```
"nbformat": 4,
"nbformat_minor": 0,
"metadata": {
  "colab": {
    "provenance": [],
    "collapsed_sections": []
  "name": "python3",
    "display_name": "Python 3"
  "language_info": {
    "name": "python"
},
"cells": [
  {
    "cell_type": "markdown",
    "source": [
      "# Basic Python"
    ],
"metadata": {
    "" "McSx
      "id": "McSxJAwc0dZ1"
    }
  },
    "cell_type": "markdown",
    "source": [
   "## 1. Split this string"
    "metadata": {
      "id": "CU48hgo40wz5"
    }
  },
{
    "cell_type": "code",
    "source": [
      "s = \"Hi there Sam!\""
    "metadata": {
      "id": "s07c7JK70qt-"
    "outputs": []
  },
    "cell_type": "code",
    "source": [
      "# Splits at space\n",
      "s.split()"
    "metadata": {
      "id": "6mGVa3SQYLkb",
      "colab": {
        "base_uri": "https://localhost:8080/"
      "outputId": "2efb0e53-98a3-43fc-893d-7524509e7b2d"
    },
```

```
"execution_count": 2,
  "outputs": [
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "['Hi', 'there', 'Sam!']"
      },
"metadata": {},
      "execution_count": 2
  ]
},
  "cell_type": "markdown",
  "source": [
    "## 2. Use .format() to print the following string. \n",
    "### Output should be: The diameter of Earth is 12742 kilometers."
  ],
"metadata": {
"' "GH1Q
    "id": "GH10Bn8HP375"
  }
<u>}</u>,
  "cell_type": "code",
  "source": [
    "planet = \TEarth\T",
    'diameter = 12742"
  "metadata": {
    "id": "_ZHoml3kPqic"
  "execution_count": 3,
  "outputs": []
},
  "cell_type": "code",
  "source": [
    "# Reverse the index numbers with the\n",
    "# parameters of the placeholders\n",
    "'The diameter of {0} is {1} kilometer'.format(planet, diameter)"
  ],
  "metadata": {
    "id": "HyRyJv6CYPb4",
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 35
    },
"outputId": "9eb70175-4c89-4cbe-eb99-d08f991306db"
  "execution_count": 4,
  "outputs": [
      "output_type": "execute_result",
      "data": {
        "text/plain": [
           "'The diameter of Earth is 12742 kilometer'"
```

```
],
"application/vnd.google.colaboratory.intrinsic+json": {
          },
          "metadata": {},
          "execution_count": 4
      ]
    },
      "cell_type": "markdown",
      "source": [
        "## 3. In this nest dictionary grab the word \"hello\""
      "metadata": {
        "id": "KE74ZEwkRExZ"
      }
   },
      "cell_type": "code",
      "source": [
        "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':
[1,2,3,'hello']}]}]
      ],
      "metadata": {
        "id": "fcVwbCc1QrQI"
      },
"execution_count": 5,
      "outputs": []
   <u>}</u>,
    {
      "cell_type": "code",
      "source": [
        "#In this nest dictionary grabing the word \"hello\"\n",
        "print(d[\"k1\"][3][\"tricky\"][3][\"target\"][3])"
      ],
"metadata": {
.... "Mvbkl
        "id": "MvbkMZpXYRaw",
        "colab": {
          "base_uri": "https://localhost:8080/"
        "outputId": "a01f22d7-69f1-45f6-a955-c2f6702c4974"
      "execution_count": 6,
      "outputs": [
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "hello\n"
        }
      ]
   },
{
      "cell_type": "markdown",
      "source": [
        "# Numpy"
```

```
"metadata": {
    "id": "bw0vVp-9ddjv"
},
  "cell_type": "code",
  "source": [
    "import numpy as np"
  "metadata": {
    "id": "LLiE_TYrhA10"
  },
"execution_count": 7,
  "outputs": []
},
{
  "cell_type": "markdown",
  "source": [
    "## 4.1 Create an array of 10 zeros? \n",
    "## 4.2 Create an array of 10 fives?"
  ],
"metadata": {
    "id": "wOg8hinbgx30"
  }
},
{
  "cell_type": "code",
  "source": [
    "#array of 10 zeros\n"
    "array1=np.zeros(10)\n",
    "print(array1)"
  ],
"metadata": {
"' "NHri
    "id": "NHrirmgCYXvU",
    "colab": {
      "base_uri": "https://localhost:8080/"
    "outputId": "1bbf080e-8459-479b-da49-8dd05975223d"
  },
"execution_count": 8,
  "outputs": [
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "[0. 0. 0. 0. 0. 0. 0. 0. 0.]\n"
    }
  ]
},
  "cell_type": "code",
  "source": [
    "# array of 10 fives\n",
    "array2=np.ones(10)*5\n",
    "print(array2)"
  "metadata": {
```

```
"id": "e4005lsTYXxx",
    "colab": {
      "base_uri": "https://localhost:8080/"
    "outputId": "d5727fbc-5582-43b1-874a-e1ccd6a0d03f"
  },
"execution_count": 9,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "[5. 5. 5. 5. 5. 5. 5. 5. 5.]\n"
    }
  ]
},
  "cell_type": "markdown",
  "source": [
    "## 5. Create an array of all the even integers from 20 to 35"
  ],
"metadata": {
"" "a7HH
    "id": "gZHHDUBvrMX4"
  }
},
{
  "cell_type": "code",
  "source": [
    "#array of all the even integers from 20 to 35\n",
    "array3=np.arange(20,36,2)\n",
    "print(array3) "
  ],
"metadata": {
"'' "oAI2
    "id": "oAI2tbU2Yag-",
    "colab": {
      "base_uri": "https://localhost:8080/"
    "outputId": "da959a05-96ed-414b-f615-b20d79bcf3a6"
  },
"execution_count": 10,
  "outputs": [
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "[20 22 24 26 28 30 32 34]\n"
    }
  ]
},
  "cell_type": "markdown",
  "source": [
    "## 6. Create a 3x3 matrix with values ranging from 0 to 8"
  "metadata": {
    "id": "NaOM308NsRpZ"
  }
```

```
"cell_type": "code",
  "source": [
