{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"provenance": []

},

"kernelspec": {

"name": "python3",

"display\_name": "Python 3"

},

"language\_info": {

"name": "python"

}

},

"cells": [

{

"cell\_type": "code",

"execution\_count": 1,

"metadata": {

"id": "uPxUkuMU\_9L1"

},

"outputs": [],

"source": [

"import pandas as pd\n",

"import numpy as np \n",

"import seaborn as sns"

]

},

{

"cell\_type": "markdown",

"source": [

"split this string\n"

],

"metadata": {

"id": "YD1QHXgTAOQs"

}

},

{

"cell\_type": "code",

"source": [

"\n",

"s=\"Hi there Sam!\"\n",

"x = s.split()\n",

"\n",

"print(x)"

],

"metadata": {

"colab": {

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

},

"id": "lfwlObfFG-Qj",

"outputId": "d416843c-fa47-434b-8c32-e606909f99c2"

},

"execution\_count": 2,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"['Hi', 'there', 'Sam!']\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"use.format() to print the following string\n",

"\n",

"output should be: The deameter of Earth is 12742 kms."

],

"metadata": {

"id": "X6Jvvd1iAUek"

}

},

{

"cell\_type": "code",

"source": [

"\n",

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

"diameter = 12742\n",

"print( 'The diameter of {} is {} kilometers.' .format(planet,diameter));"

],

"metadata": {

"colab": {

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

},

"id": "hgctiCE6G35e",

"outputId": "bcd5db66-a063-4a48-a756-3c328b3da38a"

},

"execution\_count": 3,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"The diameter of Earth is 12742 kilometers.\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"IN this nest dicftionary grab the word\"hello\""

],

"metadata": {

"id": "UIYV2ZgQAf67"

}

},

{

"cell\_type": "code",

"source": [

" \n",

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

"print(d['k1'][3][\"tricky\"][3]['target'][3])"

],

"metadata": {

"colab": {

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

},

"id": "Up\_eM\_frHhBf",

"outputId": "3bd34f9f-a1d7-456a-c0b0-1bb26014fc4c"

},

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"hello\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"create ab array of 10 zeros?"

],

"metadata": {

"id": "zmvKtx\_AAj0V"

}

},

{

"cell\_type": "code",

"source": [

"\n",

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

"print(\"An array of 10 zeros:\")\n",

"print(array)\n",

"\n",

"\n"

],

"metadata": {

"colab": {

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

},

"id": "EV\_XZpFBJPMi",

"outputId": "43a34220-a216-439a-de2d-4323faf00384"

},

"execution\_count": 4,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"An array of 10 zeros:\n",

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

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"create ab array of 10 fives?"

],

"metadata": {

"id": "j3Zsf9\_oApGz"

}

},

{

"cell\_type": "code",

"source": [

"\n",

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

"print(\"An array of 10 fives:\")\n",

"print(array)"

],

"metadata": {

"colab": {

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

},

"id": "M7qfobNOMh1A",

"outputId": "81271a02-26fc-4043-b60d-94379ced4c1e"

},

"execution\_count": 5,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"An array of 10 fives:\n",

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

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"create an array of all the even integers from 20 to 35\n"

],

"metadata": {

"id": "xckeFQ1CAuRL"

}

},

{

"cell\_type": "code",

"source": [

"import numpy as np\n",

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

"print(\"Array of all the even integers from 20 to 35\")\n",

"print(array) "

],

"metadata": {

"colab": {

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

},

"id": "SgmMFkG6Jg5U",

"outputId": "559bb9bd-db1d-4d1f-ddf9-500dca1b3458"

},

"execution\_count": 6,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Array of all the even integers from 20 to 35\n",

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

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"create a 3\*3 matrix with values ranging from 0 to 8\n"

],

"metadata": {

"id": "1ZjWjniCA1L4"

}

},

{

"cell\_type": "code",

"source": [

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

"print(x)"

],

"metadata": {

"colab": {

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

},

"id": "RIEiDIqtJotT",

"outputId": "224073b0-f01c-4542-c0ed-d5c345792d16"

},

"execution\_count": 7,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"[[0 1 2]\n",

" [3 4 5]\n",

" [6 7 8]]\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"concatinate a and b"

],

"metadata": {

"id": "\_67F5rLHA7-c"

}

},

{

"cell\_type": "code",

"source": [

"\n",

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

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

"c = np.concatenate((a,b))\n",

"print(c)"

],

"metadata": {

"colab": {

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

},

"id": "Z9ifLBDoJ0S-",

"outputId": "cf5475d6-d2f4-456d-e4c4-50df6650b13c"

},

"execution\_count": 8,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

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

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"\n",

"create a data frame with 3 rows and 2 columns"

],

"metadata": {

"id": "7mpOrILLA\_\_g"

}

},

{

"cell\_type": "code",

"source": [

"\n",

"\n",

"data = [['sindhu', 21], ['yazhini', 20], ['vishal', 22]]\n",

"df = pd.DataFrame(data, columns=['Name', 'Age'])\n",

"df\n"

],

"metadata": {

"colab": {

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

"height": 143

},

"id": "hUDGYrc0IJDB",

"outputId": "e131ca82-2598-434a-8adf-c3c46517b8ac"

