{

"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": {

"executionInfo": {

"elapsed": 6,

"status": "ok",

"timestamp": 1663077340733,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "s07c7JK7Oqt-"

},

"outputs": [],

"source": [

"s = \"Hi there San!\""

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"executionInfo": {

"elapsed": 12,

"status": "ok",

"timestamp": 1663077344454,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "6mGVa3SQYLkb",

"outputId": "bb200ad1-a6cb-485d-a972-b1bdabecf090"

},

"outputs": [

{

"data": {

"text/plain": [

"['Hi', 'there', 'San!']"

]

},

"execution\_count": 2,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"s.split()"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "GH1QBn8HP375"

},

"source": [

"## 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": {

"executionInfo": {

"elapsed": 384,

"status": "ok",

"timestamp": 1663077363417,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "\_ZHoml3kPqic"

},

"outputs": [],

"source": [

"planet = \"Earth\"\n",

"diameter = 12742"

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"executionInfo": {

"elapsed": 420,

"status": "ok",

"timestamp": 1663077365819,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "HyRyJv6CYPb4",

"outputId": "8b79b956-8e2c-488a-f6d2-691fa01daf91"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"The diameter of Earth is 12742 Kilometers\n"

]

}

],

"source": [

"print(f\"The diameter of {planet} is {diameter} Kilometers\")"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "KE74ZEwkRExZ"

},

"source": [

"## 3. In this nest dictionary grab the word \"hello\""

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"metadata": {

"executionInfo": {

"elapsed": 355,

"status": "ok",

"timestamp": 1663077371203,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "fcVwbCc1QrQI"

},

"outputs": [],

"source": [

"d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"executionInfo": {

"elapsed": 371,

"status": "ok",

"timestamp": 1663077383357,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "MvbkMZpXYRaw",

"outputId": "17ad9c19-cadb-4182-a752-a70fb465664f"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"hello\n"

]

}

],

"source": [

"print(d['k1'][-1]['tricky'][-1]['target'][-1])"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "bw0vVp-9ddjv"

},

"source": [

"# Numpy"

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"metadata": {

"executionInfo": {

"elapsed": 407,

"status": "ok",

"timestamp": 1663077388174,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"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": 9,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"executionInfo": {

"elapsed": 370,

"status": "ok",

"timestamp": 1663077435670,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "NHrirmgCYXvU",

"outputId": "56108681-ba84-4f8c-c296-f7076eb8ba0f"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]\n"

]

}

],

"source": [

"arrayZ=np.zeros(10)\n",

"print(arrayZ)"

]

},

{

"cell\_type": "code",

"execution\_count": 10,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"executionInfo": {

"elapsed": 504,

"status": "ok",

"timestamp": 1663077486112,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "e4005lsTYXxx",

"outputId": "f69379ab-297a-43b5-ec16-f1e3e66c7195"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]\n"

]

}

],

"source": [

"arrayF=np.ones(10)\*5\n",

"print(arrayF)"

]

},

{

"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": 11,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"executionInfo": {

"elapsed": 12,

"status": "ok",

"timestamp": 1663077499783,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "oAI2tbU2Yag-",

"outputId": "d537b0c7-bf54-4902-b4d5-da6d5dc0bba6"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"[20 22 24 26 28 30 32 34]\n"

]

}

],

"source": [

"even=np.arange(20,35,2)\n",

"print(even)"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "NaOM308NsRpZ"

},

"source": [

"## 6. Create a 3x3 matrix with values ranging from 0 to 8"

]

},

{

"cell\_type": "code",

"execution\_count": 12,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"executionInfo": {

"elapsed": 346,

"status": "ok",

"timestamp": 1663077512678,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "tOlEVH7BYceE",

"outputId": "0b98882c-cef1-4b45-ddb0-87ea1b954c4c"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"[[0 1 2]\n",

" [3 4 5]\n",

" [6 7 8]]\n"

]

}

],

"source": [

"mat=np.arange(0, 9).reshape(3,3)\n",

"print(mat)"

]

