

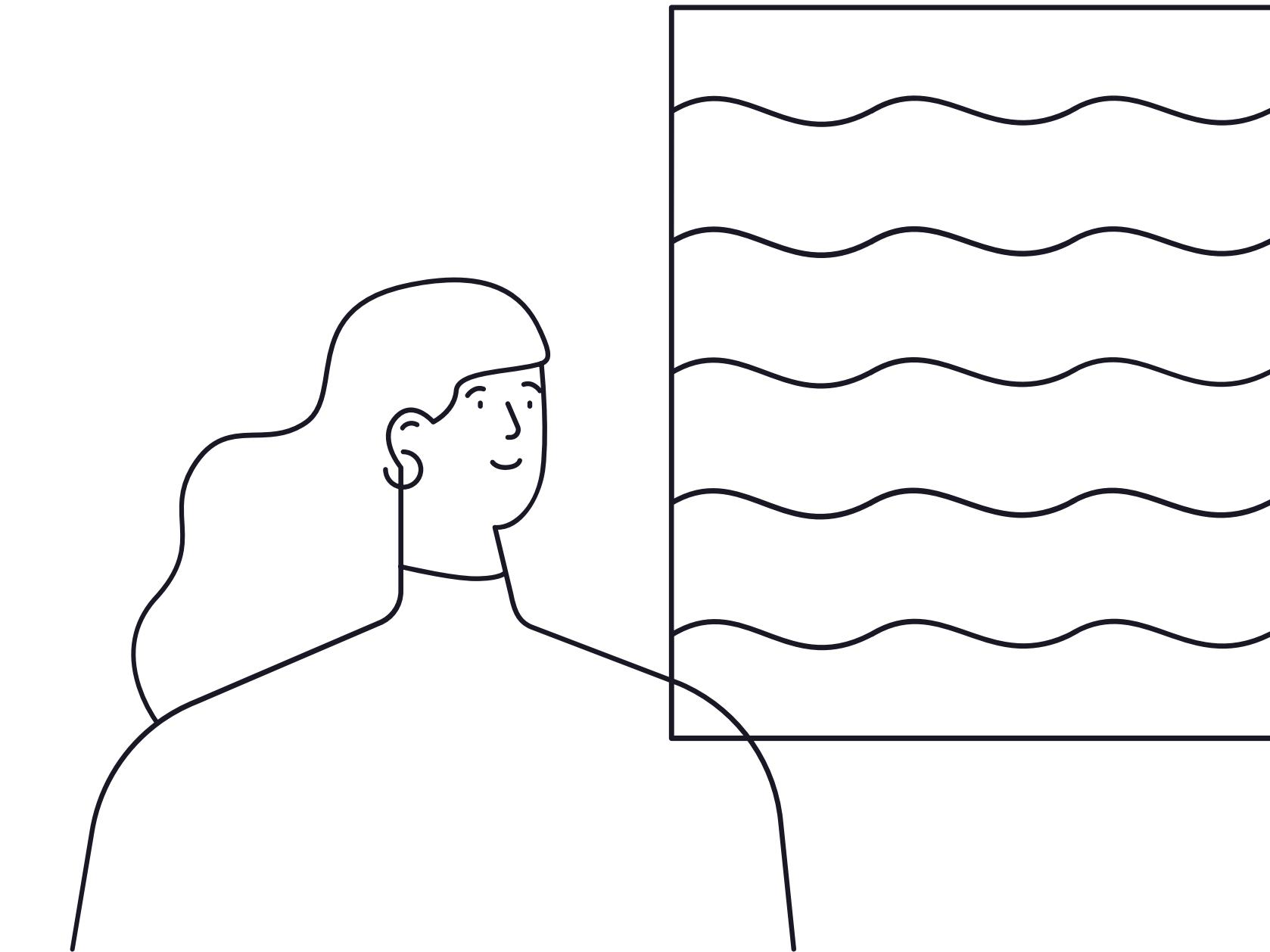
# Revisão - Python



- Tópicos:

02

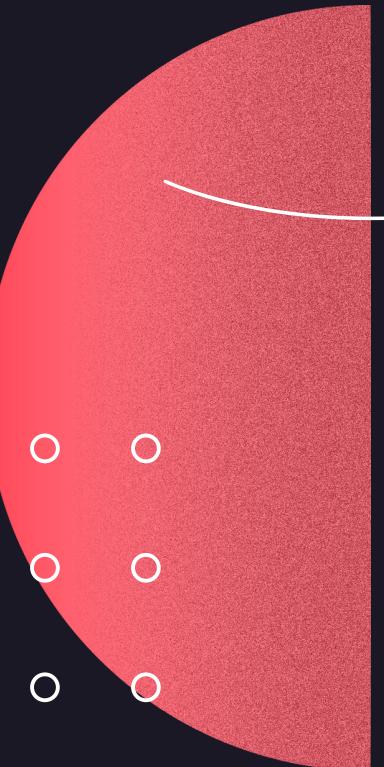
1. Introdução
2. IDE'S
3. Comentários
4. Variáveis e tipos primitivos
5. Operadores
6. Strings e indexação
7. Inputs
8. Boas práticas
9. Questão OBI



# 1. *Introdução*

Que tipo de linguagem é o Python

- Linguagem Interpretada
  - Código fonte > bytecode > interpretador > código de máquina



Preste atenção na indentação!

03

## 2. IDE's



- O que é uma IDE?

do inglês Integrated Development Environment ou Ambiente de Desenvolvimento Integrado, é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo.

