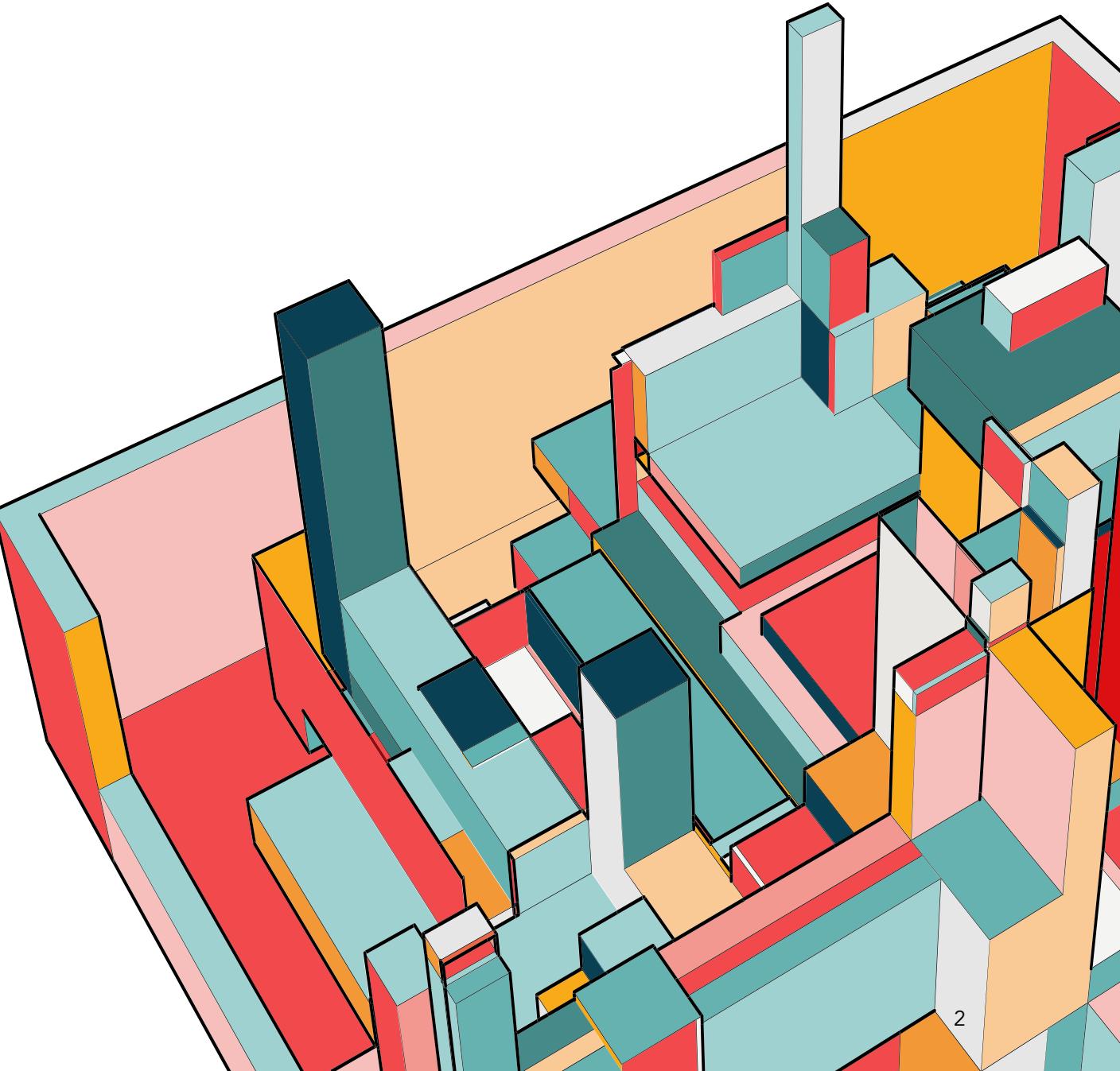


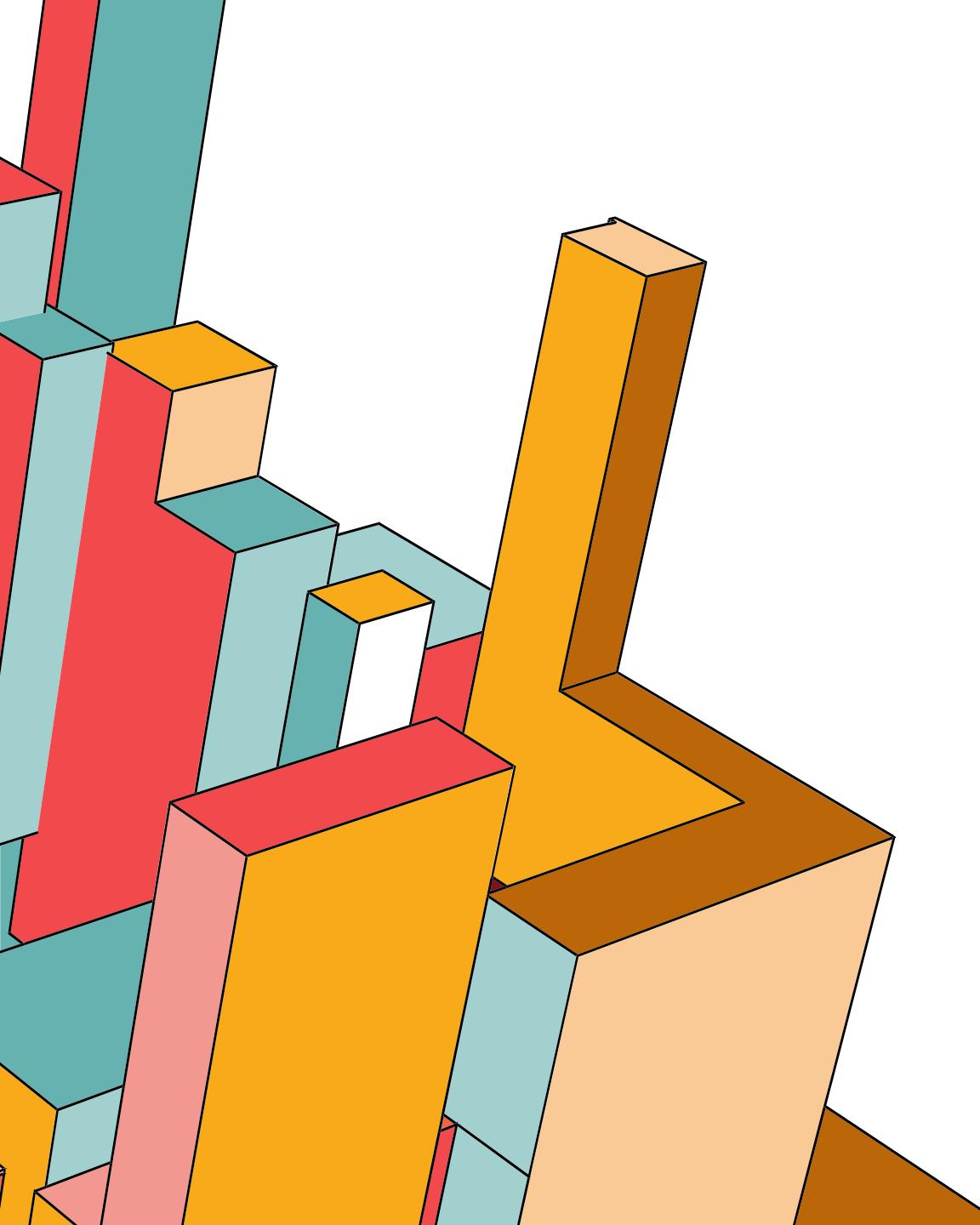
PROGRAMAÇÃO EM PYTHON



SOBRE O CURSO

O curso de Aperfeiçoamento Profissional em Programação em Python tem como objetivo desenvolver as competências relacionadas a estruturas bases da linguagem de programação Python, permitindo a criação de programas para a solução de problemas e scripts de automação.





