

# Fatiamento



**frase[9]**

Apenas o numero 9

**frase**

**C u r s o q u e m i d e o P y t h o n**

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

# Fatiamento

Do 9 ate o numero  
13 ele elimina a ultima casa



**frase[9:13]**

**[9:14]**

**frase**

**C u r s o q m P y t h o n**

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

■ADAMANTIUM■



- Diego Turchetto (RS) - Diogenes Soares (PE) - Douglas Henrique Fraga (SC) - Eder

# • Fatiamento



## Frase[9:21]

o 21 não existe entao ele para  
na ultima casa

frase



■ADAMANTIUM■



- Marcio Gemaque (PA) - Marcos Antonio Ribeiro (SP) - Marcos Binek (SP) - Marcus

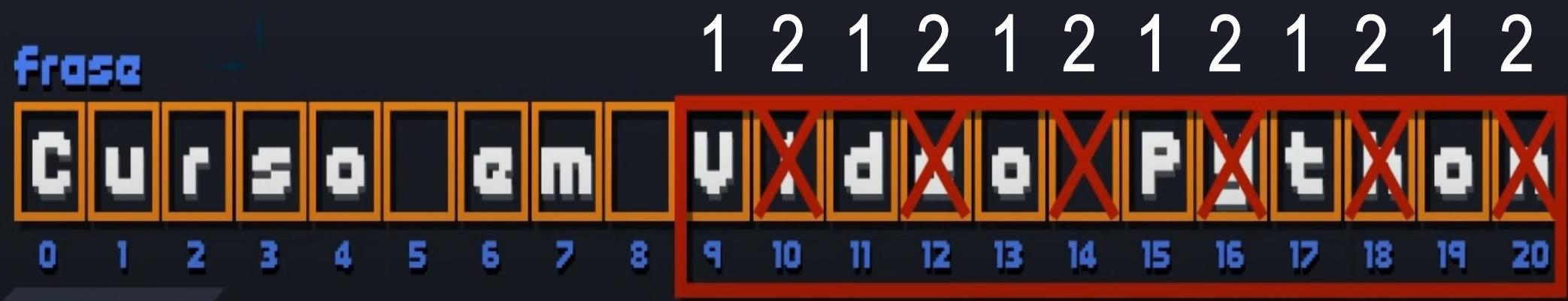


# Fatiamento



frase[9:21:2]

:2 casa sim casa nao



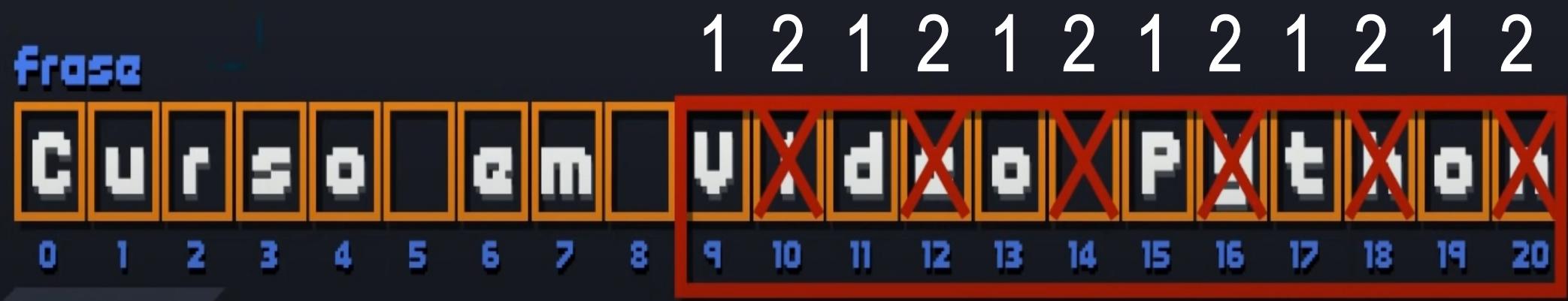
ADAMANTUM

# Fatiamento



frase[9:21:2]