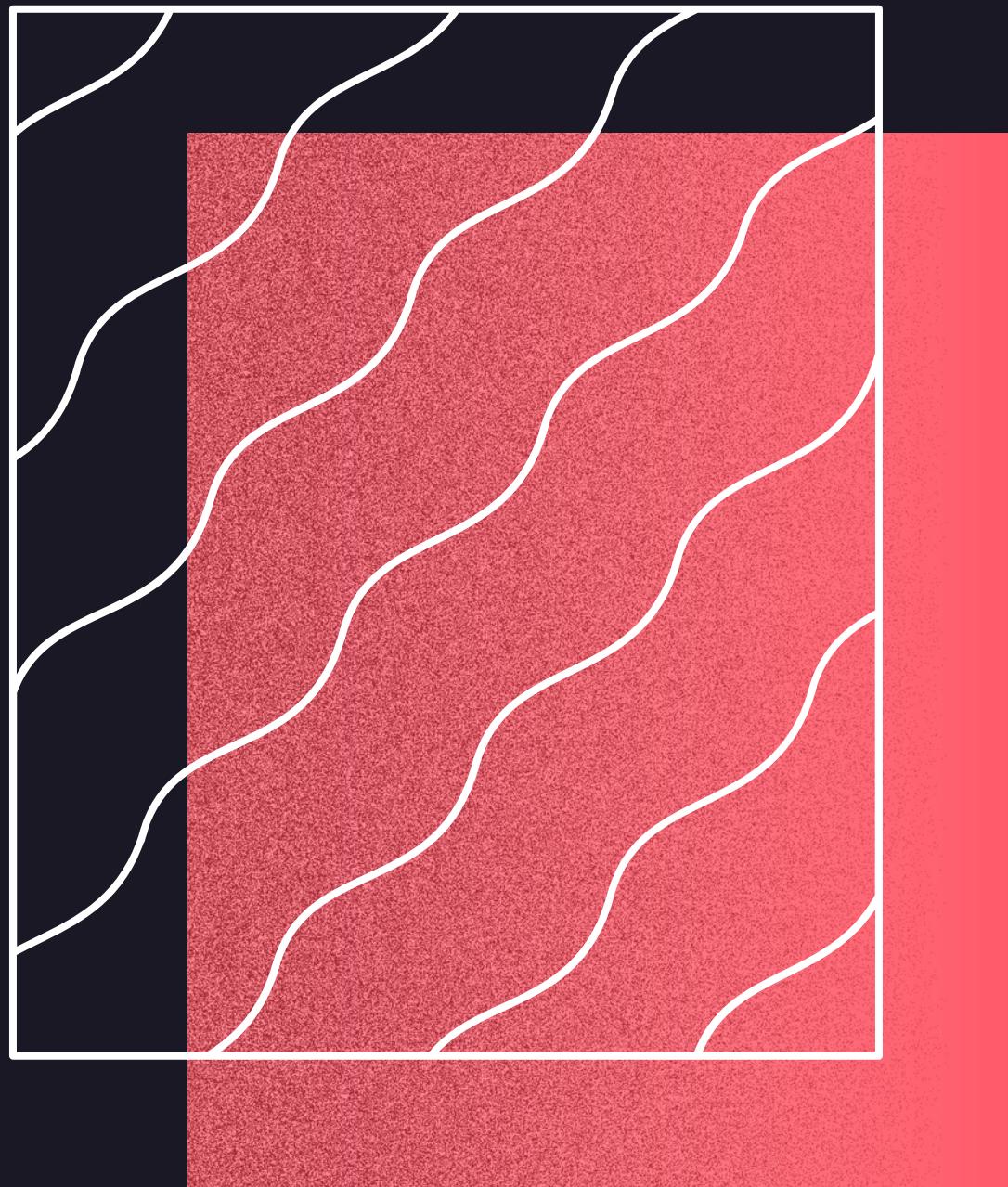
- Tipos de IDE

- Multilinguagem
- Desenvolvimento móvel
- Baseado na nuvem

- Quais iremos utilizar?



### 3. COMENTÁRIOS



O que é comentar algo?



- São importantes para facilitar a compreensão do código, tanto para si próprio, como para quem lerá ele.

1<sup>a</sup> forma:

```
#Isso aqui é um comentário

#Na linha abaixo, foi feita a soma das variáveis 'a' e 'b'
soma = a + b
```

2<sup>a</sup> forma:

```
if (ci <= ca) and (ce >= ca):
    print (ca)
else:
    print (ca)

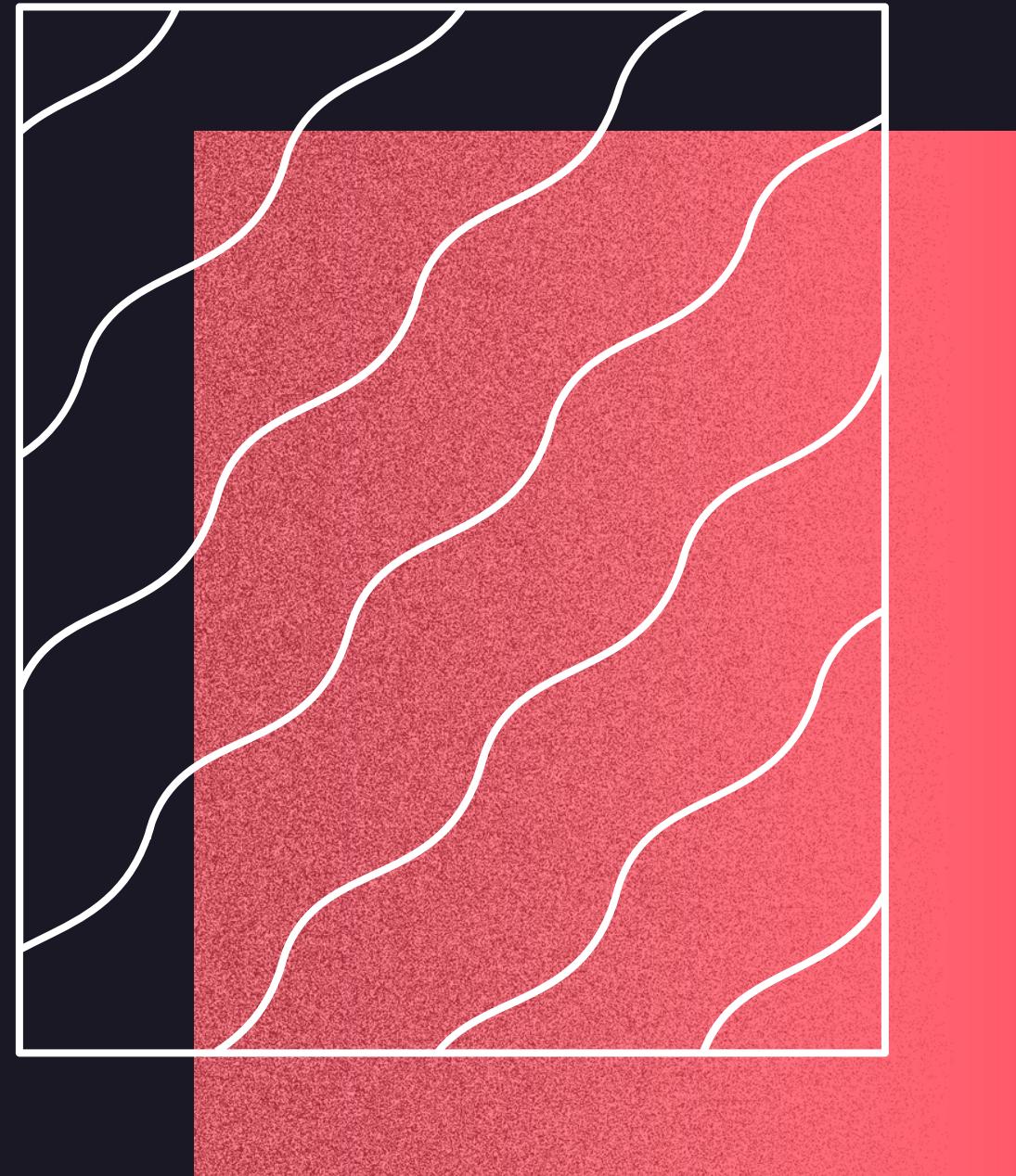
"""
Isso aqui é um comentário.
Pode passar de uma linha, e abranger um determinado bloco de códigos.
Veja no exemplo abaixo:
"""

"""

if (ci <= ca) and (ce >= ca):
    print (ca)
else:
    print (ca)

..."
```

Obs.: Não é obrigatório fazer comentários nas questões da OBI



# 4. VARIÁVEIS E TIPOS PRIMITIVOS DE DADOS

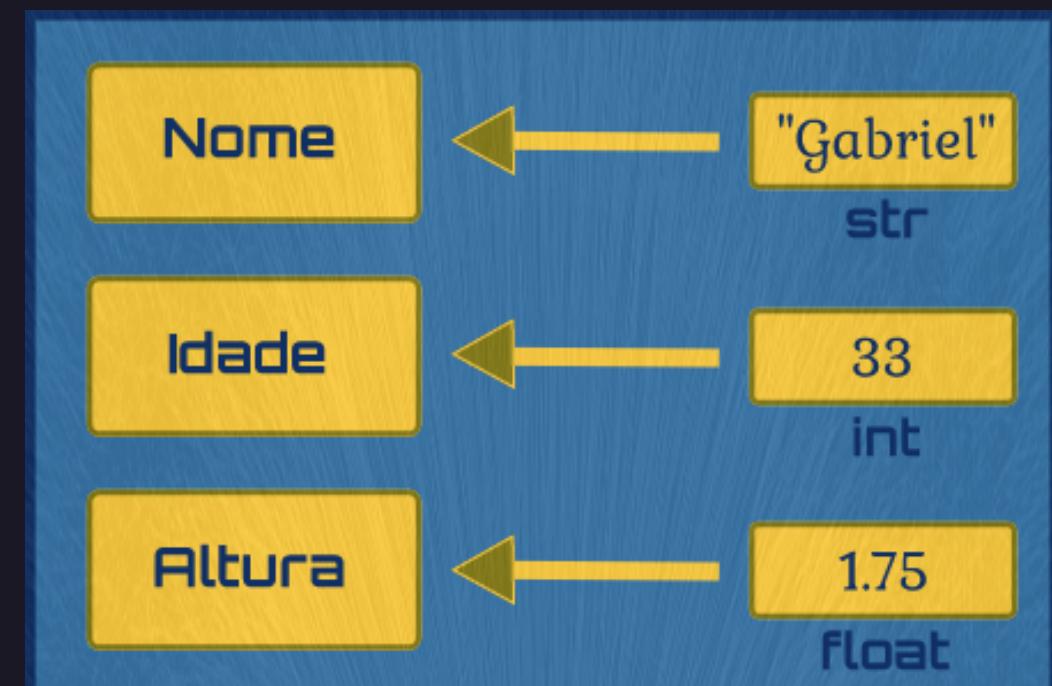
O que são variáveis?