    "#3x3 matrix with values ranging from 0 to 8\n",
    x = np.arange(0, 9).reshape(3,3)\n'',
    "print(x)"
  ],
  "metadata": {
    "id": "t0lEVH7BYceE",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
"outputId": "cdb0fdca-d44f-4b83-85d8-c20b35e0c734"
 },
"execution_count": 11,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "[[0 1 2]\n",
        " [3 4 5]\n",
        " [6 7 8]]\n"
    }
 ]
},
  "cell_type": "markdown",
  "source": [
    "## 7. Concatenate a and b n,
    "## a = np.array([1, 2, 3]), b = np.array([4, 5, 6])"
  "metadata": {
    "id": "hOOdnhAOuU p"
},
  "cell_type": "code",
  "source": [
    "a = np.array([1, 2, 3])\n",
    "b = np.array([4, 5, 6])\n",
    "#Concatenate \n"
    "np.concatenate((a,b),axis=None)"
  ],
  "metadata": {
    "id": "rAPSw97aYfE0",
    "colab": {
      "base uri": "https://localhost:8080/"
    "outputId": "40b37ecc-d377-46fe-e289-4986466e859f"
 },
"execution_count": 12,
  "outputs": [
      "output_type": "execute_result",
      "data": {
        "text/plain": [
```

```
"array([1, 2, 3, 4, 5, 6])"
         ]
       "metadata": {},
       "execution_count": 12
    }
  ]
},
  "cell_type": "markdown",
  "source": [
    "# Pandas"
  ],
"metadata": {
    '" "dlpe`
    "id": "dlPEY9DRwZga"
  }
  "cell_type": "markdown",
  "source": [
    "## 8. Create a dataframe with 3 rows and 2 columns"
  ],
"metadata": {
    "" "iioY\
    "id": "ijoYW51zwr87"
  }
},
  "cell_type": "code",
  "source": [
    "import pandas as pd\n"
  "metadata": {
    "id": "T50xJRZ8uvR7"
  "execution_count": 13,
  "outputs": []
},
  "cell_type": "markdown",
  "source": [
    "1.Numpy arrays"
  "metadata": {
    "id": "vqi06luLbbrZ"
  }
<u>}</u>,
  "cell_type": "code",
  "source": [
    "A = np.random.randint(10, size=(3,2))\n",
    "#dataframe\n",
    "df = pd.DataFrame(A, columns=['cola', 'colb'])\n",
    "df"
  ],
"metadata": {
"'` "xNpI
    "id": "xNpI_XXoYhs0",
    "colab": {
       "base_uri": "https://localhost:8080/",
       "height": 143
```

```
},
"outputId": "e1297944-dda7-4fed-ea3a-b1f65827d116"
},
"execution_count": 14,
"outputs": [
 {
   "output_type": "execute_result",
   "data": {
     "text/plain": [
          cola colb\n",
       "⊙
             1
                  5\n",
       "1
             5
                  5\n",
       "2
                  2"
             2
     ],
"text/html": [
       "\n",
       " <div id=\"df-710b8382-7ad4-4a2d-ae18-813116e562ea\">\n",
       11
           <div class=\"colab-df-container\">\n",
       11
             < div > n",
       "<style scoped>\n",
           .dataframe tbody tr th:only-of-type {\n",
       п
              vertical-align: middle;\n",
       11
           }\n",
       "\n"
       11
           .dataframe tbody tr th \{\n'',
       11
              vertical-align: top;\n",
       11
           }\n",
       "\n"
       11
           .dataframe thead th \{\n''\}
       11
              text-align: right; \n",
       п
           }\n"
       "</style>\n",
       "\n",
         <thead>\n",
       11
           \n",
       11
             \n",
       11
             cola\n"
       п
             colb\n",
       11
           \n",
       11
         </thead>\n"
       п
         \n",
       11
           \n",
       11
             0\n",
       11
             1\n",
       11
             5\n",
       11
           \n",
       11
           <tr>\n",
       п
             1\n",
       11
             5\n"
       11
             5\n",
       п
           \n",
       11
           \n",
             2\n",
       11
             2\n",
       11
             2\n",
           \n"
         \n",
       "\n",
       "</div>\n",
```

```
<button class=\"colab-df-convert\"</pre>
onclick=\"convertToInteractive('df-710b8382-7ad4-4a2d-ae18-813116e562ea')\"\n",
                               title=\"Convert this dataframe to an interactive
table.\"\n",
               11
                               style=\"display:none;\">\n",
                        \n",
               11
                  <svg xmlns=\"http://www.w3.org/2000/svg\"</pre>
height=\"24px\"viewBox=\"0 0 24 24\"\n"
                       width=\"24px\">\n"
                    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
                    <path d=\"M18.56 5.441.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-</pre>
2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-
2.06.94zm10 101.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94
2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-
1.04.2-1.43.59L10.3 9.451-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59
1.41.59.51 0 1.02-.2 1.41-.5917.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4
18.5917.72-7.72 1.47 1.35L5.41 20z\"/>\n",
                  </svg>\n",
               11
                      </button>\n",
               11
                      \n",
               п
                  <style>\n"
               11
                    .colab-df-container {\n",
                      display:flex;\n",
               11
                      flex-wrap:wrap;\n",
               11
                      gap: 12px;\n",
               11
                    }\n",
               "\n".
