



**INSTITUTO FEDERAL**  
Paraíba  
Campus João Pessoa

# BEM VINDOS



# Strings

# Strings

- Strings são sequências de caracteres (geralmente usada para textos)
- Elas são delimitadas por aspas duplas ("" ) ou aspas simples('')
- Funciona como um array de caracteres, acessados por index :)
- Strings são **imutáveis**, ou seja, seus valores não podem ser alterados.

# Strings - Imutabilidade

- No caso de reatribuição na mesma variável de string, o python entende como uma nova criação, e não uma alteração.

```
1 nome = 'Valéria'
2 print(id(nome))
3 nome = 'Zé Neto'
4 print(id(nome))
```



Output:  
140719467532464  
140719467538704

- No exemplo acima, apesar das variáveis terem o mesmo nome, o seu identificador é diferente, logo, ela foi criada novamente na linha 3

# Strings - Imutabilidade

- Por serem imutáveis, as strings não aceitam alteração direta



```
1 nome = 'Zé'  
2 nome[1] = 'u'
```



```
nome[1] = 'u'  
TypeError: 'str' object does not support item assignment
```

# Strings - Acesso

- As strings podem ser acessadas via index :), como os arrays.

```
1 nome = 'Valéria'  
2 print(nome[4])  
3 print(nome[:4])  
4 print(nome[4:])
```



Output:  
r  
Valé  
ria

Inversão da string

```
1 nome = 'Valéria'  
2 print(nome[::-1])
```



Output:  
airélaV

# Strings - Ascii

- As strings podem ser acessadas via códigos números da tabela ascii (prox slide) :D
- No Python utilizamos o `ord()` para retornar o código decimal a partir do sinal gráfico e `chr()` para retornar o sinal gráfico a partir do código decimal



```
1 print(chr(65))
2 print(ord('A'))
3 print(ord('a'))
4 print(chr(100))
```



Output:

A  
65  
97  
d

# ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	



# Strings - Ascii

- Também é possível fazer a comparação direto das strings :)
  - Exemplo:

```
1 string = '0 dia esta lindo'
2
3 for i in string:
4     if i >= 'A' and i <= 'Z':
5         print(f'{i} é uma letra maiuscula')
6     elif i >= 'a' and i <= 'z':
7         print(f'{i} é uma letra minuscula')
8     elif i.isdigit():
9         print(f'{i} é um número :0')
10    else:
11        print(f'{i} é um caracter especial')
```

```
0 é uma letra maiuscula
 é um caracter especial
d é uma letra minuscula
i é uma letra minuscula
a é uma letra minuscula
 é um caracter especial
e é uma letra minuscula
s é uma letra minuscula
t é uma letra minuscula
a é uma letra minuscula
 é um caracter especial
l é uma letra minuscula
i é uma letra minuscula
n é uma letra minuscula
d é uma letra minuscula
o é uma letra minuscula
```

# Strings - Operadores

- Alguns operadores podem ser utilizados em string, + serve para concatenação, ==, !=, is, is not para comparação.

```
1 a = 'Zé'
2 b = 'Zé'
3 c = 'Shingeki'
4 d = 'Ruim'
5
6 if a == b: #a is b
7     print('Strings Iguais :')
8 else:
9     print('Strings Diferentes :')
10 print()
11 if c is not d: #a != b
12     print(f'{c} não é {d}')
13 else:
14     print(f'{c} é {d}')
```

Output:  
Strings Iguais :)

Shingeki não é Ruim

```
1 strings = 'Verde, Vermelho, Azul'
2 strings += ', Amarelo'
3 print(strings)
4 print()
5
6 lista_strings = ['Olá', 'Tudo', 'Bem?']
7 string_unica = ''
8 for i in lista_strings:
9     string_unica += i + ' '
10 print(string_unica)
```

Output:  
Verde, Vermelho,  
Azul, Amarelo

Olá Tudo Bem?

# Strings - Funções

- **len(string)** - Retorna o tamanho da string :)

```
1 string = 'IFPB'
2
3 print(len(string))
```

→ 4

- **replace(old, new)** - Retorna uma cópia da string com trocando as substrings

```
1 string = 'IFPB'
2 print(string.replace('B', 'E'))
3 print(string.replace('I', 'U'))
```

→ IFPE  
UFPB

- **find(string)** - Retorna a posição da substring procurada  
(-1 para não encontrada)

```
1 string = 'IFPB'
2 print(string.find('a'))
3 print(string.find('P'))
```

→ -1  
2

- **split()** - Desmembra a string (em uma lista)  
(-1 para não encontrada)

```
1 string = 'POP é agro POP é tech POP é tudo'
2 print(string.split())
```

→ ['POP', 'é', 'agro', 'POP', 'é', 'tech', 'POP', 'é', 'tudo']

# Strings - Funções

- **strip()** - Retorna uma cópia retirando (por padrão) os espaços do começo e do fim

```
1 string = '      POP é agro POP é tech POP é tudo      '  
2 print(string.strip())
```

POP é agro POP é tech POP é tudo

- **rstrip()** e **lstrip()** - Retorna uma cópia retirando os espaços (padrão) da direita e da esquerda, respectivamente

```
1 string = '      POP é agro POP é tech POP é tudo      '  
2 print(string.rstrip())  
3 print(string.lstrip())
```

POP é agro POP é tech POP é tudo  
POP é agro POP é tech POP é tudo

- **upper()** - Retorna uma cópia da string com todas as letras maiúsculas

```
1 string = 'ifpb'  
2 print(string.upper())
```

IFPB

- **lower()** - Retorna uma cópia da string com todas as letras minúsculas

```
1 string = 'IFPB'  
2 print(string.lower())
```

ifpb

# Strings - Funções

- **isalpha()** - Retorna True se todos os caracteres são alfabéticos

```
1 string = 'IFPB'
2 print(string.isalpha())
3 string = 'IFPB 12312312'
4 print(string.isalpha())
```

True  
False

- **isdigit()** - Retorna True se todos os caracteres são dígitos

```
1 string = 'IFPB'
2 print(string.isdigit())
3 string = '12312312'
4 print(string.isdigit())
```

False  
True

- **count(sub)** - Retorna a quantidade de vezes que a substring é encontrada na string

```
1 string = 'O sol está lindo hoje'
2 print(string.count('o'))
3 print(string.count('e'))
4 print(string.count(' '))
```

3  
2  
4

- **join()** - Une as strings passadas em uma só string

```
1 lista_string = ['Olá', 'tudo', 'bem?']
2 print(''.join(lista_string))
3
4 lista_string = ['8', '+', '8', '=', '16']
5 print(' '.join(lista_string))
```

Out: Olátudo bem?  
8 + 8 = 16

# Exercícios :)

- **1272 - Mensagem Oculta**
- **1257 - Array Hash**
- **1235 - De Dentro para Fora**
- **1024 - Criptografia**

**OBRIGADO :D**