```
{
  "cells": [
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "McSxJAwcOdZ1"
      },
      "source": [
        "# Basic Python"
      ]
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "CU48hgo4Owz5"
      },
      "source": [
        "## 1. Split this string"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 1,
      "metadata": {
        "id": "s07c7JK7Oqt-"
      },
```

```
  "outputs": [],
  "source": [
    "s = \"Hi there Sam!\""
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 2,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "6mGVa3SQYLkb",
    "outputId": "ba07984e-2b3e-4631-e7b9-128fd9ddef95"
  },
  "outputs": [
    {
      "data": {
        "text/plain": [
          "['Hi', 'there', 'Sam!']"
        ]
      },
      "execution_count": 2,
      "metadata": {},
      "output_type": "execute_result"
    }
```

```
  ],
  "source": [
    "s.split(\" \")"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "GH1QBn8HP375"
  },
  "source": [
    "*`italicized text`*## 2. Use .format() to print the following string. \n",
    "\n",
    "### Output should be: The diameter of Earth is 12742 kilometers."
  ]
},
{
  "cell_type": "code",
  "execution_count": 3,
  "metadata": {
    "id": "_ZHoml3kPqic"
  },
  "outputs": [],
  "source": [
    "planet = \"Earth\"\n",
    "diameter = 12742"
```

```
    ]
   },
   {
    "cell_type": "code",
    "execution_count": 4,
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/"
     },
     "id": "HyRyJv6CYPb4",
     "outputId": "f7131284-c21d-480e-f91e-ea85ae222b00"
    },
    "outputs": [
     {
      "name": "stdout",
      "output_type": "stream",
      "text": [
       "The diameter of Earth is 12742 kilometer\n"
      ]
     }
    ],
    "source": [
     "print(\"The diameter of {planet} is {diameter} kilometer\".format(planet=planet,diameter=diameter))"
    ]
   },
```

```
{
  "cell_type": "markdown",
  "metadata": {
    "id": "KE74ZEwkRExZ"
  },
  "source": [
    "## 3. In this nest dictionary grab the word \"hello\""
  ]
},
{
  "cell_type": "code",
  "execution_count": 5,
  "metadata": {
    "id": "fcVwbCc1QrQl"
  },
  "outputs": [],
  "source": [
    "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}"
  ]
},
{
  "cell_type": "code",
  "execution_count": 6,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
```

```
    },
    "id": "MvbkMZpXYRaw",
    "outputId": "0e3d48e2-017a-42c1-9cd4-106729992256"
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "hello\n"
     ]
    }
   ],
   "source": [
    "print(d['k1'][3]['tricky'][3]['target'][3])"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "bw0vVp-9ddjv"
   },
   "source": [
    "# Numpy"
   ]
  },
```

```
{
  "cell_type": "code",
  "execution_count": 7,
  "metadata": {
    "id": "LLiE_TYrhA1O"
  },
  "outputs": [],
  "source": [
    "import numpy as np"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "wOg8hinbgx30"
  },
  "source": [
    "## 4.1 Create an array of 10 zeros? \n",
    "## 4.2 Create an array of 10 fives?"
  ]
},
{
  "cell_type": "code",
  "execution_count": 8,
  "metadata": {
    "colab": {
```

```
      "base_uri": "https://localhost:8080/"
    },
    "id": "NHrirmgCYXvU",
    "outputId": "ff2c7690-60fe-4647-98ae-344f2ce70650"
  },
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]\n"
      ]
    }
  ],
  "source": [
    "array=np.zeros(10)\n",
    "print(array)"
  ]
},
{
  "cell_type": "code",
  "execution_count": 9,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
```

```
    "id": "e4005lsTYXxx",
    "outputId": "97ee2b76-46d3-4841-d145-3f4bc40756b6"
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]\n"
     ]
    }
   ],
   "source": [
    "array=np.ones(10)*5\n",
    "print(array)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "gZHHDUBvrMX4"
   },
   "source": [
    "## 5. Create an array of all the even integers from 20 to 35"
   ]
  },
```

```
{
  "cell_type": "code",
  "execution_count": 10,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "oAI2tbU2Yag-",
    "outputId": "bf34326a-0b18-465a-e3d8-f0d2a6edca22"
