Assignment-1

Date	10-09-2022
Team ID	PNT2022TMID36138
Project Name	Project Real Time Communication System Powered by AI for Specially Abeled

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": [
"s=\"Hi there sam\"\n",
"s=s.split()\n",
"print(s)"
],
"metadata": {
"id": "6mGVa3SQYLkb",
"colab": {
"base_uri": "https://localhost:8080/"
},

"outputId": "c178bff6-ea36-4545-b4de-f4032a691387"
},
"execution_count": 2,
"outputs": [
{
"output_type": "stream",
"name": "stdout",
"text": [
"['Hi', 'there', 'sam']\n"
]
}
]
},
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": [
"planet=\"Earth\"\n",
"diameter=12724\n",
"print(\"the diameter of\",planet,\" is \" , diameter,\"kilometers\")"
],
"metadata": {
"id": "HyRyJv6CYPb4",
"colab": {
"base_uri": "https://localhost:8080/"
},
"outputId": "02d2f5ad-b855-49d9-e0f8-aff8187f758b"

},
"execution_count": 6,
"outputs": [
{
"output_type": "stream",
"name": "stdout",
"text": [
"the diameter of Earth is 12724 kilometers\n"
]
}
]
},
{
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": null,
"outputs": []
},
{
"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])"
],
"metadata": {
"id": "MvbkMZpXYRaw",
"colab": {
"base_uri": "https://localhost:8080/"
},
"outputId": "11ee257d-7371-43c6-efa0-a8d44916dde6"
},
"execution_count": 7,
"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": 9,
"outputs": []
},
{
4.1 Create an array of 10 zeros?
4.2 Create an array of 10 fives?"
],
"metadata": {
"id": "wOg8hinbgx30"
}
},
{
"cell_type": "code",
"source": [
"np.zeros(10)"
],
"metadata": {
"id": "NHrirmgCYXvU",
"colab": {
"base_uri": "https://localhost:8080/"
},

"outputId": "a429ef8e-dfd7-453a-a720-6f98081fa2ac"
},
"execution_count": 10,
"outputs": [
{
"output_type": "execute_result",
"data": {
"text/plain": [
"array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])"
]
},
"metadata": {},
"execution_count": 10
}
]
},
{
"cell_type": "code",
"source": [
"np.ones(10)*5"
],
"metadata": {
"id": "e4005lsTYXxx",
"colab": {
"base_uri": "https://localhost:8080/"
},
"outputId": "4a8f7ecf-baeb-4f94-def9-3dc830f85181"
},
"execution_count": 11,
"outputs": [
{
"output_type": "execute_result",
"data": {
"text/plain": [
"array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])"
]
},
"metadata": {},
"execution_count": 11
}
]
},
{
"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)\n",
"print(np.arange(20,35,2))"
],
"metadata": {
"id": "oAI2tbU2Yag-",
"colab": {
"base_uri": "https://localhost:8080/"
},
"outputId": "2ebe1d63-0311-4f22-ec66-ae28745529f4"
},
"execution_count": 15,
"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": [
"np.arange(0,9).reshape((3,3))"
],
"metadata": {
"id": "tOlEVH7BYceE",
"colab": {

"base_uri": "https://localhost:8080/"
},
"outputId": "9bb132f9-6317-4aec-8318-553c0112393d"
},
"execution_count": 16,
"outputs": [
{
"output_type": "execute_result",
"data": {
"text/plain": [
"array([[0, 1, 2],\n",
" [3, 4, 5],\n",
" [6, 7, 8]])"
]
},
"metadata": {},
"execution_count": 16
}
]
},
{
"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",
"a=np.concatenate((a,b))\n",
"print(a)"
],
"metadata": {
"id": "rAPSw97aYfE0",
"colab": {
"base_uri": "https://localhost:8080/"
},
"outputId": "ad06f382-d177-4b81-ea00-7e9915b8c296"
},
"execution_count": 17,

"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": 18,
"outputs": []
},
{
"cell_type": "code",
"source": [
"data=[{'a':1,'b':2,'c':3},{'a':10,'b':20,'c':30}]\n",
"df=pd.DataFrame(data)\n",
"df"
],

```
"metadata": {
"id": "xNpI_XXoYhs0",
"colab": {
"base_uri": "https://localhost:8080/",
"height": 112
},
"outputId": "8e4e89a0-b513-462a-e1ac-37c6ef6d0873"
},
"execution_count": 19,
"outputs": [
{
"output_type": "execute_result",
"data": {
"text/plain": [
" a b c\n",
"0 1 2 3\n",
"1 10 20 30"
],
"text/html": [
"\n",
" <div id=\"df-72220aea-f1ed-4645-a1ae-73f2550741a1\">\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>a</th>\n",
" <th>b</th>\n",
" <th>c</th>\n",
" </tr>\n",
" </thead>\n",
" <tbody>\n",
```

" <tr>\n",
" <th>0</th>\n",
" <td>1</td>\n",
" <td>2</td>\n",
" <td>3</td>\n",
" </tr>\n",
" <tr>\n",
" <th>1</th>\n",
" <td>10</td>\n",
" <td>20</td>\n",
" <td>30</td>\n",
" </tr>\n",
" </tbody>\n",
"</table>\n",
" \n",
" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-72220aea-f1ed-4645-a1ae-73f2550741a1')\"\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: 12p…
9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023"
],
"metadata": {
"id": "UXSmdNclyJQD"
}
},
{
"cell_type": "code",
"source": [

"per1=pd.date_range(start='2023-1-1',end='2023-2-10')\n",
"for val in per1:\n",
" print(val)"
],
"metadata": {
"id": "dgyC0JhVYl4F",
"colab": {
"base_uri": "https://localhost:8080/"
},
"outputId": "3bfdadc2-c917-49e5-9f70-b7b289ec2130"
},
"execution_count": 27,
"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": 28,
"outputs": []
},
{
"cell_type": "code",
"source": [
"df=pd.DataFrame(lists,columns=['roll no','name','number'])\n",
"print(df)"
],

```
"metadata": {
"id": "knH76sDKYsVX",
"colab": {
"base_uri": "https://localhost:8080/"
},
"outputId": "b458e584-461c-4fd0-95d1-07866e59cf55"
},
"execution_count": 30,
"outputs": [
{
"output_type": "stream",
"name": "stdout",
"text": [
" roll no name number\n",
"0 1 aaa 22\n",
"1 2 bbb 25\n",
"2 3 ccc 24\n"
]
}
]
}
]
}
```