```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": [],
      "collapsed_sections": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "source": [
        "# Basic Python"
      ],
      "metadata": {
        "id": "McSxJAwcOdZ1"
      }
    },
    {
      "cell_type": "markdown",
      "source": [
        "## 1. Split this string"
      ],
      "metadata": {
        "id": "CU48hgo4Owz5"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "s = \"Hi there Sam!\""
      ],
      "metadata": {
        "id": "s07c7JK7Oqt-"
      },
      "execution_count": 1,
      "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",
    "\n",
    "### Output should be: The diameter of Earth is 12742 kilometers."
  ],
  "metadata": {
    "id": "GH1QBn8HP375"
  }
},
{
  "cell_type": "code",
  "source": [
    "planet = \"Earth\"\n",
    "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": {
              "type": "string"
            }
          },
          "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": []
    },
    {
      "cell_type": "code",
      "source": [
        "#In this nest dictionary grabing the word \"hello\"\n",
        "print(d[\"k1\"][3][\"tricky\"][3][\"target\"][3])"
      ],
      "metadata": {
        "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_TYrhA1O"
  },
  "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": {
    "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. 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. 5.]\n"
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "## 5. Create an array of all the even integers from 20 to 35"
  ],
  "metadata": {
    "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": {
    "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": "tOlEVH7BYceE",
    "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": "hQ0dnhAQuU_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": {
    "id": "dlPEY9DRwZga"
  }
},
{
  "cell_type": "markdown",
  "source": [
    "## 8. Create a dataframe with 3 rows and 2 columns"
  ],
  "metadata": {
    "id": "ijoYW51zwr87"
  }
},
{
  "cell_type": "code",
  "source": [
    "import pandas as pd\n"
  ],
  "metadata": {
    "id": "T5OxJRZ8uvR7"
  },
  "execution_count": 13,
  "outputs": []
},
{
  "cell_type": "markdown",
  "source": [
    "1.Numpy arrays"
  ],
  "metadata": {
    "id": "vqi06luLbbrZ"
  }
},
{
  "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": {
    "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",
        "0     1     5\n",
        "1     5     5\n",
        "2     2     2"
      ],
      "text/html": [
        "\n",
        "  <div id=\"df-710b8382-7ad4-4a2d-ae18-813116e562ea\">\n",
        "    <div class=\"colab-df-container\">\n",
        "      <div>\n",
        "<style scoped>\n",
        "    .dataframe tbody tr th:only-of-type {\n",
        "        vertical-align: middle;\n",
        "    }\n",
        "\n",
        "    .dataframe tbody tr th {\n",
        "        vertical-align: top;\n",
        "    }\n",
        "\n",
        "    .dataframe thead th {\n",
        "        text-align: right;\n",
        "    }\n",
        "</style>\n",
        "<table border=\"1\" class=\"dataframe\">\n",
        "  <thead>\n",
        "    <tr style=\"text-align: right;\">\n",
        "      <th></th>\n",
        "      <th>cola</th>\n",
        "      <th>colb</th>\n",
        "    </tr>\n",
        "  </thead>\n",
        "  <tbody>\n",
        "    <tr>\n",
        "      <th>0</th>\n",
        "      <td>1</td>\n",
        "      <td>5</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>1</th>\n",
        "      <td>5</td>\n",
        "      <td>5</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>2</th>\n",
        "      <td>2</td>\n",
        "      <td>2</td>\n",
        "    </tr>\n",
        "  </tbody>\n",
        "</table>\n",
        "</div>\n",
```

```
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-710b8382-7ad4-4a2d-ae18-813116e562ea')\"\n",
              "              title=\"Convert this dataframe to an interactive
table.\"\n",
              "              style=\"display:none;\">\n",
              "        \n",
              "  <svg xmlns=\"http://www.w3.org/2000/svg\"
height=\"24px\"viewBox=\"0 0 24 24\"\n",
              "       width=\"24px\">\n",
              "    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
              "    <path d=\"M18.56 5.44l.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 10l.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 9.45l-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-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4
18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
              "  </svg>\n",
              "      </button>\n",
              "      \n",
              "  <style>\n",