               п
                    .colab-df-convert {\n",
               11
                      background-color: #E8F0FE;\n",
               11
                      border: none; \n",
               11
                      border-radius: 50%;\n",
               11
                      cursor: pointer;\n",
               11
                      display: none; \n",
               11
                      fill: #1967D2;\n"
               11
                      height: 32px;\n",
               11
                      padding: 0 0 0 0;\n",
               11
                      width: 32px;\n",
               11
                    }\n",
               "\n"
               11
                    .colab-df-convert:hover {\n",
               11
                      background-color: #E2EBFA;\n",
               11
                      box-shadow: Opx 1px 2px rgba(60, 64, 67, 0.3), Opx 1px 3px 1px
rgba(60, 64, 67,
                 0.15);\n",
                      fill: #174EA6;\n",
               11
                    }\n",
               11
                    [theme=dark] .colab-df-convert {\n",
               п
                      background-color: #3B4455;\n",
               11
                      fill: #D2E3FC;\n",
               11
                    }\n",
               "\n"
               11
                    [theme=dark] .colab-df-convert:hover {\n",
               11
                      background-color: #434B5C;\n",
               п
                      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
               11
                      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
               11
                      fill: #FFFFFF;\n",
                    }\n",
                  </style>\n",
               "\n",
```

```
п
                      <script>\n",
               11
                         const buttonEl =\n",
                           document.querySelector('#df-710b8382-7ad4-4a2d-ae18-
813116e562ea button.colab-df-convert'); \n",
               11
                        buttonEl.style.display =\n",
               11
                           google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
               "\n",
               ш
                         async function convertToInteractive(key) {\n",
               11
                           const element = document.guerySelector('#df-710b8382-7ad4-
4a2d-ae18-813116e562ea'); \n",
               п
                           const dataTable =\n",
                             await
google.colab.kernel.invokeFunction('convertToInteractive', \n",
                                                                          [key], {});\n",
               11
                           if (!dataTable) return;\n",
               "\n",
               11
                           const docLinkHtml = 'Like what you see? Visit the ' +\n",
                             '<a target=\"_blank\"</pre>
href=https://colab.research.google.com/notebooks/data_table.ipynb>data table
notebook</a>'\n",
                             + ' to learn more about interactive tables.';\n",
                           element.innerHTML = '';\n",
dataTable['output_type'] = 'display_data';\n",
               п
                           await google.colab.output.renderOutput(dataTable,
element); \n",
                           const docLink = document.createElement('div');\n",
               11
                           docLink.innerHTML = docLinkHtml;\n",
               п
                           element.appendChild(docLink); \n",
               11
                        }\n",
               11
                      </script>\n",
               п
                    </div>\n",
               11
                  </div>\n",
               11
             ]
           },
           "metadata": {},
           "execution_count": 14
        }
      ]
    },
      "cell_type": "markdown",
      "source": [
        "2.Dictionary"
      "metadata": {
        "id": "SsMmKaHbfCTd"
      }
    },
      "cell_type": "code",
      "source": [
        "dict_a = \{ \n'',
             'col_a':[1,2,3], \n",
             'col_b': [2,5,6], \n",
        "}\n",
        "#dataframe\n",
        "df = pd.DataFrame(dict_a)\n",
        "df"
```

```
"metadata": {
 "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 143
 },
"id": "eSwpYhbMfIcq",
"" "152002hf
 "outputId": "1b2cc2bf-ea20-41c3-a9e7-38a30489e0e4"
},
"execution_count": 15,
"outputs": [
 {
   "output_type": "execute_result",
   "data": {
     "text/plain": [
           col_a col_b\n",
       "⊙
              1
                     2\n"
       "1
                     5\n",
              2
       "2
                     6"
              3
     ],
"text/html": [
       "\n",
          <div id=\"df-eaf6c25c-1330-4cb2-ab36-4419dc9e4e0d\">\n",
       11
            <div class=\"colab-df-container\">\n",
       11
              <div>\n",
       "<style scoped>\n",
       11
            .dataframe tbody tr th:only-of-type {\n",
       11
               vertical-align: middle; \n",
       11
            }\n",
       "\n"
       11
            .dataframe tbody tr th \{\n'',
       11
               vertical-align: top;\n",
       11
            }\n",
       "\n",
       п
            .dataframe thead th \{\n'',
       11
                text-align: right;\n",
            }\n",
       "</style>\n",
       "\n",
       11
          <thead>\n",
       п
            \n",
       11
              \n",
       11
              <th>col_a\n"
       11
              col_b\n",
       11
            \n",
          </thead>\n",
       11
          \n",
       11
            \n",
       11
              0\n",
       11
              1\n"
       п
             2\n",
       11
            \n",
            \n",
       11
             1\n",
       11
             2\n",
       11
             5\n",
       11
            \n",
       11
            \n",
              2\n",
```

```
п
                      3\n"
              11
                      6\n",
                    \n"
                 \n"
              "\n",
              "</div>\n",
                      <button class=\"colab-df-convert\"</pre>
onclick=\"convertToInteractive('df-eaf6c25c-1330-4cb2-ab36-4419dc9e4e0d')\"\n",
                              title=\"Convert this dataframe to an interactive
table.\"\n",
              11
                              style=\"display:none;\">\n",
              11
                        \n",
                 <svg xmlns=\"http://www.w3.org/2000/svg\"</pre>
height=\"24px\"viewBox=\"0 0 24 24\"\n",
                       width=\"24px\">\n"
                   <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
                    <path d=\"M18.56 5.441.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-</p>
2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-
2.06.94zm10 101.94 2.06.94-2.06 2.06-.94-2.06-.94-2.06-.94
2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-
1.04.2-1.43.59L10.3 9.451-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59
1.41.59.51 0 1.02-.2 1.41-.5917.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4
18.5917.72-7.72 1.47 1.35L5.41 20z\"/>\n",
                 </svg>\n",
              11
                      </button>\n",
