

Programação em Python



CEFET-MG



Nomeando Variáveis



- **Regras (Obrigatório)**
 - Evitar nomes reservados (Óbvio!).
 - Nomes de variáveis podem conter
 - Letras.
 - Números.
 - Underscores.
 - Podem **começar** com uma letra ou underscore, mas **não** com um **número**.
 - **Espaços são proibidos.**

- **Boas Práticas (PEP 8 - Não é Obrigatório, mas...)**
 - Nomes de variáveis devem ser:
 - **Concisos e descritivos**
 - **Evite abreviações**
 - **Letras minúsculas.**
 - **Palavras separadas por underscore.**
 - **Singular.**
 - **Indique o tipo se necessário (booleano).**
 - **Para CONSTANTES, use MAIÚSCULAS com underline.**
 - **Comente o óbvio**

- **Boas Práticas (PEP 8 - Não é Obrigatório, mas...)**
 - Ex.:
 - **nome** melhor que n, no, nm.
 - **student_name** melhor que s_n, sn, studentn, sname.
 - **name_length** melhor que length_of_persons_name.
 - **is_admin** melhor que admin
 - **tem_estoque** melhor que estoque
 - **usuario_ativo** melhor que **usuario**

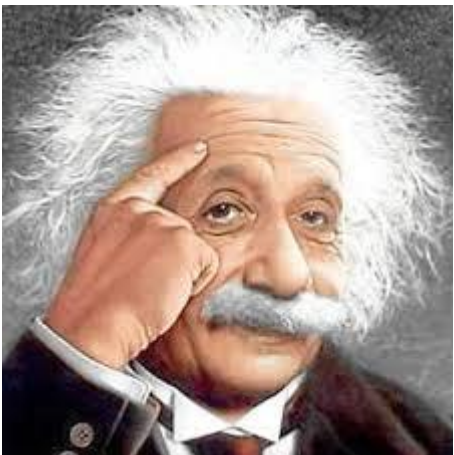
- **Boas Práticas (PEP 8 - Não é Obrigatório, mas...)**
 - Ex.:
 - **PI**
 - **RAIO**
 - **TAXA_JUROS**
 - **INFINITO**
 - **LIMITE_MAXIMO**

Nomeando Variáveis

- Boas Práticas (**PEP 8** - Não é Obrigatório, mas...)

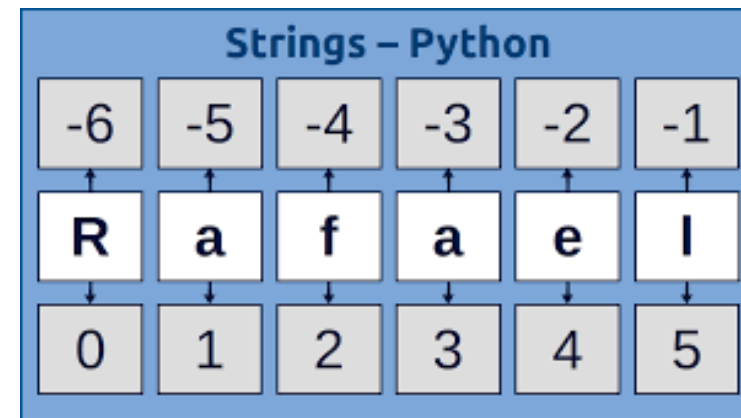
Bom programador não é aquele que
acerta sempre, mas que responde a esses
erros de modo eficiente!

TODOS ERRAM!!



- **Boas Práticas (PEP 8 - Não é Obrigatório, mas...)**
 - **Primeiro Erro**
 - **NameError** (Será importante reconhecer o tipo de erro no futuro).
 - **Digitação.**
 - **Ex.:**
 - mensagem = “Este é meu primeiro erro”
 - `print(mensagem)`

Manipulação de strings



- Série de Caracteres.
- Caractere único.
- Tudo entre “aspas” ou ‘apóstrofes’.
- Ex:
 - frase = “Python é minha linguagem favorita”
 - frase = ‘Até que ‘C’ não das piores’
 - frase = ‘Se começar com “apóstofre” não pode repetir como string (Erro de Sintaxe)!’

- **Funções principais** (Não são todas!)

- Modificação de texto:

- `.lower()` `"Python".lower()` `'python'`
- `.upper()` `"Python".upper()` `'PYTHON'`
- `.capitalize()` `"python".capitalize()` `'Python'`
- `.title()` `"python é legal".title()` `'Python É Legal'`

- **Funções principais (Não são todas!)**
 - Modificação de texto:
 - Ex.:
 - texto = "Python"
 - print(texto.lower())
 - frase = 'c é melhor que python, kkkkk'
 - print(frase.upper())
 - texto = 'ada lovelace'
 - print(texto.title())

- **Funções principais (Não são todas!)**

- Modificação de texto:

- `.strip()` `" texto ".strip()` `'texto'`

- `.lstrip()` `" texto ".lstrip()` `'texto '`

- `.rstrip()` `" texto ".rstrip()` `' texto'`

- `.replace(a, b)` `"ana".replace("a", "o")` `'ono'`

- **Funções principais (Não são todas!)**
 - Modificação de texto:
 - Ex.:
 - texto = " Python "
 - print(texto.strip())
 - frase = 'c **ainda** é melhor que python, kkkkk'
 - print(frase.strip())

- **Funções principais** (Não são todas!)