},

"execution\_count": 9,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" Name Age\n",

"0 sindhu 21\n",

"1 yazhini 20\n",

"2 vishal 22"

],

"text/html": [

"\n",

" <div id=\"df-60adf5a9-2c02-47c2-abc7-24e81f66ffa8\">\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>sindhu</td>\n",

" <td>21</td>\n",

" </tr>\n",

" <tr>\n",

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

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

" <td>20</td>\n",

" </tr>\n",

" <tr>\n",

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

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

" <td>22</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-60adf5a9-2c02-47c2-abc7-24e81f66ffa8')\"\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-60adf5a9-2c02-47c2-abc7-24e81f66ffa8 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-60adf5a9-2c02-47c2-abc7-24e81f66ffa8');\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": 9

}

]

},

{

"cell\_type": "markdown",

"source": [

"generate the series of dates from 1st jan, 2023 to 10th feb, 2023"

],

"metadata": {

"id": "rf1YtjD0BDNx"

}

},

{

"cell\_type": "code",

"source": [

"\n",

"import datetime\n",

"pd.date\_range(start=\"2023-01-01\",end=\"2023-02-10\").to\_pydatetime().tolist()\n"

],

"metadata": {

"colab": {

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

},

"id": "bJ322oQ-Ilho",

"outputId": "95a78285-7c66-4209-fb2d-9921ba06da81"

},

"execution\_count": 10,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"[datetime.datetime(2023, 1, 1, 0, 0),\n",

" datetime.datetime(2023, 1, 2, 0, 0),\n",

" datetime.datetime(2023, 1, 3, 0, 0),\n",

" datetime.datetime(2023, 1, 4, 0, 0),\n",

" datetime.datetime(2023, 1, 5, 0, 0),\n",

" datetime.datetime(2023, 1, 6, 0, 0),\n",

" datetime.datetime(2023, 1, 7, 0, 0),\n",

" datetime.datetime(2023, 1, 8, 0, 0),\n",

" datetime.datetime(2023, 1, 9, 0, 0),\n",

" datetime.datetime(2023, 1, 10, 0, 0),\n",

" datetime.datetime(2023, 1, 11, 0, 0),\n",

" datetime.datetime(2023, 1, 12, 0, 0),\n",

" datetime.datetime(2023, 1, 13, 0, 0),\n",

" datetime.datetime(2023, 1, 14, 0, 0),\n",

" datetime.datetime(2023, 1, 15, 0, 0),\n",

" datetime.datetime(2023, 1, 16, 0, 0),\n",

" datetime.datetime(2023, 1, 17, 0, 0),\n",

" datetime.datetime(2023, 1, 18, 0, 0),\n",

" datetime.datetime(2023, 1, 19, 0, 0),\n",

" datetime.datetime(2023, 1, 20, 0, 0),\n",

" datetime.datetime(2023, 1, 21, 0, 0),\n",

" datetime.datetime(2023, 1, 22, 0, 0),\n",

" datetime.datetime(2023, 1, 23, 0, 0),\n",

" datetime.datetime(2023, 1, 24, 0, 0),\n",

" datetime.datetime(2023, 1, 25, 0, 0),\n",

" datetime.datetime(2023, 1, 26, 0, 0),\n",

" datetime.datetime(2023, 1, 27, 0, 0),\n",

" datetime.datetime(2023, 1, 28, 0, 0),\n",

" datetime.datetime(2023, 1, 29, 0, 0),\n",

" datetime.datetime(2023, 1, 30, 0, 0),\n",

" datetime.datetime(2023, 1, 31, 0, 0),\n",

" datetime.datetime(2023, 2, 1, 0, 0),\n",

" datetime.datetime(2023, 2, 2, 0, 0),\n",

" datetime.datetime(2023, 2, 3, 0, 0),\n",

" datetime.datetime(2023, 2, 4, 0, 0),\n",

" datetime.datetime(2023, 2, 5, 0, 0),\n",

" datetime.datetime(2023, 2, 6, 0, 0),\n",

" datetime.datetime(2023, 2, 7, 0, 0),\n",

" datetime.datetime(2023, 2, 8, 0, 0),\n",

" datetime.datetime(2023, 2, 9, 0, 0),\n",

" datetime.datetime(2023, 2, 10, 0, 0)]"

]

},

"metadata": {},

"execution\_count": 10

}

]

},

{

"cell\_type": "markdown",

"source": [

"create 2D list to dataframe"

],

"metadata": {

"id": "zqlddA0VBIfl"

}

},

{

"cell\_type": "code",

"source": [

"\n",

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

"df = pd.DataFrame(lists, columns =['Id', 'Name','Age']) \n",

"print(df )"

],

"metadata": {

"colab": {

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

},

"id": "IO0upKctIvap",

"outputId": "d1a1adb9-9f3e-4a95-bfd9-c5355d4867b3"

},

"execution\_count": 11,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

" Id Name Age\n",

"0 1 aaa 22\n",

"1 2 bbb 25\n",

"2 3 ccc 24\n"

]

}

]

},

{

"cell\_type": "code",

"source": [],

"metadata": {

"id": "hZ8GdR-GBlFv"

},

"execution\_count": null,

"outputs": []

}

]

}