

Programação Estruturada e Orientada a Objetos com Python

Módulo 1

Preâmbulo: Neste módulo você fará o setup do ambiente de trabalho, irá se familizar com o linux e ambiente de entrega de projetos da 42 e aprenderá os básicos da linguagem de programação Python.

Versão: 1.0

Sumário

T	instruções importantes	4
II	Antes de começar	4
ш	Exercício 1: Início	6
IV	Exercício 2: Variáveis	7
\mathbf{V}	Exercício 3: Concatenação de strings	8
VI	Exercício 4: Operações com inteiros	9
VII	Exercício 5: Valores Booleanos	10
VIII	Exercício 6: Conversão de Tipos	11
IX	Validação	12
\mathbf{X}	Exercício 7: Classes possuem métodos	13
XI	Exercício 8: Loop	14
XII	Entrega e Avaliação entre Pares	15

Capítulo I

Instruções Importantes

- Use esta página como sua única referência. Não escute rumores ou especulações sobre como preparar sua solução.
- Leia todo o documento com atenção antes de iniciar. Dicas valiosas podem estar ao final de cada exercício.
- Leia os Exemplos com Atenção. Eles podem exigir coisas que não estão especificadas no enunciado.
- Para ser avaliada, sua solução deve estar no seu repositório GIT.
- Suas soluções serão avaliadas pelos seus colegas de curso.
- Você **não deve deixar outros arquivos** no seu diretório além daqueles explicitamente especificados nas instruções do exercício. É recomendado criar um arquivo .gitignore para evitar acidentes.
- Quando precisar obter uma saída precisa nos seus programas, **você não deverá** exibir uma saída pré-calculada em vez de realizar o exercício corretamente.
- Tem uma dúvida? Pergunte para o seu colega à direita. Se isso não funcionar, tente o colega à esquerda.
- Como materiais de referência, você pode utilizar o seus pares, o manual das ferramentas, e internet em geral.



Aqui, o aprendizado acontece por meio da experimentação, da tentativa e erro e da colaboração com seus colegas. Se estiver com dificuldades, peça ajuda a seus colegas e, juntos, busquem fontes confiáveis.

Evite usar IA para gerar código ou resolver exercícios por você. Aprender a enfrentar desafios sem atalhos fortalecerá suas habilidades de maneira duradoura.

- Aqui e adiante, usamos o **Python 3.10** como referência. Você pode checar a versão com o comando python3 -V
- A menos que especificado de forma diferente em determinados exercícios, todos os componentes da **biblioteca padrão do python** podem ser utilizados.
- O uso das bibliotecas e ferramentas externas ruff, mypy e pytest também é permitido (e mandatório, a partir de determinados exercícios.)
- O uso de demais bibliotecas externas deverá estar explícito no enunciado do exercício.

E que a Força esteja com você!

Capítulo II

Antes de começar

Seus projetos serão sempre entregues utilizando repositórios Git. Para que você possa acessar estes repositórios, criados automaticamente pela Intranet da 42, você deve criar um par de chaves SSH, e inserir a chave pública na página dedicada às chaves SSH em seu perfil na intranet. Para isto, siga as instruções abaixo:

- 1. Abra o terminal e digite: ssh-keygen -t ed25519
- 2. Pressione a tecla ENTER três vezes (para utilizar as configurações padrão), até que você perceba algo semelhante à isto:

```
+--[ED25519 256]--+
| ..+oo+++o|
| ...++o.|
| .o.+..o|
| .o.+..o|
| .o.B. |
| .oo.oEo. = |
| .=o+...+ o |
| .oo*o.o.o
```

3. Agora, execute o seguinte comando para visualizar sua chave pública:

```
?> cat ~/.ssh/id_ed25519.pub
```

- 4. Pegue a chave SSH exibida e copie. Você deve colocá-la no seu perfil da Intra.
 - Na página do seu perfil na Intra, selecione o seu login no canto superior direito > Settings > SSH Key. Ou, acesse diretamente por este link.
 - Clique em New SSH Key, cole a chave SSH no campo e envie.
- 5. Por fim, configure seu git para utilizar, globalmente, seu nome e email para registrar cada commit:

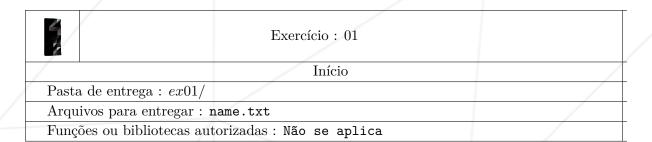
```
?> git config --global user.name "<SEU NOME>"
?> git config --global user.email "<SEU EMAIL>"
```



Utilizamos '?>' como um marcador visual para comandos executáveis, sempre que houver um exemplo de uso do terminal.

Esse prefixo serve apenas como indicação - **ele não deve ser digitado** junto com o comando.

Capítulo III



- 1. Crie uma pasta chamada semana-1 no seu diretório de usuário, e acesse-a.
- 2. Usando o commando git, clone seu repositório deste primeiro projeto para esta pasta. O endereço do repositório está na página de seu projeto na intranet. Se preferir, dê a ele um nome mais amigável, como modulo-1.
- 3. Dentro desta pasta, crie uma pasta chamada ex01.
- 4. Adicione a esta um arquivo name.txt, contendo seu nome e sobrenome.

Exemplo de Saída Experada:

?> cat name.txt
Ford Prefect
?>



Para que este exercício cumpra seu objetivo, realize-o no terminal, utilizando os comandos: 'mkdir', 'cd', 'echo' e 'git'.

