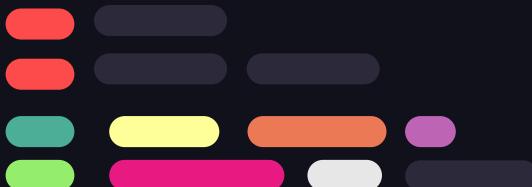




03 { ..

# Manipulação de Variáveis e Dados

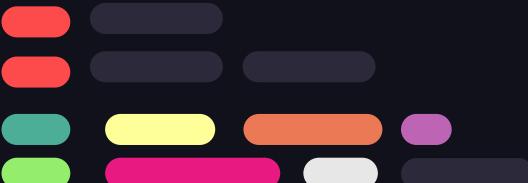


**Formador:** Ricardo Mourão

# Manipulação de Variáveis e Dados

## { Concatenação de Strings

A concatenação é o processo de juntar dois ou mais fragmentos de texto (ou strings) para formar uma única string. Em programação, isso é frequentemente usado para criar mensagens ou informações que combinam texto fixo com variáveis.

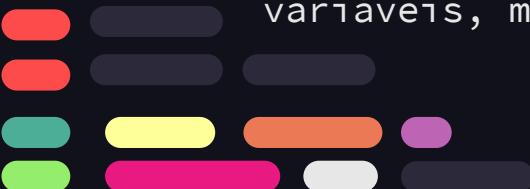


# Manipulação de Variáveis e Dados

## { Concatenação com vírgula

```
nome = "Guido van Rossum"  
print("O nome é", nome)
```

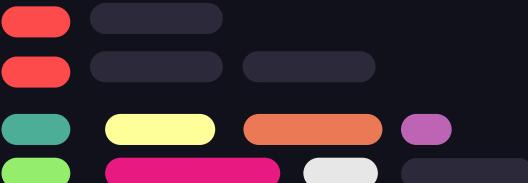
Neste caso, a função ***print()*** aceita múltiplos argumentos e concatena-os automaticamente com um espaço entre eles. É a forma mais direta e fácil de combinar strings com outras variáveis, mas oferece menos controlo sobre a formatação.



# Manipulação de Variáveis e Dados

## { String Interpolation

***String interpolation*** é uma técnica usada na programação para inserir ou substituir valores dentro de uma string. Esta técnica permite que se crie strings que incluem valores de variáveis ou expressões dentro delas, tornando o código mais legível e muitas vezes mais eficiente.



# Manipulação de Variáveis e Dados

## { Interpolação com método *.format()*

```
nome = "Guido van Rossum"  
idade = 67  
peso = 90.52
```

```
Print("O nome é {}, a idade é {} e o peso é {}".format(nome, idade, peso))
```

}



# Manipulação de Variáveis e Dados

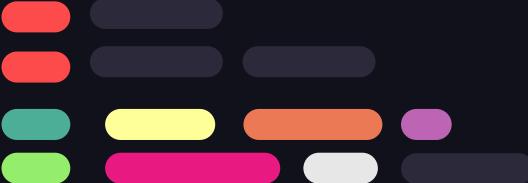
{

## Interpolação com *f-strings*

```
nome = "Guido van Rossum"  
idade = 67  
peso = 90.52
```

```
Print(f"O nome é {nome}, a idade é {idade} e o peso é {peso}.")
```

}

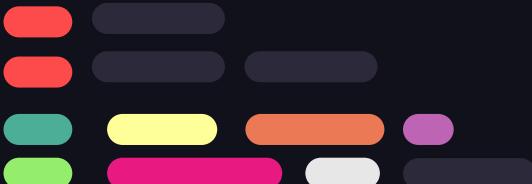




# Manipulação de Variáveis e Dados

## { String Interpolation

*String interpolation* é uma técnica bastante flexível e oferece muitas opções para formatar valores que estão a ser interpolados numa string.

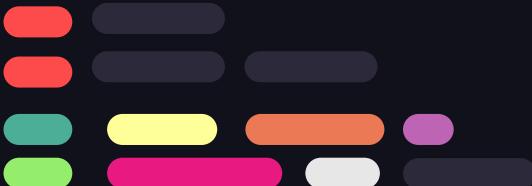


# Manipulação de Variáveis e Dados

## { String Interpolation

Precisão de ponto flutuante: {:.2f}. Isto irá formatar um número flutuante para ter duas casas decimais.

