{

"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": [

"s.split()"

],

"metadata": {

"id": "6mGVa3SQYLkb",

"outputId": "c637ce2d-751b-48fd-eee0-de80ec814a2c",

"colab": {

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

}

},

"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": [

"a=\"The diameter of {} is {} kilometers\".format(planet,diameter)\n",

"print(a)"

],

"metadata": {

"id": "HyRyJv6CYPb4",

"outputId": "84822d74-fdb0-4e34-939d-6df7551500b7",

"colab": {

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

}

},

"execution\_count": 4,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"The diameter of Earth is 12742 kilometers\n"

]

}

]

},

{

"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": [

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

],

"metadata": {

"id": "MvbkMZpXYRaw",

"outputId": "eba81713-ceec-4340-9e04-1a04d47521b4",

"colab": {

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

}

},

"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": [

"np.zeros(10)"

],

"metadata": {

"id": "NHrirmgCYXvU",

"outputId": "604aa210-4c93-4f06-a32b-b8227060b1d4",

"colab": {

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

}

},

"execution\_count": 8,

"outputs": [

{

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

"data": {

"text/plain": [

"array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])"

]

},

"metadata": {},

"execution\_count": 8

}

]

},

{

"cell\_type": "code",

"source": [

"np.ones(10)\*5"

],

"metadata": {

"id": "e4005lsTYXxx",

"outputId": "8b719610-0a7b-4f21-937e-6a5224667437",

"colab": {

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

}

},

"execution\_count": 9,

"outputs": [

{

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

"data": {

"text/plain": [

"array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])"

]

},

"metadata": {},

"execution\_count": 9

}

]

},

{

"cell\_type": "markdown",

"source": [

"## 5. Create an array of all the even integers from 20 to 35"

],

"metadata": {

"id": "gZHHDUBvrMX4"

}

},

{

"cell\_type": "code",

"source": [

"np.arange(20,35,2)"

],

"metadata": {

"id": "oAI2tbU2Yag-",

"outputId": "4c14c25d-6bcf-416c-e068-d1652b322e3b",

"colab": {

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

}

},

"execution\_count": 10,

"outputs": [

{

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

"data": {

"text/plain": [

"array([20, 22, 24, 26, 28, 30, 32, 34])"

]

},

"metadata": {},

"execution\_count": 10

}

]

},

{

"cell\_type": "markdown",

"source": [

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

],

"metadata": {

"id": "NaOM308NsRpZ"

}

},

{

"cell\_type": "code",

"source": [

"np.array([[0,1,2],[3,4,5],[6,7,8]])"

],

"metadata": {

"id": "tOlEVH7BYceE",

"outputId": "fb8ed5f7-5737-49c2-ef2c-07c8daa81ed5",

"colab": {

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

}

},

"execution\_count": 11,

"outputs": [

{

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

"data": {

"text/plain": [

"array([[0, 1, 2],\n",

" [3, 4, 5],\n",

" [6, 7, 8]])"

]

},

"metadata": {},

"execution\_count": 11

}

]

},

{

"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",

"np.concatenate((a,b))"

],

"metadata": {

"id": "rAPSw97aYfE0",

"outputId": "7588074a-58fa-4d31-8fe9-d13c267da138",

"colab": {

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

}

},

"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": "code",

"source": [

"data = {\n",

" \"calories\": [420, 380, 390],\n",

" \"duration\": [50, 40, 45]\n",

"}\n",

"\n",

"#load data into a DataFrame object:\n",

"df = pd.DataFrame(data)"

],

"metadata": {

"id": "xNpI\_XXoYhs0"

},

"execution\_count": 14,

"outputs": []

},

{

"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": [

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

],

"metadata": {

"id": "dgyC0JhVYl4F",

"outputId": "e3ed85d9-66a4-4c47-9386-025b244aaf35",

"colab": {

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

}

},

"execution\_count": 15,

"outputs": [

{

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

"data": {

"text/plain": [

"DatetimeIndex(['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04',\n",

" '2023-01-05', '2023-01-06', '2023-01-07', '2023-01-08',\n",

" '2023-01-09', '2023-01-10', '2023-01-11', '2023-01-12',\n",

" '2023-01-13', '2023-01-14', '2023-01-15', '2023-01-16',\n",

" '2023-01-17', '2023-01-18', '2023-01-19', '2023-01-20',\n",

" '2023-01-21', '2023-01-22', '2023-01-23', '2023-01-24',\n",

" '2023-01-25', '2023-01-26', '2023-01-27', '2023-01-28',\n",

" '2023-01-29', '2023-01-30', '2023-01-31', '2023-02-01',\n",

" '2023-02-02', '2023-02-03', '2023-02-04', '2023-02-05',\n",

" '2023-02-06', '2023-02-07', '2023-02-08', '2023-02-09',\n",

" '2023-02-10'],\n",

" dtype='datetime64[ns]', freq='D')"

]

},

"metadata": {},

"execution\_count": 15

}

]

},

{

"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": 16,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"pd.DataFrame(lists)"

],

"metadata": {

"id": "knH76sDKYsVX",

"outputId": "70f0be9e-d9ac-4c51-b184-cd4bae96b46a",

"colab": {

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

"height": 143

}

},

"execution\_count": 17,

"outputs": [

{

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

"data": {

"text/plain": [

" 0 1 2\n",

"0 1 aaa 22\n",

"1 2 bbb 25\n",

"2 3 ccc 24"

],

"text/html": [

"\n",

" <div id=\"df-e05eb2e8-e9b9-451a-895c-41aeddfc1d8b\">\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-e05eb2e8-e9b9-451a-895c-41aeddfc1d8b')\"\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-e05eb2e8-e9b9-451a-895c-41aeddfc1d8b 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-e05eb2e8-e9b9-451a-895c-41aeddfc1d8b');\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

}

]

}

]

}