Usadas para armazenar valores que queremos usar posteriormente.

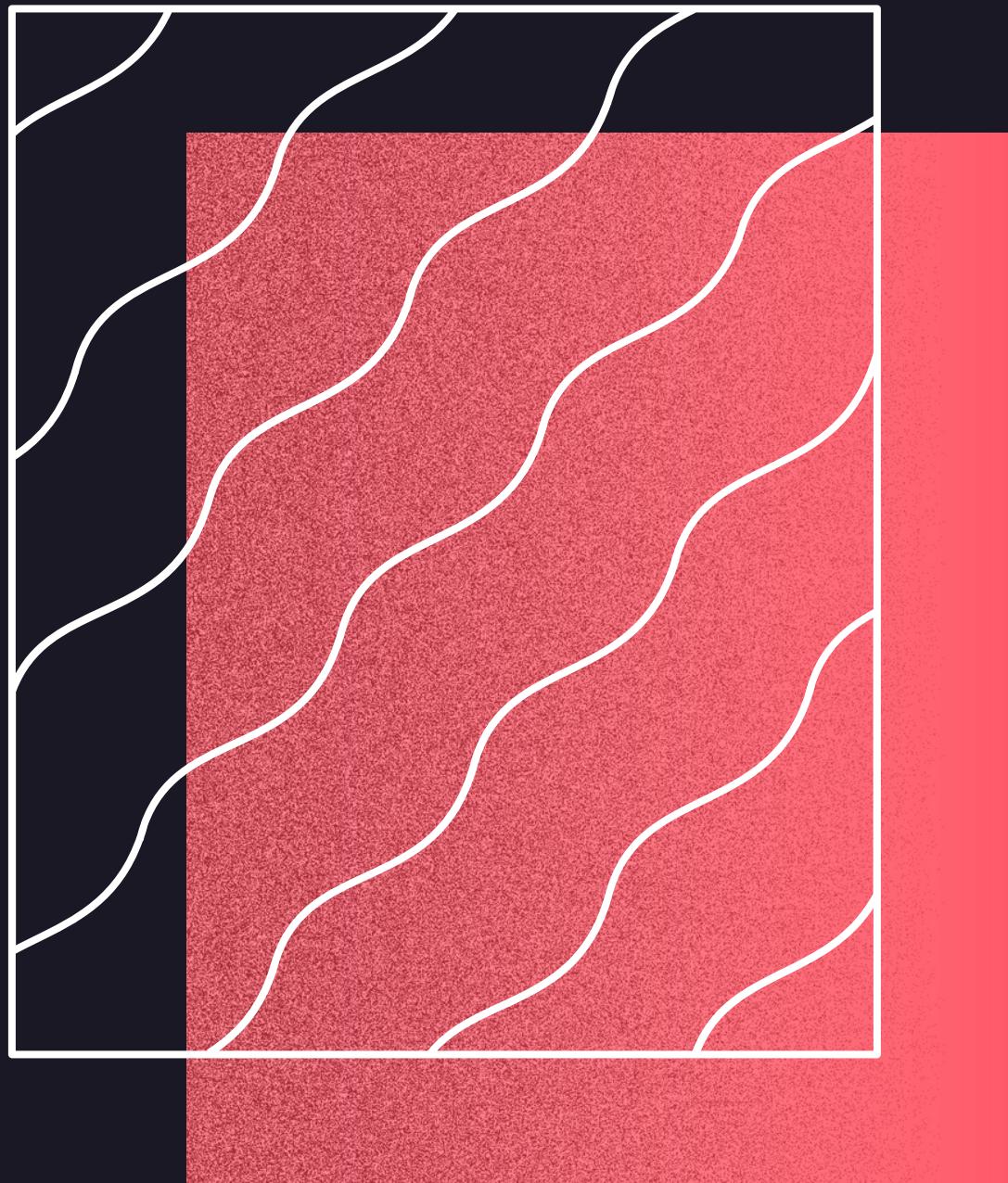
nome = "Gabriel"

idade = 33

altura = 1.75



# Regras de nomeação de variáveis

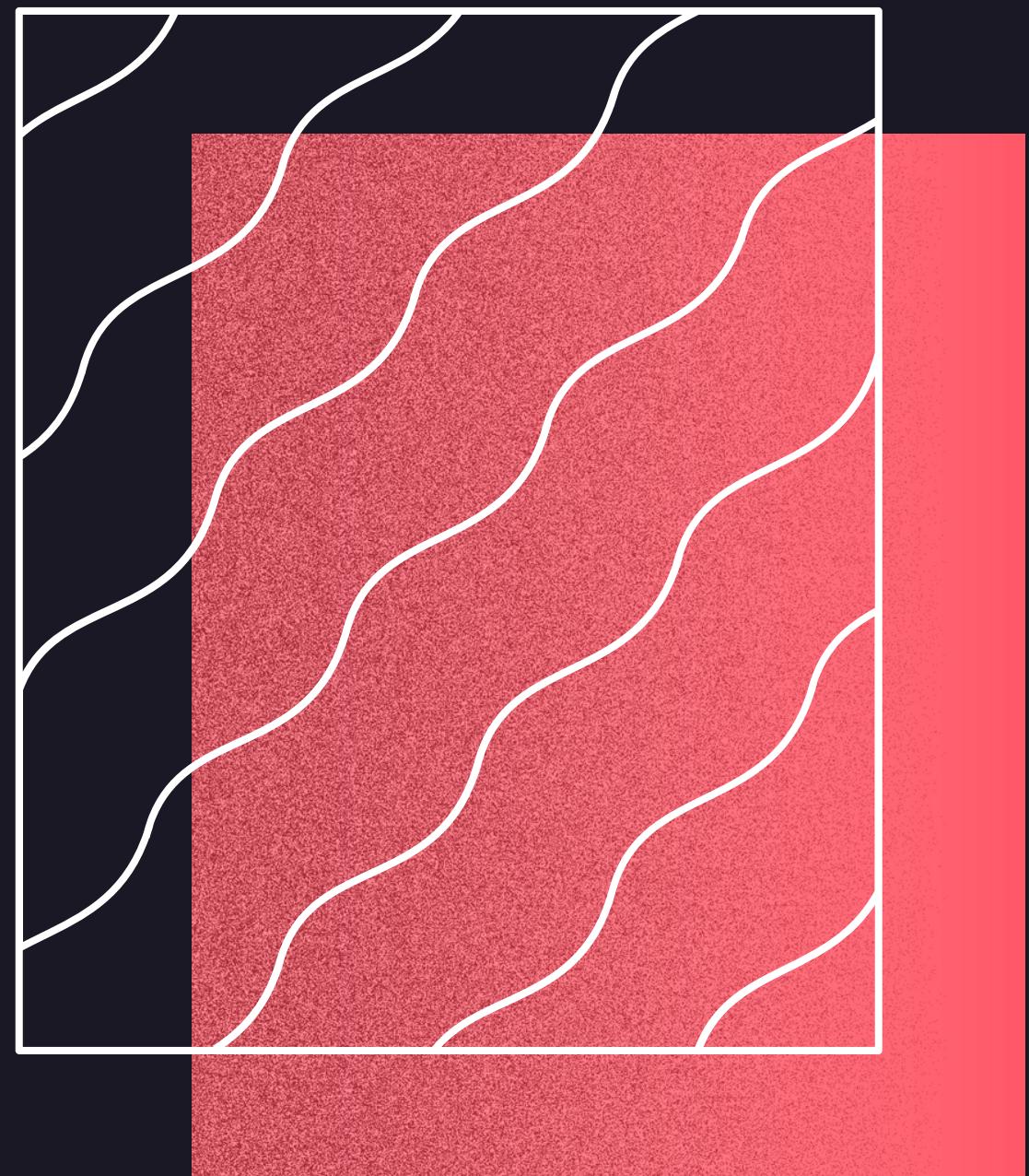


- Não podem começar com número
  - `1idade = 17`
- Não pode haver espaço no nome
  - `Nome pessoa = "Fausto"`
- Não é possível utilizar a maioria dos símbolos especiais (é comum usar underline)
- Não se pode usar palavras reservadas