CONTEÚDO PROGRAMÁTICO

1. INTRODUÇÃO AO PYTHON

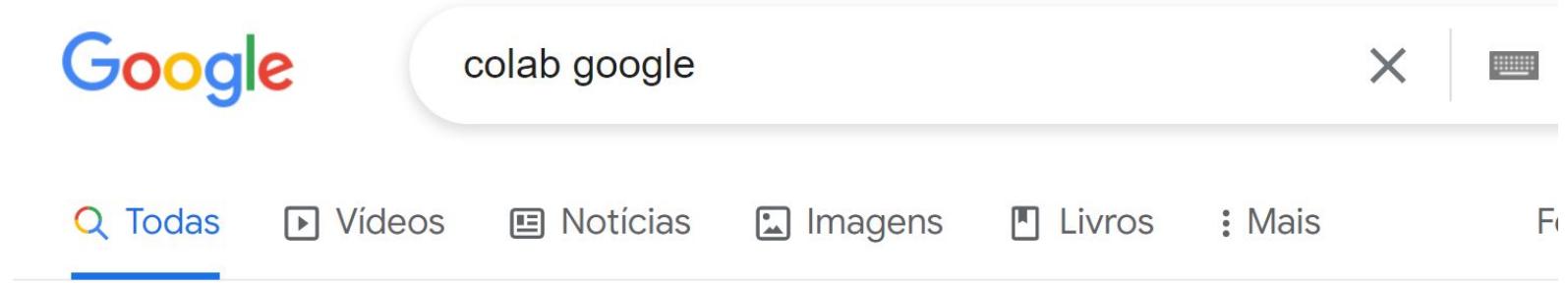
- Histórico geral
- Instalação dos aplicativos
- Ambientação

2. Conceitos de Programação em Python

- Variáveis, operadores e operações de E/S
- Listas, estruturas de decisão e laços
- Funções, tuplas e dicionários
- Módulos e pacotes
- Exceções e strings
- Geradores e iteradores
- Gerenciamento de arquivos
- Módulos diversificados.

3. Introdução a programação orientada a objetos em Python

VAMOS PARA A PRÁTICA



A screenshot of a Google search results page. The search bar at the top contains the query "colab google". Below the search bar, there are several navigation links: "Todas" (selected), "Vídeos", "Notícias", "Imagens", "Livros", and "Mais". The main content area displays the search results, starting with a snippet from the "Welcome To Colaboratory - Google Research" page, which includes a link to "https://research.google.com › colab...".

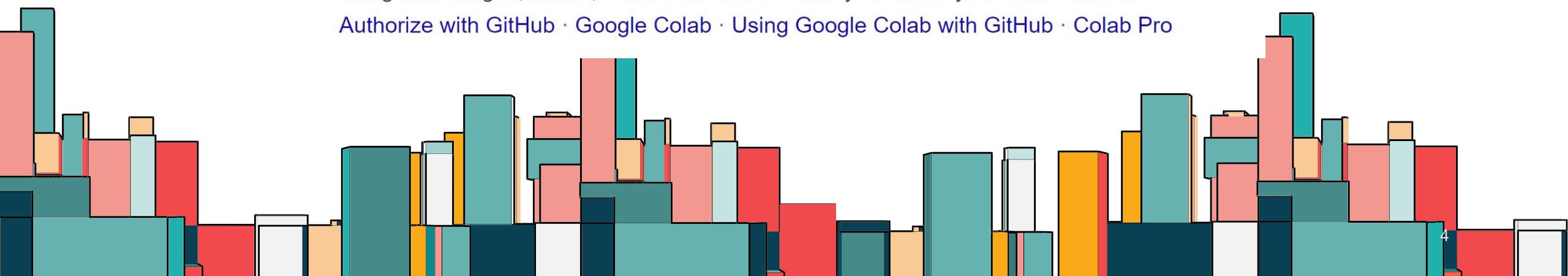
Aproximadamente 30.100.000 resultados (0,36 segundos)

