



pythonTM

Aula 02 - Entrada e Saída

UNIP - CIÊNCIA DA COMPUTAÇÃO

DISCIPLINA: IPE

PROFESSORES: CÉLIO e LUCIANA

Instaladores necessários

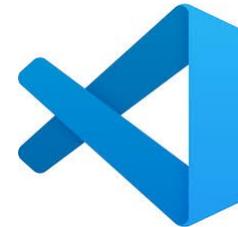
- Visual Studio Code (ambiente de desenvolvimento - IDE)
- Python (linguagem de programação)

Os tutoriais de instalação se encontram na pasta da disciplina - Aula 02

Ou...

Usar a IDE online:

<https://www.online-python.com/>



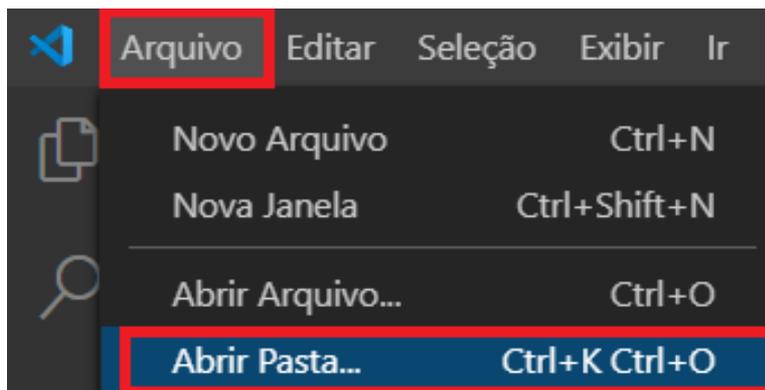
Oi Mundo – VS Code

Passos:

1. Abrir o VS Code



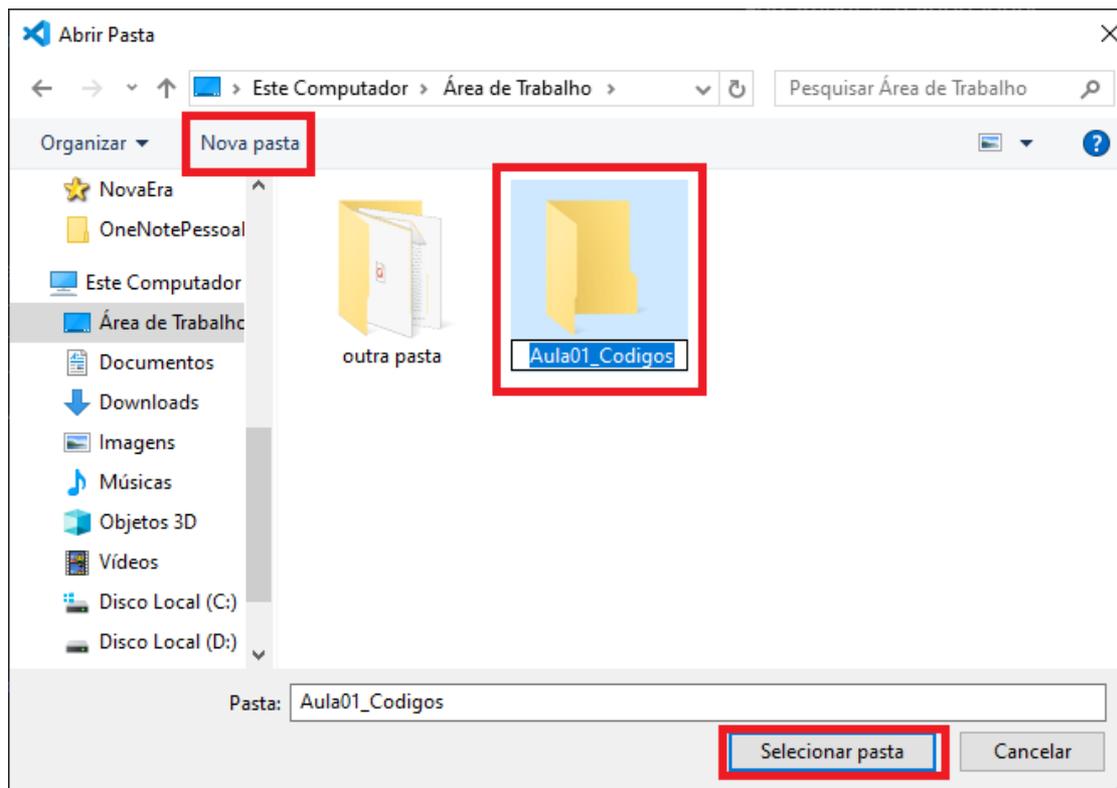
2. Arquivo > Abrir pasta...