:2 casa sim casa nao



ADAMANTUM

## Fatiamento



# frase[:5]

:5 do começo ate uma casa  
anterior aqui no caso ate o 4

frase

C	u	r	s	o	e	m	V	i	d	o	P	y	t	h	o	n				
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

# • Fatiamento



**frase[15:]**

15: do 15 ate o final

**frase**

Curso em Vídeo Python

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----





# frost[9:3]

**9::3** começa no 9 vai ate o final e  
pula em 3 e 3

Fraser

# Análise



## len(frase)

len(frase) quantos carcteres tem

21 caractares

frase

C u r s o q m V i d e o P y t h o n

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Curso em Vídeo

INSCRITO



## Análise



**frase.count('o')**

conta quantas letras ( o ) minúsculos tem

**frase**

Curso em Vídeo Python

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----



## Análise



**frase.count('o', 0, 13)**

'o', 0, 13) ele conta quantos o tem do 13 para traz



- Análise



**frase.**find('deo')****

.find (encontrar) ('deo')

**frase**

Curso em Vídeo Python

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

exemplo

A diagram illustrating the string "Curso em Vídeo Python" as a sequence of 21 boxes. The boxes are arranged horizontally, each containing a character from the string. Boxes 11, 12, and 13 are highlighted with a red border, indicating the substring being searched for by the .find method. Above the boxes, the word "exemplo" is written. Below the boxes, a numerical scale from 0 to 20 is shown, with the numbers 0 through 10 in blue and 11 through 20 in white.

## Análise



# frase.find('Android')

caso ele não encontre ele da o resultado de -1

frase

Curso em Vídeo Python

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

OURO



(PR) - Gislene Sampaio (SP) - Gleison Gomes (MG) - Graciela Jardim (PR) - Guilherme Augusto Barreto (S)



## Análise



# 'Curso' in frase

'curso' = nome do tex pesquisado  
in = dentro da frase tem o tex 'curso'  
volta como resultado True

frase

Curso em Vídeo Python

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

OURO



Talita Ramon Freitas (BA) - Hudson Barbosa (RO) - Icler Alves (IRL) - Irozino De Carvalho (RJ) - Ivan Sam

# • Transformação

.replace

(ele substitue por outro tex)



```
frase.replace('Python', 'Android')
```

frase

Curso em Vídeo Python

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

'Android'

Python

15 16 17 18 19 20



Luis Fernando Alves (SP) - Luis Filipe Coelho (RJ) - Luis Teruya (SP) - Luiz Alberto Junior (RJ) - Luiz



- Transformação



# frase.upper()

.upper()

tudo que for letras minúsculas ele vai deixar maiúsculas

frase

C U R S O E M V I D E O P Y T H O N

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

OURO



iago Wolff (SP) - Wagner Da Silva Rodrigues (SP) - Valdeir De Melo (MT) - Valdemir Lima (SP) - Vicente

# Transformação



## frase.lower()

.lower()

deixa tudo em minusculas

frase

C u r s o e m V i d e o p y t h o n

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20



- Transformação



## **frase.capitalize()**

`.capitalize()`

ele pega e deixa apenas a primeira letra em maiusculas

<b>frase</b>																				
C	u	r	s	o		e	m		V	i	d	e	o		P	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20



## • Transformação



# frase.title()

.title()

ele conta quantas palavras tem e deixa todas  
as primeiras letras das palavras em maiúsculos

frase

C u r s o E m V i d e o P y t h o n

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20



- Transformação



## **frase.strip()**

`.strip()`

ela remove todas o campos vazios no  
começo e no final

**frase**



- Transformação



# frase.rstrip()

.rstrip (.r)

ele remove da direita e não mexe no começo

frase

			A	P	r	e	n	d	o		P	y	t	h	o	n	X	X
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18



# Transformação



## frase.lstrip()

.lstrip (.)

ele remove da esquerda e não mexe no final

frase



## • Divisão



# **frase.split()**

## .split()

**ele separa as palavras e renumera elas e coloca em grupos**



## Junção



'-'.join(frase)

'-'.join(frase) ou '' com espaço normal

ele ajunta todas as palavras separando com '' ou '-'

frase

C u r s o - e m - V i d e o - P y t h o n

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