<https://research.google.com › colab...> ▾ Traduzir esta página

Welcome To Colaboratory - Google Research

Colab notebooks allow you to combine executable code and rich text in a single document, along with images, HTML, LaTeX and more. When you create your own Colab ...

[Authorize with GitHub](#) · [Google Colab](#) · [Using Google Colab with GitHub](#) · [Colab Pro](#)





Olá, este é o Colaboratory

Arquivo Editar Ver Inserir Ambiente de execução Fechar

Novo notebook

Abrir notebook

Fazer upload de notebook

Rename

Salvar uma cópia no Drive

Salvar uma cópia como Gist do GitHub

Salvar uma cópia no GitHub

Salvar

Histórico de revisões

Fazer download

Imprimir

Ctrl+O

Ctrl+S

Ctrl+P

TIPOS DE DADOS

Dados são as informações a serem processadas por um computador

Há três principais tipos de dados primitivos

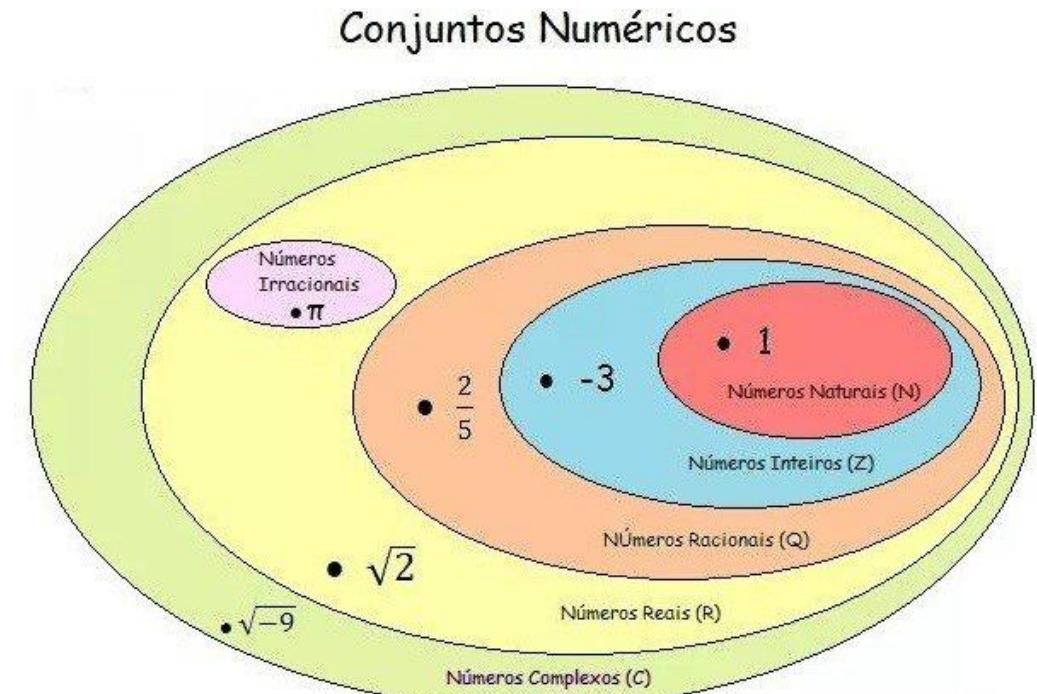
- Numéricos
- Caracteres
- Lógicos

TIPO NUMÉRICO

Dentro do aspecto computacional, os dados numéricos representáveis em um computador são divididos em apenas duas classes: os INTEIROS e os REAIS.

