{

"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": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

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

"print(x)"

],

"metadata": {

"id": "6mGVa3SQYLkb",

"colab": {

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

},

"outputId": "e9a7068f-0cc8-49d5-ac4f-52739f1c4ce8"

},

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

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

]

}

]

},

{

"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": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"print(\"The diameter of {planet} is {diameter} kilometers\".format(planet=\"Earth\",diameter=12742))"

],

"metadata": {

"id": "HyRyJv6CYPb4",

"colab": {

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

},

"outputId": "38b06c75-670f-4bbd-ccfd-784baf8e2843"

},

"execution\_count": null,

"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": 2,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"\n",

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

],

"metadata": {

"id": "MvbkMZpXYRaw",

"colab": {

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

},

"outputId": "bd26aea8-0ad3-4962-f584-a5636777777e"

},

"execution\_count": 3,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"hello\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [],

"metadata": {

"id": "MA3WC\_D7WIlL"

}

},

{

"cell\_type": "markdown",

"source": [

"# Numpy"

],

"metadata": {

"id": "bw0vVp-9ddjv"

}

},

{

"cell\_type": "code",

"source": [

"import numpy as np"

],

"metadata": {

"id": "LLiE\_TYrhA1O"

},

"execution\_count": 5,

"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": [

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

"print(a)"

],

"metadata": {

"id": "NHrirmgCYXvU",

"colab": {

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

},

"outputId": "46da1285-91a5-4bcd-980c-db05c2003ba5"

},

"execution\_count": 6,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

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

]

}

]

},

{

"cell\_type": "code",

"source": [

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

"print(b)"

],

"metadata": {

"id": "e4005lsTYXxx"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "markdown",

"source": [

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

],

"metadata": {

"id": "gZHHDUBvrMX4"

}

},

{

"cell\_type": "code",

"source": [

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

"print(c)"

],

"metadata": {

"id": "oAI2tbU2Yag-",

"colab": {

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

},

"outputId": "54772cbe-c3b5-4597-f86a-f41e30449a9f"

},

"execution\_count": 7,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

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

]

}

]

},

{

"cell\_type": "markdown",

"source": [

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

],

"metadata": {

"id": "NaOM308NsRpZ"

}

},

{

"cell\_type": "code",

"source": [

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

"print(d)"

],

"metadata": {

"id": "tOlEVH7BYceE",

"colab": {

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

},

"outputId": "64e64ed1-4dc4-4ddc-f6d6-97f1ef30eddd"

},

"execution\_count": 8,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"[[0 1 2]\n",

" [3 4 5]\n",

" [6 7 8]]\n"

]

}

]

},

{

"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",

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

"print(e)"

],

"metadata": {

"id": "rAPSw97aYfE0",

"colab": {

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

},

"outputId": "a3c8db6d-13cf-4f9f-e6ff-88214b52bb92"

},

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

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

]

}

]

},

{

"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": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"data=[['ram',21],['nick',22],['dom',26]]\n",

"f=pd.DataFrame(data,columns=['name','age'])\n",

"print(f)"

],

"metadata": {

"id": "xNpI\_XXoYhs0",

"colab": {

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

},

"outputId": "595806ad-361a-4330-faf1-6de7e083c85f"

},

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

" name age\n",

"0 ram 21\n",

"1 nick 22\n",

"2 dom 26\n"

]

}

]

},

{

"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": [

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

"for val in dates:\n",

" print(val)"

],

"metadata": {

"id": "dgyC0JhVYl4F",

"colab": {

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

},

"outputId": "ac615589-c53e-4eb9-816b-b757f47b1d70"

},

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"2023-01-01 00:00:00\n",

"2023-01-02 00:00:00\n",

"2023-01-03 00:00:00\n",

"2023-01-04 00:00:00\n",

"2023-01-05 00:00:00\n",

"2023-01-06 00:00:00\n",

"2023-01-07 00:00:00\n",

"2023-01-08 00:00:00\n",

"2023-01-09 00:00:00\n",

"2023-01-10 00:00:00\n",

"2023-01-11 00:00:00\n",

"2023-01-12 00:00:00\n",

"2023-01-13 00:00:00\n",

"2023-01-14 00:00:00\n",

"2023-01-15 00:00:00\n",

"2023-01-16 00:00:00\n",

"2023-01-17 00:00:00\n",

"2023-01-18 00:00:00\n",

"2023-01-19 00:00:00\n",

"2023-01-20 00:00:00\n",

"2023-01-21 00:00:00\n",

"2023-01-22 00:00:00\n",

"2023-01-23 00:00:00\n",

"2023-01-24 00:00:00\n",

"2023-01-25 00:00:00\n",

"2023-01-26 00:00:00\n",

"2023-01-27 00:00:00\n",

"2023-01-28 00:00:00\n",

"2023-01-29 00:00:00\n",

"2023-01-30 00:00:00\n",

"2023-01-31 00:00:00\n",

"2023-02-01 00:00:00\n",

"2023-02-02 00:00:00\n",

"2023-02-03 00:00:00\n",

"2023-02-04 00:00:00\n",

"2023-02-05 00:00:00\n",

"2023-02-06 00:00:00\n",

"2023-02-07 00:00:00\n",

"2023-02-08 00:00:00\n",

"2023-02-09 00:00:00\n",

"2023-02-10 00:00:00\n"

]

}

]

},

{

"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": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"dframe=pd.DataFrame(lists,columns=['rank','name','age'])\n",

"print(dframe)"

],

"metadata": {

"id": "knH76sDKYsVX",

"colab": {

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

},

"outputId": "65231e54-9e5a-4142-8d02-7910cc2ee91f"

},

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

" rank name age\n",

"0 1 aaa 22\n",

"1 2 bbb 25\n",

"2 3 ccc 24\n"

]

}

]

}

]

}