Oi Mundo – VS Code

Passos:

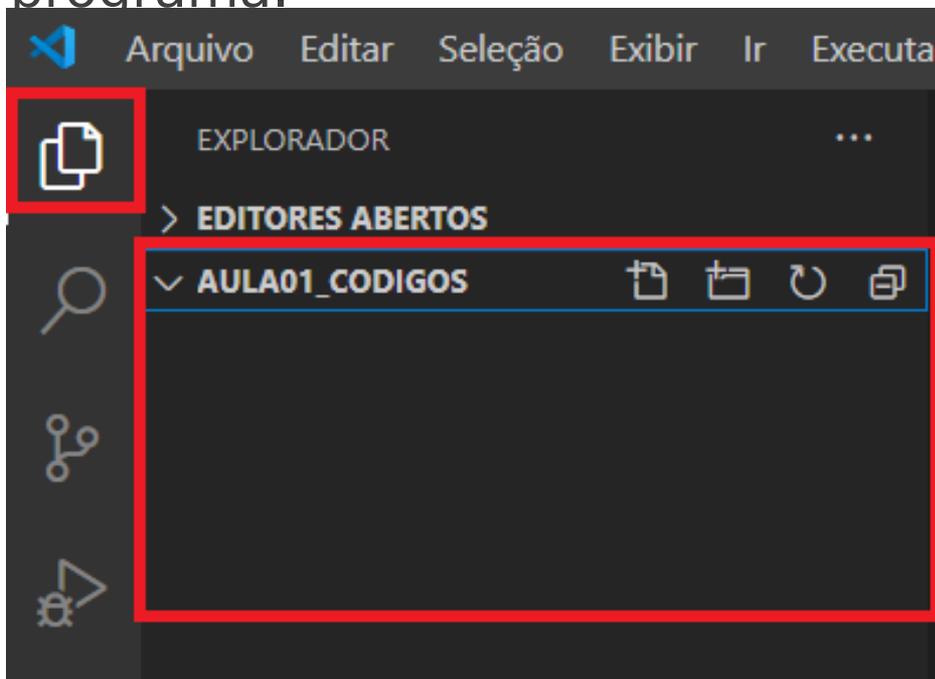
3. Criar uma nova pasta em algum lugar seguro...



Oi Mundo – VS Code

Passos:

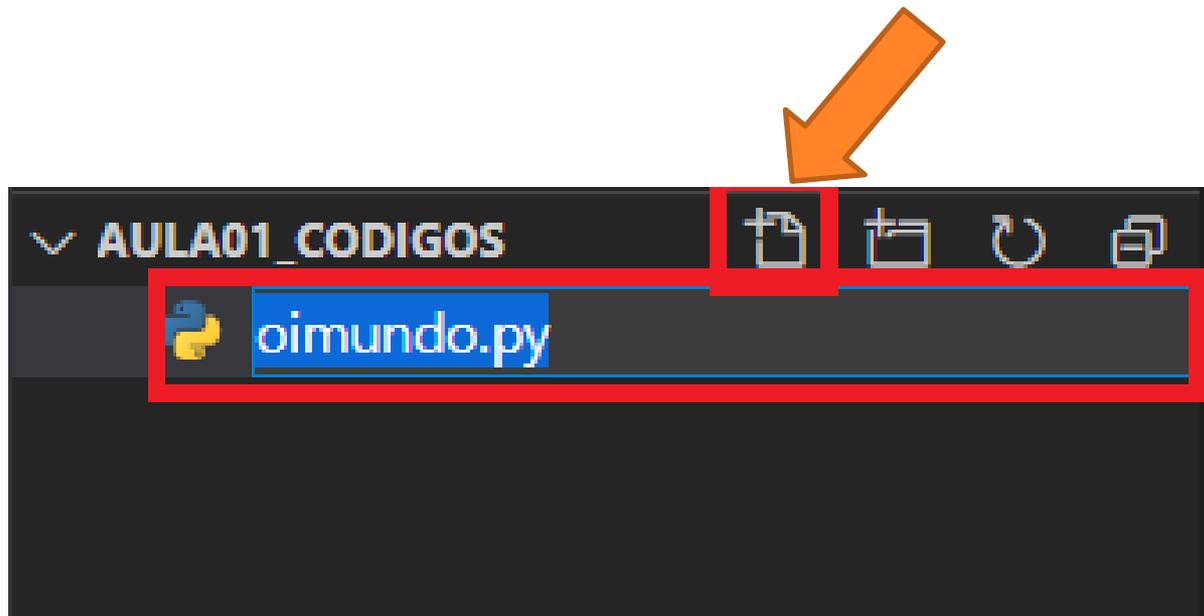
4. Agora todos nossos códigos ficarão organizados em uma única pasta. Já podemos criar o nosso primeiro programa.



Oi Mundo – VS Code

Passos:

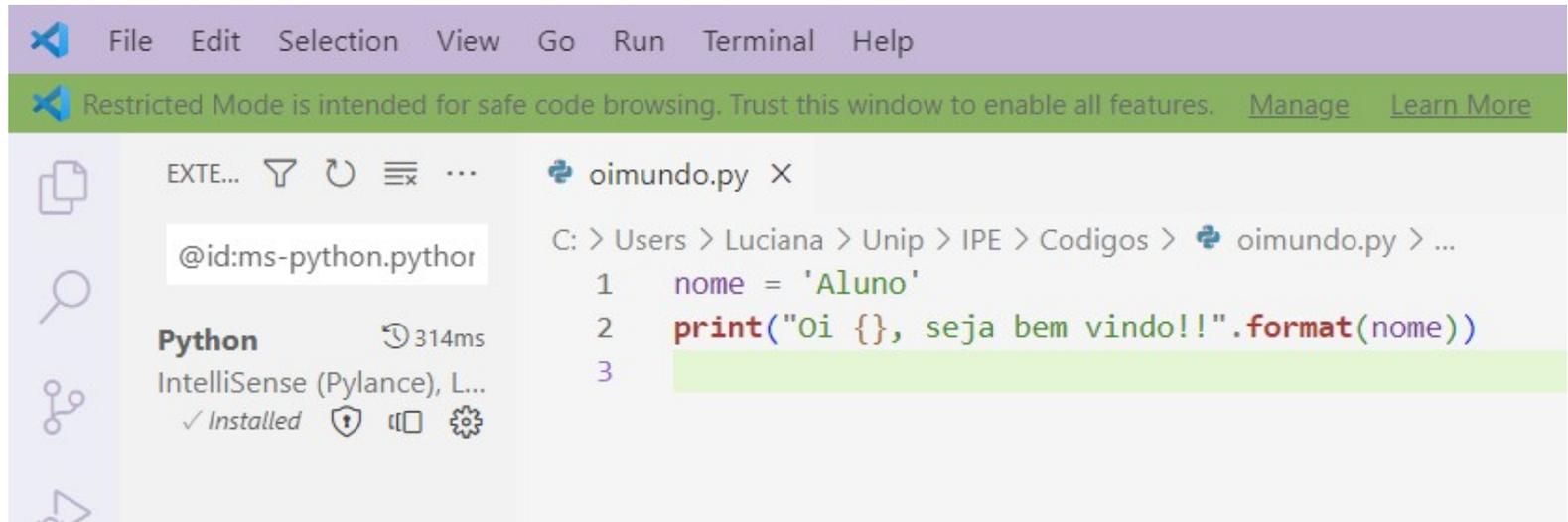
5. Clique no botão novo arquivo e coloque o nome ***oimundo.py***
É OBRIGATÓRIO colocar a extensão (*py*).



