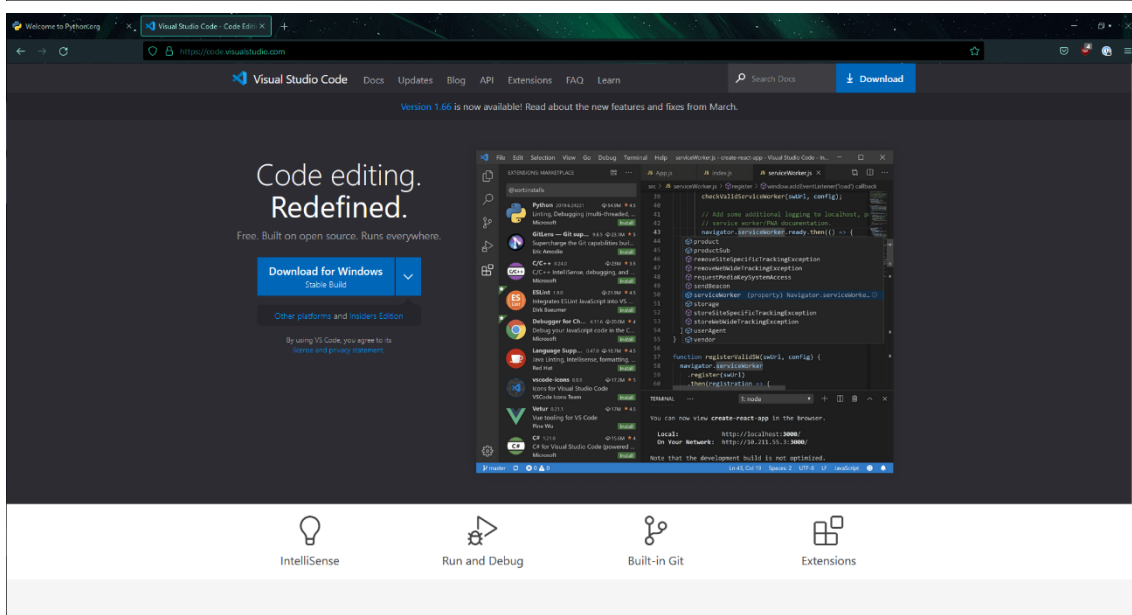
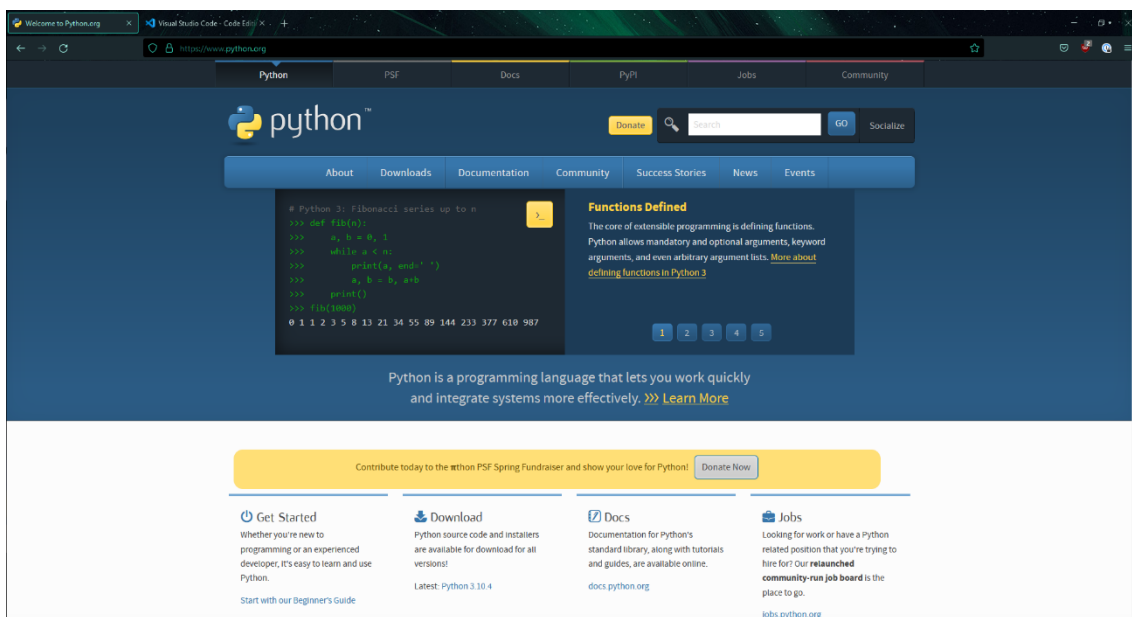


