{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"provenance": [],

"collapsed\_sections": [],

"toc\_visible": true

},

"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!\"\n",

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

"print(p)"

],

"metadata": {

"id": "s07c7JK7Oqt-",

"colab": {

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

},

"outputId": "e5602e76-674e-42bb-96ca-0fae2434141b"

},

"execution\_count": 1,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

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

]

}

]

},

{

"cell\_type": "code",

"source": [],

"metadata": {

"id": "6mGVa3SQYLkb"

},

"execution\_count": null,

"outputs": []

},

{

"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\n",

"p=\"The diameter of {planet} is {diameter} kilometers.\"\n",

"print(p.format(planet = \"Earth\",diameter = 12742))"

],

"metadata": {

"id": "\_ZHoml3kPqic",

"colab": {

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

},

"outputId": "bfc0c2a0-5ba0-4316-eb3a-8c213a139d92"

},

"execution\_count": 4,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

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

]

}

]

},

{

"cell\_type": "code",

"source": [],

"metadata": {

"id": "HyRyJv6CYPb4"

},

"execution\_count": null,

"outputs": []

},

{

"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']}]}]}\n",

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

],

"metadata": {

"id": "fcVwbCc1QrQI",

"colab": {

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

},

"outputId": "6c6ba238-7ab8-47f5-f675-31192a46daca"

},

"execution\_count": 21,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"hello\n"

]

}

]

},

{

"cell\_type": "code",

"source": [],

"metadata": {

"id": "MvbkMZpXYRaw"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "markdown",

"source": [

"# Numpy"

],

"metadata": {

"id": "bw0vVp-9ddjv"

}

},

{

"cell\_type": "code",

"source": [

"import numpy as np"

],

"metadata": {

"id": "LLiE\_TYrhA1O"

},

"execution\_count": 22,

"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": [

"import numpy as np\n",

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

"print('An array of ten zeros')\n",

"print(arr)"

],

"metadata": {

"id": "NHrirmgCYXvU",

"colab": {

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

},

"outputId": "ef027ae9-6867-4fbf-fac7-12e0ed45d5d3"

},

"execution\_count": 23,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"An array of ten zeros\n",

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

]

}

]

},

{

"cell\_type": "code",

"source": [

"import numpy as np\n",

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

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

"print(arr)"

],

"metadata": {

"id": "e4005lsTYXxx",

"colab": {

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

},

"outputId": "dda271f9-f797-4e4d-b3d9-48a319eebafb"

},

"execution\_count": 24,

"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": [

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

],

"metadata": {

"id": "gZHHDUBvrMX4"

}

},

{

"cell\_type": "code",

"source": [

"import numpy as np\n",

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

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

"print(arr)"

],

"metadata": {

"id": "oAI2tbU2Yag-",

"colab": {

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

},

"outputId": "5e60227b-89ac-479b-9bd4-eaa8cdfdb4d1"

},

"execution\_count": 25,

"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": [

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

],

"metadata": {

"id": "NaOM308NsRpZ"

}

},

{

"cell\_type": "code",

"source": [

"import numpy as np\n",

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

"print(p)"

],

"metadata": {

"id": "tOlEVH7BYceE",

"colab": {

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

},

"outputId": "fa5cbb87-5b65-4668-ff6e-9e6781fe51ad"

},

"execution\_count": 26,

"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": [

"import numpy as np\n",

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

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

"p = np.concatenate((a,b),axis=0)\n",

"print(p)"

],

"metadata": {

"id": "rAPSw97aYfE0",

"colab": {

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

},

"outputId": "4dfe14ce-2483-49a6-8c07-f72161bb5d4f"

},

"execution\_count": 29,

"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": [

"import pandas as pd\n",

"d = {'column1':[1,2,3],'column2':[1,2,3]}\n",

"p = pd.DataFrame(d)\n",

"print(p)"

],

"metadata": {

"id": "xNpI\_XXoYhs0",

"colab": {

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

},

"outputId": "702572b0-28cb-4156-bf68-45805cb47672"

},

"execution\_count": 35,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

" column1 column2\n",

"0 1 1\n",

"1 2 2\n",

"2 3 3\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": [

"import pandas as pd\n",

"pd.date\_range('01-01-2023', '02-10-2023', freq='d')\n"

],

"metadata": {

"id": "dgyC0JhVYl4F",

"colab": {

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

},

"outputId": "2cf73939-0edb-49df-86b9-a0c8e2d8f1d9"

},

"execution\_count": 39,

"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": 39

}

]

},

{

"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": [

"import pandas as pd\n",

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

"df = pd.DataFrame(lists, columns = ['c1','c2','c3'])\n",

"print(df)"

],

"metadata": {

"id": "\_XMC8aEt0llB",

"colab": {

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

},

"outputId": "68d5454a-159b-4d7b-80d1-4ec4a6f61a72"

},

"execution\_count": 45,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

" c1 c2 c3\n",

"0 1 aaa 22\n",

"1 2 bbb 25\n",

"2 3 ccc 24\n"

]

}

]

},

{

"cell\_type": "code",

"source": [],

"metadata": {

"id": "knH76sDKYsVX"

},

"execution\_count": null,

"outputs": []

}

]

}