              "    .colab-df-container {\n",
              "      display:flex;\n",
              "      flex-wrap:wrap;\n",
              "      gap: 12px;\n",
              "    }\n",
              "\n",
              "    .colab-df-convert {\n",
              "      background-color: #E8F0FE;\n",
              "      border: none;\n",
              "      border-radius: 50%;\n",
              "      cursor: pointer;\n",
              "      display: none;\n",
              "      fill: #1967D2;\n",
              "      height: 32px;\n",
              "      padding: 0 0 0 0;\n",
              "      width: 32px;\n",
              "    }\n",
              "\n",
              "    .colab-df-convert:hover {\n",
              "      background-color: #E2EBFA;\n",
              "      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",
              "    }\n",
              "\n",
              "    [theme=dark] .colab-df-convert {\n",
              "      background-color: #3B4455;\n",
              "      fill: #D2E3FC;\n",
              "    }\n",
              "\n",
              "    [theme=dark] .colab-df-convert:hover {\n",
              "      background-color: #434B5C;\n",
              "      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
              "      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
              "      fill: #FFFFFF;\n",
              "    }\n",
              "  </style>\n",
              "\n",
```

```
              "      <script>\n",
              "        const buttonEl =\n",
              "          document.querySelector('#df-710b8382-7ad4-4a2d-ae18-
813116e562ea button.colab-df-convert');\n",
              "        buttonEl.style.display =\n",
              "          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
              "\n",
              "        async function convertToInteractive(key) {\n",
              "          const element = document.querySelector('#df-710b8382-7ad4-
4a2d-ae18-813116e562ea');\n",
              "          const dataTable =\n",
              "            await
google.colab.kernel.invokeFunction('convertToInteractive',\n",
              "                                                     [key], {});\n",
              "          if (!dataTable) return;\n",
              "\n",
              "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
              "            '<a target=\"_blank\"
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",
              "          await google.colab.output.renderOutput(dataTable,
element);\n",
              "          const docLink = document.createElement('div');\n",
              "          docLink.innerHTML = docLinkHtml;\n",
              "          element.appendChild(docLink);\n",
              "        }\n",
              "      </script>\n",
              "    </div>\n",
              "  </div>\n",
              "  "
            ]
          },
          "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",
  "outputId": "1b2cc2bf-ea20-41c3-a9e7-38a30489e0e4"
},
"execution_count": 15,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "   col_a  col_b\n",
        "0      1      2\n",
        "1      2      5\n",
        "2      3      6"
      ],
      "text/html": [
        "\n",
        "  <div id=\"df-eaf6c25c-1330-4cb2-ab36-4419dc9e4e0d\">\n",
        "    <div class=\"colab-df-container\">\n",
        "      <div>\n",
        "<style scoped>\n",
        "    .dataframe tbody tr th:only-of-type {\n",
        "        vertical-align: middle;\n",
        "    }\n",
        "\n",
        "    .dataframe tbody tr th {\n",
        "        vertical-align: top;\n",
        "    }\n",
        "\n",
        "    .dataframe thead th {\n",
        "        text-align: right;\n",
        "    }\n",
        "</style>\n",
        "<table border=\"1\" class=\"dataframe\">\n",
        "  <thead>\n",
        "    <tr style=\"text-align: right;\">\n",
        "      <th></th>\n",
        "      <th>col_a</th>\n",
        "      <th>col_b</th>\n",
        "    </tr>\n",
        "  </thead>\n",
        "  <tbody>\n",
        "    <tr>\n",
        "      <th>0</th>\n",
        "      <td>1</td>\n",
        "      <td>2</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>1</th>\n",
        "      <td>2</td>\n",
        "      <td>5</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>2</th>\n",
        "
```

```
              "      <td>3</td>\n",
              "      <td>6</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-eaf6c25c-1330-4cb2-ab36-4419dc9e4e0d')\"\n",
              "              title=\"Convert this dataframe to an interactive
table.\"\n",
              "              style=\"display:none;\">\n",
              "        \n",
              "  <svg xmlns=\"http://www.w3.org/2000/svg\"
height=\"24px\"viewBox=\"0 0 24 24\"\n",