              11
                      n'',
              п
                 <style>\n"
              11
                    .colab-df-container \{\n'',
              11
                      display:flex;\n",
              11
                      flex-wrap:wrap;\n",
              11
                      gap: 12px;\n",
              11
                    }\n",
              "\n",
              11
                    .colab-df-convert {\n",
              11
                      background-color: #E8F0FE;\n",
              11
                      border: none; \n",
              11
                      border-radius: 50%;\n",
              п
                      cursor: pointer;\n",
                      display: none; \n",
              11
                      fill: #1967D2;\n",
              11
                      height: 32px;\n",
              11
                      padding: 0 0 0 0;\n",
              11
                      width: 32px;\n",
              п
                    }\n",
              "\n"
              11
                    .colab-df-convert:hover {\n",
              11
                      background-color: #E2EBFA;\n",
              11
                      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px
rgba(60, 64,
             67,
                 0.15);\n",
                      fill: #174EA6;\n",
              п
                   }\n",
              "\n"
                    [theme=dark] .colab-df-convert {\n",
              11
                      background-color: #3B4455;\n",
              11
                      fill: #D2E3FC;\n",
              11
                    }\n",
              "\n"
              11
                    [theme=dark] .colab-df-convert:hover {\n",
                      background-color: #434B5C;\n",
```

```
п
                      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
               11
                      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
               11
                      fill: #FFFFFF;\n",
               11
                    }\n",
               11
                 </style>\n'',
              "\n",
               11
                      <script>\n",
               п
                        const buttonEl =\n",
                          document.guerySelector('#df-eaf6c25c-1330-4cb2-ab36-
4419dc9e4e0d button.colab-df-convert');\n",
                        buttonEl.style.display =\n",
               11
                          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
              "\n",
               п
                        async function convertToInteractive(key) {\n",
               п
                          const element = document.querySelector('#df-eaf6c25c-1330-
4cb2-ab36-4419dc9e4e0d');\n",
                          const dataTable =\n",
                            await
google.colab.kernel.invokeFunction('convertToInteractive', \n",
                                                                        [key], {});\n",
              11
                          if (!dataTable) return;\n",
              "\n",
                          const docLinkHtml = 'Like what you see? Visit the ' +\n",
                             '<a target=\"_blank\"</pre>
href=https://colab.research.google.com/notebooks/data_table.ipynb>data table
notebook</a>'\n",
                            + ' to learn more about interactive tables.';\n",
               п
                          element.innerHTML = '';\n",
               11
                          dataTable['output_type'] = 'display_data';\n",
                          await google.colab.output.renderOutput(dataTable,
element); \n",
                          const docLink = document.createElement('div');\n",
               11
                          docLink.innerHTML = docLinkHtml;\n",
               11
                          element.appendChild(docLink); \n",
               11
                        }\n",
                      </script>\n",
              11
                    </div>\n",
              11
                  </div>\n",
               11
            ]
          },
          "metadata": {},
          "execution_count": 15
        }
      ]
    },
      "cell_type": "markdown",
      "source": [
        "3.List"
      "metadata": {
        "id": "ahecadWfffsa"
      }
      "cell_type": "code",
      "source": [
        "lst_a = [['John', 23], ['Jane', 25], ['Mary', 21]] \n",
```

```
"#dataframe\n",
 "df = pd.DataFrame(lst_a,columns=['Name', 'Age'])\n",
 "df"
],
"metadata": {
 "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 143
 "outputId": "ac942318-5a61-4439-8c38-698afc1cadf0"
},
"execution_count": 16,
"outputs": [
  {
   "output_type": "execute_result",
   "data": {
     "text/plain": [
          Name Age\n",
       "0 John
                23\n",
       "1
          Jane
                25\n",
       "2
                21"
         Mary
     "text/html": [
       "\n",
         <div id=\"df-ee1cbbd8-98a1-459b-9355-84d07c6ba3bb\">\n",
       11
           <div class=\"colab-df-container\">\n",
       11
             <div>\n"
       "<style scoped>\n",
           .dataframe tbody tr th:only-of-type {\n",
       11
               vertical-align: middle;\n",
       11
           }\n",
       "\n",
       11
           .dataframe thody tr th \{\n'',
       п
               vertical-align: top;\n",
       11
           }\n",
       "\n"
       п
            .dataframe thead th {\n"
       11
               text-align: right;\n",
       11
           }\n",
       "</style>\n"
       "\n",
         <thead>n'',
       п
           \n",
       11
             \n",
       11
             Name\n",
       11
             Age\n",
       11
           \n"
       11
         </thead>\n"
       11
         \n",
       п
           \n",
       11
             0\n",
       11
             John\n",
       11
             23\n",
       11
           \n",
       11
           \n",
       11
             1\n",
       п
             Jane\n",
             25\n",
```

```
11
                   \n",
              11
                    \n",
              11
                      2\n"
              11
                      Mary\n",
              11
                      21\n",
              11
                    \n"
              п
                 \n",
              "\n",
              "</div>\n",
                      <button class=\"colab-df-convert\"</pre>
onclick=\"convertToInteractive('df-ee1cbbd8-98a1-459b-9355-84d07c6ba3bb')\"\n",
                              title=\"Convert this dataframe to an interactive
table.\"\n",
              11
                              style=\"display:none;\">\n",
              п
                        \n",
              11
                 <svg xmlns=\"http://www.w3.org/2000/svg\"</pre>
height=\"24px\"viewBox=\"0 0 24 24\"\n"
                       width=\"24px\">\n"
              11
                    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
              11
                    <path d=\"M18.56 5.441.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-
2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-
2.06.94zm10 101.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94
2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-