  },
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "[20 22 24 26 28 30 32 34]\n"
      ]
    }
  ],
  "source": [
    "array=np.arange(20,35,2)\n",
    "print(array) "
  ]
},
{
  "cell_type": "markdown",
```

```
  "metadata": {
    "id": "NaOM308NsRpZ"
  },
  "source": [
    "## 6. Create a 3x3 matrix with values ranging from 0 to 8"
  ]
},
{
  "cell_type": "code",
  "execution_count": 11,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "tOlEVH7BYceE",
    "outputId": "6a2aa614-123b-487e-d577-483b2d8a0f68"
  },
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "[[0 1 2]\n",
        " [3 4 5]\n",
        " [6 7 8]]\n"
      ]
```

```
    }
  ],
  "source": [
    "x =     np.arange(0, 9).reshape(3,3)\n",
    "print(x)"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "hQ0dnhAQuU_p"
  },
  "source": [
    "## 7. Concatinate a and b \n",
    "## a = np.array([1, 2, 3]), b = np.array([4, 5, 6])"
  ]
},
{
  "cell_type": "code",
  "execution_count": 12,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "rAPSw97aYfE0",
    "outputId": "dfa17040-abd8-4d94-e37a-8ea79fed06c5"
```

```
  },
  "outputs": [
    {
      "data": {
        "text/plain": [
          "array([1, 2, 3, 4, 5, 6])"
        ]
      },
      "execution_count": 12,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "a = np.array([1, 2,3])\n",
    "b = np.array([4,5, 6])\n",
    "np.concatenate((a, b), axis=0)"
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {
    "id": "dlPEY9DRwZga"
  },
  "source": [
    "# Pandas"
```

```
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "ijoYW51zwr87"
  },
  "source": [
    "## 8. Create a dataframe with 3 rows and 2 columns"
  ]
},
{
  "cell_type": "code",
  "execution_count": 13,
  "metadata": {
    "id": "T5OxJRZ8uvR7"
  },
  "outputs": [],
  "source": [
    "import pandas as pd\n"
  ]
},
{
  "cell_type": "code",
  "execution_count": 15,
  "metadata": {
```

```
  "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 143
  },
  "id": "xNpI_XXoYhs0",
  "outputId": "b9c78d52-c29d-47e2-cb7e-62673f74ab54"
},
"outputs": [
  {
    "data": {
      "text/html": [
        "\n",
        "    <div id=\"df-162525ea-3a49-48a8-ab26-95176a944ada\">\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>69</td>\n",
"            <td>88</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>1</th>\n",
"            <td>17</td>\n",
"            <td>90</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>2</th>\n",
"            <td>80</td>\n",
"            <td>37</td>\n",
"        </tr>\n",
```

```
"    </tbody>\n",
"</table>\n",
"</div>\n",
"           <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-162525ea-3a49-48a8-ab26-95176a944ada')\"\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-162525ea-3a49-48a8-ab26-95176a944ada 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-162525ea-3a49-48a8-ab26-95176a944ada');\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",
        "    "
       ],
       "text/plain": [
        "      cola    colb\n",
        "0       69       88\n",
        "1       17       90\n",
        "2       80       37"
       ]
      },
     "execution_count": 15,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "A = np.random.randint(100, size=(3,2))\n",
    "df = pd.DataFrame(A,columns=['cola', 'colb'])\n",
    "df\n"
   ]
```

```
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "UXSmdNclyJQD"
  },
  "source": [
    "## 9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023"
  ]
},
{
  "cell_type": "code",
  "execution_count": 21,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 1000
    },
    "id": "dgyC0JhVYl4F",
    "outputId": "ba698c0f-e487-45b2-88c2-373a28e78567"
  },
  "outputs": [
    {
      "data": {
        "text/html": [
          "\n",
```

```
"    <div id=\"df-ace4b22d-efb9-4b83-9a7f-96c83284c02d\">\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>date</th>\n",
"        </tr>\n",
"    </thead>\n",
"    <tbody>\n",
"        <tr>\n",
"            <th>0</th>\n",
```

