```
{
  "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": [
        "# Basic Python"
      ],
      "metadata": {
        "id": "McSxJAwcOdZ1"
      }
    },
    {
      "cell_type": "markdown",
      "source": [
        "## 1. Split this string"
      ],
   "metadata": {
    "id": "CU48hgo4Owz5"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "s = \"Hi there Sam!\"\n"
   ],
   "metadata": {
    "id": "s07c7JK7Oqt-"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "s = \"Hi there Sam!\"\n",
    "xx = s.split()\n",
    "print(xx)"
   ],
   "metadata": {
    "id": "6mGVa3SQYLkb",
    "outputId": "fb95b0b6-0e18-436e-f0a9-72bbb01604c4",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "['Hi', 'there', 'Sam!']\n"
     ]
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "## 2. Use .format() to print the following string. \n",
    "\n",
    "### Output should be: The diameter of Earth is 12742 kilometers."
   ],
   "metadata": {
    "id": "GH1QBn8HP375"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "planet = \" The diameter of Earth \"\n",
    "diameter = \" is 12742 kilometers \"\n",
    "print(planet + diameter.format())\n"
   ],
   "metadata": {
    "id": "_ZHoml3kPqic",
    "outputId": "3b387ee3-519e-4244-b633-631f306cdd13",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      " The diameter of Earth  is 12742 kilometers \n"
     ]
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [],
   "metadata": {
    "id": "HyRyJv6CYPb4"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "markdown",
   "source": [
    "## 3. In this nest dictionary grab the word \"hello\""
   ],
   "metadata": {
    "id": "KE74ZEwkRExZ"
   }
```

```
},
{
 "cell_type": "code",
 "source": [
  "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}\n",
  "d['k1'][3]['tricky'][3]['target'][3]"
 ],
 "metadata": {
  "id": "fcVwbCc1QrQI",
  "outputId": "0bbfcd9a-211a-46e6-a193-6cd07198a8f5",
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 36
  }
 },
 "execution_count": null,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "'hello'"
    ],
    "application/vnd.google.colaboratory.intrinsic+json": {
     "type": "string"
    }
   },
   "metadata": {},
   "execution_count": 62
  }
 ]
```

```
},
{
 "cell_type": "code",
 "source": [
  "print(\"hello\")"
 ],
 "metadata": {
  "id": "MvbkMZpXYRaw",
  "outputId": "cbb286d6-00bf-4ae5-d2bb-8a7931298018",
  "colab": {
   "base_uri": "https://localhost:8080/"
  }
 },
 "execution_count": null,
 "outputs": [
  {
   "output_type": "stream",
   "name": "stdout",
   "text": [
    "hello\n"
   ]
  }
 ]
},
{
 "cell_type": "markdown",
 "source": [
  "# Numpy"
 ],
 "metadata": {
  "id": "bw0vVp-9ddjv"
 }
},
{
 "cell_type": "code",
 "source": [
  "import numpy as np"
 ],
 "metadata": {
  "id": "LLiE_TYrhA1O"
 },
 "execution_count": null,
 "outputs": []
},
{
 "cell_type": "markdown",
 "source": [
  "## 4.1 Create an array of 10 zeros? \n",
  "## 4.2 Create an array of 10 fives?"
