```
{
 "nbformat": 4,
 "nbformat_minor": 0,
 "metadata": {
  "colab": {
   "provenance": [],
   "collapsed_sections": []
  },
  "kernelspec": {
   "name": "python3",
   "display_name": "Python 3"
  },
  "language_info": {
   "name": "python"
  }
 },
 "cells": [
  {
   "cell_type": "markdown",
   "source": [
    "# 1. Split this string"
   ],
   "metadata": {
    "id": "TAvbq87dneJ1"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "s = \"Hi there Sam!\""
   ],
   "metadata": {
    "id": "0DI3jIEwoM-N"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "s.split()"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "l9ewcw_AoUYL",
    "outputId": "4a7bf440-6f87-4cc8-f152-97073d7aa888"
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "['Hi', 'there', 'Sam!']"
      ]
```

```
      },
      "metadata": {},
      "execution_count": 6
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "#  2. Use .format() to print the following string\n",
    "\n",
    "## The output should be: The diameter of Earth is 12742 kilometers"
  ],
  "metadata": {
    "id": "bJE6gW3ApAzi"
  }
},
{
  "cell_type": "code",
  "source": [
    "planet = \"Earth\"\n",
    "diameter = 12742"
  ],
  "metadata": {
    "id": "2to42RZariXc"
  },
  "execution_count": 5,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "word = \"The diameter of {} is {} kilometers\"\n",
    "print(word.format(planet,diameter))"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "RhzZblBdrs9D",
    "outputId": "69b22637-591a-419e-a900-8952cfaa789d"
  },
  "execution_count": 6,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "The diameter of Earth is 12742 kilometers\n"
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "# 3. In this nest dictionary grab the word \"hello\""
```

```
  ],
  "metadata": {
    "id": "MzYcu_SBvXiG"
  }
},
{
  "cell_type": "code",
  "source": [
    "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}"
  ],
  "metadata": {
    "id": "TU6A6BImvxmv"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "print(d['k1'][3]['tricky'][3]['target'][3])"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "LE1tHcyywJvi",
    "outputId": "88bbaba0-a873-4e81-9840-4f326bb538f3"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "hello\n"
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "# Numpy"
  ],
  "metadata": {
    "id": "_GiyaJQTA3UT"
  }
},
{
  "cell_type": "code",
  "source": [
    "import numpy as np"
  ],
  "metadata": {
    "id": "KqhWyLUrA7p2"
  },
  "execution_count": null,
```

```
  "outputs": []
},
{
  "cell_type": "markdown",
  "source": [
    "# 4.1 Create an array of 10 zeros"
  ],
  "metadata": {
    "id": "IVtSo6mmBA_d"
  }
},
{
  "cell_type": "code",
  "source": [
    "np.zeros(10)"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "YDKXkQC1BL_b",
    "outputId": "b889a83e-6695-40b4-a138-0d38c783a28a"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])"
        ]
      },
      "metadata": {},
      "execution_count": 50
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "# 4.2 Create an array of 10 fives"
  ],
  "metadata": {
    "id": "DOx2NdsjCCLt"
  }
},
{
  "cell_type": "code",
  "source": [
    "np.ones(10)*5"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "2MYP_0CmCI_b",
    "outputId": "aa6816aa-2113-4e45-8ebe-b9ab13416345"
```

```
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "array([5., 5., 5., 5., 5., 5., 5., 5., 5.])"
