{

"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": null,

"metadata": {

"id": "s07c7JK7Oqt-"

},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "6mGVa3SQYLkb",

"outputId": "ef398f1a-a4ea-4086-a518-86b5e151ee89"

},

"outputs": [

{

"data": {

"text/plain": [

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

]

},

"execution\_count": 3,

"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": null,

"metadata": {

"id": "\_ZHoml3kPqic"

},

"outputs": [],

"source": [

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

"diameter = 12742"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "HyRyJv6CYPb4",

"outputId": "243c3104-7891-4976-9bd5-ecfef438e469"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"The diameter of Earth is 12742 kilometers\n"

]

}

],

"source": [

"print(\"The diameter of %s is %d kilometers\"%(planet,diameter))"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "KE74ZEwkRExZ"

},

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "fcVwbCc1QrQI"

},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "MvbkMZpXYRaw",

"outputId": "d9578bf4-393c-483a-c050-55f8409b64c3"

},

"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": null,

"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": null,

"metadata": {

"id": "NHrirmgCYXvU",

"outputId": "419d4a2e-fc2b-4eea-cd31-f121abea05f0"

},

"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": null,

"metadata": {

"id": "e4005lsTYXxx",

"outputId": "1a8dc36e-1f0d-4b4f-9c9d-af3613e661f3"

},

"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": null,

"metadata": {

"id": "oAI2tbU2Yag-",

"outputId": "24cfb916-7981-4f7e-bd4b-2d0e58b995ec"

},

"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": null,

"metadata": {

"id": "tOlEVH7BYceE",

"outputId": "800f8c98-5e87-4bee-d00e-db55bebaa3c2"

},

"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": null,

"metadata": {

"id": "rAPSw97aYfE0",

"outputId": "31d2f788-f737-4b1a-eeff-e6fb718448f3"

},

"outputs": [

{

"data": {

"text/plain": [

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

]

},

"execution\_count": 30,

"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": null,

"metadata": {

"id": "T5OxJRZ8uvR7"

},

"outputs": [],

"source": [

"import pandas as pd\n"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "xNpI\_XXoYhs0",

"outputId": "0bb173b7-e4ee-4539-8865-c0dcc9a0d7dc"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

" Name Age\n",

"0 george 10\n",

"1 jack 15\n",

"2 jhon 14\n"

]

}

],

"source": [

"data = [['george', 10], ['jack', 15], ['jhon', 14]]\n",

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

"print(df)"

]

},

{

"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": null,

"metadata": {

"id": "1IJO4Kz5AXKG"

},

"outputs": [],

"source": [

"import datetime"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "dgyC0JhVYl4F",

"outputId": "19ac1c77-9ae5-40de-8741-e612d56a8829"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"01-01-2023\n",

"02-01-2023\n",

"03-01-2023\n",

"04-01-2023\n",

"05-01-2023\n",

"06-01-2023\n",

"07-01-2023\n",

"08-01-2023\n",

"09-01-2023\n",

"10-01-2023\n",

"11-01-2023\n",

"12-01-2023\n",

"13-01-2023\n",

"14-01-2023\n",

"15-01-2023\n",

"16-01-2023\n",

"17-01-2023\n",

"18-01-2023\n",

"19-01-2023\n",

"20-01-2023\n",

"21-01-2023\n",

"22-01-2023\n",

"23-01-2023\n",

"24-01-2023\n",

"25-01-2023\n",

"26-01-2023\n",

"27-01-2023\n",

"28-01-2023\n",

"29-01-2023\n",

"30-01-2023\n",

"31-01-2023\n",

"01-02-2023\n",

"02-02-2023\n",

"03-02-2023\n",

"04-02-2023\n",

"05-02-2023\n",

"06-02-2023\n",

"07-02-2023\n",

"08-02-2023\n",

"09-02-2023\n"

]

}

],

"source": [

"\n",

"start = datetime.datetime.strptime(\"01-01-2023\", \"%d-%m-%Y\")\n",

"end = datetime.datetime.strptime(\"10-02-2023\", \"%d-%m-%Y\")\n",

"date\_generated = [start + datetime.timedelta(days=x) for x in range(0, (end-start).days)]\n",

"\n",

"for date in date\_generated:\n",

" print (date.strftime(\"%d-%m-%Y\"))\n"

]

},

{

"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": null,

"metadata": {

"id": "\_XMC8aEt0llB"

},

"outputs": [],

"source": [

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

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "knH76sDKYsVX"

},

"outputs": [],

"source": [

"df = pd.DataFrame(lists, columns =['id', 'tag','number']) "

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "AREYrSliAXbP",

"outputId": "a62d6ad8-7d38-473d-875a-12af3d3aee42"

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

" id tag number\n",

"0 1 aaa 22\n",

"1 2 bbb 25\n",

"2 3 ccc 24\n"

]

}

],

"source": [

"print(df)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "X3KYvfHTAXbQ"

},

"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": 0

}