

AULA 04

LINGUAGEM DE PROGRAMAÇÃO II - LP212

CONCEITOS IMPORTANTES

PROFA. ANA PAULA MÜLLER GIANCOLI

Instituto Federal de São Paulo - IFSP

Campus Bragança Paulista

AGENDA

Importante String

Importante Listas

```
self.file.  
self.fingerprints.  
  
def request_seen(self, request)  
    fp = self.request_fingerprint  
    if fp in self.fingerprints:  
        return True
```

**PARA APROFUNDAMENTO
CONSULTE O SITE DE DOCUMENTAÇÃO
(VERSÃO ESTÁVEL)**

[HTTPS://DOCS.PYTHON.ORG/3.7/](https://docs.python.org/3.7/)

IMPORTANTE STRING

IMPORTANTE

STRINGS

- As strings podem ser concatenadas por meio da utilização do **símbolo +**.

```
>>> nome = "Ana Paula"
>>> sobrenome = "Müller"
>>> ultimo = "Giancoli"
>>> nome + sobrenome + ultimo
'Ana PaulaMüllerGiancoli'
>>> |
```

IMPORTANTE

STRINGS

```
>>> nome = "Ana Paula"
>>> sobrenome = "Müller"
>>> ultimo = "Giancoli"
>>> nome + " " + sobrenome + " " + ultimo
'Ana Paula Müller Giancoli'
>>> |
```

IMPORTANTE

STRINGS

- As strings podem ser repetidas por meio da utilização do **símbolo ***.

```
>>> nome = "Ana Paula"
>>> 3 * nome
'Ana PaulaAna PaulaAna Paula'
>>> |
```

STRINGS

>>> |

IMPORTANTE

STRINGS

- **Duas ou mais strings próximas** uma da outra, são concatenadas automaticamente.

```
>>> "Ana Paula" "Müller" "Giancoli"
```

```
'Ana PaulaMüllerGiancoli'
```

```
>>> "Ana Paula" ' ' "Müller" ' ' "Giancoli"
```

```
'Ana Paula Müller Giancoli'
```

```
>>> |
```

IMPORTANTE

STRINGS

- Podemos **indexar** as sequências de caracteres (strings), sendo que o **1o. caracter tenha o índice 0 (zero)**.

+---+---+---+---+---+---+---+---+---+---+																
	G		i		a		n		c		o		l		i	
+---+---+---+---+---+---+---+---+---+---+																
0	1	2	3	4	5	6	7	8								

IMPORTANTE

```
>>> sobrenome = "Giancoli"
```

```
>>> sobrenome[0]
```

```
'G'
```

```
>>> sobrenome[5]
```

```
'o'
```

```
>>> sobrenome[7]
```

```
'i'
```

```
>>> sobrenome[8]
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
IndexError: string index out of range
```

```
>>> |
```

