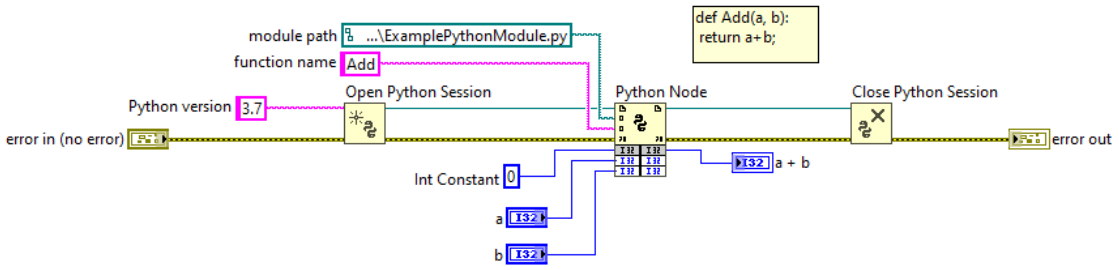
Usando Códigos de Python no LabVIEW

O LabVIEW possui funções para interpretar códigos “.py”, que são as relacionadas ao *Python Node*, como demonstrado no simples exemplo a seguir:



No arquivo “*ExamplePythonModule.py*”, está presente a função descrita no bloco de documentação, chamada “*Add*”:

```
def Add(a, b):
    return a+b;
```

As funções dependem do LabVIEW versão 64 bits e Python instalado para funcionarem. O procedimento é o seguinte:

A sessão para o uso do *Python Node* precisa ser iniciada pela função “*Open Python Session*” (*palheta de funções do diagrama de blocos > Connectivity > Python > Open Python Session*), e indicando nela a versão do Python a ser utilizada, nesse exemplo, na constante string “*Python Version*”, com valor “3.7”.

Após aberta a sessão, a função “*Python Node*” (*palheta de funções do diagrama de blocos > Connectivity > Python > Python Node*), executa as linhas de código correspondentes a função definida pela constante do exemplo “*function name”*, que deve estar presente no arquivo “.py” indicado pela constante “*module path*”. Deve ser indicado no primeiro terminal de entrada da função *Python Node*, qual o tipo de dado de resposta da função em Python, nesse caso aqui, um inteiro, representado pelo “*Int Constant”*, e com retorno via indicador “*a+b*”. Logo abaixo da entrada de tipo de dado de resposta, entram os dados a serem processados, em ordem, de acordo com a função “*Add*” em python.

A função “*Close Python Session”* (*palheta de funções do diagrama de blocos > Connectivity > Python > Close Python Session*), encerra a sessão de Python, removendo da memória após o uso dentro do programa.

O exemplo foca em mostrar os processos relacionados a execução de um código de Python, e não na complexidade do código escrito. Porém é possível integrar códigos bem complexos, com diversas bibliotecas e pacotes em Python.

Outro ponto interessante é a capacidade de integração com artifícios do LabVIEW, como os controles, indicadores, gráficos do painel frontal e bibliotecas de comunicação com hardwares, e a possiblidade da verificação de erros, em Python, que pode ser visualizada no LabVIEW.