{

"cells": [

{

"cell\_type": "code",

"execution\_count": 2,

"id": "62f52ef6",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

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

]

}

],

"source": [

"## 1. Split this string\n",

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

"print(s.split())"

]

},

{

"cell\_type": "code",

"execution\_count": 102,

"id": "c0a30c44",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

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

]

}

],

"source": [

"## 2. Use .format() to print the following string. \n",

"\n",

"### Output should be: The diameter of Earth is 12742 kilometers.\n",

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

"diameter = 12742\n",

"print(\"The diameter of {0} is {1} kilometers.\".format(planet,diameter))"

]

},

{

"cell\_type": "code",

"execution\_count": 76,

"id": "e4fbb0d6",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"hello\n"

]

}

],

"source": [

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

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

"a = d['k1']\n",

"b = a[3]\n",

"c = b['tricky']\n",

"d = c[3]\n",

"e = d['target'][3]\n",

"print(e)"

]

},

{

"cell\_type": "code",

"execution\_count": 81,

"id": "2ebe863a",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

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

"[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]\n"

]

}

],

"source": [

"import numpy as np\n",

"##4.1 Create an array of 10 zeros? \n",

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

"\n",

"print(arr)\n",

"## 4.2 Create an array of 10 fives?\n",

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

"print(arr)"

]

},

{

"cell\_type": "code",

"execution\_count": 38,

"id": "e944ee25",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

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

]

}

],

"source": [

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

"arr =np.array([i for i in range(20,36) if(i%2 ==0) ])\n",

"print(np.array(arr))"

]

},

{

"cell\_type": "code",

"execution\_count": 61,

"id": "ba97e5cb",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"[[0 1 2]\n",

" [3 4 5]\n",

" [6 7 8]]\n"

]

}

],

"source": [

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

"arr = np.array([range(0,9)])\n",

"print(arr.reshape(3,3))\n"

]

},

{

"cell\_type": "code",

"execution\_count": 84,

"id": "623b7075",

"metadata": {

"scrolled": true

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

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

]

}

],

"source": [

"## 7. Concatenate a and b \n",

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

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

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

"print(c)"

]

},

{

"cell\_type": "code",

"execution\_count": 96,

"id": "e563d81a",

"metadata": {

"scrolled": true

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

" a b\n",

"0 1 2\n",

"1 4 5\n",

"2 6 7\n"

]

}

],

"source": [

"## 8. Create a dataframe with 3 rows and 2 columns\n",

"import pandas as pd\n",

"data = [[1,2],[4,5],[6,7]]\n",

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

"print(df)"

]

},

{

"cell\_type": "code",

"execution\_count": 101,

"id": "94e0174c",

"metadata": {},

"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', '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')"

]

},

"execution\_count": 101,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"## 9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023\n",

"min\_date = \"2023-01-01\"\n",

"max\_date =\"2023-02-10\"\n",

"dt = pd.date\_range(min\_date,max\_date)\n",

"dt"

]

},

{

"cell\_type": "code",

"execution\_count": 39,

"id": "8e43fee7",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

" 0 1 2\n",

"0 1 aaa 22\n",

"1 2 bbb 25\n",

"2 3 ccc 24\n"

]

}

],

"source": [

"## 10. Create 2D list to DataFrame\n",

"import pandas as pd\n",

"lists = [[1, 'aaa', 22],\n",

" [2, 'bbb', 25],\n",

" [3, 'ccc', 24]]\n",

"df = pd.DataFrame(lists)\n",

"print(df)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"id": "ccb3e4a9",

"metadata": {},

"outputs": [],

"source": []

}

],

"metadata": {

"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": 5

}