1.04.2-1.43.59L10.3 9.451-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59
1.41.59.51 0 1.02-.2 1.41-.5917.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4
18.5917.72-7.72 1.47 1.35L5.41 20z\"/>\n",
                 </svg>\n",
              11
                      </button>\n",
              11
                      \n",
              11
                 <style>\n"
              11
                    .colab-df-container {\n",
              11
                      display:flex;\n",
              11
                      flex-wrap:wrap;\n",
              11
                      gap: 12px;\n",
              11
                    }\n",
              "\n",
              11
                    .colab-df-convert {\n",
              п
                      background-color: #E8F0FE;\n",
              11
                      border: none;\n",
              11
                      border-radius: 50%;\n",
              11
                      cursor: pointer;\n",
              11
                      display: none; \n",
              11
                      fill: #1967D2;\n",
              п
                      height: 32px;\n",
              11
                      padding: 0 0 0 0;\n",
              11
                     width: 32px;\n",
              11
                    }\n",
              "\n",
              11
                    .colab-df-convert:hover {\n",
              11
                      background-color: #E2EBFA;\n"
              п
                      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px
rgba(60, 64,
             67,
                 0.15);\n",
                      fill: #174EA6;\n",
              11
                   }\n",
              "\n"
                    [theme=dark] .colab-df-convert {\n",
              11
                     background-color: #3B4455;\n",
              11
                      fill: #D2E3FC;\n",
                    }\n",
```

```
"\n",
                    [theme=dark] .colab-df-convert:hover {\n",
                      background-color: #434B5C;\n",
               11
                      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
               11
                      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
               11
                      fill: #FFFFFF;\n",
               11
                    }\n",
               п
                 </style>\n",
              "\n",
               11
                      <script>\n",
               11
                        const buttonEl =\n",
               11
                          document.querySelector('#df-ee1cbbd8-98a1-459b-9355-
84d07c6ba3bb button.colab-df-convert');\n",
                        buttonEl.style.display =\n",
               п
                          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
               "\n",
               11
                        async function convertToInteractive(key) {\n",
               11
                          const element = document.guerySelector('#df-ee1cbbd8-98a1-
459b-9355-84d07c6ba3bb');\n",
              п
                          const dataTable =\n",
              ш
                            await
google.colab.kernel.invokeFunction('convertToInteractive', \n",
                                                                        [key], {});\n",
               11
                          if (!dataTable) return;\n",
              "\n",
                          const docLinkHtml = 'Like what you see? Visit the ' +\n'',
                             '<a target=\"_blank\"</pre>
href=https://colab.research.google.com/notebooks/data_table.ipynb>data table
notebook</a>'\n",
                            + ' to learn more about interactive tables.';\n",
               11
                          element.innerHTML = '';\n",
               11
                          dataTable['output_type'] = 'display_data';\n",
                          await google.colab.output.renderOutput(dataTable,
element); \n",
                          const docLink = document.createElement('div');\n",
                          docLink.innerHTML = docLinkHtml;\n",
               11
                          element.appendChild(docLink); \n",
               п
                        }\n",
               11
                      </script>\n",
              11
                    </div>\n",
              11
                 </div>\n",
               11
            ]
          },
          "metadata": {},
          "execution_count": 16
        }
      ]
    },
      "cell_type": "markdown",
      "source": [],
      "metadata": {
        "id": "olNxcHbhbg9Q"
      }
    },
      "cell_type": "markdown",
      "source": [
```

```
"## 9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023"
      "metadata": {
        "id": "UXSmdNclyJQD"
     }
    },
      "cell_type": "code",
      "source": [
        "import pandas as pd\n",
        "\n",
        "# calling DataFrame constructor\n",
       "df = pd.DataFrame()\n",
        "\n",
       "# Create 6 dates\n",
        "df['time'] = pd.date_range(start=\"1/1/2023\", end=\"2/10/2023\", freq
='24H')\n",
        " # print dataframe\n",
        " \n",
        "# Extract features - year, month, day, hour, and minute\n",
        "df['year'] = df['time'].dt.year\n"
        "df['month'] = df['time'].dt.month\n",
        "df['day'] = df['time'].dt.day\n",
        "\n",
        "\n",
       "# Show six rows\n",
        "df.head(len(df[\"time\"]))"
      "metadata": {
       "id": "dgyC0JhVYl4F",
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 1000
        "outputId": "7a8c7745-ce98-4fb4-8358-e9626144a9d3"
      "execution_count": 17,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
                                   month day\n",
                       time year
              "0 2023-01-01 2023
                                   1
                                            1\n",
              "1
                                            2\n",
                 2023-01-02 2023
                                       1
              "2 2023-01-03 2023
                                            3\n"
                                       1
              "3 2023-01-04 2023
                                            4\n"
                                       1
              "4 2023-01-05 2023
                                            5\n"
                                       1
              "5 2023-01-06 2023
                                            6\n"
                                       1
                 2023-01-07 2023
                                       1
                                            7\n"
              "7
                 2023-01-08 2023
                                       1
                                            8\n",
              "8 2023-01-09 2023
                                       1
                                           9\n"
              "9 2023-01-10 2023
                                       1 10\n"
              "10 2023-01-11 2023
                                       1 11\n"
              "11 2023-01-12 2023
                                       1 12\n"
              "12 2023-01-13 2023
                                       1