- Busca e Verificação:

- `.startswith("var")`

- `"abcdef".startswith("abc")`

True

- `.endswith("var ")`

- `"testando.py".endswith(".py")`

True

- `.find("var")`

- `"exemplo".find("emp")`

2 (posição)

- **Funções principais** (Não são todas!)

- Busca e Verificação:

- Ex.:

- `texto = "Python"`
- `print(texto.startswith("P"))`
- `print(texto.startswith("p"))`
- `print(texto.find('o'))`
- `print(texto.find('a'))`
- `print(texto.find('p'))`
- `print(texto.find('P'))`

- **Funções principais** (Não são todas!)

- Busca e Verificação:

- `.index("var")`

- `"exemplo".index("e")`

0 (posição, mas ValueError se não achar)

- `.count("var")`

- `"banana".count("a")`

3

- `len('texto')`

- `len('Frase ou texto')`

14 (quant. de caracteres)

- **Funções principais (Não são todas!)**

- Busca e Verificação:

- Ex.:

- `texto = "Python"`
- `print(texto.index("P"))`
- `print(texto.index("p"))`
- `print(len(texto))`

- **Funções principais** (Não são todas!)

- Outras:

- `.split()`

- `"um dois três".split()` `['um', 'dois', 'três']` (*lista[]*)

- `"a,b,c".split(",")` `['a', 'b', 'c']`

- `.partition('sep')`

- `'email@dominio.com'.partition('@')` `('email', '@', 'dominio.com')` (**tupla**)

- `.join()`

- `" ".join(['um', 'dois'])` `'um dois'`

- **Funções principais (Não são todas!)**
 - Outras:
 - Ex.:
 - texto = 'Não adianta , C continua melhor que Python'
 - `print(texto.split())`
 - texto = '1000,00'
 - `print(texto.split(','))`
 - texto = "email@dominio.com"
 - `print(texto.partition('@'))`

- **Funções principais** (Não são todas!)

- Outras:

- `.center(10, ".")`

`"oi".center(10, ".")`

`'....oi....'`

- `.ljust(10, "-")`

`"oi".ljust(10, "-")`

`'oi-----'`

- `.rjust(10, "*")`

`"oi".rjust(10, "*")`

`'*****oi'`

- `.zfill(5)`

`"42".zfill(5)`

`'00042'`

- `.isnumeric()`

`"42.0".isnumeric()`

`False`

- **Funções principais (Não são todas!)**

- Outras:

- Ex.:

- `texto = "oi".center(9)`

- `print(texto)`

- `texto = '10'`

- `print(texto.zfill(4))`

- `texto = 10`

- `print(texto.zfill(4))`

- `AttributeError: 'int' object has no attribute 'zfill'`

Vamos Programar!!
WOO HOO!!



- **Exercício: Separador de Nome**
- Peça que o usuário digite o nome completo. O programa deve exibir:
 - O nome com todas as letras maiúsculas.
 - O nome com todas as letras minúsculas.
 - Quantos caracteres há sem considerar espaços.
 - Quantas letras tem o primeiro nome.
- Ex.:
 - Entrada: Digite seu nome completo: Ana Clara Souza
 - Saída: Maiúsculas: ANA CLARA SOUZA; Minúsculas: ana clara Souza; Total de letras (sem espaços): 14; Primeiro nome tem 3 letras

• Exercício: Separador de Nome

- `nome_completo = input("Digite seu nome completo: ")`
- `print("Maiúsculas:", nome_completo.upper())`
- `print("Minúsculas:", nome_completo.lower())`

- `# Remove espaços para contar caracteres`
- `total_sem_espacos = len(nome_completo.replace(" ", ""))`
- `print("Total de letras (sem espaços):", total_sem_espacos)`

- `# Divide em partes e pega o primeiro nome`
- `primeiro_nome = nome_completo.split()[0]`
- `print(f"Primeiro nome tem {len(primeiro_nome)} letras")`

- **Exercício: Contador de Letras Específicas**
- Peça ao usuário que digite uma frase e conte quantas vezes a letra “a” (ou “A”) aparece.
 - Exemplo de entrada: Digite uma frase: Amanhã é dia de aula
 - Saída esperada: Número de letras 'a' ou 'A': 5

- **Exercício: Contador de Letras Específicas**

- frase = input("Digite uma frase: ")
- # Converte tudo para minúsculo e conta quantos 'a' aparecem
- quantidade = frase.lower().count('a')
- print("Número de letras 'a' ou 'A':", quantidade)