Oi Mundo – VS Code

Passos:

6. Digite o código abaixo (coloque seu nome entre as aspas).



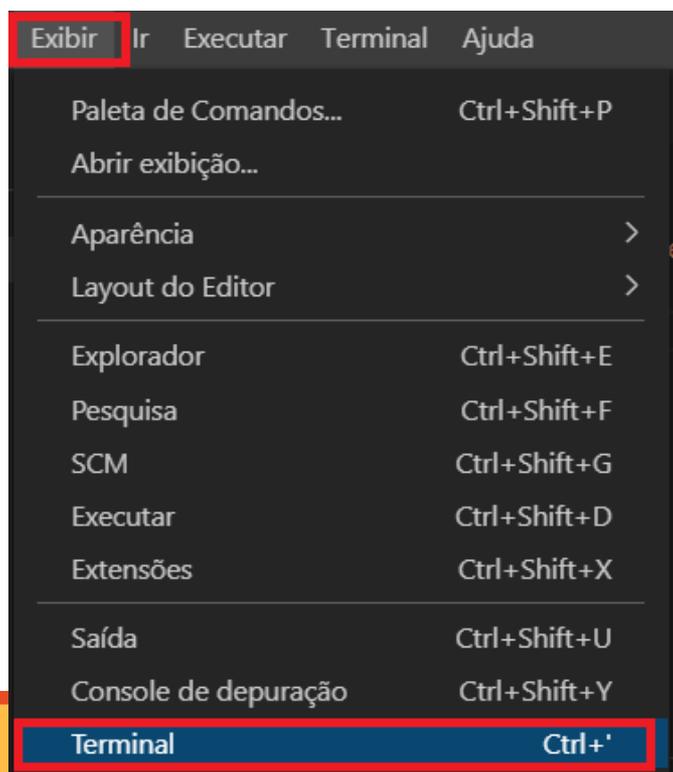
```
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
oimundo.py x
C: > Users > Luciana > Unip > IPE > Codigos > oimundo.py > ...
1 nome = 'Aluno'
2 print("Oi {}, seja bem vindo!!".format(nome))
3
```

Oi Mundo – VS Code

Passos:

7. Salve o arquivo (Ctrl+S)

Depois, abra o terminal (Ctrl+ `)



Oi Mundo – VS Code

Passos:

8. Verifique se o terminal está na pasta correta e depois execute o comando:

python nomeDoArquivo.py



The image shows a screenshot of the Visual Studio Code interface. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'JUPYTER'. The 'TERMINAL' tab is selected and underlined. Below the tabs, the terminal shows two lines of text: 'PS C:\Users\Luciana\Unip\IPE> cd Codigos' and 'PS C:\Users\Luciana\Unip\IPE\Codigos> python oi mundo.py'. The second line is highlighted with a red rectangular box, and a black cursor is visible at the end of the command.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

PS C:\Users\Luciana\Unip\IPE> cd Codigos
PS C:\Users\Luciana\Unip\IPE\Codigos> python oi mundo.py
```

Oi Mundo – VS Code

Passos:

9. Visualize o resultado:

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