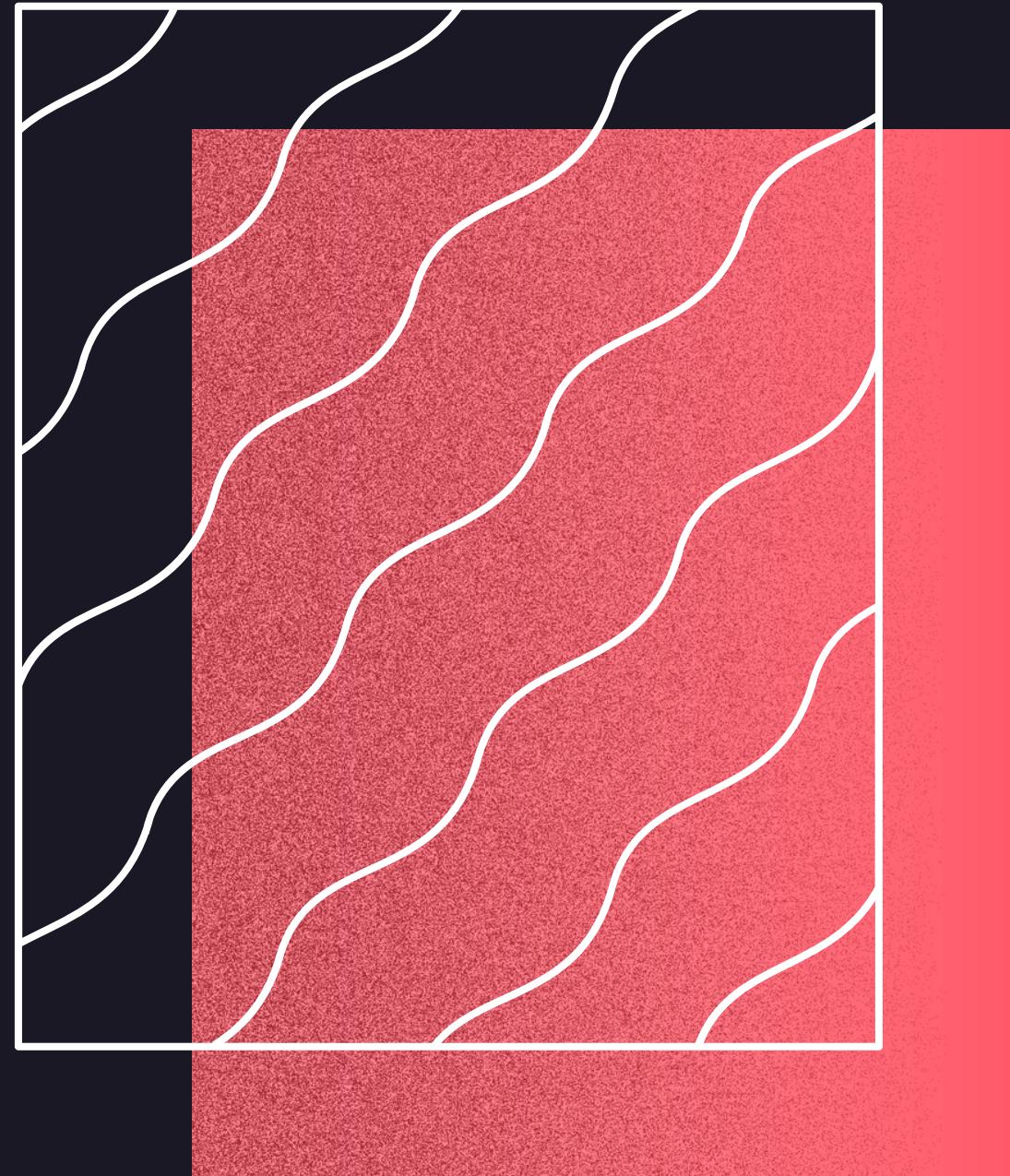
# O que são tipos primitivos?

São maneiras de classificar determinadas informações.

- int
- float
- str
- bool



# 5. OPERADORES



- Aritméticos
- Relacional
- Atribuição

# Operadores Aritméticos

Operador	Significado	Exemplo
<code>+</code>	Soma	$2 + 2 \rightarrow 4$
<code>-</code>	Subtração	$3 - 2 \rightarrow 1$
<code>*</code>	Multiplicação	$2 * 3 \rightarrow 6$
<code>/</code>	Divisão	$10 / 2 \rightarrow 5$
<code>%</code>	Módulo	$5 \% 2 \rightarrow 1$
<code>**</code>	Potência	$4 ** 2 \rightarrow 16$
<code>int()</code>	Converte para inteiro	<code>int(3.2) → 3</code>
<code>float()</code>	Converte para float	<code>float(2) → 2.0</code>

# Operadores Relacionais

Operador	Significado
<code>==</code>	Igualdade / equivalência
<code>!=</code>	Desigualdade / Inequivalência
<code>&gt;</code>	Maior que
<code>&lt;</code>	Menor que
<code>&gt;=</code>	Maior que ou igual a
<code>&lt;=</code>	Menor que ou igual a

# Operadores de Atribuição

Operador	Significado	Exemplo
=	Atribuição	$z = 10$
+=	Soma	$z += 10$ (equivalente a $z = z + 10$ )
-=	Subtração	$z -= 10$ (equivalente a $z = z - 10$ )
*=	Multiplicação	$z *= 10$ (equivalente a $z = z * 10$ )
/=	Divisão	$z /= 10$ (equivalente a $z = z / 10$ )
%=	Módulo	$z %= 10$ (equivalente a $z = z \% 10$ )
**=	Potência	$z **= 10$ (equivalente a $z = z ** 10$ )
//=	Divisão inteira	$z //= 10$ (equivalente a $z = z // 10$ )

# Praticando...

1º) Represente em linguagem de programação as operações matemáticas a baixo:

- a)  $3 + 2$
- b)  $6 - 2$
- c)  $6 \times 3$
- d) 10 divido por 2
- e) 2 elevado a potência 3
- f) Conversão de 4.5 para inteiro
- g) conversão de 8 para float

2º) Seguindo os operadores relacionais utilizados em Python, preencha corretamente:

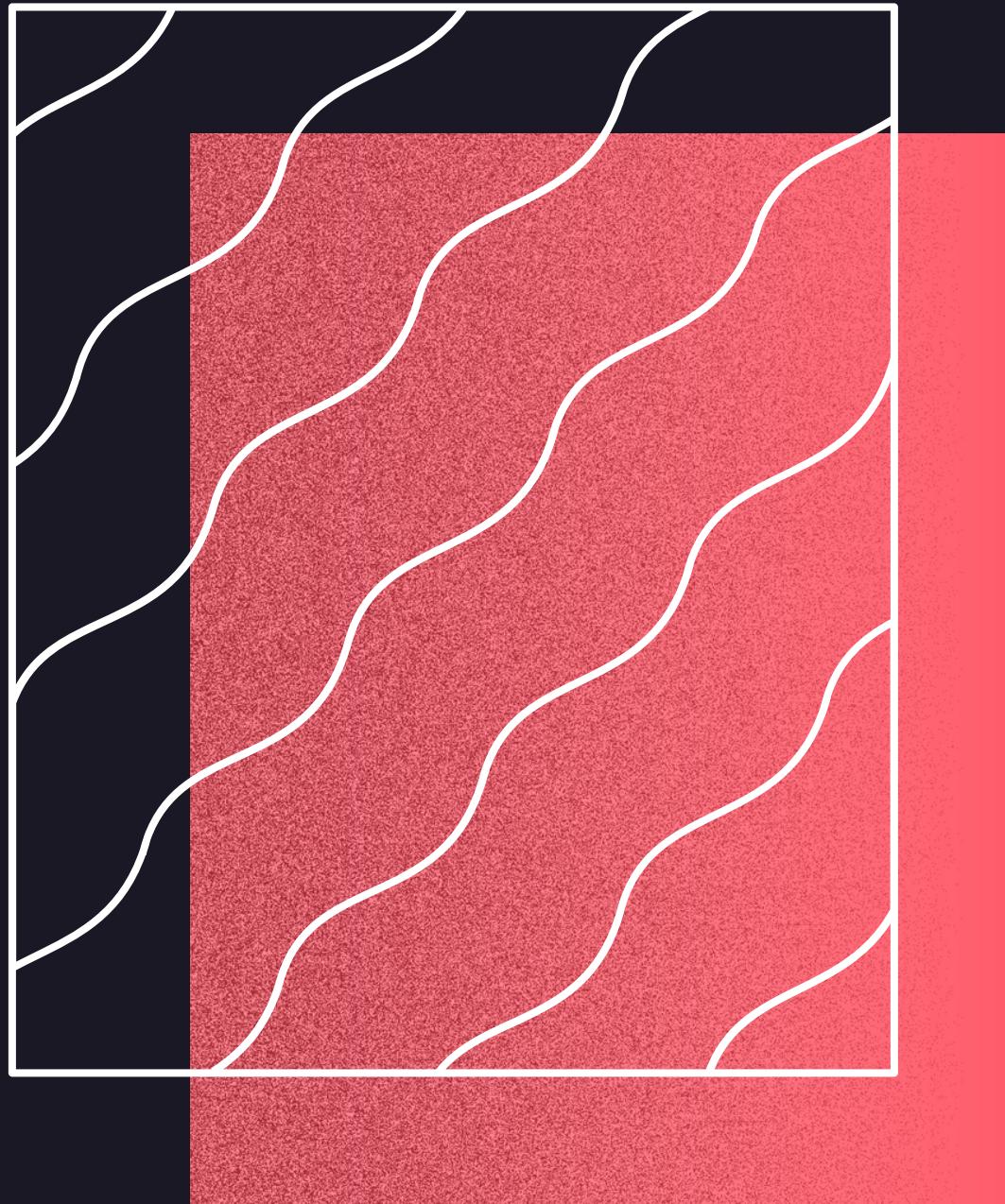
- a) Represente que dois números são desiguais:
- b) Represente que dois números são iguais:
- c) Represente que um número é **menor** que outro. Depois, que um número é **maior** que outro.
- d) Crie uma variável, e demonstre como seria a verificação para saber se ela é menor ou igual/maior ou igual a um determinado valor de referência

# Praticando...

3º) Seguindo as sentenças de atribuição a baixo, demonstre a equivalência delas:

- a)  $A = 10$
- b)  $A += 10$
- c)  $A -= 10$
- d)  $A *= 10$
- e)  $A /= 10$
- f)  $A \%= 10$
- g)  $A **= 10$

# 6. STRINGS E INDEXAÇÃO



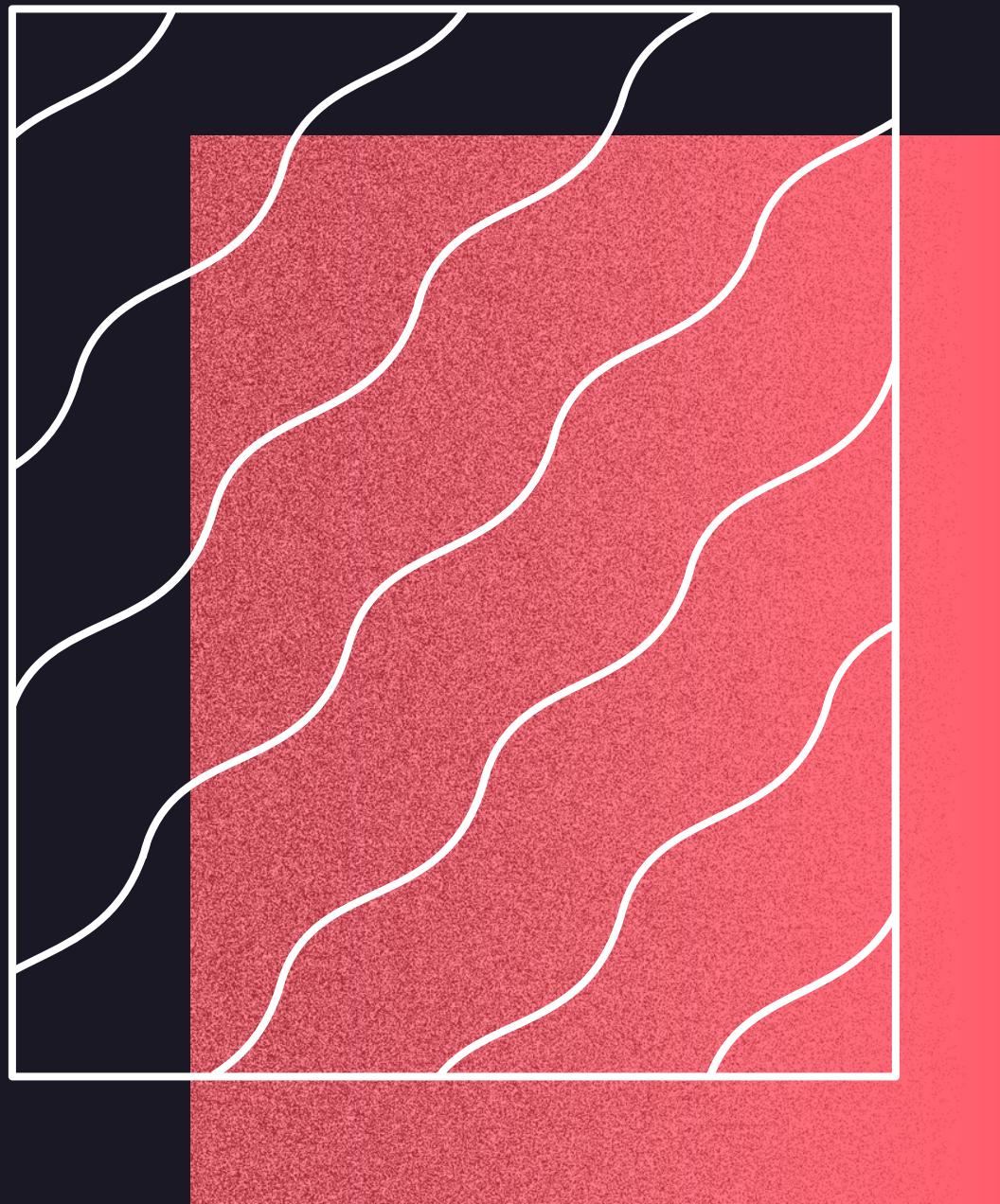
- Strings são um conjunto de caracteres!
- Sempre seguidas de aspas
- Podemos nos referir a um caractere específico na string