Capítulo IV



Exercício: 02

Variáveis

Pasta de entrega : ex02/

Arquivos para entregar : name.py

Funções ou bibliotecas autorizadas : Biblioteca padrão

- 1. Crie um script chamado name.py
- 2. Defina uma variável chamada first_name e a inicialize com o seu primeiro nome.
- 3. Defina uma variável chamada last_name e a inicialize com o seu último nome.
- 4. Mostre ambas variáveis na saída padrão do terminal, seguida por uma quebra de linha.

Exemplo de Saída Experada:

?> python3 name.py
Ford Prefect
?>

Capítulo V



Exercício: 03

Concatenação de strings

Pasta de entrega : ex03/

Arquivos para entregar : full_name.py

Funções ou bibliotecas autorizadas : Biblioteca padrão

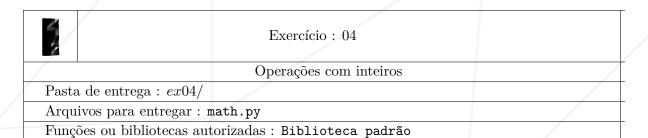
- 1. Crie um script chamado full_name.py contendo ambas variáveis definidas no exercício anterior.
- 2. Defina uma variável chamada full_name que contenha a concatenação de first_name e last_name, separadas por um espaço.
- 3. Mostre o valor da variável full_name no terminal.

Exemplo de Saída Experada:

?> python3 full_name.py
Ford Prefect

?>

Capítulo VI



- 1. Crie um script chamado math.py.
- 2. Defina duas variáveis chamadas a e b, atribuindo a elas valores inteiros de sua escolha.
- 3. Defina as seguintes variáveis com os resultados das operações matemáticas correspondentes:
 - sum_result para armazenar a soma de a e b.
 - diff_result para armazenar a subtração de a por b.
 - prod result para armazenar o produto de a e b.
 - quot_result para armazenar a divisão inteira de a por b.

Exemplo de Saída Experada:

```
?> python3 math.py
Soma: 15
Subtração: 5
Produto: 50
Divisão inteira: 2
?>
```



aproveite este exercício para explorar o modo interativo do python e os **operadores de divisão** desta linguagem.

Capítulo VII



Exercício: 05

Valores Booleanos

Pasta de entrega : ex05/

Arquivos para entregar : boolean.py

Funções ou bibliotecas autorizadas : Biblioteca padrão

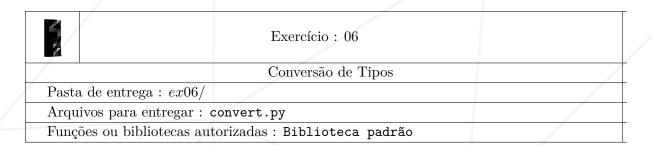
Utilizando apenas duas variáveis, a e b, alcance a saída esperada:

?> python3 boolean.py É igual? False É maior? True É menor? False



Neste exercício você deve utilizar operadores de comparação.

Capítulo VIII



- 1. Defina uma variável num_str contendo o número 42 como uma string.
- 2. Crie um script que alcance a saída esperada:

```
?> python3 convert.py
Valor original 42 <class 'str'>
Como int 42 <class 'int'>
Como float 42.0 <class 'float'>
Como bool True <class 'bool'>
?>
```



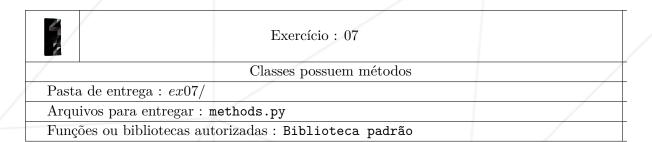
help(type)

Capítulo IX

Validação

- Se você chegou até aqui e executou os exercícios com rigor, já é possível validar este módulo com o resultado final de 80% e já detém os conhecimentos necessários para avançar para os módulos seguintes.
- Os próximos exercícios possibilitam alcançar o resultado de 100% e 125% respectivamente. Avalie a realização destes exercícios considerando a facilidade/dificuldade encontrada na realização dos exercícios anteriores. Busque o equilíbrio entre se desafiar e avançar para os próximos módulos.

Capítulo X

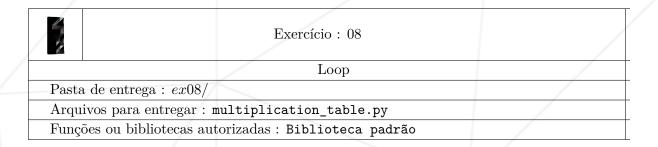


- 1. Defina uma variável contendo uma string com seu nome e sobrenome.
- 2. Utilize os métodos implementados no tipo str para alcançar o seguinte resultado:

```
?> python3 methods.py
É título? True
São maíusculas? False
É númerico? False
É ascii? True
É alfanumérico? False
?>
```

Capítulo XI

Bônus



Alcance a saída esperada ao receber um argumento de linha de comando cujo único valor é um inteiro:

```
?> python3 multiplication_table.py 9
0 x 9 = 0
1 x 9 = 9
2 x 9 = 18
3 x 9 = 27
4 x 9 = 36
5 x 9 = 45
6 x 9 = 54
7 x 9 = 63
8 x 9 = 72
9 x 9 = 81
10 x 9 = 90
?>
```

Capítulo XII

Entrega e Avaliação entre Pares

- Entregue seu projeto em seu repositório *Git* disponível na página do projeto na intranet.
- Apenas o trabalho dentro do seu repositório será avaliado durante a defesa. Não hesite em verificar os nomes de seus arquivos e pastas para garantir que estejam corretos.
- No horário da avaliação, o avaliado se dirigirá à estação de trabalho do aluno avaliador para realizar os testes. Um clone do repositório deverá ser realizado em uma nova pasta, e estes são os arquivos que serão avaliados.