              "       width=\"24px\">\n",
              "    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
              "    <path d=\"M18.56 5.44l.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 10l.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 9.45l-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-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4
18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
              "  </svg>\n",
              "      </button>\n",
              "      \n",
              "  <style>\n",
              "    .colab-df-container {\n",
              "      display:flex;\n",
              "      flex-wrap:wrap;\n",
              "      gap: 12px;\n",
              "    }\n",
              "\n",
              "    .colab-df-convert {\n",
              "      background-color: #E8F0FE;\n",
              "      border: none;\n",
              "      border-radius: 50%;\n",
              "      cursor: pointer;\n",
              "      display: none;\n",
              "      fill: #1967D2;\n",
              "      height: 32px;\n",
              "      padding: 0 0 0 0;\n",
              "      width: 32px;\n",
              "    }\n",
              "\n",
              "    .colab-df-convert:hover {\n",
              "      background-color: #E2EBFA;\n",
              "      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",
              "    }\n",
              "\n",
              "    [theme=dark] .colab-df-convert {\n",
              "      background-color: #3B4455;\n",
              "      fill: #D2E3FC;\n",
              "    }\n",
              "\n",
              "    [theme=dark] .colab-df-convert:hover {\n",
              "      background-color: #434B5C;\n",
```

```
              "      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
              "      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
              "      fill: #FFFFFF;\n",
              "    }\n",
              "  </style>\n",
              "\n",
              "      <script>\n",
              "        const buttonEl =\n",
              "          document.querySelector('#df-eaf6c25c-1330-4cb2-ab36-
4419dc9e4e0d button.colab-df-convert');\n",
              "        buttonEl.style.display =\n",
              "          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
              "\n",
              "        async function convertToInteractive(key) {\n",
              "          const element = document.querySelector('#df-eaf6c25c-1330-
4cb2-ab36-4419dc9e4e0d');\n",
              "          const dataTable =\n",
              "            await
google.colab.kernel.invokeFunction('convertToInteractive',\n",
              "                                                     [key], {});\n",
              "          if (!dataTable) return;\n",
              "\n",
              "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
              "            '<a target=\"_blank\"
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",
              "          await google.colab.output.renderOutput(dataTable,
element);\n",
              "          const docLink = document.createElement('div');\n",
              "          docLink.innerHTML = docLinkHtml;\n",
              "          element.appendChild(docLink);\n",
              "        }\n",
              "      </script>\n",
              "    </div>\n",
              "  </div>\n",
              "  "
            ]
          },
          "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
  },
  "id": "CuQ4jV7GfjVC",
  "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  Mary   21"
      ],
      "text/html": [
        "\n",
        "  <div id=\"df-ee1cbbd8-98a1-459b-9355-84d07c6ba3bb\">\n",
        "    <div class=\"colab-df-container\">\n",
        "      <div>\n",
        "<style scoped>\n",
        "    .dataframe tbody tr th:only-of-type {\n",
        "        vertical-align: middle;\n",
        "    }\n",
        "\n",
        "    .dataframe tbody tr th {\n",
        "        vertical-align: top;\n",
        "    }\n",
        "\n",
        "    .dataframe thead th {\n",
        "        text-align: right;\n",
        "    }\n",
        "</style>\n",
        "<table border=\"1\" class=\"dataframe\">\n",
        "  <thead>\n",
        "    <tr style=\"text-align: right;\">\n",
        "      <th></th>\n",
        "      <th>Name</th>\n",
        "      <th>Age</th>\n",
        "    </tr>\n",
        "  </thead>\n",
        "  <tbody>\n",
        "    <tr>\n",
        "      <th>0</th>\n",
        "      <td>John</td>\n",
        "      <td>23</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>1</th>\n",
        "      <td>Jane</td>\n",
        "      <td>25</td>\n",
```

```
              "    </tr>\n",
              "    <tr>\n",
