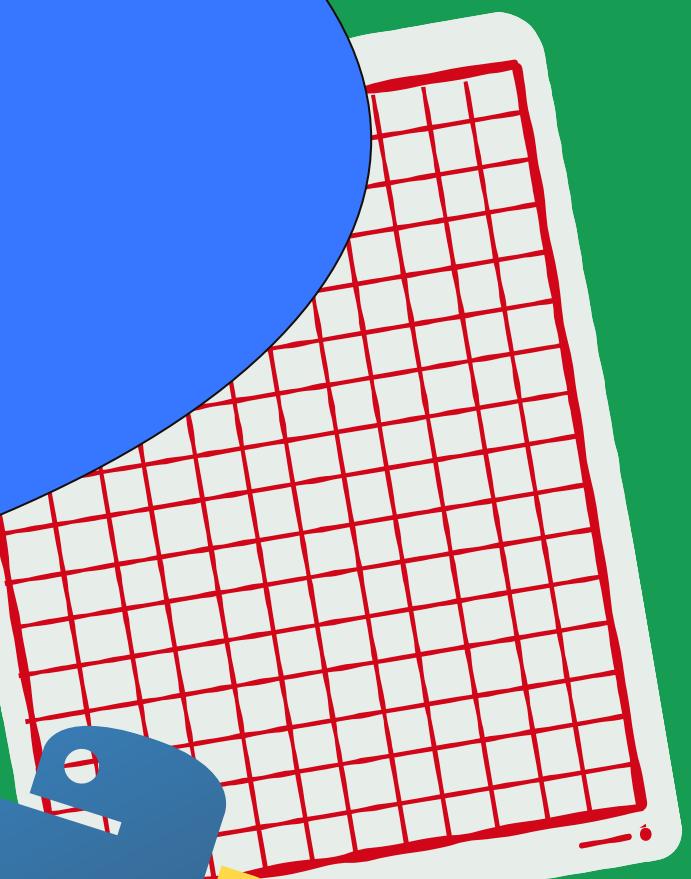
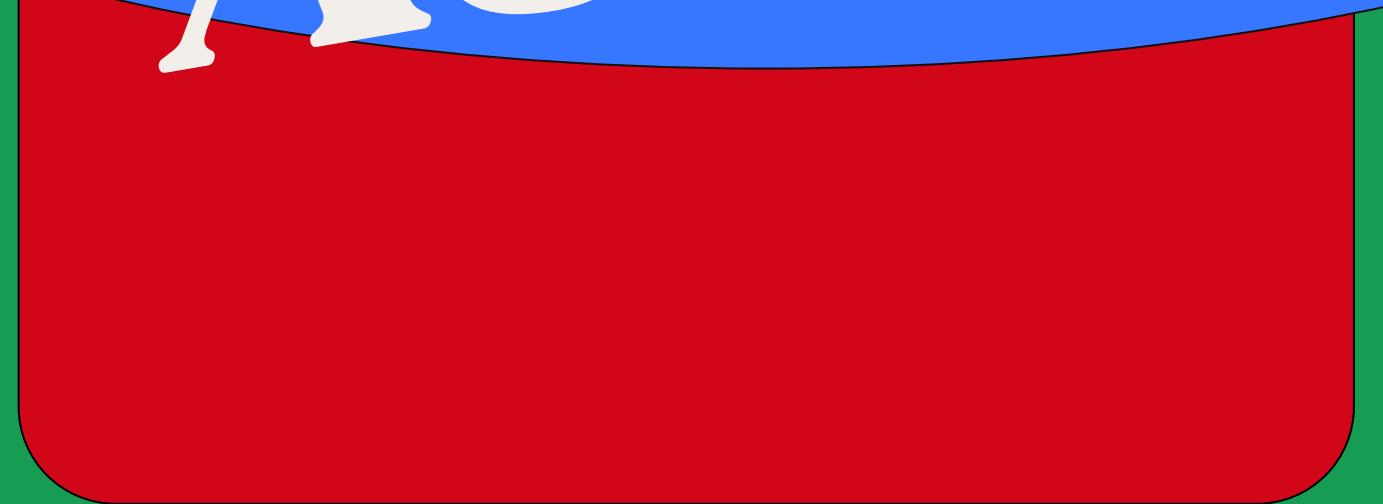


# PYTHON

## AULA 2





## AGENDA

- Hoje veremos dois pilares da programação: **variáveis** e **entrada e saída de dados**.