Aula 2

Preparação do ambiente

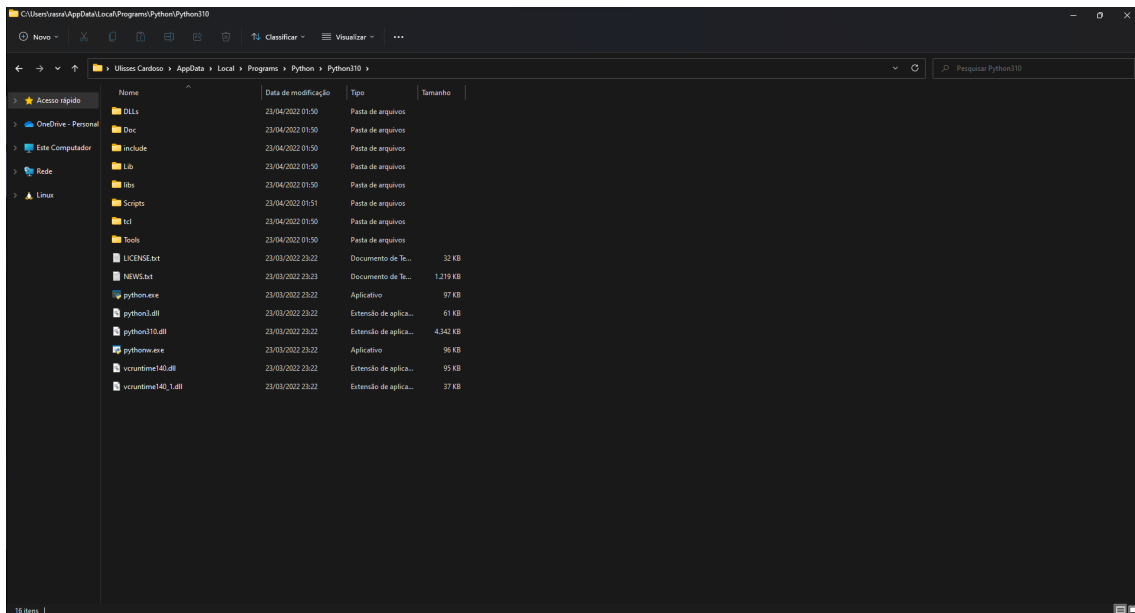
Download e Instalação do Python e Visual Studio Code

Para instalar o python precisa acessar o site: www.python.org. Para baixar o visual studio code precisa acessar o site <https://code.visualstudio.com>.



Depois de baixado é só seguir o passo a passo e já está instalado.