Inteiro (int): números inteiros, positivos e negativos

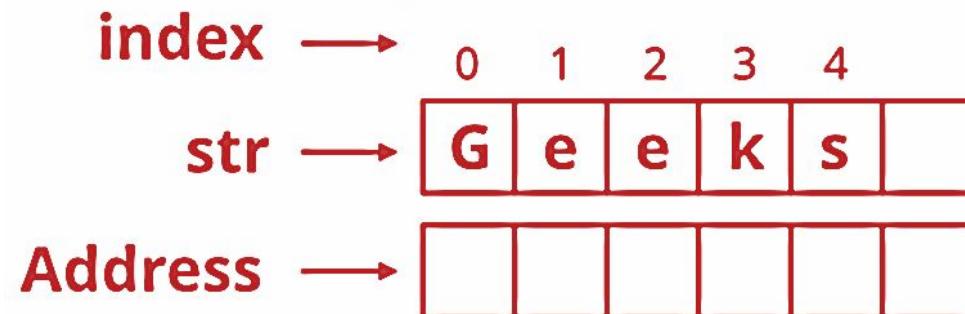
Real (float): números fracionários, positivos e negativos



TIPO CARACTERE (String - str)

Também conhecido como tipo literal

- Se refere a uma sequência que contenham letras, números e símbolos
- Os caracteres devem ser sempre representados com aspas no código
- Pode se falar também em cadeia, string, alfanumérico ou char



TIPO LÓGICO (Booleano - bool)

São dados cujos os valores somente podem assumir

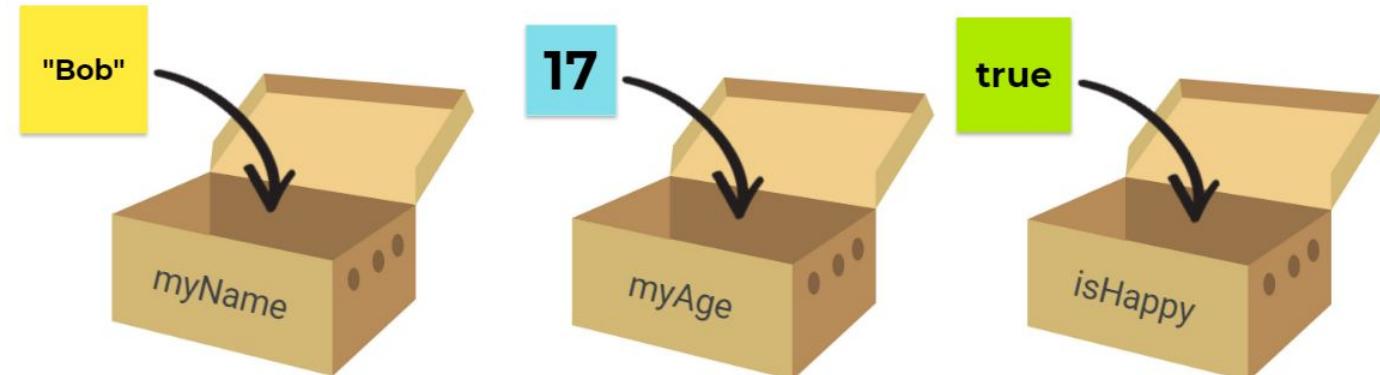
- Verdadeiro (True)
- Falso (False)

Também conhecido como tipo Booleano



VARIÁVEIS

- Uma variável é uma localização na memória RAM do computador que é utilizada para armazenar temporariamente os dados que são utilizados pelo programa
- As variáveis possuem características como:
 - Identificação
 - Endereço
 - Tipo
 - Tamanho
 - Valor



NOMENCLATURA DE VARIÁVEIS

- Um nome de variável é uma sequência de letras ($a \rightarrow z$, $A \rightarrow Z$) e números ($0 \rightarrow 9$), que devem sempre começar com uma letra. Apenas letras comuns são permitidas.
- Letras acentuadas, cedilhas, espaços, caracteres especiais como \$, #, @, etc. são proibidos, exceto para o caractere _ (sublinhado/underline).
- Além dessa regra é importante também estar atento às palavras reservadas da linguagem

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

CONSTANTES

É uma posição na memória cujo o valor não muda durante a execução do programa;

Por exemplo, o valor Pi é uma constante, pois possui sempre o mesmo valor (3,1415...)