                                           13\n",
              "13 2023-01-14 2023
                                       1 14\n",
              "14 2023-01-15 2023
                                       1
                                           15\n",
              "15 2023-01-16 2023
                                       1
                                           16\n",
```

```
"16 2023-01-17 2023
                         1
                             17\n",
 "17 2023-01-18 2023
                             18\n"
                         1
 "18 2023-01-19 2023
                         1
                             19\n"
 "19 2023-01-20 2023
                         1 20\n"
 "20 2023-01-21 2023
                         1 21\n",
 "21 2023-01-22 2023
                         1 22\n"
                             23\n"
 "22 2023-01-23 2023
                         1
 "23 2023-01-24
                             24\n"
               2023
                         1
 "24 2023-01-25
                2023
                         1
                             25\n"
 "25 2023-01-26 2023
                         1
                             26\n"
 "26 2023-01-27
                         1 27\n"
               2023
 "27 2023-01-28 2023
                         1 28\n"
 "28 2023-01-29
                2023
                         1
                             29\n"
 "29 2023-01-30
                         1
                2023
                             30\n"
 "30 2023-01-31
                             31\n"
                2023
                         1
 "31 2023-02-01
                2023
                         2
                             1\n"
 "32 2023-02-02 2023
                         2
                             2\n"
 "33 2023-02-03 2023
                         2
                              3\n"
 "34 2023-02-04 2023
                         2
                             4\n"
 "35 2023-02-05 2023
                         2
                              5\n"
 "36 2023-02-06
                         2
                              6\n"
                2023
 "37 2023-02-07
                         2
                             7\n"
                2023
 "38 2023-02-08
                         2
                2023
                             8\n"
 "39 2023-02-09 2023
                         2
                             9\n",
 "40 2023-02-10 2023
                             10"
"text/html": [
 "\n",
    <div id=\"df-5ffea3b0-d3fb-4993-b0df-8086a405ad62\">\n",
      <div class=\"colab-df-container\">\n",
 11
        <div>\n"
 "<style scoped>\n",
      .dataframe tbody tr th:only-of-type {\n",
 11
         vertical-align: middle;\n",
 11
      }\n",
 "\n"
 11
      .dataframe thody tr th \{\n'',
 11
         vertical-align: top;\n",
 11
      }\n",
 "\n",
 11
      .dataframe thead th \{\n''\}
 11
         text-align: right;\n",
      }\n",
 "</style>\n"
 "\n",
    <thead>\n",
 11
      \n",
 п
        \n",
 11
        time
 11
        year\n"
 п
        month\n",
 11
        <th>day\n",
 11
      \n",
 п
    </thead>\n",
 11
    \n",
 11
      \n",
 11
        0\n",
 11
        2023-01-01\n",
        2023\n",
```

```
11
    1\n",
11
    1\n",
11
   \n",
11
   \n",
11
    1\n",
    2023-01-02\n",
11
    2023\n",
п
    1\n",
11
    2\n",
11
   \n",
п
   \n",
11
    2\n",
11
    2023-01-03\n",
п
    2023\n",
п
    1\n",
11
    3\n",
11
   \n",
11
   \n",
11
    3\n",
11
    2023-01-04\n",
п
    2023\n",
11
    1\n"
    4\n",
11
   \n",
11
   \n",
    4\n",
11
    2023-01-05\n",
11
    2023\n",
11
    1\n"
11
    5\n"
11
   \n",
11
   <tr>\n",
11
    5\n",
11
    2023-01-06\n",
п
    2023\n",
    1\n"
11
    6\n",
11
   \n",
   \n",
11
    6\n",
п
    2023-01-07\n",
11
    2023\n",
    1\n"
11
    7\n",
11
   \n",
11
   \n",
11
    7\n",
п
    2023-01-08\n",
11
    2023\n",
11
    1\n"
п
    8\n",
11
   \n",
   \n",
п
    8\n",
11
    2023-01-09\n",
    2023\n",
11
    1\n"
11
    9\n",
   \n",
```

```
11
   \n",
11
    9\n"
11
    2023-01-10\n",
11
    2023\n",
11
    1\n"
11
    10\n",
11
   \n",
11
   \n",
11
    10\n",
11
    2023-01-11\n",
п
    2023\n",
11
    1\n",
11
    11\n",
п
   \n",
п
   \n",
11
    11\n",
11
    2023-01-12\n",
11
    2023\n",
11
    1\n",
11
    12\n",
п
   \n",
11
   \n",
    12\n",
п
    2023-01-13\n",
11
    2023\n",
    1\n"
11
    13\n"
11
   \n",
11
   \n",
11
    13\n",
11
    2023-01-14\n",
11
    2023\n",
11
    1\n"
11
    14\n"
11
   \n",
11
   \n",
11
    14\n",
п
    2023-01-15\n",
    2023\n",
11
    1\n"
п
    15\n"
11
   \n",
11
   <tr>\n"
11
    15\n",
11
    2023-01-16\n",
    2023\n",
11
    1\n"
11
    16\n",
11
   \n",
11
   \n",
п
    16\n",
11
    2023-01-17\n",
    2023\n",
п
    1\n"
11
    17\n"
   \n",
п
   \n",
п
    17\n",
    2023-01-18\n",
```

```
11
    2023\n",
11
    1\n"
11
    18\n"
   \n",
11
11
   \n",
    18\n",
п
    2023-01-19\n",
п
    2023\n",
11
    1\n"
11
    19\n"
   \n",
п
11
   \n",
11
    19\n",
п
    2023-01-20\n",
п
    2023\n",
11
    1\n"
11
    20\n",
11
   \n",
11
   <tr>\n",
11
    20\n",
п
    2023-01-21\n",
11
    2023\n",
    1\n"
11
    21\n",
11
   \n",
   \n",
п
    21\n",
п
    2023-01-22\n",
11
    2023\n",
11
    1\n"
11
    22\n",
11
   \n",
11
   \n",
11
    22\n",
п
    2023-01-23\n",
    2023\n",
11
    1\n"
п
    23\n",
   \n",
11
    \n'',
п
    23\n",
11
    2023-01-24\n",
    2023\n",
п
    1\n"
11
    24\n",
   \n",
11
   <tr>\n",
11
    24\n",
11
    2023-01-25\n",
11
    2023\n",
п
    1\n"
11
    25\n",
   \n",
п
   \n",
11
    25\n",
    2023-01-26\n",
п
    2023\n",
п
    1\n",
    26\n",
```

```
11
   \n",
11
   <tr>\n"
11
    26\n",
11
    2023-01-27\n",
11
    2023\n",
    1\n"
11
    27\n",
п
   \n",
11
   \n",
11
    27\n",
п
    2023-01-28\n",
11
    2023\n",
11
    1\n"
п
    28\n"
п
   \n",
11
   \n",
11
    28\n",
11
    2023-01-29\n",
11
    2023\n",
11
    1\n",
п
    29\n"
11
   \n",
   \n",
11
    29\n",
11
    2023-01-30\n",
    2023\n",
11
    1\n"
11
    30\n",
11
   \n",
11
   \n",
11
    30\n",
11
    2023-01-31\n",
11
    2023\n",
11
    1\n",
11
    31\n",
11
   \n",
11
   \n",
п
    31\n",
    2023-02-01\n",
11
    2023\n",
п
    2\n",
11
    1\n",
11
   \n",
11
   <tr>\n"
11
    32\n",
    2023-02-02\n",
11
    2023\n",
п
    2\n",
11
    2\n"
11
   \n",
п
   \n",
11
    33\n",
    2023-02-03\n",
п
    2023\n",
11
    2\n",
11
    3\n"
11
   \n",
11
   \n",
    34\n",
```

```
2023-02-04\n",
           11
                2023\n",
           11
                2\n"
           11
                4\n",
           11
               \n",
           11
               \n"
           11
                35\n",
           п
                2023-02-05\n",
           11
                2023\n",
           11
                2\n",
           п
                5\n",
           11
               \n",
           11
               \n",
           п
                36\n",
           п
                2023-02-06\n",
           11
                2023\n",
           11
                2\n"
           11
                6\n",
           11
               \n",
           11
                \n'',
           п
                37\n",
           11
                2023-02-07\n",
                2023\n",
           п
                2\n",
           11