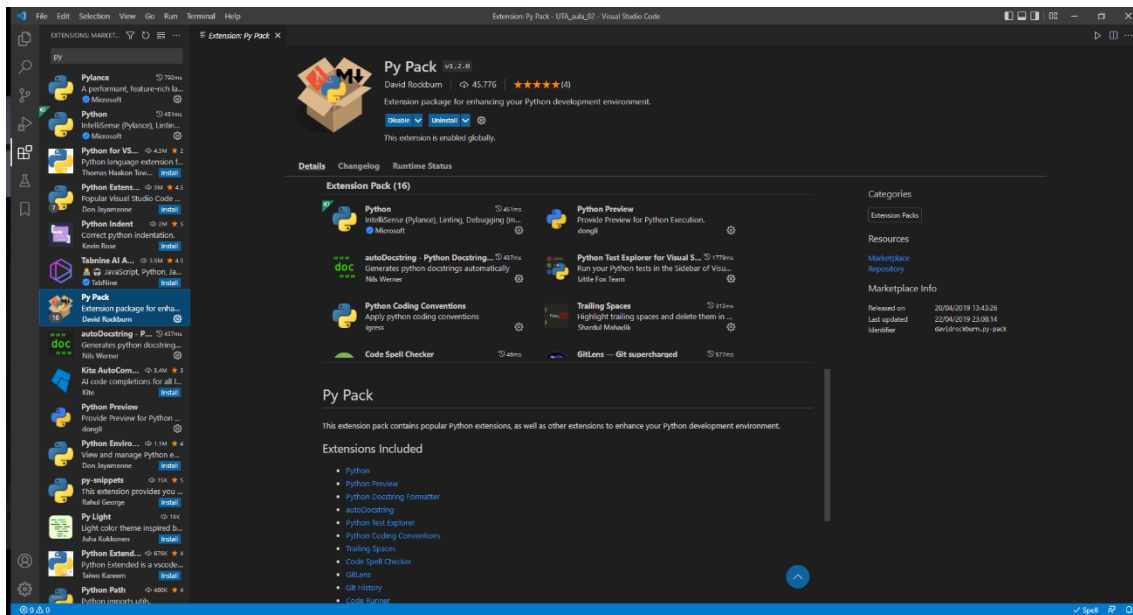
No python é preciso adicionar a variável de ambiente PATH, para isso se precisa abrir o Windows Explorer e na barra de pesquisa digite “%appdata%”. Depois voltar uma pasta atrás e acessar Local/Programs/Python/Python[NumerodaVersão]/.



Na busca do windows digitar “variáveis de ambiente” e clicar no “enter”.

Procura pelo item PATH, clica em “Novo”, cola o endereço que copiou e clica no “ok” três vezes.

No visual studio code, vai na parte de extensão e pesquisa por “py” e procura pela extensão “py pack” que vai instalar o que precisa para usar o python no vscode