Largura e alinhamento: {:<10}. Isto irá alinhar o valor à esquerda dentro de um espaço de 10 caracteres.

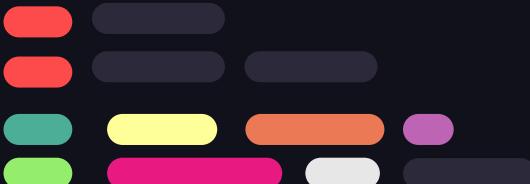


# Manipulação de Variáveis e Dados

## { String Interpolation

Largura e alinhamento à direita: {:>10}. Isto irá alinhar o valor à direita dentro de um espaço de 10 caracteres.

Largura e alinhamento ao centro: {:^10}. Isto irá centrar o valor dentro de um espaço de 10 caracteres.



# Manipulação de Variáveis e Dados

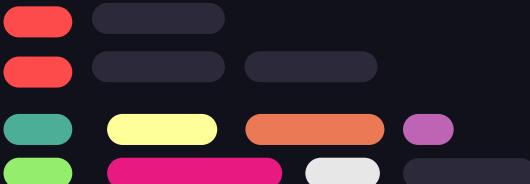
## { String Interpolation

Preenchimento com caracteres: `{:x<10}` ou `{:0>10}`. Isto irá preencher o espaço com o caractere especificado.

Exemplo: `"{:x<10}"`.format("texto") resulta em 'textoxxxxx'.

Exemplo: `"{:0>10}"`.format(5) resulta em '0000000005'.

}

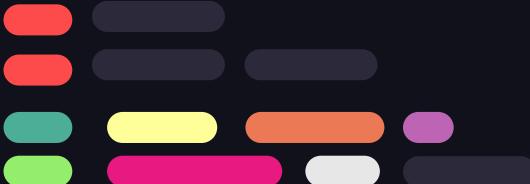


# Manipulação de Variáveis e Dados

## { String Interpolation

Formatação de percentagem: {:.1%}. Isto irá formatar um número flutuante como percentagem.

Exemplo: "{:.1%}".format(0.25) resulta em '25.0'

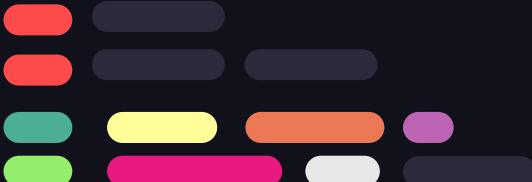


# Manipulação de Variáveis e Dados

## { String Interpolation

Separador de milhares: {:,}. Isto irá formatar um número com separadores de milhar.

Exemplo: "{:,}".format(1000000) resulta em '1,000,000'.

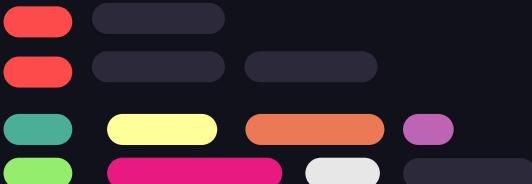


# Manipulação de Variáveis e Dados

## { Fatiamento de Strings

Quando é atribuído um valor a uma variável, é criado um espaço de memória que vai armazenar esses valores.

Ex: frase = "Curso de Python"



# Manipulação de Variáveis e Dados

## { Manipulação de Strings

frase:

C u r s o d e P y t h o n

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14



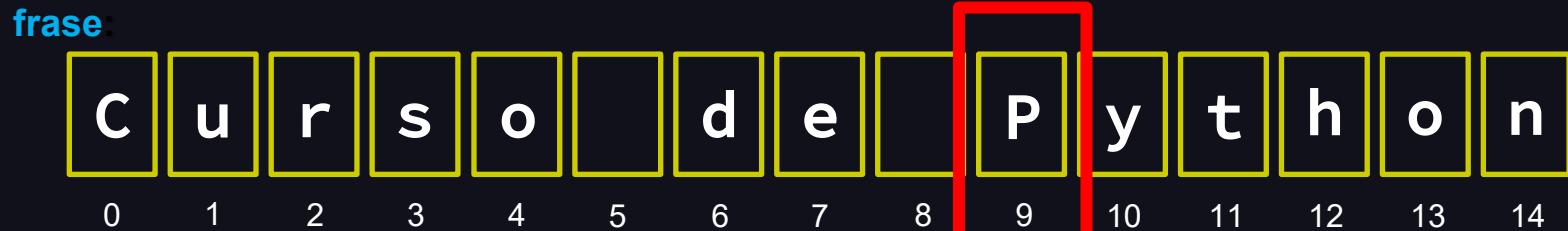
{}



# Manipulação de Variáveis e Dados

## { Fatiamento

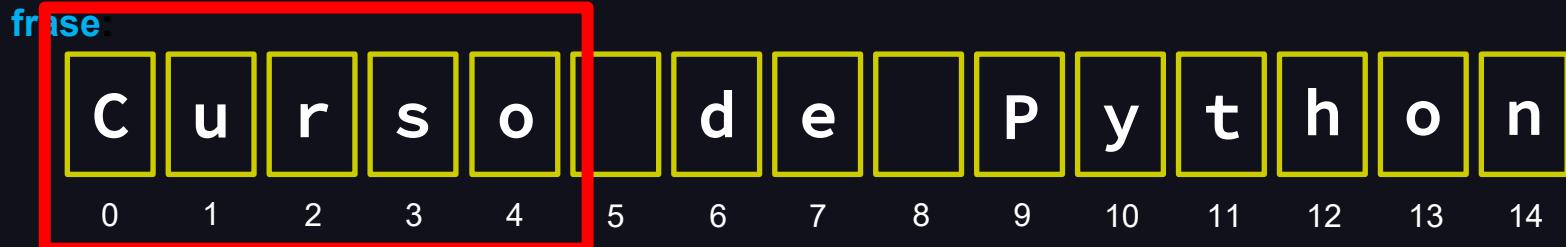
**frase[9]**



# Manipulação de Variáveis e Dados

## { Fatiamento

**frase[0:5]**



# Manipulação de Variáveis e Dados

## { Fatiamento

**frase[9:15]**

**frase:**



# Manipulação de Variáveis e Dados

## { Fatiamento

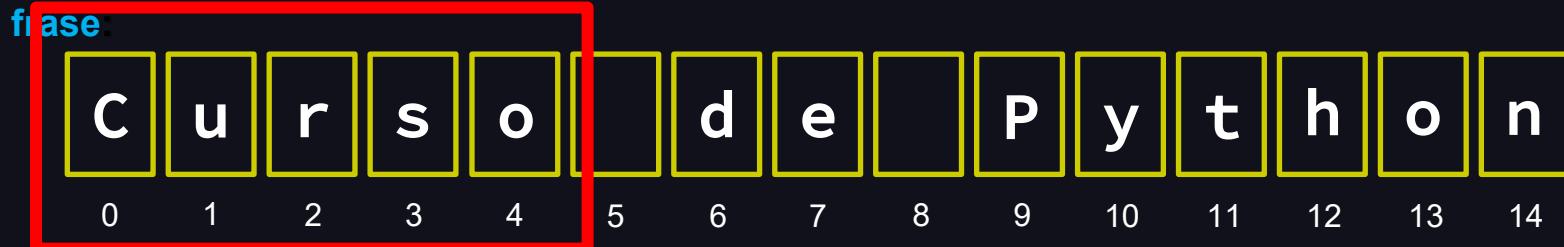
**frase[0:15:2]**



# Manipulação de Variáveis e Dados

## { Fatiamento

**frase[:5]**



# Manipulação de Variáveis e Dados

## { Fatiamento

**frase[9:]**

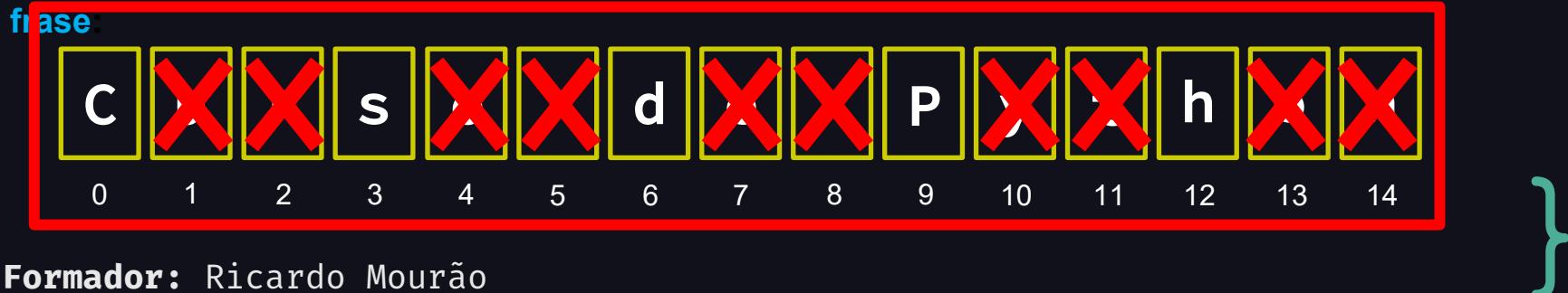
**frase:**



# Manipulação de Variáveis e Dados

## { Fatiamento

**frase[0::3]**



# Manipulação de Variáveis e Dados

{Análise

