```
{
 "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": null,
   "metadata": {
    "id": "s07c7JK7Oqt-"
   },
   "outputs": [],
   "source": [
    "s = \"Hi there Sam!\""
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "6mGVa3SQYLkb"
   },
   "outputs": [],
   "source": []
  },
  {
   "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": null,
   "metadata": {
    "id": "_ZHoml3kPqic"
   },
   "outputs": [],
   "source": [
    "planet = \"Earth\"\n",
    "diameter = 12742"
   ]
  },
  {
```

```
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "HyRyJv6CYPb4"
   },
   "outputs": [],
   "source": []
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "KE74ZEwkRExZ"
   },
   "source": [
    "## 3. In this nest dictionary grab the word \"hello\""
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "fcVwbCc1QrQI"
   },
   "outputs": [],
   "source": [
    "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "MvbkMZpXYRaw"
   },
   "outputs": [],
   "source": []
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "bw0vVp-9ddjv"
   },
   "source": [
    "# Numpy"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "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": null,
   "metadata": {
    "id": "NHrirmgCYXvU"
   },
   "outputs": [],
   "source": []
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "e4005lsTYXxx"
   },
   "outputs": [],
   "source": []
  },
  {
   "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": null,
   "metadata": {
    "id": "oAI2tbU2Yag-"
   },
   "outputs": [],
   "source": []
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "NaOM308NsRpZ"
   },
   "source": [
    "## 6. Create a 3x3 matrix with values ranging from 0 to 8"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "tOlEVH7BYceE"
   },
   "outputs": [],
   "source": []
  },
  {
   "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": null,
    "metadata": {
     "id": "rAPSw97aYfE0"
    },
    "outputs": [],
    "source": []
   },
   {
    "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": null,
    "metadata": {
     "id": "T5OxJRZ8uvR7"
    },
    "outputs": [],
    "source": [
     "import pandas as pd\n"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
     "id": "xNpI_XXoYhs0"
    },
    "outputs": [],
    "source": []
   },
   {
    "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": null,
    "metadata": {
     "id": "dgyC0JhVYl4F"
    },
    "outputs": [],
    "source": []
```

```
   },
   {
    "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": null,
    "metadata": {
     "id": "_XMC8aEt0llB"
    },
    "outputs": [],
    "source": [
     "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
     "id": "knH76sDKYsVX"
    },
    "outputs": [],
    "source": []
   }
  ],
  "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
 }
```