{

"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": 5,

"metadata": {

"id": "s07c7JK7Oqt-"

},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"metadata": {

"id": "6mGVa3SQYLkb"

},

"outputs": [

{

"data": {

"text/plain": [

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

]

},

"execution\_count": 7,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"s1=s.split()\n",

"s1"

]

},

{

"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": 8,

"metadata": {

"id": "\_ZHoml3kPqic"

},

"outputs": [],

"source": [

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

"diameter = 12742"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"metadata": {

"id": "HyRyJv6CYPb4"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

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

]

}

],

"source": [

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

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "KE74ZEwkRExZ"

},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 10,

"metadata": {

"id": "fcVwbCc1QrQI"

},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 14,

"metadata": {

"id": "MvbkMZpXYRaw"

},

"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": 15,

"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": 17,

"metadata": {

"id": "NHrirmgCYXvU"

},

"outputs": [

{

"data": {

"text/plain": [

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

]

},

"execution\_count": 17,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

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

"s"

]

},

{

"cell\_type": "code",

"execution\_count": 18,

"metadata": {

"id": "e4005lsTYXxx"

},

"outputs": [

{

"data": {

"text/plain": [

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

]

},

"execution\_count": 18,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"s1=np.array([5]\*10)\n",

"s1"

]

},

{

"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": 19,

"metadata": {

"id": "oAI2tbU2Yag-"

},

"outputs": [

{

"data": {

"text/plain": [

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

]

},

"execution\_count": 19,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"s2=np.arange(20,36,2)\n",

"s2"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "NaOM308NsRpZ"

},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 22,

"metadata": {

"id": "tOlEVH7BYceE"

},

"outputs": [

{

"data": {

"text/plain": [

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

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

" [6, 7, 8]])"

]

},

"execution\_count": 22,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"s3=np.arange(0,9).reshape(3,3)\n",

"s3"

]

},

{

"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": 24,

"metadata": {

"id": "rAPSw97aYfE0"

},

"outputs": [

{

"data": {

"text/plain": [

"array([1, 2, 3, 4, 5, 6])"

]

},

"execution\_count": 24,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

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

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

"s4=np.concatenate((a,b))\n",

"s4"

]

},

{

"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": 25,

"metadata": {

"id": "T5OxJRZ8uvR7"

},

"outputs": [],

"source": [

"import pandas as pd\n"

]

},

{

"cell\_type": "code",

"execution\_count": 30,

"metadata": {

"id": "xNpI\_XXoYhs0"

},

"outputs": [

{

"data": {

"text/html": [

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

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

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

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

" <td>2021</td>\n",

" </tr>\n",

" <tr>\n",

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

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

" <td>2016</td>\n",

" </tr>\n",

" <tr>\n",

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

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

" <td>2016</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Name Year\n",

"0 Happiness 2021\n",

"1 W 2016\n",

"2 Goblin 2016"

]

},

"execution\_count": 30,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"s5=pd.DataFrame({'Name':['Happiness','W','Goblin'],'Year':[2021,2016,2016]})\n",

"s5"

]

},

{

"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": 31,

"metadata": {

"id": "dgyC0JhVYl4F"

},

"outputs": [

{

"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',\n",

" ...\n",

" '2023-09-23', '2023-09-24', '2023-09-25', '2023-09-26',\n",

" '2023-09-27', '2023-09-28', '2023-09-29', '2023-09-30',\n",

" '2023-10-01', '2023-10-02'],\n",

" dtype='datetime64[ns]', length=275, freq='D')"

]

},

"execution\_count": 31,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"s6=pd.date\_range(start='1-1-2023',end='10-2-2023',freq='d')\n",

"s6"

]

},

{

"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": 32,

"metadata": {

"id": "\_XMC8aEt0llB"

},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": 34,

"metadata": {

"id": "knH76sDKYsVX"

},

"outputs": [

{

"data": {

"text/html": [

"<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>"

],

"text/plain": [

" 0 1 2\n",

"0 1 aaa 22\n",

"1 2 bbb 25\n",

"2 3 ccc 24"

]

},

"execution\_count": 34,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"s7=pd.DataFrame.from\_records(lists)\n",

"s7"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

}

],

"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"

}

},

"nbformat": 4,

"nbformat\_minor": 1

}