```
{
 "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!\""
  ],
  "metadata": {
    "id": "s07c7JK7Oqt-"
  },
  "execution_count": 34,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
    "s=s.split(\" \")\n",
    "print(s)"
  ],
  "metadata": {
    "id": "6mGVa3SQYLkb",
    "outputId": "3fde57d7-ed61-4394-df6c-e1015ad30b53",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "execution_count": 35,
  "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 = \"Earth\"\n",
    "diameter = 12742"
  ],
  "metadata": {
    "id": "_ZHoml3kPqic"
  },
  "execution_count": 36,
  "outputs": []
 },
```

```
{
 "cell_type": "code",
 "source": [
  "print(\"The diameter of {} is {} kilometers\".format(planet,diameter));"
 ],
 "metadata": {
  "id": "HyRyJv6CYPb4",
  "outputId": "54f1955c-7cbe-4a29-8096-fcbe48e0f75a",
  "colab": {
   "base_uri": "https://localhost:8080/"
  }
 },
 "execution_count": 39,
 "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": "KE74ZEwkRExZ"
 }
```

```
},
{
 "cell_type": "code",
 "source": [
  "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}"
 ],
 "metadata": {
  "id": "fcVwbCc1QrQI"
 },
 "execution_count": null,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "print(d['k1'][3][\"tricky\"][3]['target'][3])"
 ],
 "metadata": {
  "id": "MvbkMZpXYRaw",
  "outputId": "c18c87a3-61ad-49c1-8f98-beeb9aaf629b",
  "colab": {
   "base_uri": "https://localhost:8080/"
  }
 },
 "execution_count": 40,
 "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": [
   "a=np.zeros(10)\n",
   "print(b)"
  ],
  "metadata": {
   "id": "NHrirmgCYXvU",
   "outputId": "fb11a83a-ee86-4a98-8c51-8814082048f5",
   "colab": {
    "base_uri": "https://localhost:8080/"
   }
  },
  "execution_count": 41,
  "outputs": [
   {
    "output_type": "stream",
    "name": "stdout",
    "text": [
     "[4 5 6]\n"
    ]
   }
  ]
 },
 {
  "cell_type": "code",
  "source": [
   "b=np.ones(10)*5\n",
```

```
    "print(b)"
   ],
   "metadata": {
    "id": "e4005lsTYXxx",
    "outputId": "6abaab42-12fc-4163-d419-4f70b1499fb3",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": 42,
   "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": [
   "import numpy as np\n",
   "x = np.arange(20,37,2).reshape(3,3)\n",
   "print(x)"
  ],
  "metadata": {
   "id": "oAI2tbU2Yag-",
   "outputId": "1d2c622d-6a2f-4824-bf94-9f7b416645b8",
   "colab": {
    "base_uri": "https://localhost:8080/"
   }
  },
  "execution_count": 43,
  "outputs": [
   {
    "output_type": "stream",
    "name": "stdout",
    "text": [
     "[[20 22 24]\n",
     " [26 28 30]\n",
     " [32 34 36]]\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": "5a6121ef-96e2-4fac-e100-1e4a136812b6",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": 45,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "[[0 1 2]\n",
      " [3 4 5]\n",
      " [6 7 8]]\n"
     ]
    }
   ]
  },
  {
```

```
 "cell_type": "markdown",
 "source": [
  "## 7. Concatinate a and b \n",
  "## a = np.array([1, 2, 3]), b = np.array([4, 5, 6])"
 ],
 "metadata": {
  "id": "hQ0dnhAQuU_p"
 }
},
{
 "cell_type": "code",
 "source": [
  "import numpy as np\n",
  "a = np.array([1,2,3])\n",
  "b = np.array([4,5,6])\n",
  "c=np.concatenate((a,b))\n",
  "print(c)"
 ],
 "metadata": {
  "id": "rAPSw97aYfE0",
  "outputId": "03d5d94a-c451-4f1d-d36d-982a6cb815b1",
  "colab": {
   "base_uri": "https://localhost:8080/"
  }
 },
 "execution_count": 46,
 "outputs": [
  {
   "output_type": "stream",
   "name": "stdout",
   "text": [
```

```
        "[1 2 3 4 5 6]\n"
      ]
    }
  ]
 },
 {
  "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"
  ],
  "metadata": {
    "id": "T5OxJRZ8uvR7"
  },
```

```
  "execution_count": 47,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "a=[[1,'a'],[2,'b'],[3,'c']]\n",
   "df=pd.DataFrame(a,columns=[\"id\",\"name\"])\n",
   "print(df)"
  ],
  "metadata": {
   "id": "xNpI_XXoYhs0",
   "outputId": "b75e9551-35eb-4aa7-845d-719fc5c3a4a2",
   "colab": {
    "base_uri": "https://localhost:8080/"
   }
  },
  "execution_count": 48,
  "outputs": [
   {
    "output_type": "stream",
    "name": "stdout",
    "text": [
     "   id name\n",
     "0   1    a\n",
     "1   2    b\n",
     "2   3    c\n"
    ]
   }
  ]
 },
```

```
{
 "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": [
  "a=pd.date_range(start='1-1-2023',end='2-10-2023',freq='1D')\n",
  "for i in a:\n",
  "  print(i)"
 ],
 "metadata": {
  "id": "dgyC0JhVYl4F",
  "outputId": "ae293953-b4e3-4d03-cce4-c216774c904c",
  "colab": {
   "base_uri": "https://localhost:8080/"
  }
 },
 "execution_count": 50,
 "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",
    "\n",
    "lists = [[1, 'aaa', 22],\n",
    ""        [2, 'bbb', 25],\n",
    ""        [3, 'ccc', 24]]"
  ],
  "metadata": {
    "id": "ZizSetD-y5az"
  }
 },
 {
  "cell_type": "code",
  "source": [
    "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]"
  ],
  "metadata": {
```

```
   "id": "_XMC8aEt0llB"
  },
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "df=pd.DataFrame(lists,columns=[\"tokens\",'Id',\"value\"])\n",
   "print(df)"
  ],
  "metadata": {
   "id": "knH76sDKYsVX",
   "outputId": "1f846ffb-2938-4003-c479-3ccc4986fe5e",
   "colab": {
    "base_uri": "https://localhost:8080/"
   }
  },
  "execution_count": 49,
  "outputs": [
   {
    "output_type": "stream",
    "name": "stdout",
    "text": [
     "   tokens   Id  value\n",
     "0      1  aaa     22\n",
     "1      2  bbb     25\n",
     "2      3  ccc     24\n"
    ]
   }
  ]
```

```
    }
  ]
}
```