```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "McSxJAwcOdZ1"
   },
   "source": [
    "# Basic Python"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "CU48hgo4Owz5"
   },
   "source": [
    "## 1. Split this string"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 27,
   "metadata": {
    "id": "s07c7JK7Oqt-"
   },
   "outputs": [],
   "source": [
    "s = \"Hi there Sam!\""
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 28,
   "metadata": {
    "id": "6mGVa3SQYLkb"
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "['Hi', 'there', 'Sam!']\n"
     ]
    }
   ],
   "source": [
    "a=s.split();\n",
    "print(a)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
```

```
  "id": "GH1QBn8HP375"
 },
 "source": [
  "## 2. Use .format() to print the following string. \n",
  "\n",
  "### Output should be: The diameter of Earth is 12742 kilometers."
 ]
},
{
 "cell_type": "code",
 "execution_count": 29,
 "metadata": {
  "id": "_ZHoml3kPqic"
 },
 "outputs": [],
 "source": [
  "planet = \"Earth\"\n",
  "diameter = 12742"
 ]
},
{
 "cell_type": "code",
 "execution_count": 30,
 "metadata": {
  "id": "HyRyJv6CYPb4"
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "The diameter of earth is 12742 kilometers!\n"
   ]
  }
 ],
 "source": [
  "txt = \"The diameter of earth is {price:g} kilometers!\"\n",
  "print(txt.format(price = 12742))"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "KE74ZEwkRExZ"
 },
 "source": [
  "## 3. In this nest dictionary grab the word \"hello\""
 ]
},
{
 "cell_type": "code",
 "execution_count": 31,
 "metadata": {
  "id": "fcVwbCc1QrQI"
```

```
   },
   "outputs": [],
   "source": [
    "d =
{'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}
]}]}"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 32,
   "metadata": {
    "id": "MvbkMZpXYRaw"
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "hello\n"
     ]
    }
   ],
   "source": [
    "print (d['k1'][3]['tricky'][3]['target'][3])"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "bw0vVp-9ddjv"
   },
   "source": [
    "# Numpy"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 33,
   "metadata": {
    "id": "LLiE_TYrhA1O"
   },
   "outputs": [],
   "source": [
    "import numpy as np"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "wOg8hinbgx30"
   },
   "source": [
    "## 4.1 Create an array of 10 zeros? \n",
```

```
  "## 4.2 Create an array of 10 fives?"
 ]
},
{
 "cell_type": "code",
 "execution_count": 34,
 "metadata": {
  "id": "NHrirmgCYXvU"
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]\n"
   ]
  }
 ],
 "source": [
  "array=np.zeros(10)\n",
  "print(array)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 35,
 "metadata": {
  "id": "e4005lsTYXxx"
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]\n"
   ]
  }
 ],
 "source": [
  "array=np.ones(10)*5\n",
  "print(array)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "gZHHDUBvrMX4"
 },
 "source": [
  "## 5. Create an array of all the even integers from 20 to 35"
 ]
},
{
 "cell_type": "code",
```

```
 "execution_count": 36,
 "metadata": {
  "id": "oAI2tbU2Yag-"
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "[20 22 24 26 28 30 32 34]\n"
   ]
  }
 ],
 "source": [
  "array=np.arange(20,36,2)\n",
  "print(array)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "NaOM308NsRpZ"
 },
 "source": [
  "## 6. Create a 3x3 matrix with values ranging from 0 to 8"
 ]
},
{
 "cell_type": "code",
 "execution_count": 37,
 "metadata": {
  "id": "tOlEVH7BYceE"
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "[[0 1 2]\n",
    " [3 4 5]\n",
    " [6 7 8]]\n"
   ]
  }
 ],
 "source": [
  "array=np.arange(0,9).reshape(3,3)\n",
  "print(array)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "hQ0dnhAQuU_p"
 },
```

```
 "source": [
  "## 7. Concatinate a and b \n",
  "## a = np.array([1, 2, 3]), b = np.array([4, 5, 6])"
 ]
},
{
 "cell_type": "code",
 "execution_count": 38,
 "metadata": {
  "id": "rAPSw97aYfE0"
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "[1 2 3 4 5 6]\n"
   ]
  }
 ],
 "source": [
  "array1 = np.array([1, 2, 3])\n",
  "array2 = np.array([4, 5, 6])\n",
  "array = np.concatenate((array1,array2))\n",
  "print(array)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "dlPEY9DRwZga"
 },
 "source": [
  "# Pandas"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "ijoYW51zwr87"
 },
 "source": [
  "## 8. Create a dataframe with 3 rows and 2 columns"
 ]
},
{
 "cell_type": "code",
 "execution_count": 39,
 "metadata": {
  "id": "T5OxJRZ8uvR7"
 },
 "outputs": [],
 "source": [
  "import pandas as pd"
```

```
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 40,
   "metadata": {
    "id": "xNpI_XXoYhs0"
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "   0  1\n",
      "0  1  5\n",
      "1  4  0\n",
      "2  0  4\n"
     ]
    }
   ],
   "source": [
    "A = np.random.randint(6, size=(3,2))\n",
    "df = pd.DataFrame(A)\n",
    "print (df)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "UXSmdNclyJQD"
   },
   "source": [
    "## 9. Generate the series of dates from 1st Jan, 2023 to 10th Feb,
2023"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 41,
   "metadata": {
    "id": "dgyC0JhVYl4F"
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "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"
     ]
    }
   ],
   "source": [
    "period = pd.date_range(start ='1-01-2023',end ='2-10-2023',freq
='24H') \n",
    "for val in period: \n",
    "    print(val)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "ZizSetD-y5az"
   },
   "source": [
    "## 10. Create 2D list to DataFrame\n",
    "\n",
    "lists = [[1, 'aaa', 22],\n",
    "         [2, 'bbb', 25],\n",
```

```
   "         [3, 'ccc', 24]]"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 42,
  "metadata": {
   "id": "_XMC8aEt0llB"
  },
  "outputs": [],
  "source": [
   "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 43,
  "metadata": {
   "id": "knH76sDKYsVX"
  },
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "   s.no  Tag  number\n",
     "0     1  aaa      22\n",
     "1     2  bbb      25\n",
     "2     3  ccc      24\n"
    ]
   }
  ],
  "source": [
   "df = pd.DataFrame(lists, columns =['s.no','Tag', 'number'])  \n",
   "print(df )"
  ]
 }
],
"metadata": {
 "colab": {
  "collapsed_sections": [],
  "provenance": []
 },
 "kernelspec": {
  "display_name": "Python 3",
  "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": "3.8.8"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 1
}
```