 ],
 "metadata": {
  "id": "wOg8hinbgx30"
 }
},
{
 "cell_type": "code",
 "source": [
  "import numpy as np\n",
  "np.zeros(10)"
 ],
 "metadata": {
  "id": "NHrirmgCYXvU",
    "outputId": "807f0687-9dd7-4291-f35e-4cf5c27fbc28",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "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": 6
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "DineshAlone=np.ones(10)*5\n",
    "print(DineshAlone)"
   ],
   "metadata": {
    "id": "e4005lsTYXxx",
    "outputId": "205e5608-1beb-4f05-a433-92ca560eae80",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]\n"
     ]
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "## 5. Create an array of all the even integers from 20 to 35"
   ],
   "metadata": {
    "id": "gZHHDUBvrMX4"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "array=np.arange(20,35,2)\n",
    "print(array)"
   ],
   "metadata": {
    "id": "oAI2tbU2Yag-",
    "outputId": "134cf12c-827e-41c6-bee5-fb05f315dbbd",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "[20 22 24 26 28 30 32 34]\n"
     ]
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "## 6. Create a 3x3 matrix with values ranging from 0 to 8"
   ],
   "metadata": {
    "id": "NaOM308NsRpZ"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "import numpy as np\n",
    "x =  np.arange(0, 9).reshape(3,3)\n",
    "print(x)\n"
   ],
   "metadata": {
    "id": "tOlEVH7BYceE",
    "outputId": "6eeaed0d-525f-42dd-d81d-dce7cbc494e0",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "[[0 1 2]\n",
      " [3 4 5]\n",
      " [6 7 8]]\n"
     ]
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "## 7. Concatenate a and b \n",
    "## a = np.array([1, 2, 3]), b = np.array([4, 5, 6])"
   ],
   "metadata": {
    "id": "hQ0dnhAQuU_p"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "a = np.array([1, 2, 3])\n",
    "b = np.array([4, 5, 6])\n",
    "np.concatenate((a, b), axis=0)\n"
   ],
   "metadata": {
    "id": "rAPSw97aYfE0",
    "outputId": "eb039375-2dea-4cae-f412-1b4454cad39c",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "array([1, 2, 3, 4, 5, 6])"
      ]
     },
     "metadata": {},
     "execution_count": 18
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "# Pandas"
   ],
   "metadata": {
    "id": "dlPEY9DRwZga"
   }
  },
  {
   "cell_type": "markdown",
   "source": [
    "## 8. Create a dataframe with 3 rows and 2 columns"
   ],
   "metadata": {
    "id": "ijoYW51zwr87"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "import pandas as pd\n",
    "\n",
    "df = {'col_1': [0, 1, 2, 3],\n",
    "      'col_2': [4, 5, 6, 7]}\n",
    "df = pd.DataFrame(df)\n",
    "print(df)\n"
   ],
   "metadata": {
    "id": "T5OxJRZ8uvR7",
    "outputId": "6e62dc11-b9bb-4b67-f12c-4a2ec853ba9c",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "   col_1  col_2\n",
      "0      0      4\n",
      "1      1      5\n",
      "2      2      6\n",
      "3      3      7\n"
     ]
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [],
   "metadata": {
    "id": "xNpI_XXoYhs0"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "markdown",
   "source": [
    "## 9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023"
   ],
   "metadata": {
    "id": "UXSmdNclyJQD"
   }
```

```
},
{
 "cell_type": "code",
 "source": [
  "import pandas as pd\n",
  "dRan1 = pd.date_range(start ='1-1-2023',\n",
  "           end ='10-01-2023', freq ='M')\n",
  "\n",
  "  \n",
  "print(dRan1)"
 ],
 "metadata": {
  "id": "dgyC0JhVYl4F",
  "outputId": "33c402be-7880-4b03-cad6-95d5e729bbb1",
  "colab": {
   "base_uri": "https://localhost:8080/"
  }
 },
 "execution_count": null,
 "outputs": [
  {
   "output_type": "stream",
   "name": "stdout",
   "text": [
    "DatetimeIndex(['2023-01-31', '2023-02-28', '2023-03-31', '2023-04-30',\n",
    "               '2023-05-31', '2023-06-30', '2023-07-31', '2023-08-31',\n",
    "               '2023-09-30'],\n",
    "              dtype='datetime64[ns]', freq='M')\n"
   ]
  }
 ]
```

```
},
{
 "cell_type": "markdown",
 "source": [
  "## 10. Create 2D list to DataFrame\n",
  "\n",
  "lists = [[1, 'aaa', 22],\n",
  "        [2, 'bbb', 25],\n",
  "        [3, 'ccc', 24]]"
 ],
 "metadata": {
  "id": "ZizSetD-y5az"
 }
},
{
 "cell_type": "code",
 "source": [
  "\n",
  "import pandas as pd\n",
  "arr= np.arraylists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]\n",
  "df=pd.DataFrame(arr)\n",
  "print(df)"
 ],
 "metadata": {
  "id": "_XMC8aEt0llB",
  "outputId": "1ba05646-35cc-4cb3-a668-4b3a4ec46399",
  "colab": {
   "base_uri": "https://localhost:8080/"
  }
 },
 "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"
     ]
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [],
   "metadata": {
    "id": "knH76sDKYsVX"
   },
   "execution_count": null,
   "outputs": []
  }
 ]
}
```