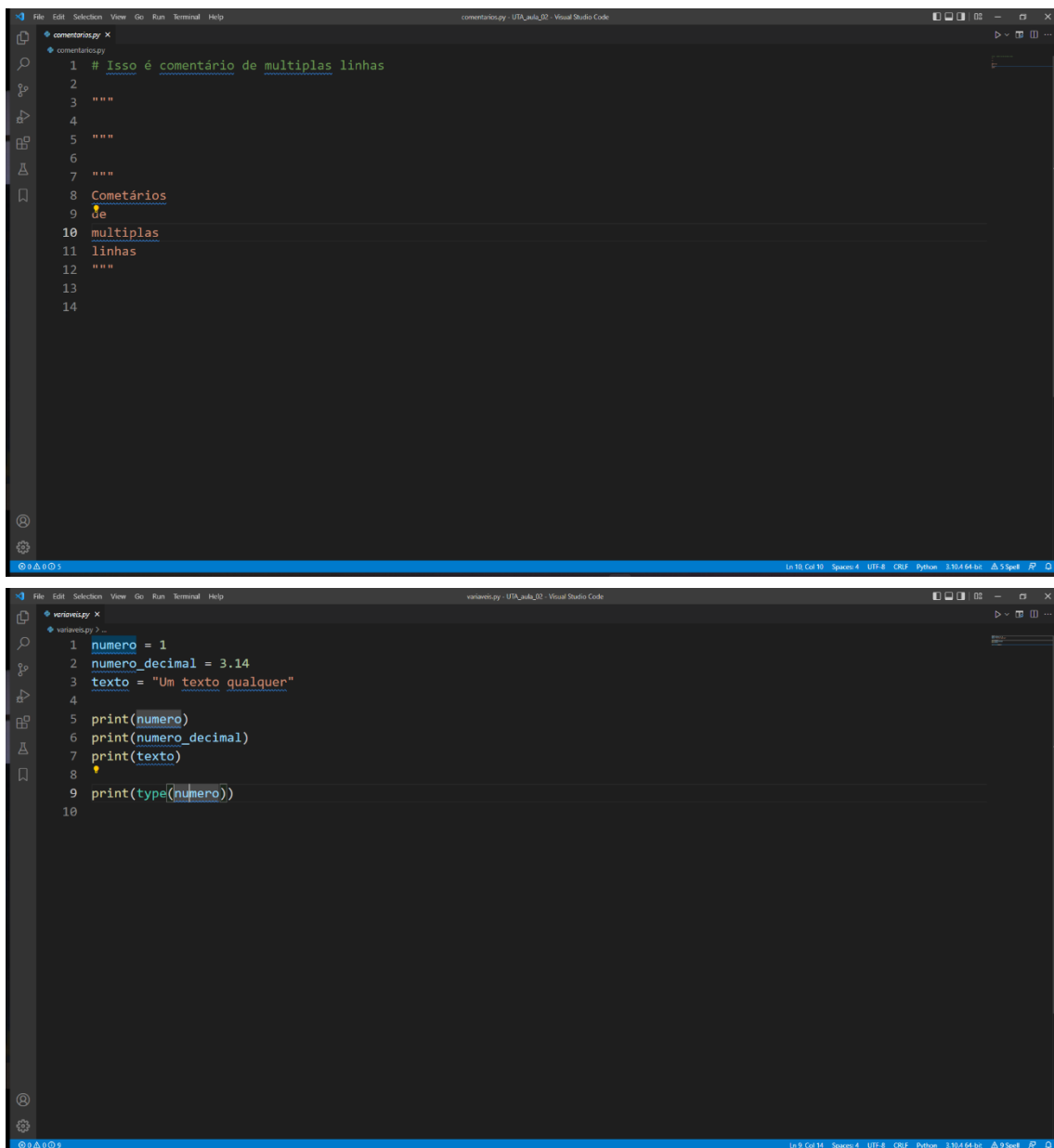


Variáveis, comentários e tipos de dados básicos

Os comentários são linhas que serão ignoradas pelo interpretador durante a execução do programa. Para uma linha é usado a “#” e para múltiplas linhas pode usar seis aspas.

As variáveis são usadas para armazenar na memória valores que posteriormente serão usados em outras partes do programa, por exemplo armazenar int (números sem parte decimal), float (números decimais) e strings (texto). Para saber o tipo da variável usa-se o comando “type()”.

Por exemplo, usar a instrução “input()” para pegar o texto digitado pelo usuário e o armazenar em uma variável, depois usa o comando “print()” para mostrar a variável na tela.