## len(frase)

15 caracteres

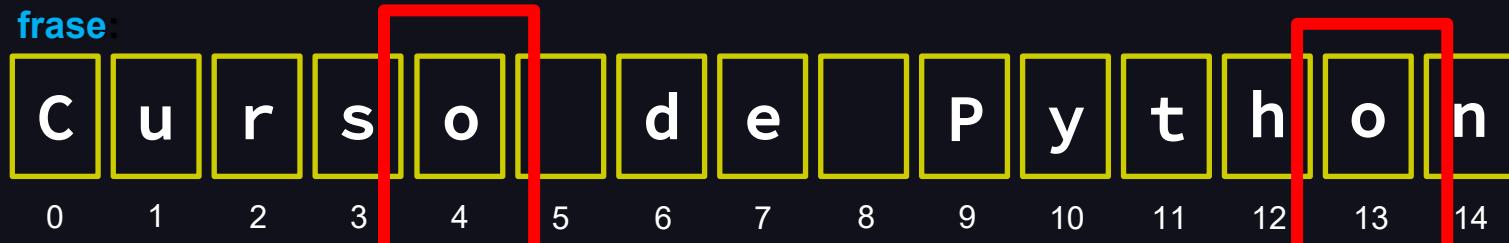
frase:

C	u	r	s	o		d	e		P	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

# Manipulação de Variáveis e Dados

{Análise

**frase.count("o")**



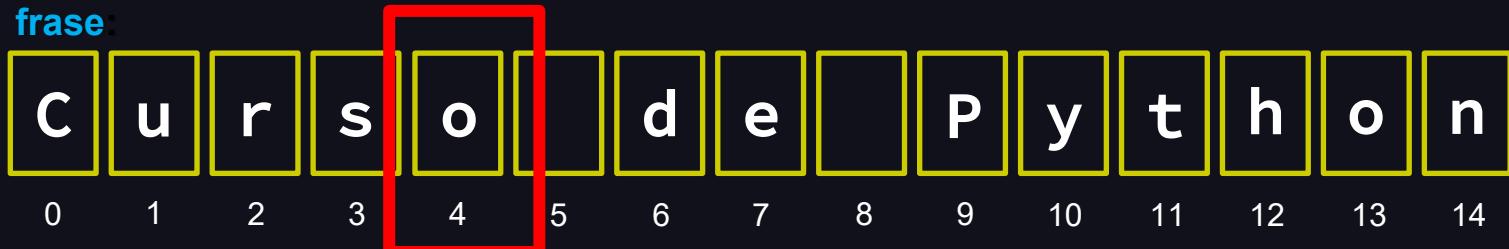
}

# Manipulação de Variáveis e Dados

{Análise

**frase.count("o", 0, 5)**

**frase:**



# Manipulação de Variáveis e Dados

{Análise

**frase.find("th")**

**frase:**

A diagram illustrating the string "Curso de Python" as an array of characters. The string is split into individual characters: C, u, r, s, o, , d, e, , P, y, t, h, o, n. These characters are arranged in a row of 15 boxes, indexed from 0 to 14 below them. The characters 't' and 'h' are highlighted with a red box, indicating their positions at indices 11 and 12.