              "      <th>2</th>\n",
              "      <td>Mary</td>\n",
              "      <td>21</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-ee1cbbd8-98a1-459b-9355-84d07c6ba3bb')\"\n",
              "              title=\"Convert this dataframe to an interactive
table.\"\n",
              "              style=\"display:none;\">\n",
              "        \n",
              "  <svg xmlns=\"http://www.w3.org/2000/svg\"
height=\"24px\"viewBox=\"0 0 24 24\"\n",
              "       width=\"24px\">\n",
              "    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
              "    <path d=\"M18.56 5.44l.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 10l.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 9.45l-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-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4
18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
              "  </svg>\n",
              "      </button>\n",
              "      \n",
              "  <style>\n",
              "    .colab-df-container {\n",
              "      display:flex;\n",
              "      flex-wrap:wrap;\n",
              "      gap: 12px;\n",
              "    }\n",
              "\n",
              "    .colab-df-convert {\n",
              "      background-color: #E8F0FE;\n",
              "      border: none;\n",
              "      border-radius: 50%;\n",
              "      cursor: pointer;\n",
              "      display: none;\n",
              "      fill: #1967D2;\n",
              "      height: 32px;\n",
              "      padding: 0 0 0 0;\n",
              "      width: 32px;\n",
              "    }\n",
              "\n",
              "    .colab-df-convert:hover {\n",
              "      background-color: #E2EBFA;\n",
              "      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",
              "    }\n",
              "\n",
              "    [theme=dark] .colab-df-convert {\n",
              "      background-color: #3B4455;\n",
              "      fill: #D2E3FC;\n",
              "    }\n",
```

```
              "\n",
              "    [theme=dark] .colab-df-convert:hover {\n",
              "      background-color: #434B5C;\n",
              "      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
              "      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
              "      fill: #FFFFFF;\n",
              "    }\n",
              "  </style>\n",
              "\n",
              "      <script>\n",
              "        const buttonEl =\n",
              "          document.querySelector('#df-ee1cbbd8-98a1-459b-9355-
84d07c6ba3bb button.colab-df-convert');\n",
              "        buttonEl.style.display =\n",
              "          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
              "\n",
              "        async function convertToInteractive(key) {\n",
              "          const element = document.querySelector('#df-ee1cbbd8-98a1-
459b-9355-84d07c6ba3bb');\n",
              "          const dataTable =\n",
              "            await
google.colab.kernel.invokeFunction('convertToInteractive',\n",
              "                                                     [key], {});\n",
              "          if (!dataTable) return;\n",
              "\n",
              "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
              "            '<a target=\"_blank\"
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",
              "          await google.colab.output.renderOutput(dataTable,
element);\n",
              "          const docLink = document.createElement('div');\n",
              "          docLink.innerHTML = docLinkHtml;\n",
              "          element.appendChild(docLink);\n",
              "        }\n",
              "      </script>\n",
              "    </div>\n",
              "  </div>\n",
              "  "
            ]
          },
          "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": [
              "         time  year  month  day\n",
              "0  2023-01-01  2023      1    1\n",
              "1  2023-01-02  2023      1    2\n",
              "2  2023-01-03  2023      1    3\n",
              "3  2023-01-04  2023      1    4\n",
              "4  2023-01-05  2023      1    5\n",
              "5  2023-01-06  2023      1    6\n",
              "6  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      1   18\n",
  "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",
  "22 2023-01-23  2023      1   23\n",
  "23 2023-01-24  2023      1   24\n",
  "24 2023-01-25  2023      1   25\n",
  "25 2023-01-26  2023      1   26\n",
  "26 2023-01-27  2023      1   27\n",
  "27 2023-01-28  2023      1   28\n",
  "28 2023-01-29  2023      1   29\n",
  "29 2023-01-30  2023      1   30\n",