# VARIÁVEIS: Recapitulando...

Na última aula nós aprendemos que, quando um programa está rodando, ele precisa se lembrar de algumas informações. Para isso usamos as variáveis.

Semana passada, usamos a analogia da caixa com uma etiqueta para ilustrar esse conceito.

idade

13

# VARIÁVEIS: Dentro do computador

Quando criamos uma variável em um programa, o computador precisa guardar esse valor em sua memória.

A memória do computador é como uma grande estante cheia de espaços numerados. Cada espaço pode guardar uma informação.



# VARIÁVEIS: Dentro do computador

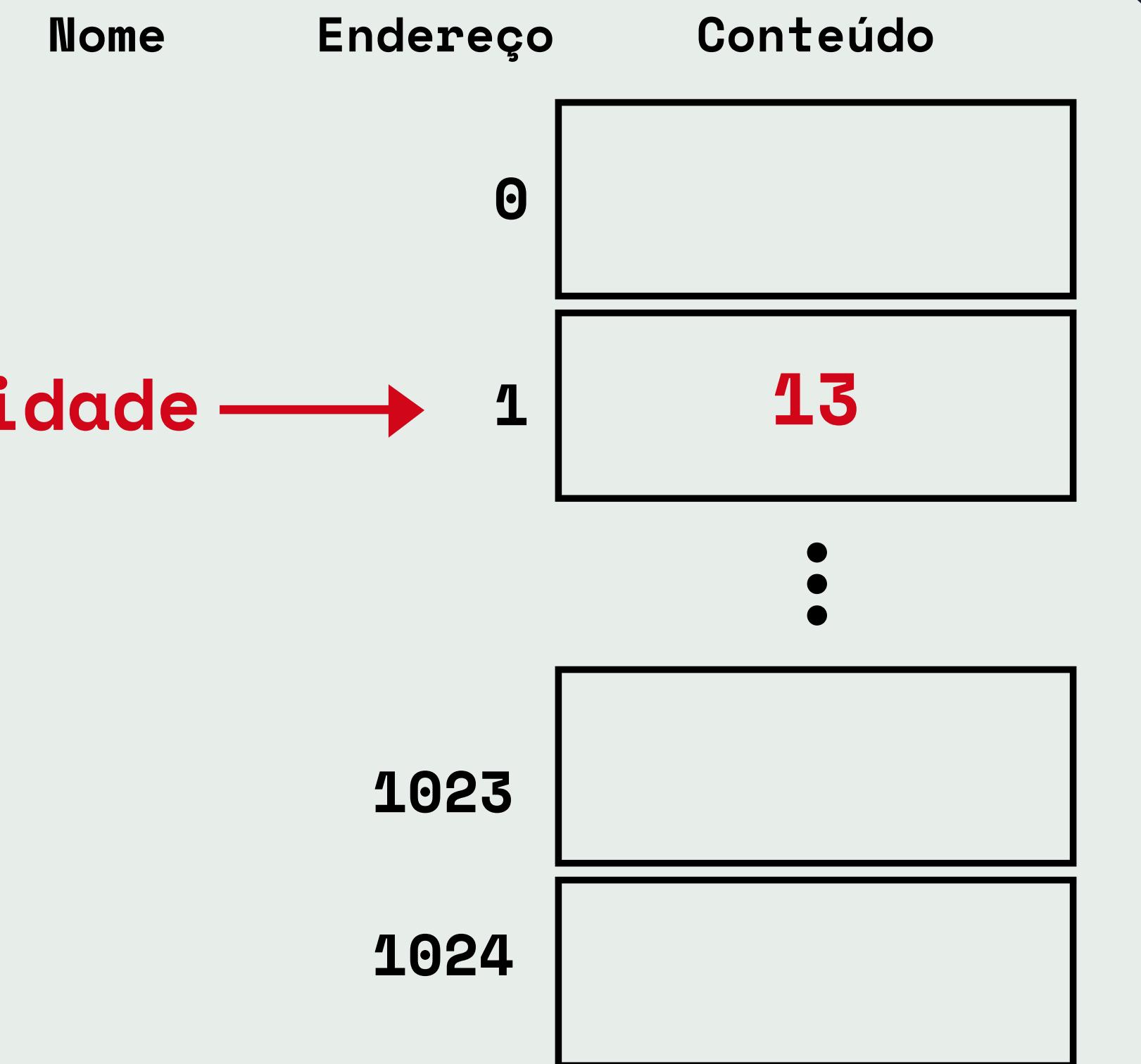
Por exemplo, quando escrevemos:

**idade = 13**

O computador reserva um espaço nessa estante para guardar o número 13.