                7\n",
               \n",
           11
               \n",
           11
                38\n",
           11
                2023-02-08\n",
           11
                2023\n",
           11
                2\n",
           11
                8\n",
           11
               \n",
           11
               \n",
           11
                39\n",
           11
                2023-02-09\n",
           11
                2023\n",
           п
                2\n",
                9\n",
           11
               \n",
           п
               <tr>\n"
           11
                40\n",
           11
                2023-02-10\n",
           11
                2023\n",
           11
                2\n",
           11
                10\n",
           11
              \n"
           11
             \n",
           "\n",
           "</div>\n",
                <button class=\"colab-df-convert\"</pre>
onclick=\"convertToInteractive('df-5ffea3b0-d3fb-4993-b0df-8086a405ad62')\"\n",
                       title=\"Convert this dataframe to an interactive
table.\"\n",
           11
                       style=\"display:none;\">\n",
           11
                  \n",
             <svg xmlns=\"http://www.w3.org/2000/svg\"</pre>
height=\"24px\"viewBox=\"0 0 24 24\"\n"
                 width=\"24px\">\n",
```

11

```
<path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
                    <path d=\"M18.56 5.441.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-</p>
2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-
2.06.94zm10 101.94 2.06.94-2.06 2.06-.94-2.06-.94-2.06-.94
2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-
1.04.2-1.43.59L10.3 9.451-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59
1.41.59.51 0 1.02-.2 1.41-.5917.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4
18.5917.72-7.72 1.47 1.35L5.41 20z\"/>\n",
                  </svg>\n",
               11
                      </button>\n",
               п
                      n'',
               11
                  <style>\n",
               11
                    .colab-df-container {\n",
               п
                      display:flex; \n",
               п
                      flex-wrap:wrap;\n",
               11
                      gap: 12px;\n",
               11
                    }\n",
               "\n",
               11
                    .colab-df-convert {\n",
               11
                      background-color: #E8F0FE;\n",
               п
                      border: none; \n",
               11
                      border-radius: 50%;\n",
                      cursor: pointer;\n",
               11
                      display: none;\n"
               11
                      fill: #1967D2;\n",
                      height: 32px;\n",
               п
                      padding: 0 0 0 0;\n",
               п
                      width: 32px;\n",
               11
                    }\n",
               "\n"
               11
                    .colab-df-convert:hover {\n"
               11
                      background-color: #E2EBFA;\n",
               11
                      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px
                 0.15);\n",
rgba(60, 64,
             67,
                      fill: #174EA6;\n",
                    }\n",
              "\n"
               п
                    [theme=dark] .colab-df-convert {\n",
               11
                      background-color: #3B4455;\n",
               11
                      fill: #D2E3FC;\n",
               п
                    }\n",
               "\n"
               11
                    [theme=dark] .colab-df-convert:hover {\n",
               п
                      background-color: #434B5C;\n",
               11
                      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
               11
                      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
               11
                      fill: #FFFFFF;\n",
               11
                    }\n",
               11
                  </style>\n",
              "\n",
               п
                      <script>\n",
               11
                        const buttonEl =\n'',
                          document.querySelector('#df-5ffea3b0-d3fb-4993-b0df-
8086a405ad62 button.colab-df-convert'); \n",
                        buttonEl.style.display =\n",
               11
                          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
              "\n",
               п
                        async function convertToInteractive(key) {\n",
                          const element = document.querySelector('#df-5ffea3b0-d3fb-
```

```
4993-b0df-8086a405ad62');\n",
                          const dataTable =\n",
                            await
google.colab.kernel.invokeFunction('convertToInteractive', \n",
                                                                        [key], {});\n",
              11
                          if (!dataTable) return;\n",
              "\n",
               ш
                          const docLinkHtml = 'Like what you see? Visit the ' +\n",
                             '<a target=\"_blank\"</pre>
href=https://colab.research.google.com/notebooks/data_table.ipynb>data table
notebook </a > ' n",
                            + ' to learn more about interactive tables.';\n",
                          element.innerHTML = '';\n",
               11
              п
                          dataTable['output_type'] = 'display_data';\n",
               11
                          await google.colab.output.renderOutput(dataTable,
element); \n",
                          const docLink = document.createElement('div');\n",
               11
                          docLink.innerHTML = docLinkHtml;\n",
               11
                          element.appendChild(docLink);\n",
               11
                        }\n",
               п
                      </script>\n",
               11
                    </div>\n",
                  </div>\n",
              11
            ]
          },
          "metadata": {},
          "execution_count": 17
        }
      ]
    },
      "cell_type": "markdown",
      "source": [
        "## 10. Create 2D list to DataFrame\n",
        "lists = [[1, 'aaa', 22], n",
                   [2, 'bbb', 25],\n",
        11
                   [3, 'ccc', 24]]"
      "metadata": {
        "id": "ZizSetD-y5az"
      }
    },
      "cell_type": "code",
      "source": [
        "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]"
      "metadata": {
        "id": " XMC8aEt0llB"
      "execution_count": 18,
      "outputs": []
      "cell_type": "code",
      "source": [
        "#2D list to DataFrame\n",
```

```
"df = pd.DataFrame(lists, columns = ['col1', \"col2\", \"col3\"]) \n",
 "df"
],
"metadata": {
    "": "knH70
 "id": "knH76sDKYsVX",
  "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 143
 },
"outputId": "a5f2f164-791b-4a7e-edda-0beaf40dddff"
"execution_count": 19,
"outputs": [
  {
    "output_type": "execute_result",
   "data": {
     "text/plain": [
           col1 col2
                     col3\n",
                       22\n",
       "⊙