`escola = "IFAL Campus Palmeira dos Índios"`

`input: escola[0]`  
`output: I`

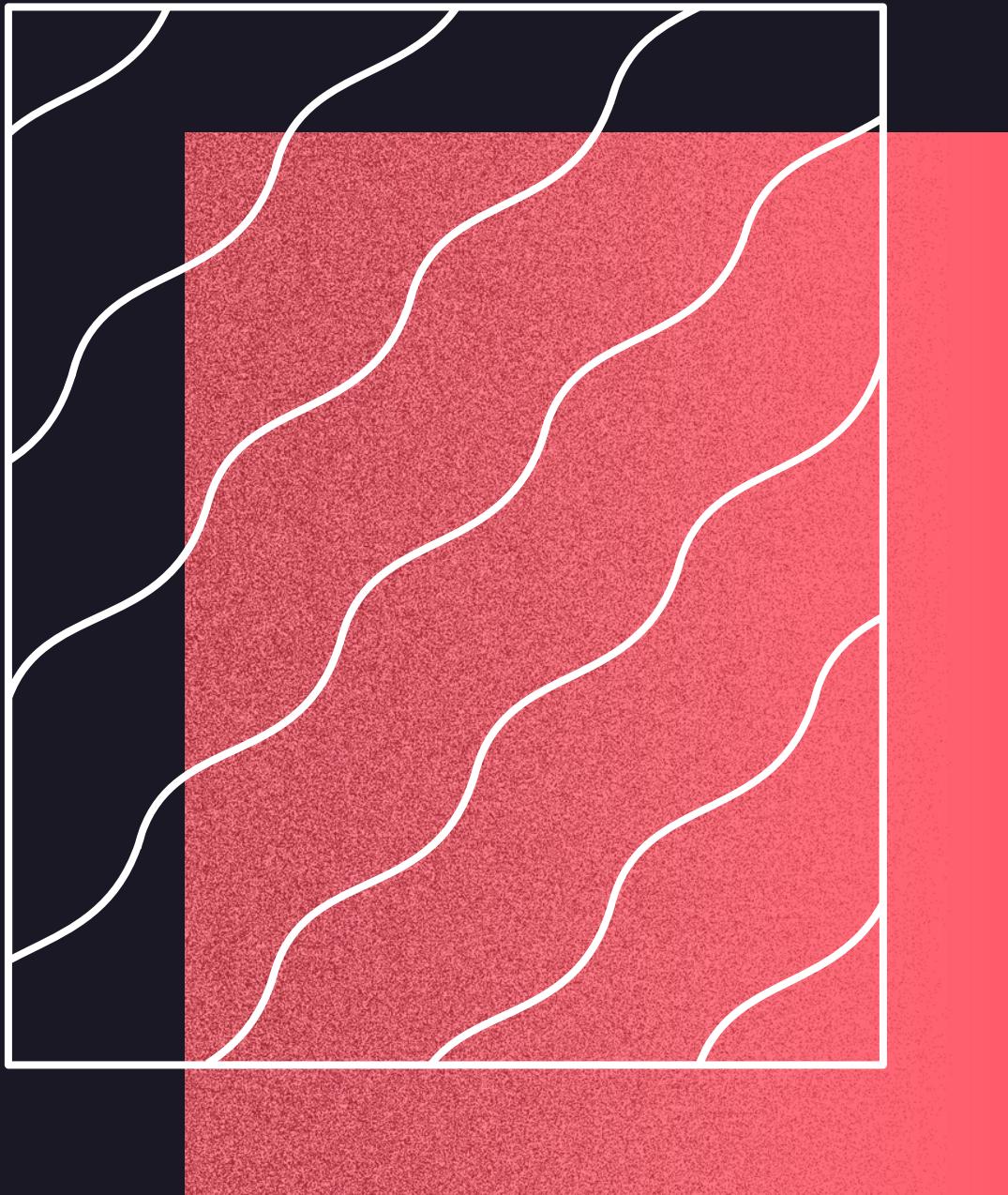
`input: escola[5]`  
`output: C`

# Imutabilidade



- Os elementos de uma string não podem ser mudados!
- exemplo: `escola[7] = "F"`
- No entanto, a variável pode ser reatribuída.

exemplo: `escola = "Massachusetts Institute of Technology"`



# Fatiamento

- Podemos pegar uma substring dentro da própria string!

Exemplos:

`escola = "Massachusetts Institute of Technology"`

`input: escola[27:]`

`output: Technology`

`input: escola[:13]`

`output: Massachusetts`

`input: escola[-1]`

`output: y`

`input: escola[14:26]`

`output: Institute of`

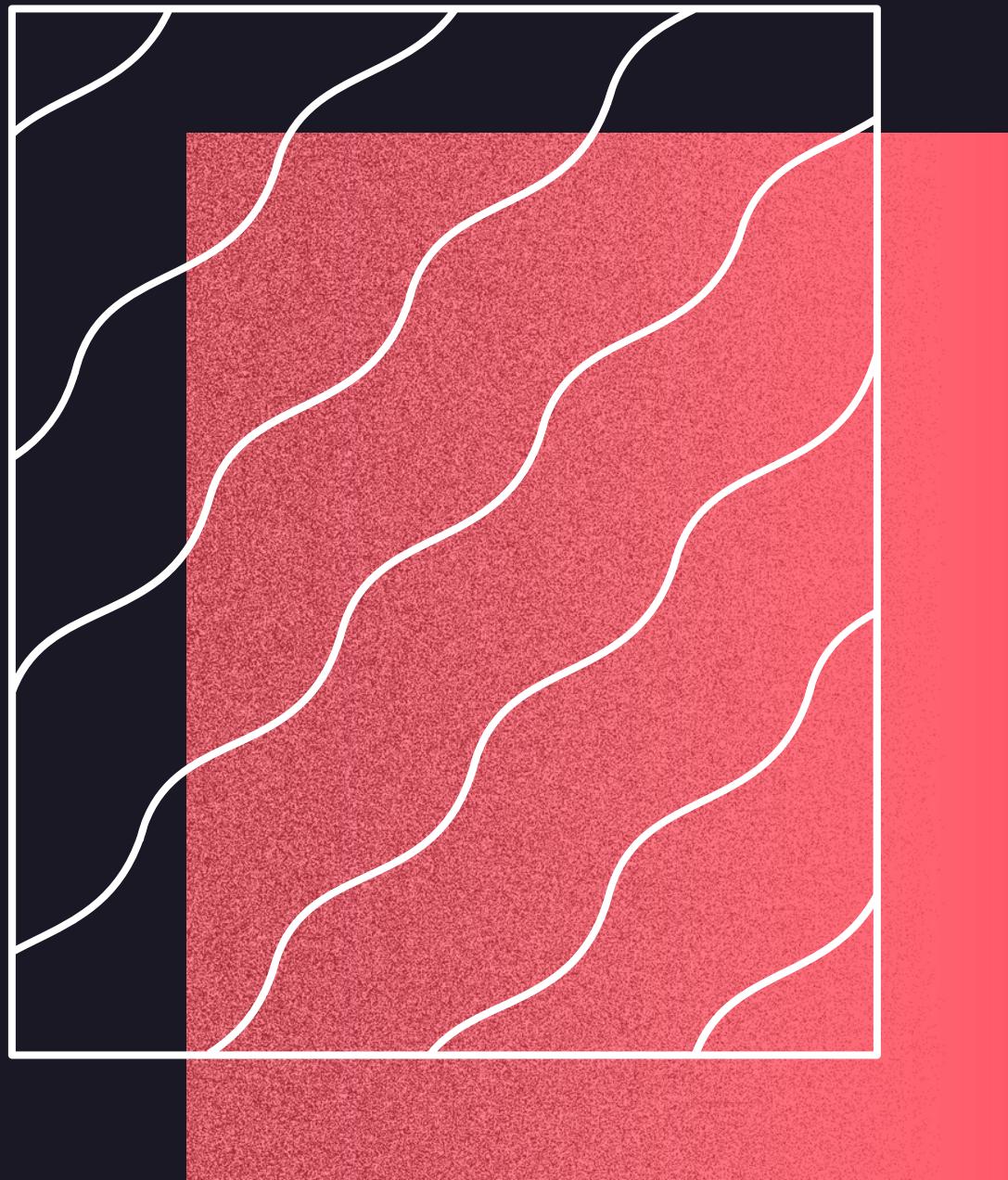
`input: escola[0:1]`

`output: M`

`input: escola[::-2]`

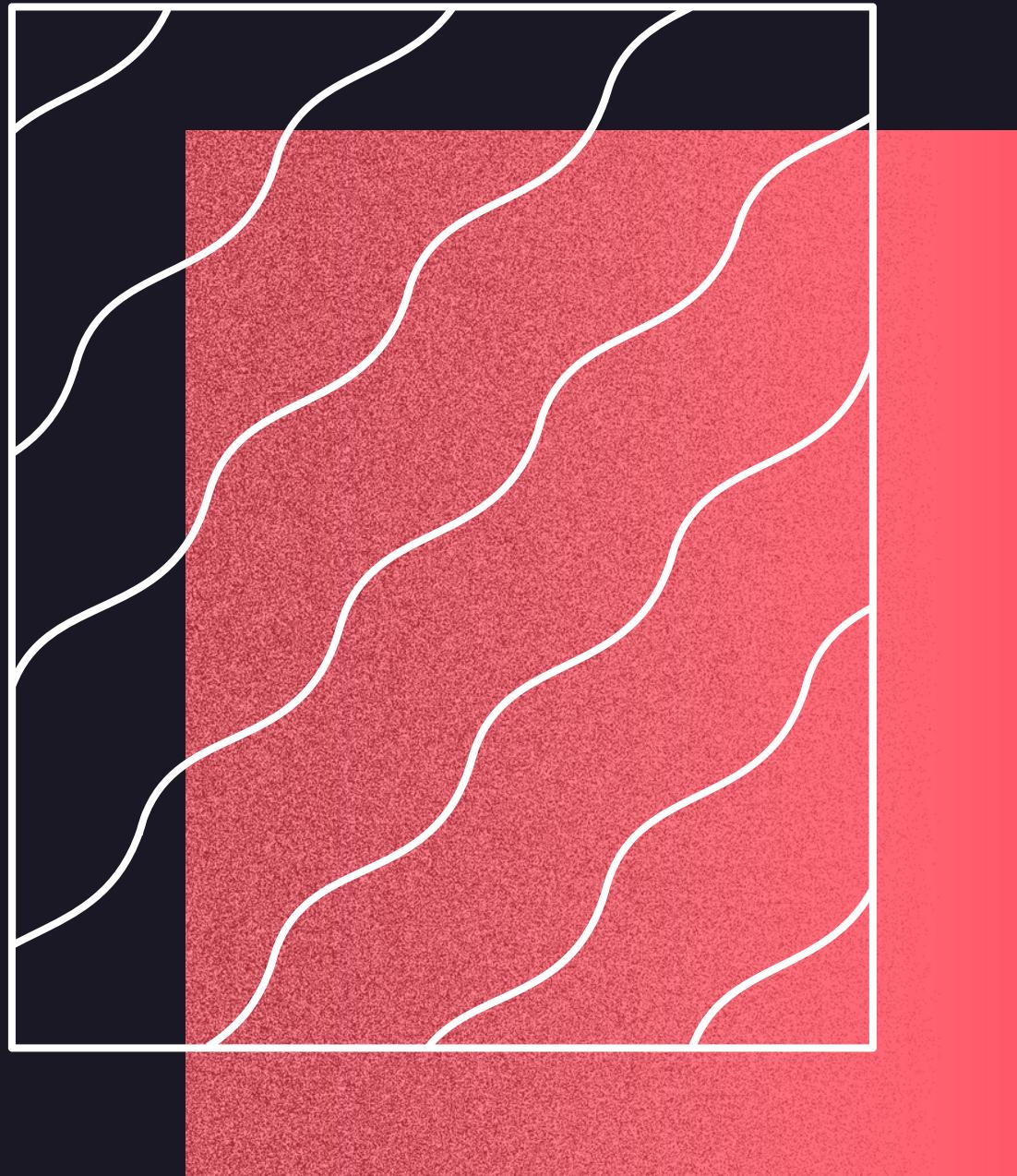
`output: Msahstslsiueo ehooy`

# Funções Built-in



- Python é uma linguagem Orientada a Objetos
- Todo tipo de variável é um objeto pertencente a uma classe
  - A variável "escola" é um objeto de classe string
- Um objeto pode (ou não) conter **atributos** e **métodos**
- Atributos são as propriedades de determinado objeto
- Métodos são as ações que podem ser feitas pelo objeto

# Funções Built-in



Exemplos:

`string = "Isso é uma string"`

`str.capitalize()`

`str.count("a")`

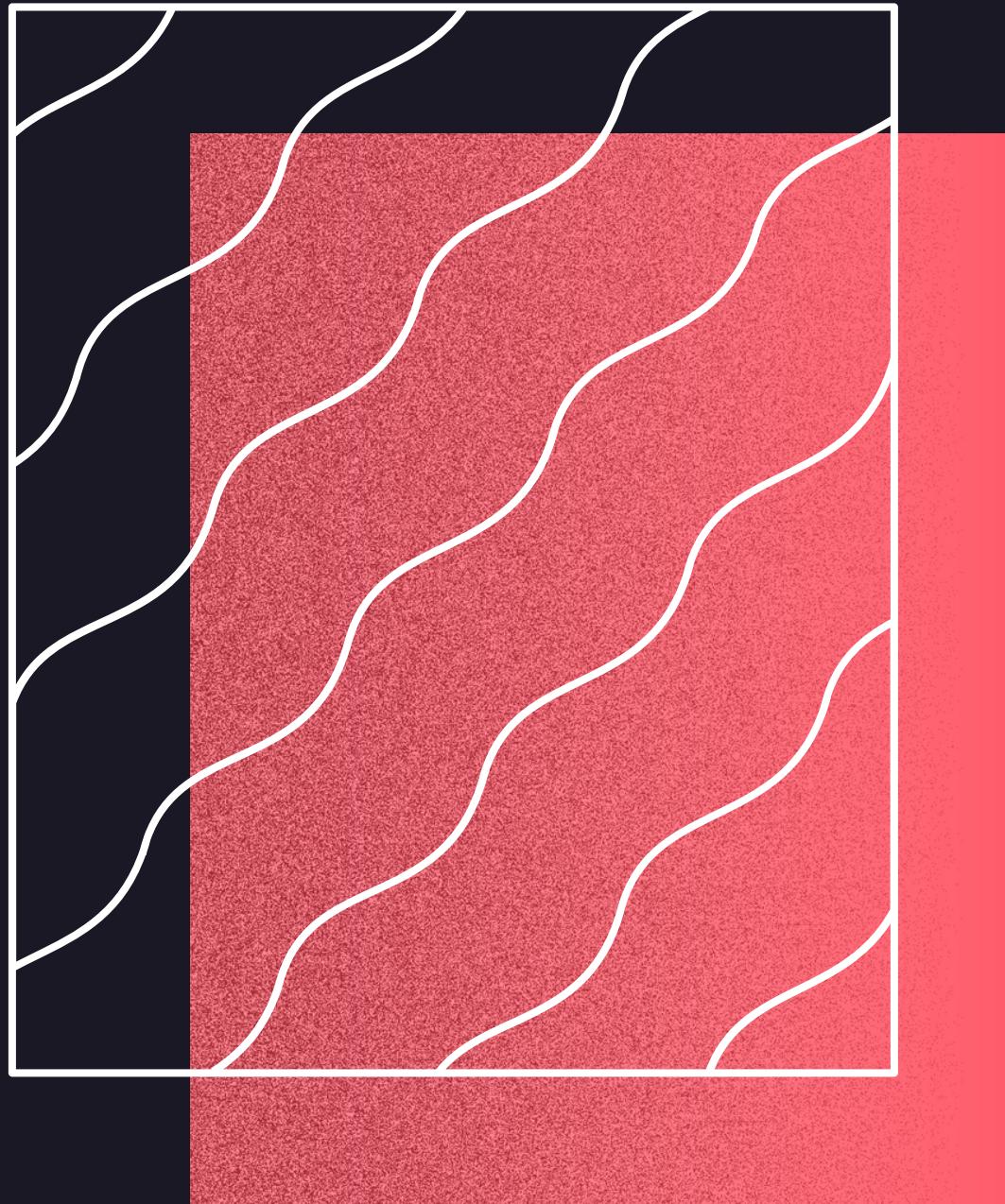
`str.isspace()`

`str.split()`

`str.join()`

`str.find("u")`

# 7. INPUTS



- E se precisarmos receber algum valor do usuário, como fazemos isso?
- Input!! É bem simples:  
`nome = input()`
- Resolvido! Neste exemplo, iremos receber um valor que será atribuída à variável "nome".
- Mas é importante lembrar: o input trata os valores como texto (string)! Se precisarmos receber algum número, um inteiro, por exemplo, temos que converter o valor, da seguinte forma:  
`idade = int(input())`