Depois, ele anota que o nome “idade” está ligado àquele espaço.

Assim, sempre que o programa precisar usar a variável “idade”, o computador vai direto ao espaço reservado e pega o valor que está lá dentro.



# VARIÁVEIS: Tipos

As variáveis podem guardar diferentes tipos de informações

Por exemplo:

`idade = 13`

`altura = 1.65`

`nome = "Maria"`

`chovendo = True`



## Tipos numéricos:

- **int**: Representam números inteiros
  - `idade = 13`
- **float**: Representam números com casas decimais
  - `altura = 1.65`

# VARIÁVEIS: Tipos

As variáveis podem guardar diferentes tipos de informações

Por exemplo:

idade = 13

altura = 1.65

nome = "Maria"

chovendo = True



Tipo de texto:

- **str:** Representa sequências de caracteres. São declaradas entre aspas simples ou duplas

- nome = "Maria" ✓
- nome = Maria ✗

# VARIÁVEIS: Tipos

As variáveis podem guardar diferentes tipos de informações

Por exemplo:

```
idade = 13
```

```
altura = 1.65
```

```
nome = "Maria"
```

```
chovendo = True
```

Tipo booleano:

- **bool**: Representa valores lógicos, podendo ser **True** (verdadeiro) ou **False** (falso).

# VARIÁVEIS: Regras

Como vimos anteriormente, quando criamos uma variável, precisamos escolher um nome para ela. Esse nome deve seguir algumas regras.

- 1) Não podem começar com números
  - (`idade2` , `2idade` ).
- 2) Não podem conter espaços
  - (`minha_idade` , `minha idade` ).
- 3) Só podem usar letras, números e o caractere de sublinhado (underline: `_`)
  - (`aula_2` ).
- 4) Não podem ter acentos ou símbolos especiais (`á`, `!`, `?` etc.).
- 5) Não podem usar palavras reservadas do Python (como `for`, `if`, `while`).

# VARIÁVEIS: Podem mudar de valor

Nós podemos alterar os valores das variáveis ao longo do programa. Lembra das nossas caixinhas com etiquetas? Podemos pegar o que está dentro e colocar um novo valor, sem precisar criar uma caixinha nova!

Por exemplo:

`idade = 13`



Neste momento, o programa usará a variável “idade” com o valor 13.

`idade = 14`



Neste momento, o programa usará a variável “idade” com o valor 14.

# VARIÁVEIS: Podem receber outras variáveis

Variáveis também podem receber o valor de outras variáveis. Nesse caso, o valor de uma variável será copiado para a variável que a recebe, mantendo o valor em ambas.

Por exemplo:

```
sua_idade = 13
```

```
idade_aluno = sua_idade
```



idade\_aluno terá o valor 13.

# VARIÁVEIS: Podem receber outras variáveis

Copiar o valor de uma variável para outra é útil em várias situações.  
Por exemplo:

- Guardar um registro antigo: se queremos salvar o valor atual de uma variável antes de atualizá-la, podemos copiá-la para outra variável e assim manter o histórico.
- Trabalhar com cálculos intermediários: podemos usar uma cópia para fazer operações sem alterar o valor original.
- Organizar dados diferentes: às vezes precisamos que duas variáveis com nomes diferentes guardem o mesmo valor inicial, mas cada uma será usada de forma distinta depois.

# EXERCÍCIO

Diga se cada nome de variável é válido (V) ou inválido (F) em Python:

1. prova2
2. 2a\_lista
3. cor\_favorita
4. mês
5. meuNúmero
6. minha altura
7. nome@usuario

# EXERCÍCIO

Diga se cada nome de variável é válido (V) ou inválido (F) em Python:

1. prova2 **(V)**
2. 2a\_lista **(F)**
3. cor\_favorita **(V)**
4. mês **(F)**
5. meuNúmero **(V)**
6. minha altura **(F)**
7. nome@usuario **(F)**

# EXERCÍCIO

Qual o tipo da variável?

1. dia = 26
2. sobrenome = 'Silva'
3. aprovado = True
4. aprovado = "True"
5. nota = 9.0

# EXERCÍCIO

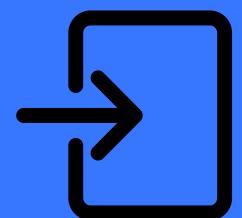
Qual o tipo da variável?