        ]
      },
      "metadata": {},
      "execution_count": 49
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "# 5. Create an array of all the even integers from 20 to 35"
  ],
  "metadata": {
    "id": "-Pd9KXv1CZdj"
  }
},
{
  "cell_type": "code",
  "source": [
    "np.arange(20,36,2)"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "6FR9XaKECk5w",
    "outputId": "3d470bd2-a0c7-46a2-a3dc-d7d880a8f65e"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "array([20, 22, 24, 26, 28, 30, 32, 34])"
        ]
      },
      "metadata": {},
      "execution_count": 48
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "# 6. Create a 3x3 matrix with values ranging from 0 to 8"
  ],
  "metadata": {
    "id": "7w-oaE4dDe0y"
```

```
  }
 },
 {
  "cell_type": "code",
  "source": [
   "np.arange(0,9).reshape(3,3)"
  ],
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/"
   },
   "id": "PuhLw4qJDrzA",
   "outputId": "6862d85e-eb39-400e-e250-15d251f93f6f"
  },
  "execution_count": null,
  "outputs": [
   {
    "output_type": "execute_result",
    "data": {
     "text/plain": [
      "array([[0, 1, 2],\n",
      "       [3, 4, 5],\n",
      "       [6, 7, 8]])"
     ]
    },
    "metadata": {},
    "execution_count": 52
   }
  ]
 },
 {
  "cell_type": "markdown",
  "source": [
   "# 7. Concatenate a and b\n",
   "# a = np.array([1, 2, 3]), b = np.array([4, 5, 6])"
  ],
  "metadata": {
   "id": "q7TqfbzkE8a6"
  }
 },
 {
  "cell_type": "code",
  "source": [
   "a = np.array([1,2,3])\n",
   "b = np.array([4,5,6])\n",
   "np.concatenate((a,b),axis = None)"
  ],
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/"
   },
   "id": "qIS4o4tmFFL1",
   "outputId": "08dd42b6-9c07-4c66-c14d-5f0576414553"
  },
  "execution_count": null,
  "outputs": [
   {
```

```
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "array([1, 2, 3, 4, 5, 6])"
        ]
      },
      "metadata": {},
      "execution_count": 51
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "# Pandas"
  ],
  "metadata": {
    "id": "6_yZdQgaHoZy"
  }
},
{
  "cell_type": "markdown",
  "source": [
    "# 8. Create a dataframe with 3 rows and 2 columns"
  ],
  "metadata": {
    "id": "uWvm_H-PI1O6"
  }
},
{
  "cell_type": "code",
  "source": [
    "import pandas as pd"
  ],
  "metadata": {
    "id": "LFLsNFbzI822"
  },
  "execution_count": 2,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "d = {\"Name\":[\"Sam\",\"Krishna\",\"Ibrahim\"],\"GraduationYear\":[2023,2023,2023]}\n",
    "print(pd.DataFrame(d))"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "fqyoKDvvJyCM",
    "outputId": "d0bc0737-d777-4cdf-b197-95f70d5ecc77"
  },
  "execution_count": 3,
  "outputs": [
    {
      "output_type": "stream",
```

```
      "name": "stdout",
      "text": [
        "      Name  GraduationYear\n",
        "0      Sam            2023\n",
        "1  Krishna            2023\n",
        "2  Ibrahim            2023\n"
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "# 9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023"
  ],
  "metadata": {
    "id": "Qqh8tQrvLe-8"
  }
},
{
  "cell_type": "code",
  "source": [
    "date = pd.date_range(start = \"1-1-2023\" , end=\"2-10-2023\")\n",
    "for val in date:\n",
    "  print(val)"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "HhhBGmdLLpi7",
    "outputId": "634e869d-dfcf-49c0-b395-3d383102a61f"
  },
  "execution_count": null,
  "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",
    "lists = [[1, 'aaa', 22],\n",
    "         [2, 'bbb', 25],\n",
    "         [3, 'ccc', 24]]"
  ],
  "metadata": {
    "id": "yUh4MGepNncY"
  }
},
{
  "cell_type": "code",
  "source": [
    "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]"
  ],
  "metadata": {
    "id": "Jl3fVM-ZN2Su"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "print(pd.DataFrame(lists))"
  ],
  "metadata": {
    "colab": {
```

```
        "base_uri": "https://localhost:8080/"
       },
       "id": "tFPQi2H5OD5O",
       "outputId": "ba8ffed6-8786-48d0-b75a-57b0d27ee666"
      },
      "execution_count": null,
      "outputs": [
       {
        "output_type": "stream",
        "name": "stdout",
        "text": [
         "   0    1   2\n",
         "0  1  aaa  22\n",
         "1  2  bbb  25\n",
         "2  3  ccc  24\n"
        ]
       }
      ]
     }
    ]
}
```