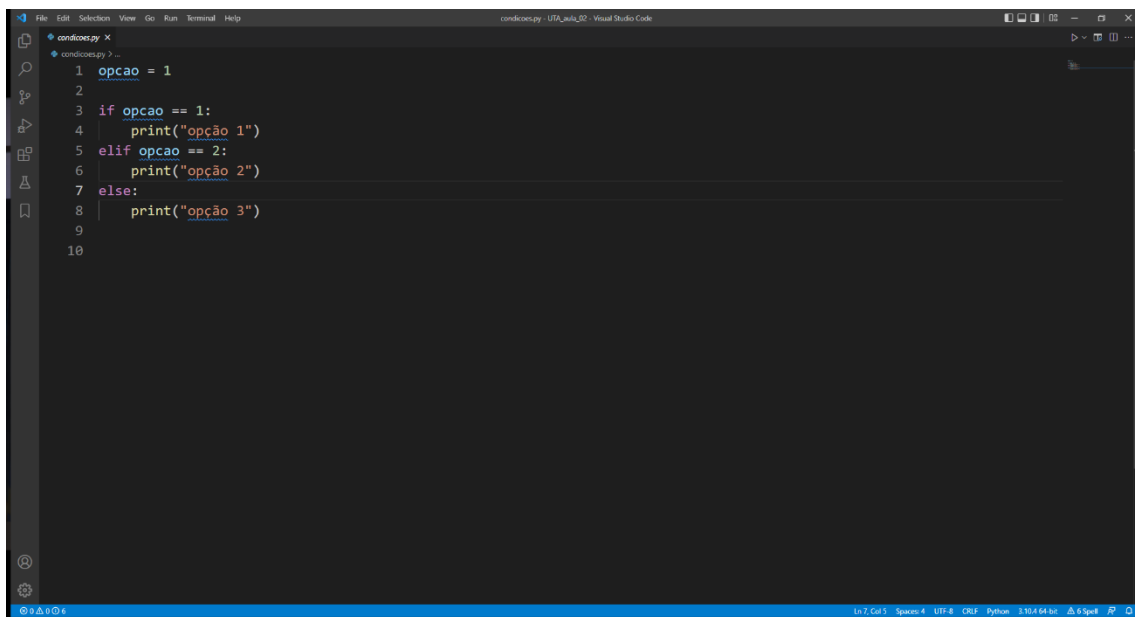


Estruturas de Controle de Fluxo (Condições e Loops)

Como o interpretador do python lê um arquivo de código fonte de linha a linha, às vezes não se precisa ler alguma linha de código ou repetir um conjunto de instruções. Para desviar da linha que não precisa em uma determinada parte pode usar condições com “if-else” e para repetir linhas pode-se usar o loop “for” ou “while”

Condições

O comando “if” serve para executar algumas linhas de código se determinada condição for verdadeira, para isso se usa os operadores condicionais: “==”, “!=”, “>”, “<”, “>=”, “<=” e outros. O comando “else” serve para quando o “if” cair na condição falsa executar outro conjunto de instruções. O “elif” serve para adicionar outras condições no bloco “if-else”.

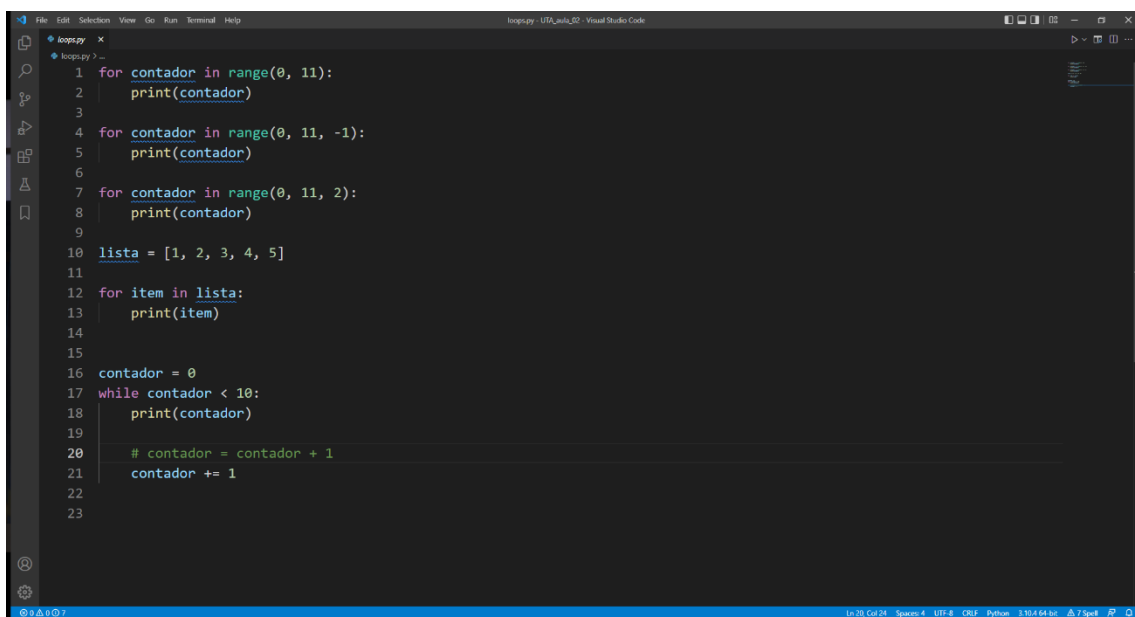


```
1 opcao = 1
2
3 if opcao == 1:
4     print("opção 1")
5 elif opcao == 2:
6     print("opção 2")
7 else:
8     print("opção 3")
9
10
```

Loops

Os loops são usados para quando se quer repetir determinado conjunto de instruções. O loop “for” serve para repetir quando se sabe a quantidade de repetições que serão executadas. Como um intervalo de valores, a função “range()” serve para isso, uma lista de valores, uma variável que armazena outras variáveis.