+---+---+---+---+---+---+---+---+---+---+---+---+

| G | i | a | n | c | o | l | i |

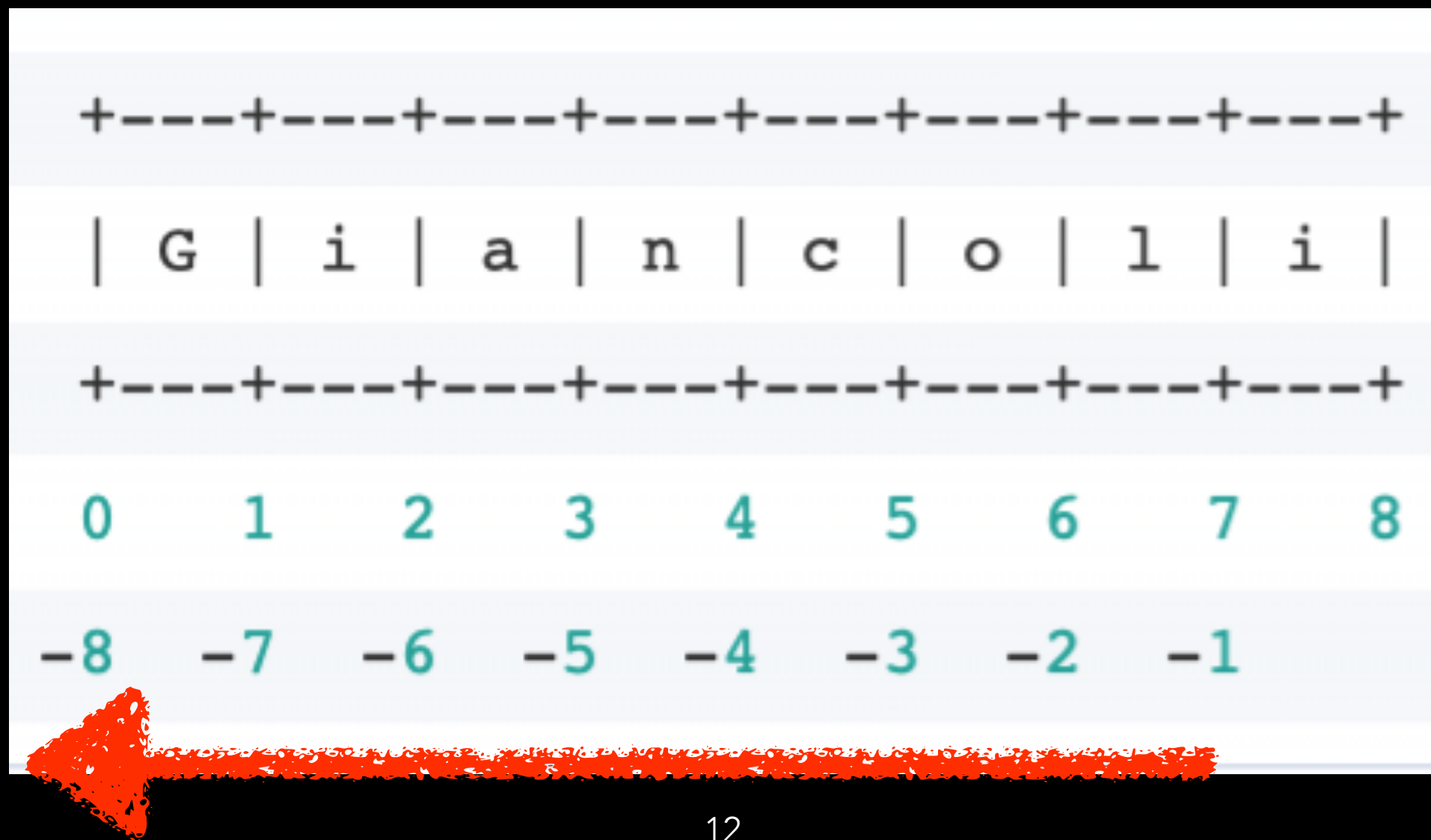
+---+---+---+---+---+---+---+---+---+---+---+---+

0 1 2 3 4 5 6 7 8

IMPORTANTE

STRINGS

- Os **índices** podem ser **negativos**, neste caso, iniciam a contagem da direita pra esquerda, a partir do último caracter.



IMPORTANTE

```
>>> sobrenome = "Giancoli"
```

```
>>> sobrenome[-1]
```

```
'i'
```

```
>>> sobrenome[-5]
```

```
'n'
```

```
>>> sobrenome[-8]
```

```
'G'
```

```
>>> sobrenome[0]
```

```
'G'
```

```
>>> sobrenome[-0]
```

```
'G'
```

```
>>> |
```

+	-	-	-	+	-	-	-	+	-	-	-	+	-	-	-	+	-	-	-	+
	G		i		a		n		c		o		l		i					
+	-	-	-	+	-	-	-	+	-	-	-	+	-	-	-	+	-	-	-	+
0	1	2	3	4	5	6	7	8												
-8	-7	-6	-5	-4	-3	-2	-1													

OBS: -0 É A MESMA COISA QUE 0.