CONSTANTES FÍSICAS

Constante física	Símbolo	Valor
Aceleração da gravidade na superfície terrestre	g	9,80 m/s ²
Constante eletrostática do vácuo	k_0	9 . 10 ⁹ N.m ² /C ²
Carga elementar	e	1,6 . 10 ⁻¹⁹ C
Elétron-volt	eV	1,6 . 10 ⁻¹⁹ J
Massa do elétron	m_e	9,1 . 10 ⁻³¹ kg
Massa do próton	m_p	1,67 . 10 ⁻²⁷ kg
Permeabilidade magnética do vácuo	μ_0	4π . 10 ⁻⁷ T. m/A
Permitividade elétrica do vácuo	ϵ_0	8,8 . 10 ⁻¹² F/m
Velocidade da luz no vácuo	c	3 . 10 ⁸ m/s
Constante de Plank	h	6,63 . 10 ⁻³⁴ J.s ou 4,14 . 10 ⁻¹⁵ eV.s

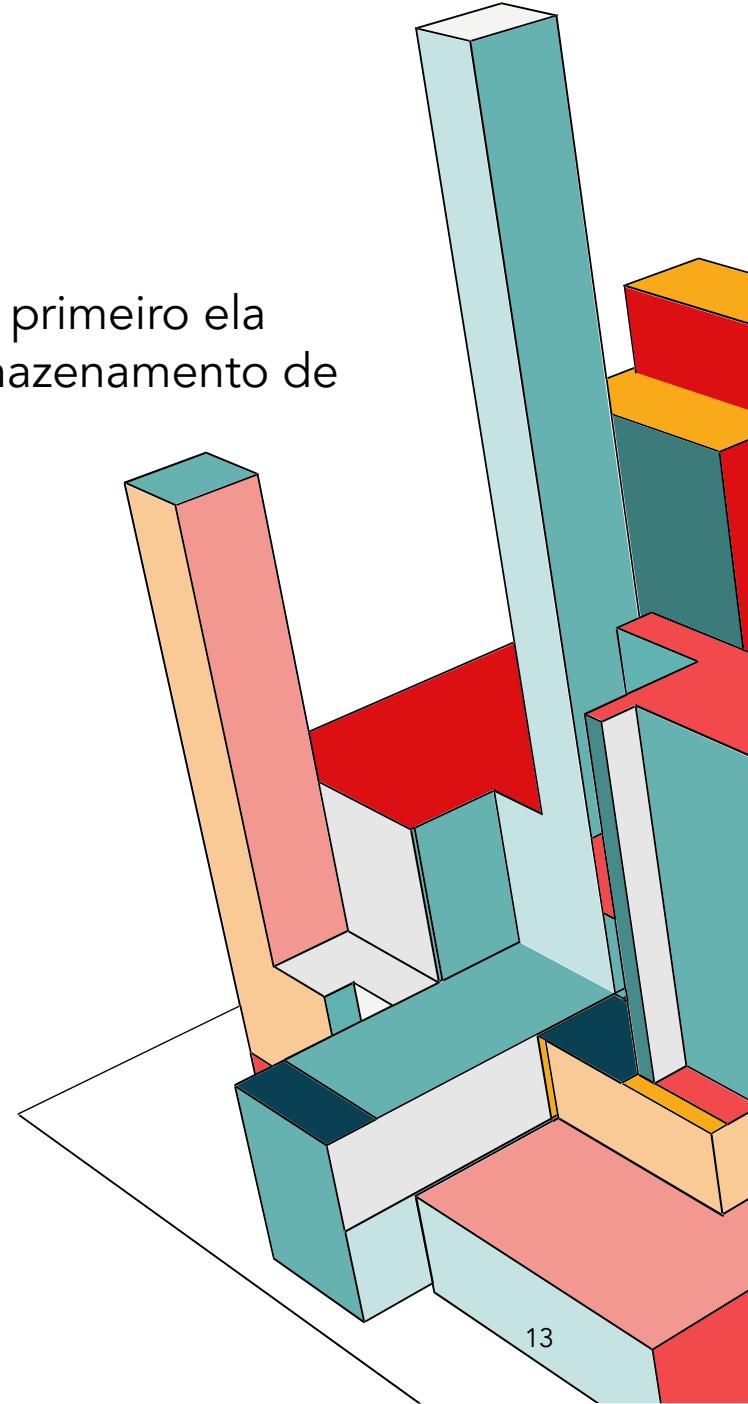
DECLARAÇÃO DE VARIÁVEIS

Para que uma variável ou uma constante possa ser utilizada pelo programa, primeiro ela deve ser declarada, para que seja reservado o espaço na memória para armazenamento de seus dados

```
texto="2" #declaração de variável string
texto1= str(2); #declaração de variável do tipo string
numero=3 #declaração de variável numerica, podendo ser Inteira ou Real
numero1= int(3) #declaração de variável numerica, do tipo inteiro
numero2= float(3) #declaração de variável numerica, do tipo Real
print(texto)
print(texto1)
print(numero)
print(numero1)
print(numero2)
```

2
2
3
3
3.0

Seguindo as boas práticas, as variáveis devem ser declaradas no início do programa



ENTRADA DE DADOS

- A função `input` possui um parâmetro do tipo `string` que é a mensagem a ser exibida ao usuário e retorna também uma `string` que é a resposta fornecida pelo usuário.