  "30 2023-01-31  2023      1   31\n",
  "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  2023      2    6\n",
  "37 2023-02-07  2023      2    7\n",
  "38 2023-02-08  2023      2    8\n",
  "39 2023-02-09  2023      2    9\n",
  "40 2023-02-10  2023      2   10"
],
"text/html": [
  "\n",
  "  <div id=\"df-5ffea3b0-d3fb-4993-b0df-8086a405ad62\">\n",
  "    <div class=\"colab-df-container\">\n",
  "      <div>\n",
  "<style scoped>\n",
  "    .dataframe tbody tr th:only-of-type {\n",
  "        vertical-align: middle;\n",
  "    }\n",
  "\n",
  "    .dataframe tbody tr th {\n",
  "        vertical-align: top;\n",
  "    }\n",
  "\n",
  "    .dataframe thead th {\n",
  "        text-align: right;\n",
  "    }\n",
  "</style>\n",
  "<table border=\"1\" class=\"dataframe\">\n",
  "  <thead>\n",
  "    <tr style=\"text-align: right;\">\n",
  "      <th></th>\n",
  "      <th>time</th>\n",
  "      <th>year</th>\n",
  "      <th>month</th>\n",
  "      <th>day</th>\n",
  "    </tr>\n",
  "  </thead>\n",
  "  <tbody>\n",
  "    <tr>\n",
  "      <th>0</th>\n",
  "      <td>2023-01-01</td>\n",
  "      <td>2023</td>\n",
```

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

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

```
"      <td>2023</td>\n",
"      <td>1</td>\n",
"      <td>18</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>18</th>\n",
"      <td>2023-01-19</td>\n",
"      <td>2023</td>\n",
"      <td>1</td>\n",
"      <td>19</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>19</th>\n",
"      <td>2023-01-20</td>\n",
"      <td>2023</td>\n",
"      <td>1</td>\n",
"      <td>20</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>20</th>\n",
"      <td>2023-01-21</td>\n",
"      <td>2023</td>\n",
"      <td>1</td>\n",
"      <td>21</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>21</th>\n",
"      <td>2023-01-22</td>\n",
"      <td>2023</td>\n",
"      <td>1</td>\n",
"      <td>22</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>22</th>\n",
"      <td>2023-01-23</td>\n",
"      <td>2023</td>\n",
"      <td>1</td>\n",
"      <td>23</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>23</th>\n",
"      <td>2023-01-24</td>\n",
"      <td>2023</td>\n",
"      <td>1</td>\n",
"      <td>24</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>24</th>\n",
"      <td>2023-01-25</td>\n",
"      <td>2023</td>\n",
"      <td>1</td>\n",
"      <td>25</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>25</th>\n",
"      <td>2023-01-26</td>\n",
"      <td>2023</td>\n",
"      <td>1</td>\n",
"      <td>26</td>\n",
```

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

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

```
              "    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
              "    <path d=\"M18.56 5.44l.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 10l.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 9.45l-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-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4
18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
              "  </svg>\n",
              "      </button>\n",
              "      \n",
              "  <style>\n",
              "    .colab-df-container {\n",
              "      display:flex;\n",
              "      flex-wrap:wrap;\n",
              "      gap: 12px;\n",
              "    }\n",
              "\n",
              "    .colab-df-convert {\n",
              "      background-color: #E8F0FE;\n",
              "      border: none;\n",
              "      border-radius: 50%;\n",
              "      cursor: pointer;\n",
              "      display: none;\n",
              "      fill: #1967D2;\n",
              "      height: 32px;\n",
              "      padding: 0 0 0 0;\n",
              "      width: 32px;\n",
              "    }\n",
              "\n",
              "    .colab-df-convert:hover {\n",
              "      background-color: #E2EBFA;\n",
              "      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",
              "    }\n",
              "\n",
              "    [theme=dark] .colab-df-convert {\n",
              "      background-color: #3B4455;\n",
              "      fill: #D2E3FC;\n",
              "    }\n",
              "\n",
              "    [theme=dark] .colab-df-convert:hover {\n",
              "      background-color: #434B5C;\n",
              "      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
              "      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
              "      fill: #FFFFFF;\n",
              "    }\n",
              "  </style>\n",
              "\n",
              "      <script>\n",
              "        const buttonEl =\n",
              "          document.querySelector('#df-5ffea3b0-d3fb-4993-b0df-