1. dia = 26 **int**
2. sobrenome =  
'Silva' **str**
3. aprovado = True **bool**
4. aprovado = "True" **str**
5. nota = 9.0 **float**

# ENTRADA E SAÍDA

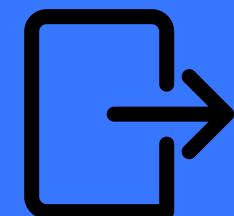
Quando usamos o computador, nós podemos dar informações para ele e também receber informações dele.

Chamamos isso de:



Entrada (**input**)

São as informações que o usuário fornece ao programa, como digitar no teclado, clicar em um botão ou falar no microfone.



Saída (**output**)

São as informações que o programa devolve ao usuário, como mostrar um texto na tela, tocar um som ou exibir uma imagem.

# ENTRADA →

Para os nossos programas, as entradas serão, principalmente, dados passados pelo teclado  
Temos dois tipos de dados de entrada:

- 
- 

Números  
Strings

## Exemplo:

```
nome = input("Qual é o seu nome? ")  
idade = input("Qual sua idade? ")
```

# ENTRADA →

**Exemplo:**

```
idade = input("Qual sua idade?")
```

Opcional!

Em Python, sempre que quisermos receber uma entrada, usamos a função: **input()**

A função **input()** vai retornar tudo aquilo que digitarmos no teclado e armazenar isso na variável “idade”

Dentro do parênteses, podemos escrever uma string, que aparecerá no terminal, ajudando o usuário a saber o que colocar como entrada

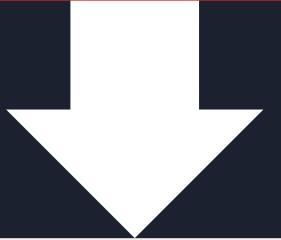
# SAÍDA ➔

- Quando programamos, a saída é o que o computador mostra para a gente depois que o código é executado. É como pedir para o computador falar com a gente

**Exemplo:**

```
print("Olá, mundo!")
```

O computador vai mostrar:



**Terminal**

Olá, mundo!

# SAÍDA ➔

Para programarmos uma saída,  
usamos a função `print()`.

Dentro dos parêntesis, podemos  
colocar strings e variáveis

**Exemplo:**

```
nome = "Manoel"  
print(nome)
```

**Terminal**

Manoel

**Exemplo:**

```
numero = 32  
print(numero)
```

**Terminal**

32

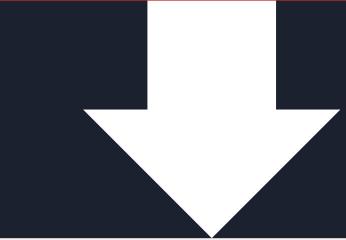
# SAÍDA ➔

Podemos misturar string e variáveis dentro dos parêntesis usando o “+”

Exemplo:

```
idade = 10  
print("Eu tenho" + idade + "anos")
```

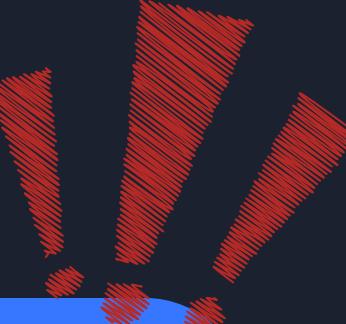
O computador vai mostrar:



Terminal

Eu tenho 10 anos

# SAÍDA ➔



Juntando as entradas com as saídas, podemos ter um programa muito mais poderoso que se encaixe em várias situações

As linhas em azul são as entradas digitadas pelo usuário no terminal

## Exemplo:

```
nome = input("Qual seu nome?")
idade = input("Qual sua idade?")
print("Olá, me chamo" + nome)
print("Tenho" + idade + "anos")
```



## Terminal

```
Qual seu nome?  
Arthur .....  
Qual sua idade?  
18 .....
```

```
Olá, me chamo Arthur  
Tenho 18 anos
```

# EXERCÍCIO

Escreva um programa que receba como entrada uma fruta e imprima a mensagem “Hmm, (fruta) é uma delícia!”

**Saída - Exemplo**

Digite uma fruta  
manga

Hmm, manga é uma delicia!

# EXERCÍCIO

Resposta

```
fruta = input("Digite uma fruta")
print("Hmm, " + fruta + "e uma delicia!")
```

# OBRIGADO!

Contem para a gente o que acharam da aula de hoje:



<https://forms.gle/Q1BYFnKxjyKuCC647>