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

```
"        <tr>\n",
"            <th>7</th>\n",
"            <td>2023-01-08</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>8</th>\n",
"            <td>2023-01-09</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>9</th>\n",
"            <td>2023-01-10</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>10</th>\n",
"            <td>2023-01-11</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>11</th>\n",
"            <td>2023-01-12</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>12</th>\n",
"            <td>2023-01-13</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>13</th>\n",
```

```
"            <td>2023-01-14</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>14</th>\n",
"            <td>2023-01-15</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>15</th>\n",
"            <td>2023-01-16</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>16</th>\n",
"            <td>2023-01-17</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>17</th>\n",
"            <td>2023-01-18</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>18</th>\n",
"            <td>2023-01-19</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>19</th>\n",
"            <td>2023-01-20</td>\n",
"        </tr>\n",
```

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

```
"            <td>2023-01-27</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>27</th>\n",
"            <td>2023-01-28</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>28</th>\n",
"            <td>2023-01-29</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>29</th>\n",
"            <td>2023-01-30</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>30</th>\n",
"            <td>2023-01-31</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>31</th>\n",
"            <td>2023-02-01</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>32</th>\n",
"            <td>2023-02-02</td>\n",
"        </tr>\n",
```

```
"        <tr>\n",
"            <th>33</th>\n",
"            <td>2023-02-03</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>34</th>\n",
"            <td>2023-02-04</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>35</th>\n",
"            <td>2023-02-05</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>36</th>\n",
"            <td>2023-02-06</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>37</th>\n",
"            <td>2023-02-07</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>38</th>\n",
"            <td>2023-02-08</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>39</th>\n",
```

```
"            <td>2023-02-09</td>\n",
"        </tr>\n",
"        <tr>\n",
"            <th>40</th>\n",
"            <td>2023-02-10</td>\n",
"        </tr>\n",
"    </tbody>\n",
"</table>\n",
"</div>\n",
"            <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-ace4b22d-efb9-4b83-9a7f-96c83284c02d')\"\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-ace4b22d-efb9-4b83-9a7f-96c83284c02d
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-ace4b22d-efb9-4b83-9a7f-96c83284c02d');\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",
      "    "
    ],
    "text/plain": [
      "                date\n",
      "0    2023-01-01\n",
      "1    2023-01-02\n",
      "2    2023-01-03\n",
      "3    2023-01-04\n",
      "4    2023-01-05\n",
      "5    2023-01-06\n",
      "6    2023-01-07\n",
      "7    2023-01-08\n",
      "8    2023-01-09\n",
```

"9 2023-01-10\n",

"10 2023-01-11\n",

"11 2023-01-12\n",

"12 2023-01-13\n",

"13 2023-01-14\n",

"14 2023-01-15\n",

"15 2023-01-16\n",

"16 2023-01-17\n",

"17 2023-01-18\n",

"18 2023-01-19\n",

"19 2023-01-20\n",

"20 2023-01-21\n",

"21 2023-01-22\n",

"22 2023-01-23\n",

"23 2023-01-24\n",

"24 2023-01-25\n",

"25 2023-01-26\n",

"26 2023-01-27\n",

"27 2023-01-28\n",

"28 2023-01-29\n",

"29 2023-01-30\n",

"30 2023-01-31\n",

"31 2023-02-01\n",

"32 2023-02-02\n",

"33 2023-02-03\n",

"34 2023-02-04\n",

```
          "35 2023-02-05\n",
          "36 2023-02-06\n",
          "37 2023-02-07\n",
          "38 2023-02-08\n",
          "39 2023-02-09\n",
          "40 2023-02-10"
        ]
      },
      "execution_count": 21,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "import pandas as pd\n",
    "    \n",
    "df = pd.DataFrame()\n",
    "\n",
    "df['date'] = pd.date_range(start=\"1/1/2023\",end=\"2/10/2023\", freq ='24H')\n",
    "\n",
    "df.head(len(df[\"date\"]))"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
```

```
    "id": "ZizSetD-y5az"
   },
   "source": [
    "## 10. Create 2D list to DataFrame\n",
    "\n",
    "lists = [[1, 'aaa', 22],\n",
    "                [2, 'bbb', 25],\n",
    "                [3, 'ccc', 24]]"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 17,
   "metadata": {
    "id": "_XMC8aEt0llB"
   },
   "outputs": [],
   "source": [
    "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 23,
   "metadata": {
    "colab": {
```

```
    "base_uri": "https://localhost:8080/",
    "height": 143
  },
  "id": "knH76sDKYsVX",
  "outputId": "e7234dd2-1abd-4f66-fde7-d918d5b86d91"
},
"outputs": [
  {
    "data": {
      "text/html": [
        "\n",
        "    <div id=\"df-3531ec0a-b30d-4b88-8bf4-9143c8cf5be4\">\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>1</th>\n",
"            <th>2</th>\n",
"            <th>3</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-3531ec0a-b30d-4b88-8bf4-9143c8cf5be4')\"\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-3531ec0a-b30d-4b88-8bf4-9143c8cf5be4
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-3531ec0a-b30d-4b88-8bf4-9143c8cf5be4');\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",
        "    "
       ],
       "text/plain": [
        "     1       2     3\n",
        "0    1    aaa    22\n",
        "1    2    bbb    25\n",
        "2    3    ccc    24"
       ]
      },
     "execution_count": 23,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df = pd.DataFrame(lists, columns =['1','2','3']) \n",
```

```
      "df"
    ]
   }
  ],
 "metadata": {
  "colab": {
   "collapsed_sections": [],
   "provenance": []
  },
  "kernelspec": {
   "display_name": "Python 3 (ipykernel)",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.9.7"
  }
```

```
  },
  "nbformat": 4,
  "nbformat_minor": 1
}
```