8086a405ad62 button.colab-df-convert');\n",
              "        buttonEl.style.display =\n",
              "          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
              "\n",
              "        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",
              "          if (!dataTable) return;\n",
              "\n",
              "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
              "            '<a target=\"_blank\"
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",
              "          await google.colab.output.renderOutput(dataTable,
element);\n",
              "          const docLink = document.createElement('div');\n",
              "          docLink.innerHTML = docLinkHtml;\n",
              "          element.appendChild(docLink);\n",
              "        }\n",
              "      </script>\n",
              "    </div>\n",
              "  </div>\n",
              "  "
            ]
          },
          "metadata": {},
          "execution_count": 17
        }
      ]
    },
    {
      "cell_type": "markdown",
      "source": [
        "## 10. Create 2D list to DataFrame\n",
        "\n",
        "lists = [[1, 'aaa', 22],\n",
        "         [2, 'bbb', 25],\n",
        "         [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": {
  "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",
        "0     1  aaa    22\n",
        "1     2  bbb    25\n",
        "2     3  ccc    24"
      ],
      "text/html": [
        "\n",
        "  <div id=\"df-5486a87f-40d0-49bd-8fe1-40ca2559ff4f\">\n",
        "    <div class=\"colab-df-container\">\n",
        "      <div>\n",
        "<style scoped>\n",
        "    .dataframe tbody tr th:only-of-type {\n",
        "        vertical-align: middle;\n",
        "    }\n",
        "\n",
        "    .dataframe tbody tr th {\n",
        "        vertical-align: top;\n",
        "    }\n",
        "\n",
        "    .dataframe thead th {\n",
        "        text-align: right;\n",
        "    }\n",
        "</style>\n",
        "<table border=\"1\" class=\"dataframe\">\n",
        "  <thead>\n",
        "    <tr style=\"text-align: right;\">\n",
        "      <th></th>\n",
        "      <th>col1</th>\n",
        "      <th>col2</th>\n",
        "      <th>col3</th>\n",
        "    </tr>\n",
        "  </thead>\n",
        "  <tbody>\n",
        "    <tr>\n",
        "      <th>0</th>\n",
        "      <td>1</td>\n",
        "      <td>aaa</td>\n",
        "      <td>22</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>1</th>\n",
        "      <td>2</td>\n",
        "```
```

```
              "      <td>bbb</td>\n",
              "      <td>25</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>2</th>\n",
              "      <td>3</td>\n",
              "      <td>ccc</td>\n",
              "      <td>24</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-5486a87f-40d0-49bd-8fe1-40ca2559ff4f')\"\n",
              "              title=\"Convert this dataframe to an interactive
table.\"\n",
              "              style=\"display:none;\">\n",
              "        \n",
              "  <svg xmlns=\"http://www.w3.org/2000/svg\"
height=\"24px\"viewBox=\"0 0 24 24\"\n",
              "       width=\"24px\">\n",
              "    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
              "    <path d=\"M18.56 5.44l.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 10l.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 9.45l-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-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4
18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
              "  </svg>\n",
              "      </button>\n",
              "      \n",
              "  <style>\n",
              "    .colab-df-container {\n",
              "      display:flex;\n",
              "      flex-wrap:wrap;\n",
              "      gap: 12px;\n",
              "    }\n",
              "\n",
              "    .colab-df-convert {\n",
              "      background-color: #E8F0FE;\n",
              "      border: none;\n",
              "      border-radius: 50%;\n",
              "      cursor: pointer;\n",
              "      display: none;\n",
              "      fill: #1967D2;\n",
              "      height: 32px;\n",
              "      padding: 0 0 0 0;\n",
              "      width: 32px;\n",
              "    }\n",
              "\n",
              "    .colab-df-convert:hover {\n",
              "      background-color: #E2EBFA;\n",
              "      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",
              "    }\n",
              "\n",
              "    [theme=dark] .colab-df-convert {\n",
```

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