```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 3,
   "id": "7d66524b",
   "metadata": {},
   "outputs": [],
   "source": [
    "#1. Split this string"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 4,
   "id": "4a6e78eb",
   "metadata": {},
   "outputs": [],
   "source": [
    "s = 'Hi there Sam!'"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 11,
   "id": "d9ea9411",
   "metadata": {},
   "outputs": [],
   "source": [
    "#2. Use .format() to print the following string.\n",
    "#Output should be: The diameter of Earth is 12742 kilometers."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 12,
   "id": "9e635ce1",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "['Hi', 'there', 'Sam!']"
      ]
     },
     "execution_count": 12,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "s.split()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 13,
```

```
  "id": "ef082dd3",
  "metadata": {},
  "outputs": [],
  "source": [
   "planet = \"Earth\"\n",
   "diameter = 12742"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 14,
  "id": "7900e2db",
  "metadata": {},
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "The diameter of Earth is 12742 kilometers.\n"
    ]
   }
  ],
  "source": [
   "print(\"The diameter of {} is {} kilometers.\".format(planet,diameter))"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 15,
  "id": "bd4a81a2",
  "metadata": {},
  "outputs": [],
  "source": [
   "#3. In this nest dictionary grab the word \"hello\""
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 16,
  "id": "50b520e9",
  "metadata": {},
  "outputs": [],
  "source": [
   "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}"
  ]
 },
 {
 "cell_type": "code",
 "execution_count": 17,
 "id": "3897081a",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "'hello'"
    ]
```

```
    },
    "execution_count": 17,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "d['k1'][3]['tricky'][3]['target'][3]"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 18,
  "id": "3d1e33b8",
  "metadata": {},
  "outputs": [],
  "source": [
   "#4.1 Create an array of 10 zeros?"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 22,
  "id": "04aa49d9",
  "metadata": {},
  "outputs": [],
  "source": [
   "import numpy as np"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 24,
  "id": "418dac1d",
  "metadata": {},
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "An array of 10 zeros:\n",
     "[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]\n"
    ]
   }
  ],
  "source": [
   "array=np.zeros(10)\n",
   "print(\"An array of 10 zeros:\")\n",
   "print(array)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 25,
  "id": "c64cc256",
  "metadata": {},
  "outputs": [],
```

```
  "source": [
   "#4.2 Create an array of 10 fives?"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 26,
  "id": "15b41fec",
  "metadata": {},
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "An array of 10 fives:\n",
     "[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]\n"
    ]
   }
  ],
  "source": [
   "array=np.ones(10)*5\n",
   "print(\"An array of 10 fives:\")\n",
   "print(array)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 27,
  "id": "4d65c4f1",
  "metadata": {},
  "outputs": [],
  "source": [
   "#5. Create an array of all the even integers from 20 to 35"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 28,
  "id": "5d5fefb8",
  "metadata": {},
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "Array of all the even integers from 30 to 70\n",
     "[20 22 24 26 28 30 32 34]\n"
    ]
   }
  ],
  "source": [
   "import numpy as np\n",
   "array=np.arange(20,35,2)\n",
   "print(\"Array of all the even integers from 30 to 70\")\n",
   "print(array) "
  ]
 },
```

```
{
 "cell_type": "code",
 "execution_count": 29,
 "id": "83d40b3e",
 "metadata": {},
 "outputs": [],
 "source": [
  "#6. Create a 3x3 matrix with values ranging from 0 to 8"
 ]
},
{
"cell_type": "code",
"execution_count": 30,
"id": "8117754d",
"metadata": {},
"outputs": [
 {
  "name": "stdout",
  "output_type": "stream",
  "text": [
   "[[0 1 2]\n",
   " [3 4 5]\n",
   " [6 7 8]]\n"
  ]
 }
],
"source": [
 "import numpy as np\n",
 "x =    np.arange(0, 9).reshape(3,3)\n",
  "print(x)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 45,
 "id": "bba26f01",
 "metadata": {},
 "outputs": [],
 "source": [
  "#7. Concatinate a and b"
 ]
},
{
 "cell_type": "code",
 "execution_count": 46,
 "id": "04da763d",
 "metadata": {},
"outputs": [
 {
  "name": "stdout",
  "output_type": "stream",
  "text": [
   "[1 2 3 4 5 6]\n"
  ]
 }
],
"source": [
```

```
    "import numpy as np\n",
    "    \n",
    "a = np.array([1, 2, 3])\n",
    "b = np.array([4, 5, 6])\n",
    "    \n",
    "new = np.concatenate((a, b), axis = 0)\n",
    "    \n",
    "print (new)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 47,
   "id": "cb3fa532",
   "metadata": {},
   "outputs": [],
   "source": [
    "#6. Create a dataframe with 3 rows and 2 columns"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 48,
   "id": "07cafb6d",
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Empty DataFrame\n",
      "Columns: []\n",
      "Index: []\n"
     ]
    }
   ],
   "source": [
    "import pandas as pd\n",
    "df = pd.DataFrame()\n",
    "print(df)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 49,
   "id": "fc086630",
   "metadata": {},
   "outputs": [],
   "source": [
    "#9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 50,
   "id": "b22e2442",
   "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": 50,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "pd.date_range(start=\"2023-01-01\",end=\"2023-02-10\")"
]
},
{
"cell_type": "code",
"execution_count": 53,
"id": "e5e570e4",
"metadata": {},
"outputs": [],
"source": [
 "#10. Create 2D list to DataFrame\n",
 "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]"
]
},
{
"cell_type": "code",
"execution_count": 62,
"id": "84aa9ad6",
"metadata": {},
"outputs": [
 {
  "name": "stdout",
  "output_type": "stream",
  "text": [
   "    S.No. Name  Age\n",
   "0      1  aaa   22\n",
   "1      2  bbb   25\n",
   "2      3  ccc   24\n"
 ]
},
{
 "name": "stderr",
 "output_type": "stream",
```

```
        "text": [
         "C:\\Users\\91948\\AppData\\Local\\Temp\\ipykernel_20204\\1580403774.py:3:
FutureWarning: Could not cast to int32, falling back to object. This behavior is deprecated. In
a future version, when a dtype is passed to 'DataFrame', either all columns will be cast to that
dtype, or a TypeError will be raised.\n",
          "    df = pd.DataFrame(lists, columns =['S.No.', 'Name', 'Age'], dtype = int)\n"
        ]
       }
      ],
      "source": [
       "import pandas as pd\n",
       "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]\n",
       "df = pd.DataFrame(lists, columns =['S.No.', 'Name', 'Age'], dtype = int)\n",
       "print(df)"
      ]
     },
     {
      "cell_type": "code",
      "execution_count": null,
      "id": "406398a0",
      "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
}
```