# 8. Boas práticas

11

- Clareza é importante. Mantenha o seu código limpo e organizado;
- Código esparso é melhor que código denso;
- Sempre documente o seu código;
- Siga os padrões. Não para criar complexidade, mas para manter a regra;
- Erros nunca serão silenciosos, a menos que propositalmente;
- Simples é melhor que complexo, e complexo é melhor que complicado;



# Praticando...

**Questões resolvidas da OBI:**

<https://olimpiada.ic.unicamp.br/pratique/p1/2020/f1/tesouro/>

**DESAFIO:**

<https://olimpiada.ic.unicamp.br/pratique/p1/2021/f1/torneio/>

Dica: usar estrutura de decisão e loops

# Questão 1: Divisão do Tesouro

*"A primeira linha da entrada contém um número inteiro A, o número de moedas na arca. A segunda linha contém um inteiro N, o número de marinheiros (não contando o Capitão)."*

Dessa forma, devemos executar dois inputs:

```
moedas = int(input())
marin = int(input())
```

Tendo a quantidade de moedas e o número de marinheiros, basta pensarmos na lógica da questão:

*"Seu programa deve produzir na saída uma única linha, contendo um único inteiro, o número de moedas que o Capitão Olho Roxo deve receber."*

# Questão 1: Divisão do Tesouro

Como o capitão receberá o dobro de moedas dos marinheiros, podemos pensar o seguinte:

Se há 2 marinheiros e 1 capitão, podemos dividir a moeda para 4 pessoas! Sendo assim, 1 parcela irá para cada marinheiro e 2 (o dobro) irá para o capitão.

Sendo assim:

$$\text{cap} = 2 * \text{moedas} // (\text{marin} + 2)$$

" $\text{moedas} // (\text{marin} + 2)$ " indica a quantidade de moedas por pessoa (contando os marinheiros, o capitão e + 1 pessoa).

Multiplica-se por 2 porque o capitão recebe 2 parcelas (a dele e a da pessoa que "não existe").

# Questão 1: Divisão do Tesouro

Nota: o símbolo "://" indica uma divisão inteira, como foi explicado no decorrer da aula. Isso foi feito porque a saída do problema tem que ser um número inteiro.

Finalmente, é só imprimir a saída:

`print(cap)`

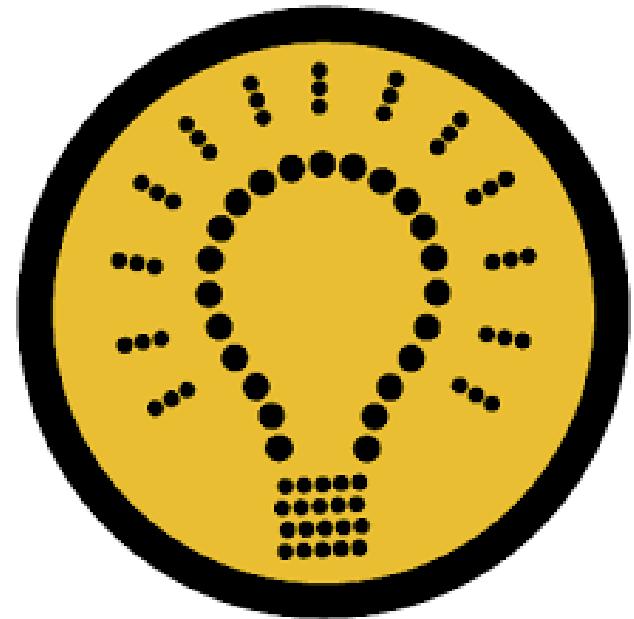
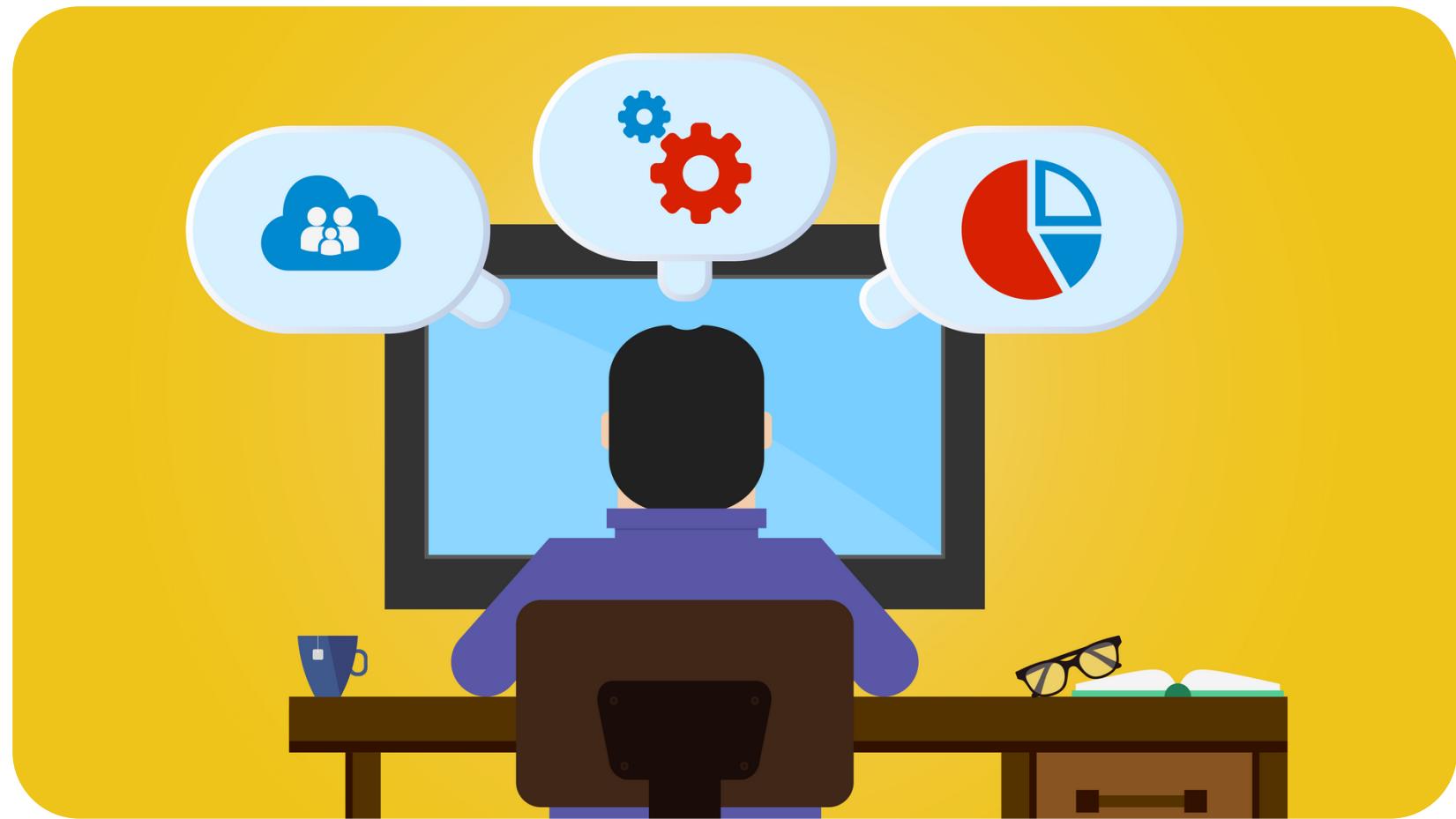
Resultado final:

```
moedas = int(input())
marin = int(input())
```

```
cap = 2 * moedas // (marin + 2)
```

`print(cap)`

# Por hoje é só!



Revisem sempre que necessário o conteúdo passado.