

print (x - y)

print (x / y)

print (x * y)

print (x % y)

print (x // y)
```

```python
print (x ** y)

# Manipulação de strings
# len - Retorna o tamanho da string
# capitalize - Retorna a string com a primeira letra maiúsculas
# upper -  Retorna uma cópia da string com as letras maiúsculas
# lower - Retorna uma cópia da string com a letras minúsculas
# title - Retorna uma cópia da string com as primeiras letras de cada pal
# strip - Remove os espaços em branco da strings
# cout - Retorna quantas vezes um caractere aparece na string

#        0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
#        c u r s o   d e   p  y  t  h  o  n
curso = 'curso de python'
print (curso)

print (len(curso))

# Fatiamento

print (curso[0:5])
print (curso [9:15])
print (curso [-1])



print(curso.capitalize())

print (curso.upper())

print (curso.lower())

print (curso.title())

col = '   Colab  '
print (col)
print (col.strip())

fruta = 'Bananada de banana'
print (fruta.count('a'))

# Formatação de string
```

```python
nome = 'Cosmo'
idade = 35

print ('Nome:',nome,'idade:',idade)
print ('Nome:{} Idade:{}'.format(nome,idade))
print (f'Nome:{nome} Idade:{idade}')

# input () -> Entrada de dados
# a função input - Retorna o tipo str

nome = input('Digite o seu nome: ')
print (type(nome))
print (nome)

idade = int (input ('Digite sua idade: '))
print (idade)

# Operadores de comparação

# Maior -          >
# Menor -          <
# Maior ou igual - >=
# Menor ou igual - <=
# igual -          ==
# Diferente -      !=

num1 = 5
num2 = 4

print (num1 > num2)

print (num1 < num2)

print (num1 >= num2)

print (num1 <= num2)

print (num1 == num2)

print (num1 != num2)

# Operadore Lógicos
```

```python
# not - não é - Inverte o resultado
# and - e     - Retorna True se as duas condições for verdadeiras
# or - ou     - Retorna True se as uma das duas condições for verdadeira

a = 10
b = 8
c = 7

print (a > b and b > c)

print (a > b or b < c)

print (not a > b)

# Estruturas condicionais
# if , elif , else

# if - else - elif
# if -> se
# else -> senão
# elif -> senão se

idade = int (input ('Digite sua idade: '))

if idade >= 18 and idade < 60:
    print ('Maior de idade!')
elif idade < 18:
    print ('Menor idade!')
else:
    print ('Melhor idade!')

# list () -> Listas

# len() ->   Retorna o tamanho da lista
# min() -> Retorna o menor valor
# max() ->  Retorna o maior valor
# append()  ->  Adiciona item no final da lista
# insert() ->  Adiciona em um índice específico
# pop()  Remove o último item da lista ou um índice específico
# del ->  Remove o item pelo índice
# sort() -> Ordena a lista
# reverse() ->  Inverte a lista
```

```python
#          0      1   2   3
lista = ['Python',10,5.5,True]
print(lista)
print (len(lista))
print (lista[0])
print (lista[-1])

lista = [2,4,3,5,7,6,9,8]
print (lista)

print (min(lista))
print (max(lista))

lista.sort()
print (lista)

lista.reverse()
print (lista)

lista [0] = 'Python'
print (lista)

lista.append('Java')
print (lista)

lista.insert(1,'Django')
print (lista)

lista.pop()
print (lista)

lista.pop(1)
print (lista)

del lista [0]
print (lista)

lista = list(range(0,21))
print (lista)

# tuple () -> Tuplas
# As tuplas são imutáveis , não pode ser alteradas
```

```python
tupla = (8,7,3,4,4,5)
print (tupla)

print (tupla[0])

print (tupla.index(8))

print (tupla.count(4))

# dict -> Dicionários

# keys() # Retorna a chave
# values() # Retorna os valores
# items() # Retorna os itens

cores = {
    'A': 'Verde',
    'B': 'Vermelho',
    'C': 'Amarelo'
    }

print (cores)

print (cores['A'])

cores ['D'] = 'Azul'
print (cores)

print (cores.keys())

print (cores.values())

print (cores.items())

# while -> Enquanto

x = 0
while x < 10:
    print (x)
    x = x + 1

senha = 123
```

```
leitura = 0

while senha != leitura:
    leitura = int (input ('Digite a senha!'))
    if senha == leitura:
         print ('Acesso liberado!')
    else:
        print ('Senha incorreta , tente novamente')

import random

senha = random.randint(0,10)
leitura = 0

while senha != leitura:
    leitura = int (input ('Digite a senha!'))
    if senha == leitura:
         print ('Acesso liberado!')
    else:
        print ('Senha incorreta , tente novamente')

# for -> para

for x in range (0,11):
    print (x)

lista = [1,2,3,4,5,6,7,8,9]
for x in lista:
    print (x)

curso = 'Curso de Python'
for letra in curso:
    print (letra)

senha = 123

for leitura in range( senha):
    leitura = int (input ('Digite a senha!'))
    if senha == leitura:
         print ('Acesso liberado!')
         break
    else:
        print ('Senha incorreta , tente novamente')
```

```python
# def () -> Funções

def mensagem():
    print ('Bom dia!')

mensagem()

def mensagem(msg):
    return msg

print (mensagem('Boa noite!'))

def soma(x,y):
    return x + y

print(soma(4,7))

def login():
    senha = 123

    for leitura in range( senha):
        leitura = int (input ('Digite a senha!'))
        if senha == leitura:
            print ('Acesso liberado!')
            break
        else:
            print ('Senha incorreta , tente novamente')

login()

# class -> Classes
# Atributos e Métodos

class Pessoa:
#     # Atributos
     nome = ''
     idade = 0

#     # Métodos
     def dados(self):
         return(f'Nome:{self.nome} Idade:{self.idade}')
```

```python
# objeto
p1 = Pessoa()
p1.nome = 'Cosmo'
p1.idade = 35
print(p1.nome,p1.idade)
print(p1.dados())

class Pessoa:
    # Método construtor ou inicial
    def __init__(self,valor_nome,valor_idade):
        self.nome = valor_nome
        self.idade = valor_idade

    def dados (self):
        return (f'Nome:{self.nome} Idade:{self.idade}')

    # get
    def get_nome(self):
        return self.nome

    def get_idade(self):
        return self.idade

    # set
    def set_nome (self,novo_nome):
        self.nome = novo_nome

    def set_idade(self,nova_idade):
        self.nome = nova_idade



p2 = Pessoa('Cosmo',35)
print (p2.dados())
print (p2.get_nome())
print (p2.set_nome('Lucas'))
print(p2.dados())

# Herança

class Animal():
    def __init__(self, nome, cor):
```

```
        self.nome = nome
        self.cor = cor

    def comer(self):
        print(f"O {self.nome} está comendo")


class Gato(Animal):
    def __init__(self, nome, cor):
        super().__init__(nome, cor)

class Cachorro(Animal):
    def __init__(self, nome, cor):
        super().__init__(nome, cor)


class Coelho(Animal):
    def __init__(self, nome, cor):
        super().__init__(nome, cor)

gato = Gato("Bichano", "Branco")
cachorro = Cachorro("Totó", "Preto")
coelho = Coelho("Pernalonga", "Cinza")

gato.comer()
cachorro.comer()
coelho.comer()
```