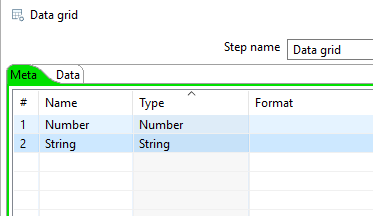
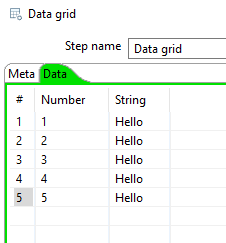
**PDI Python Step Setup Instructions**

**Prerequisites: PDI Client Tools (Spoon) should already be setup and properly configured using the GitLab repos.**

* 1. Installation instructions are located at **I:\Share\SoftwareDownloads\SIG\Pentaho\client\PDI Client Tools Installation Instructions.docx**

1. **Install Anaconda or Miniconda on the Developer Workstation (if not already present)**
   1. Anaconda <https://www.anaconda.com/products/individual> (full feature)
   2. Miniconda <https://docs.conda.io/en/latest/miniconda.html> (lighter, less included)
   3. Accept all defaults
2. **Launch the Anaconda Prompt**
   1. Finding the Anaconda Prompt
      1. Do a Windows Search (Magnifying glass or search bar on the task bar) for Anaconda Prompt
         1. The top choice should be the correct one
         2. (I like to create a shortcut or pin to the taskbar once open)
3. **Create a Python Virtual Environment** 
   1. In the open Anaconda prompt, change directory to C:\apps
      1. This should be where your java folder is for PDI
   2. Create a **venvs** folder if it doesn’t already exist (full path C:\apps\venvs)
      1. You can type **mkdir venvs** if using the command prompt
   3. cd into the venvs folder
   4. **Note: setting up the folder structure this way is essential for the PDI saved variables to work correctly**
   5. Environment Setup
      1. Linux or Mac
         1. python3 -m venv pdi\_venv
         2. source bin/activate
         3. python3 -m pip install upgrade pip
         4. pip install -r requirements.txt
         5. deactivate
      2. Windows
         1. From C:\apps\venvs> run the following commands
            1. (run command) **py -m venv pdi\_venv**
         2. Copy the requirements.txt file from I:\Share\SoftwareDownloads\SIG\Pentaho\client\python to C:\apps\venvs\pdi\_venv
         3. Continue running the following commands
            1. **cd pdi\_venv** (C:\apps\venvs\pdi\_venv)
            2. (run command) **Scripts\activate**
            3. (run command) **python -m pip install -U pip**
            4. (run command) **pip install -r requirements.txt**
            5. (run command) **deactivate**
4. **Test the Python Scripting Step in PDI (Spoon)**
   1. **Note: you will not be able to follow these steps unless PDI Client Tools are installed and configured using the directions in I:\Share\SoftwareDownloads\SIG\Pentaho\client\PDI Client Tools Installation Instructions.docx**
   2. From the command prompt in the directory C:\etlbase\projects\platform-configuration\config-pdi-local run the command **spoon** to launch Spoon
   3. Open and run the C:\pentaho\design-tools\data-integration\jb\_set\_dev\_env.kjb job to set the development environment (or equivalent setup job for production)
   4. Create a new transformation
   5. Add a **Data grid** step to the canvas
      1. Configure with the exact data in this picture to compare your output results to compare to this example, both Meta and Data tabs:

* 1. Add a **Python Executor** step to the canvas with a hop coming from the Data grid step
     1. Configure the Python Executor step
        1. Script Tab
           1. Check the **Use a Python virtual environment** box
           2. In the path box, type control+spacebar to open the list of available variables

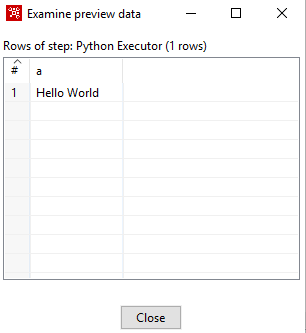
Start typing PYTHON and choose ${PYTHON\_VIRTUAL\_ENV}

If this variable isn’t available, you probably didn’t run the setup environment job or are using an outdated version of the sc\_etls.properties file (update you sc\_etls project from GitLab, then rerun the environment setup job)

* + - * 1. In the Manual Python Script dialogue, copy and paste the following:

a = list(df['String'].apply(lambda x: ' '.join([x, 'World'])))[0]

* + - 1. Input Tab
         1. Row processing select **All rows**
         2. Available variables should not be blank
         3. Change the **Variable name** to **df**
         4. Data structure should be **Pandas dataframe**
         5. Click Get fields, and your column names should appear
      2. Output Tab
         1. Output processing set as **Variable to fields**
         2. Variable: enter **a**
         3. Click **Get fields**, then click **Yes run** in the next dialoge
         4. Click OK
  1. **Test the Script**
     1. Right-click the Python Executor step -> click **Preview…**
     2. Click **Quick Launch**
     3. If everything is successful, this should be your output:

 You’re welcome for that crazy Hello World app.