},

{

"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": 13,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"executionInfo": {

"elapsed": 362,

"status": "ok",

"timestamp": 1663077534642,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "rAPSw97aYfE0",

"outputId": "49f575c7-1ccd-4df5-e7bf-536b6144edb2"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"[1 2 3 4 5 6]\n"

]

}

],

"source": [

"a=np.array([1,2,3])\n",

"b=np.array([4,5,6])\n",

"concat=np.concatenate((a,b),axis=None)\n",

"print(concat)"

]

},

{

"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": 14,

"metadata": {

"executionInfo": {

"elapsed": 370,

"status": "ok",

"timestamp": 1663077540360,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "T5OxJRZ8uvR7"

},

"outputs": [],

"source": [

"import pandas as pd\n"

]

},

{

"cell\_type": "code",

"execution\_count": 15,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 144

},

"executionInfo": {

"elapsed": 545,

"status": "ok",

"timestamp": 1663077559650,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "xNpI\_XXoYhs0",

"outputId": "f5da08e6-fb0f-4266-ea52-cec6a7e2ef43"

},

"outputs": [

{

"data": {

"text/html": [

"\n",

" <div id=\"df-85047f49-9ade-4c26-8951-3bbbe88d17ee\">\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>0</th>\n",

" <th>1</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>0</td>\n",

" <td>1</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>2</td>\n",

" <td>3</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>4</td>\n",

" <td>5</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-85047f49-9ade-4c26-8951-3bbbe88d17ee')\"\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-85047f49-9ade-4c26-8951-3bbbe88d17ee 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-85047f49-9ade-4c26-8951-3bbbe88d17ee');\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": [

" 0 1\n",

"0 0 1\n",

"1 2 3\n",

"2 4 5"

]

},

"execution\_count": 15,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataframe=pd.DataFrame(np.arange(0,6).reshape(3,2))\n",

"dataframe"

]

},

{

"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": 16,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"executionInfo": {

"elapsed": 445,

"status": "ok",

"timestamp": 1663077576287,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "dgyC0JhVYl4F",

"outputId": "aaddb98b-6ff5-4acc-ea37-113eb2c9bc52"

},

"outputs": [

{

"data": {

"text/plain": [

"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\n",

"dtype: datetime64[ns]"

]

},

"execution\_count": 16,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dates=pd.date\_range(start='1/1/2023', end='2/10/2023')\n",

"pd.Series(dates)"

]

},

{

"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": {

"executionInfo": {

"elapsed": 353,

"status": "ok",

"timestamp": 1663077580624,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "\_XMC8aEt0llB"

},

"outputs": [],

"source": [

"lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]"

]

},

{

"cell\_type": "code",

"execution\_count": 18,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 144

},

"executionInfo": {

"elapsed": 362,

"status": "ok",

"timestamp": 1663077591455,

"user": {

"displayName": "Sooraj Kumar",

"userId": "15840049306769233973"

},

"user\_tz": -330

},

"id": "knH76sDKYsVX",

"outputId": "fe11bf32-73b6-4001-a5e0-012fafd5f73d"

},

"outputs": [

{

"data": {

"text/html": [

"\n",

" <div id=\"df-059ec563-1a3f-4875-be13-43402a2c0d97\">\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>0</th>\n",

" <th>1</th>\n",

" <th>2</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-059ec563-1a3f-4875-be13-43402a2c0d97')\"\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-059ec563-1a3f-4875-be13-43402a2c0d97 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-059ec563-1a3f-4875-be13-43402a2c0d97');\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": [

" 0 1 2\n",

"0 1 aaa 22\n",

"1 2 bbb 25\n",

"2 3 ccc 24"

]

},

"execution\_count": 18,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataf = pd.DataFrame(lists)\n",

"dataf"

]

}

],

"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": "3.9.12"

},

"vscode": {

"interpreter": {

"hash": "b337b16e1f284c9fe7de692799556d56c1809887abe3f5a49ffeb9e7df151cfb"

}

}

},

"nbformat": 4,

"nbformat\_minor": 1

}