IMPORTANTE

STRINGS

- Podemos **particionar**, ou fatiar as strings, além de indexar. Enquanto que a **indexação**, seleciona um único caracter, o **fatiamento** seleciona parte da string (substring).

IMPORTANTE

STRINGS

```
>>> sobrenome = "Giancoli"
```

```
>>> sobrenome[0:2]
```

```
'Gi'
```

```
>>> sobrenome[2:6]
```

```
'anco'
```

```
>>> sobrenome[6:8]
```

```
'li'
```

```
>>> sobrenome[6:20]
```

```
'li'
```

```
>>> |
```

OBS: A DIFERENÇA ENTRE OS ÍNDICES INDICA A QUANTIDADE DE CARACTERES A SEREM SELECIONADOS. OU, O SEGUNDO ÍNDICE SEMPRE É EXCLUDENTE, EXCLUSIVE.

+---+---+---+---+---+---+---+---+---+---+																	
	G		i		a		n		c		o		l		i		
+---+---+---+---+---+---+---+---+---+---+																	
0	1	2	3	4	5	6	7	8									
-8	-7	-6	-5	-4	-3	-2	-1										

OBS: MESMO UTILIZANDO UM ÍNDICE MAIOR QUE O INTERVALO, NO PARTICIONAMENTO, NÃO OCORRE ERRO. SÃO MANIPULADOS NORMALMENTE.

IMPORTANTE

STRINGS

```
>>> sobrenome = "Giancoli"
```

```
>>> sobrenome[:2]
```

```
'Gi'
```

```
>>> sobrenome[2:]
```

```
'ancoli'
```

```
>>> sobrenome[-2:]
```

```
'li'
```

```
>>> sobrenome[-7:]
```

```
'iancoli'
```

```
>>> |
```

+	-	-	-	+	-	-	-	+	-	-	-	+	-	-	-	+	-	-	-	+
	G		i		a		n		c		o		l		i					
+	-	-	-	+	-	-	-	+	-	-	-	+	-	-	-	+	-	-	-	+
0		1		2		3		4		5		6		7		8				
-8		-7		-6		-5		-4		-3		-2		-1						

OBS: O 1º. ÍNDICE OMITIDO POR PADRÃO É 0 (ZERO).
O 2º. ÍNDICE OMITIDO POR PADRÃO É O TAMANHO DA
STRING QUE ESTÁ SENDO PARTICIONADA.

IMPORTANTE

STRINGS

- As sequências de caracteres do Python são **imutáveis**.
- **Não é possível modificar um caracter de uma string atribuindo à uma posição indexada**, pois gerará **erro**.

```
>>> sobrenome = "Giancoli"
```

```
>>> sobrenome[5]
```

```
'o'
```

```
>>> sobrenome[5] = 'i'
```

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
TypeError: 'str' object does not support item assignment
```

```
>>> |
```

IMPORTANTE

STRINGS

- Podemos utilizar o **método `replace()`** para efetuar a atribuição. Porém, **atenção**, o método não efetua a substituição na origem, ***ele gera uma cópia com a modificação.***

```
>>> sobrenome = "Giancoli"
>>> sobrenome.replace(sobrenome[5], 'i')
'Giancili'
>>> sobrenome
'Giancoli'
>>> |
```


IMPORTANTE

STRINGS

- Poderemos utilizar a função **len(variável)** para obter o tamanho da string.

```
>>> sobrenome = "Giancoli"  
>>> len(sobrenome)  
8  
>>> |
```

IMPORTANTE LISTAS

IMPORTANTE

LISTAS

- O Python possui diversos tipos de dados compostos utilizados para agrupar valores. O mais versátil, isto é, sujeito a mudanças, é a **Lista**.
- A **Lista** pode ser escrita separada por **vírgula** , e entre **colchetes** []. Pode conter **tipos de dados diferentes**, mas no geral, possuem dados do mesmo tipo.
- Exemplos:

```
>>> lab = ['A401', 'A402', 'A403', 'A404', 'A405', 'A406', 'A407', 'A408']
>>> lab
['A401', 'A402', 'A403', 'A404', 'A405', 'A406', 'A407', 'A408']
>>> |
```

IMPORTANTE

LISTAS


+-----+-----+-----+-----+-----+-----+-----+-----+								
'A401'	'A402'	'A403'	'A404'	'A405'	'A406'	'A407'	'A408'	
+-----+-----+-----+-----+-----+-----+-----+-----+								
0	1	2	3	4	5	6	7	8
-8	-7	-6	-5	-4	-3	-2	-1	

IMPORTANTE

LISTAS

- Da mesma maneira que as Strings, as Listas podem ser indexadas e particionadas. Ao serem **particionadas** geram uma nova lista.
- Exemplos:

```
>>> lab = ['A401', 'A402', 'A403', 'A404', 'A405', 'A406', 'A407', 'A408']
>>> lab[0]
'A401'
>>> lab[-1]
'A408'
>>> lab[-3:]
['A406', 'A407', 'A408']
>>> len(lab)
8
>>> |
```



+-----+-----+-----+-----+-----+-----+-----+-----+								
'A401'	'A402'	'A403'	'A404'	'A405'	'A406'	'A407'	'A408'	
+-----+-----+-----+-----+-----+-----+-----+-----+								
0	1	2	3	4	5	6	7	8
-8	-7	-6	-5	-4	-3	-2	-1	

IMPORTANTE

LISTAS

- As **Listas** também aceitam concatenação.
- Exemplos:

```
>>> seq1 = [11, 22, 33, 44]
>>> seq2 = [10, 20, 30, 40]
>>> seq1 + seq2
[11, 22, 33, 44, 10, 20, 30, 40]
>>> |
```

IMPORTANTE

LISTAS

```
>>> numeros
```

```
[11, 22, 33, 44, 10, 20, 30, 40]
```

```
>>> numeros[3]
```

```
44
```

```
>>> numeros[3] = 55
```

```
>>> numeros[3]
```

```
55
```

```
>>> numeros
```

```
[11, 22, 33, 55, 10, 20, 30, 40]
```

```
>>> |
```

AO CONTRÁRIO DAS STRINGS, AS LISTAS PERMITEM MODIFICAÇÃO POR MEIO DE ATRIBUIÇÃO AOS ÍNDICES.

IMPORTANTE

LISTAS

- Podemos adicionar novos itens ao final da **Lista**. Para tal, utilizamos o **método append()**.
- Exemplos:

```
>>> seq2 = [10, 20, 30, 40]
>>> seq2.append(50)
>>> seq2
[10, 20, 30, 40, 50]
>>> |
```

IMPORTANTE

LISTAS

- Podemos efetuar uma cópia da **Lista**.
- Exemplos:

```
>>> seq2 = [10, 20, 30, 40]
>>> seq2[:]
[10, 20, 30, 40, 50]
>>> |
```

IMPORTANTE

LISTAS

```
>>> seq3 = seq2[:]
```

```
>>> seq3
```

```
[10, 20, 30, 40, 50]
```

```
>>> seq3[2:4]
```

```
[30, 40]
```

```
>>> seq3[2:4] = []
```

```
>>> seq3
```

```
[10, 20, 50]
```

```
>>> seq3[:] = []
```

```
>>> seq3
```

```
[]
```

```
>>> |
```

PODEMOS EFETUAR UMA CÓPIA E
TAMBÉM LIMPAR A **LISTA**.

A LISTA **SEQ3** É UMA CÓPIA DA
LISTA **SEQ2**.

AO EXECUTAR A ATRIBUIÇÃO DE
COLCHETES PARA UMA PARTE DA
LISTA, ESTA PARTE É **REMOVIDA**.

AO EXECUTAR A ATRIBUIÇÃO DE
COLCHETES PARA **SEQ3[:]**, A LISTA
É **REMOVIDA**.

PERGUNTAS ?

REFERÊNCIAS

<http://python.org.br>