```
Oi Aluno, seja bem vindo!!
```

```
PS C:\Users\Luciana\Unip\IPE\Codigos> █
```

Entrada e saída

Entrada

Quando queremos ler um dado (*método input*)

input('frase') -> método que sempre retornará uma **string (texto)**

Saída

Quando queremos imprimir um dado

print(valor1, valor2, valor3 ...)

Tipos primitivos básicos no Python

Consideraremos apenas 3 tipos primitivos básicos de dados no python, são eles:

- String – Texto (frase, palavra, caractere)
- Int – inteiro, não tem casas decimais
- Float – número flutuante (real), possui casas decimais

Outras linguagens possuem outros tipos, alguns exemplos:

- Booleano – aceita apenas TRUE ou FALSE (verdadeiro ou falso)
- Caractere (char) – aceita apenas uma letra ou símbolo
- Ponteiro – aceita endereços de memória (linguagens C e C++)

Criando variáveis no Python

O python possui tipagem dinâmica. Criamos variáveis informando um nome e um valor para ela:

- `a = 1 # cria uma variável chamada a do tipo inteiro, com valor 1 atribuído a ela`
- `valor = 1.3 # cria uma variável chamada valor do tipo float, com valor 1.3 atribuído a ela`
- `meuNome = 'Igor' ou meuNome = "Igor" # cria uma variável chamada meuNome do tipo string, com valor 'Igor' atribuído a ela`

OBS: No python, não existe diferença entre usar aspas ' ou aspas duplas " para as strings

Convertendo dados (cast)

“Converter o dado” ou “fazer um cast” é quando convertemos uma variável de um tipo para outro. Alguns exemplos:

- `a = str(1) # cria uma variável do tipo string, com valor '1'`
- `a = str(1.3) # cria uma variável do tipo string, com valor '1.3'`

- `a = int('1') # cria uma variável do tipo inteiro, com valor 1`
- `a = int(1.3) # cria uma variável do tipo inteiro, com valor 1 (inteiro não aceita casas decimais)`

- `a = float('1.3') # cria uma variável do tipo float, com valor 1.3`
- `a = float(1) # cria uma variável do tipo float, com valor 1.0`

Input - Convertendo dados

Voltando ao método input()....

Assim, para leitura de um valor inteiro, usamos dois comandos:

```
valor = int(input('Digite um número: '))
```

`input('Digite um número: ')` -> faz **sempre** a leitura de uma string

`int(.....)` -> depois o número digitado será convertido para um inteiro

Comentários - # e '''

Comentários são muito importantes para documentação do código.

Qualquer linha comentada não será executada pelo interpretador.

Para realizar o comentário de uma linha usamos #

Essa linha é um comentário

Para múltiplas linhas usamos três aspas simples (abrindo e fechando) '''

''' Essas
linhas estão
comentadas '''

Atribuição

Atribuição significa definir um novo valor para uma variável. A sintaxe é a mesma da declaração (criação) de variáveis.

A = 10 # declaração variável inteiro

A = 30 # como a variável já foi criada na linha acima, aqui é uma atribuição

Diferentemente de outras linguagens, o python permite a tipagem dinâmica:

A = 10 # declaração variável inteiro

A = 'Aluno' # variável A torna-se string com atribuição do valor 'Aluno'

Operações básicas entre valores

Operação	Resultado
$x + y$	Soma
$x - y$	Subtração
$x * y$	Multiplicação
x / y	Divide x por y
$x // y$	Divide x por y e pega o piso (parte inteira) do resultado
$x \% y$	Divide x por y e pega o resto da divisão
$x ** y$	x elevado a y
<code>pow(x,y)</code>	x elevado a y (mesma operação que a linha acima)

Formatando strings

Voltando ao método print().... segue algumas formatações que podem ser utilizadas em uma string:

```
print('{ } {}'.format('primeiro', 'segundo')) # primeiro  
segundo
```

```
print('{1} {0}'.format('primeiro', 'segundo')) # segundo  
primeiro
```

```
print('{:d}'.format(42)) # 42
```

```
print('{:f}'.format(3.141592653589793)) # 3.141593
```

```
print('{:.2f}'.format(3.141592653589793)) # 3.14
```

```
print('{:.3f}'.format(3.141592653589793)) # 3.142
```

Atividades

ATIVIDADES PARA A PROXIMA SEMANA DE AULA:

- **ESTUDAR APOSTILA Python3 ATÉ A PÁGINA 10 (tópico 1.10 COMENTÁRIOS).**
- **ESTUDAR O LIVRO Python Orientação a Objetos ATÉ A PÁGINA 18 (tópico 3.4 INSTRUÇÕES);**
- **RESOLVER A LISTA 0 E LISTA 1 DE EXERCÍCIOS ENVIADA NO GRUPO.**