```
nome = input("Digite seu nome: ")
print("seu nome é: ", nome)
```

Digite seu nome: Fernando
seu nome é: Fernando

CONVERSÃO DE DADOS

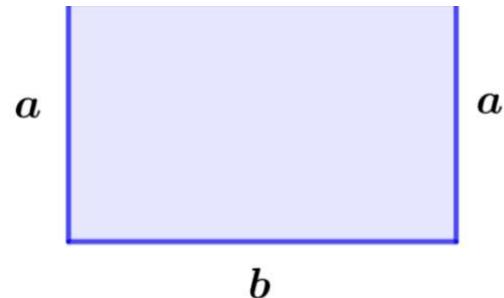
- Para converter o valor informado pelo usuário em um inteiro usamos a função int.
- Para converter em um valor com decimais usamos a função float.

```
valor_uni = float(input("Informe o valor unitário: "))
quantidade = int(input("Informe a quantidade de produtos: "))
print("O total da compra é: ", valor_uni*quantidade)
```

```
Informe o valor unitário: 3.54
Informe a quantidade de produtos: 4
O total da compra é: 14.16
```

VAMOS PRATICAR

1. Crie um programa para efetuar a leitura de um número inteiro e apresentar o resultado do quadrado deste número.
2. Escreva um programa que leia dois caracteres e imprima-os na tela da seguinte forma: "O usuário digitou {caractere1} e {caractere2}!".
3. Crie um programa que leia um número inteiro e imprimir seu sucessor e seu antecessor.
4. Crie um programa para entrar com a base e com a altura de um retângulo e imprimir respectivamente o perímetro e a área correspondente.



$$A = ab \quad p = 2(a + b)$$

5. Crie um programa que, dados o valor, a taxa e o tempo (meses), efetuar o cálculo do valor de uma prestação em atraso, utilizando a fórmula:

prestação = valor + (valor * (taxa/100) * tempo)