              1 aaa
       "1
              2
                 bbb
                       25\n",
       "2
              3 ccc
                       24"
     ],
"text/html": [
       "\n",
       11
          <div id=\"df-5486a87f-40d0-49bd-8fe1-40ca2559ff4f\">\n",
       11
            <div class=\"colab-df-container\">\n",
       11
              < div > n",
       "<style scoped>\n",
            .dataframe tbody tr th:only-of-type {\n",
       11
                vertical-align: middle;\n",
       11
            }\n",
       "\n",
       11
            .dataframe thody tr th \{\n'',
       11
                vertical-align: top;\n",
       11
            }\n",
       "\n",
       11
            .dataframe thead th \{\n''\}
       п
                text-align: right; \n",
            }\n",
       "</style>\n",
       "\n",
          <thead>n'',
       11
            \n",
       11
              \n",
       11
              <th>col1\n",
       11
              <th>col2\n",
       11
              col3\n",
       11

n"
       11
          </thead>\n",
       11
          \n",
       п
            \n",
       11
              0\n",
       11
              1\n",
       11
              aaa
       11
              22\n",
       11
            \n",
       11
            \n",
       11
              1\n",
       11
              2\n",
```

```
11
                     bbb\n",
              11
                     25\n",
              11
                   \n",
              11
                   <tr>\n"
              11
                     2\n",
              11
                     3\n"
              11
                     ccc\n",
              п
                     24\n",
              11
                   \n"
              11
                 \n"
              "\n",
              "</div>\n",
                     <button class=\"colab-df-convert\"</pre>
onclick=\"convertToInteractive('df-5486a87f-40d0-49bd-8fe1-40ca2559ff4f')\"\n",
                              title=\"Convert this dataframe to an interactive
table.\"\n",
              11
                              style=\"display:none;\">\n",
              11
                       \n",
              11
                 <svg xmlns=\"http://www.w3.org/2000/svg\"</pre>
height=\"24px\"viewBox=\"0 0 24 24\"\n",
                      width=\"24px\">\n"
              11
                   <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
                   <path d=\"M18.56 5.441.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-</p>
2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-
2.06.94zm10 101.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94
2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-
1.04.2-1.43.59L10.3 9.451-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59
1.41.59.51 0 1.02-.2 1.41-.5917.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4
18.5917.72-7.72 1.47 1.35L5.41 20z\"/>\n",
                 </svg>\n",
              11
                     </button>\n",
              11
                     \n",
              11
                 <style>\n",
              11
                    .colab-df-container {\n",
              п
                     display:flex;\n",
              11
                     flex-wrap:wrap;\n",
              11
                     gap: 12px;\n",
              11
                   }\n",
              "\n".
              11
                    .colab-df-convert {\n",
              11
                     background-color: #E8F0FE;\n",
              11
                     border: none; \n",
              11
                     border-radius: 50%;\n",
              п
                     cursor: pointer;\n",
              11
                     display: none;\n",
              11
                     fill: #1967D2;\n",
              11
                     height: 32px;\n",
              11
                     padding: 0 0 0 0;\n",
              11
                     width: 32px;\n'',
              11
                   }\n",
              "\n".
              11
                    .colab-df-convert:hover {\n",
              11
                     background-color: #E2EBFA;\n",
              п
                     box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px
                 0.15);\n",
rgba(60, 64, 67,
                     fill: #174EA6;\n",
              11
                   }\n",
              "\n",
                   [theme=dark] .colab-df-convert {\n",
```

```
11
                      background-color: #3B4455;\n",
               11
                      fill: #D2E3FC;\n",
               11
                    }\n",
               "\n"
               11
                    [theme=dark] .colab-df-convert:hover {\n",
               11
                      background-color: #434B5C;\n",
               11
                      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
               п
                      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
               11
                      fill: #FFFFFF;\n",
               11
                    }\n",
               п
                  </style>\n",
               "\n",
               11
                      <script>\n",
               п
                        const buttonEl =\n",
               п
                          document.querySelector('#df-5486a87f-40d0-49bd-8fe1-
40ca2559ff4f button.colab-df-convert');\n",
                        buttonEl.style.display =\n",
               11
               11
                          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
               "\n",
               11
                        async function convertToInteractive(key) {\n",
               ш
                          const element = document.guerySelector('#df-5486a87f-40d0-
49bd-8fe1-40ca2559ff4f'); \n",
                          const dataTable =\n",
               11
                             await
google.colab.kernel.invokeFunction('convertToInteractive', \n",
                                                                         [key], {});\n",
               11
                          if (!dataTable) return; \n",
               "\n",
                          const docLinkHtml = 'Like what you see? Visit the ' +\n",
                             '<a target=\"_blank\"</pre>
href=https://colab.research.google.com/notebooks/data_table.ipynb>data table
notebook</a>'\n",
                             + ' to learn more about interactive tables.';\n",
               11
                          element.innerHTML = '';\n"
               11
                          dataTable['output_type'] = 'display_data';\n",
                          await google.colab.output.renderOutput(dataTable,
element);\n",
                          const docLink = document.createElement('div');\n",
               11
                          docLink.innerHTML = docLinkHtml;\n",
               11
                          element.appendChild(docLink); \n",
               11
                        }\n",
               11
                      </script>\n",
               11
                    </div>\n",
               п
                  </div>\n",
               11
            ]
          "metadata": {},
          "execution_count": 19
      ]
    }
  ]
}
```