O loop “while” serve para quando não é determinado o início e fim de uma repetição, como tem uma lista de arquivos e quer remover aqueles que tem nome repetido ou remover linhas de arquivo que atendem uma determinada condição.

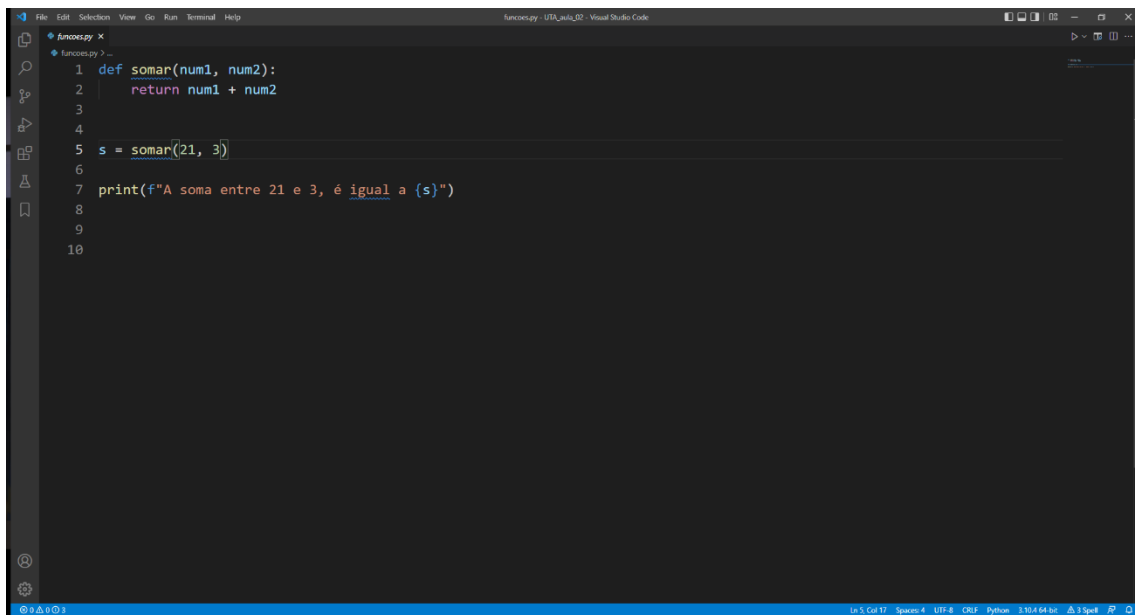


```
1 for contador in range(0, 11):
2     print(contador)
3
4 for contador in range(0, 11, -1):
5     print(contador)
6
7 for contador in range(0, 11, 2):
8     print(contador)
9
10 lista = [1, 2, 3, 4, 5]
11
12 for item in lista:
13     print(item)
14
15
16 contador = 0
17 while contador < 10:
18     print(contador)
19
20     # contador = contador + 1
21     contador += 1
22
23
```

Funções

As funções são usadas quando precisa de um trecho do código que vai ser usado em várias partes do programa ou para melhorar a legibilidade de código. São definidas com a palavra “def”, em seguida o nome e entre parênteses os parâmetros, valores que a função irá usar. A palavra “return” serve para quando vai usar o valor que a função executou em outra parte do programa, mas não é obrigatória.

Para colocar um valor de variável em uma string usa-se “fstring” que é o f na frente e em seguida a string e a variável na região “{}”



```
1 def somar(num1, num2):
2     return num1 + num2
3
4
5 s = somar(21, 3)
6
7 print(f"A soma entre 21 e 3, é igual a {s}")
8
9
10
```