C	u	r	s	o	,	d	e	,	P	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

# Manipulação de Variáveis e Dados

{Análise

**frase.find("Playstation")**

-1

frase:

C	u	r	s	o		d	e		P	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

# Manipulação de Variáveis e Dados

{Análise

“curso” in frase

False

frase:

C	u	r	s	o		d	e		P	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

# Manipulação de Variáveis e Dados

{Análise

“Curso” in frase

True

frase:

C	u	r	s	o		d	e		P	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Formador: Ricardo Mourão

}

# Manipulação de Variáveis e Dados

{ Transformação

**frase.replace(“Python”, “C++”)**

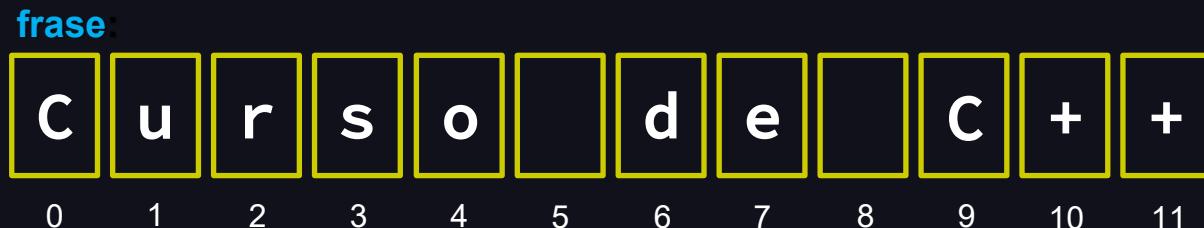
**frase:**

C	u	r	s	o		d	e		P	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

# Manipulação de Variáveis e Dados

{ Transformação

**frase.replace(“Python”, “C++”)**



# Manipulação de Variáveis e Dados

## { Transformação

### frase.upper()

frase:

C	u	r	s	o		d	e		P	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

# Manipulação de Variáveis e Dados

## { Transformação

### frase.upper()

frase:

C	U	R	S	O		D	E		P	Y	T	H	O	N
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Formador: Ricardo Mourão

{}

# Manipulação de Variáveis e Dados

## { Transformação

### frase.lower()

frase:

c	u	r	s	o		d	e		p	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

# Manipulação de Variáveis e Dados

## { Transformação

### frase.capitalize()

frase:

C	u	r	s	o		d	e		p	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

# Manipulação de Variáveis e Dados

{ Transformação

## frase.title()

frase:

C	u	r	s	o		D	e		P	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Formador: Ricardo Mourão

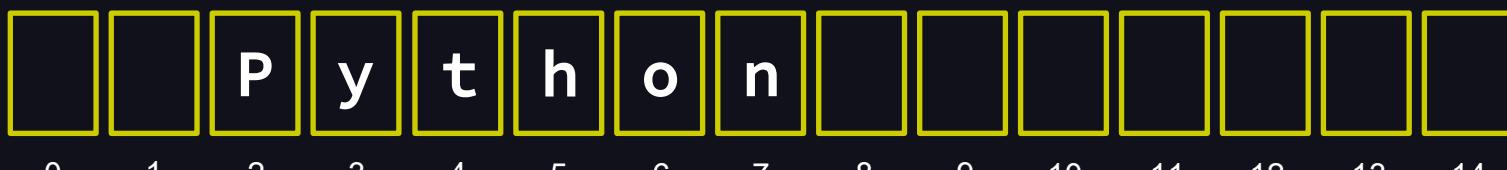
}

# Manipulação de Variáveis e Dados

## { Transformação

### **frase.strip()**

**frase:**



# Manipulação de Variáveis e Dados

{ Transformação

## frase.strip()

frase:

P	y	t	h	o	n
0	1	2	3	4	5

Formador: Ricardo Mourão

}

# Manipulação de Variáveis e Dados

## { Transformação

### frase.rstrip()

frase:



# Manipulação de Variáveis e Dados

## { Transformação

### frase.rstrip()

frase:



# Manipulação de Variáveis e Dados

{ Transformação

## frase.lstrip()

frase:



# Manipulação de Variáveis e Dados

## { Transformação

### frase.split()

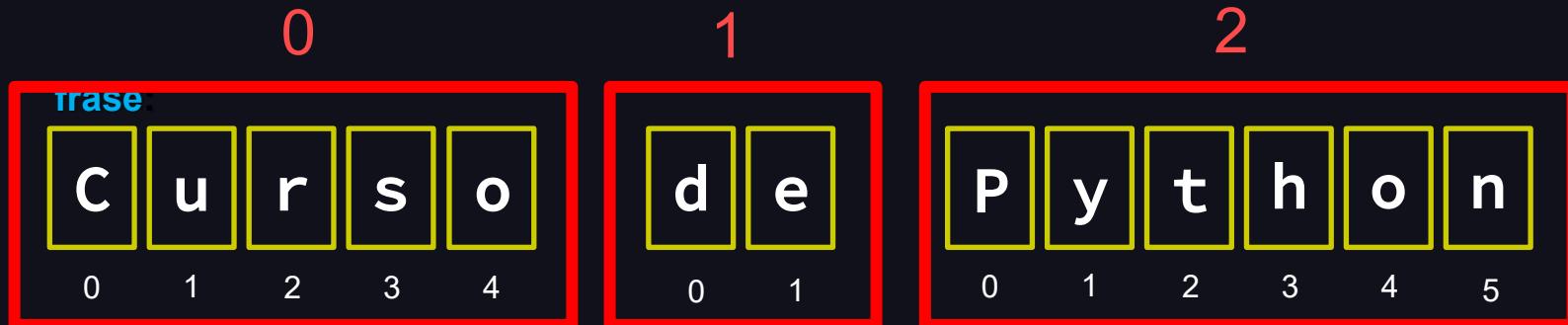
frase:



# Manipulação de Variáveis e Dados

{ Transformação

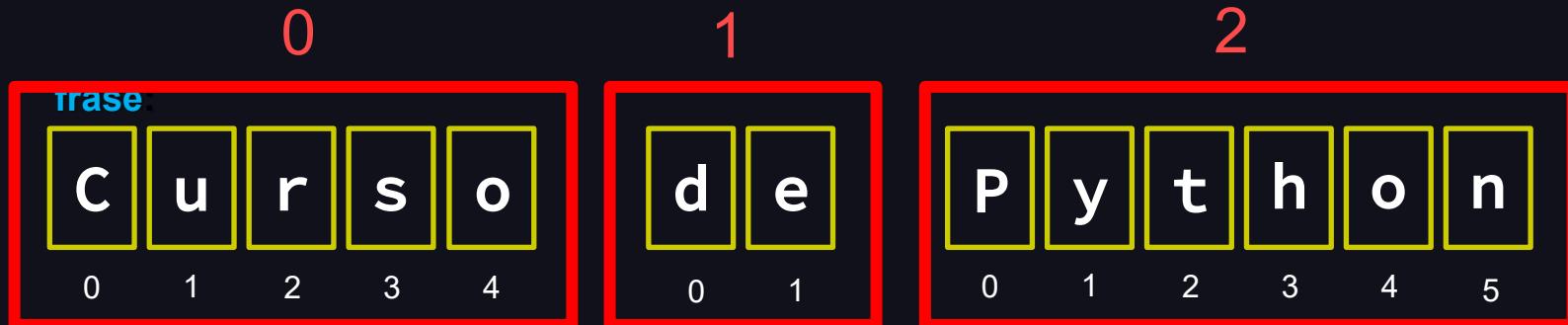
## frase.split()



# Manipulação de Variáveis e Dados

## { Transformação

`“_”.join(frase)`



# Manipulação de Variáveis e Dados

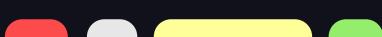
## { Transformação

“\_”.join(**frase**)

**frase**:

C	u	r	s	o	-	d	e	-	P	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	

# PRÁTICA! Exercício 10

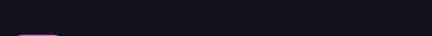
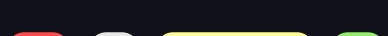


Crie um programa que leia o nome completo de uma pessoa e mostre:

- ✓ O nome com todas as letras maiúsculas;
- ✓ O nome com todas as letras minúsculas;
- ✓ Quantidade de letras sem espaços;
- ✓ Quantas letras tem o primeiro nome.



# PRÁTICA! Exercício 11

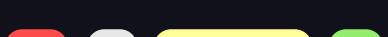
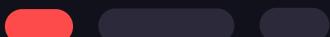


Crie um programa que leia uma frase e mostre:

- ✓ Quantas vezes aparece a letra “a/A”;
- ✓ Em que posição aparece a primeira vez;
- ✓ Em que posição aparece a última vez.



# PRÁTICA! Exercício 12



Crie um programa que leia o primeiro e último nome de uma pessoa e exiba as mensagens:

1. “Olá **nome**, o seu registo está completo.”
2. “O seu email é **p.ultimo@empresa.pt**”

Ex **email**:

Alfredo Xavier

a.xavier@empresa.pt

