

```
{
```

```
    "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": "CU48hgo40wz5"  
        }  
      },  
      {  
        "cell_type": "code",  
        "source": [  
          "s = \"Hi there Sam!\""  
        ],  
        "metadata": {  
          "id": "s07c7JK70qt-"  
        }  
      }  
    ]  
  }  
}
```

```

    },
    "execution_count": null,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "s=\"Hi there Sam!\"\n",
      "s=s.split()\n",
      "print(s);"
    ],
    "metadata": {
      "id": "6mGVa3SQYLkb",
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "outputId": "99b8d2a1-9e56-4bf4-dc82-0e0aafbddf57"
    },
    "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"
    },
    "execution_count": null,
    "outputs": [
      {
        "cell_type": "code",

```

```

"source": [
    "planet = \"Earth\\\"\\n\",
    "diameter = 12742"
],
"metadata": {
    "id": "_ZHoml3kPqic"
},
"execution_count": null,
"outputs": []
},
{
    "cell_type": "code",
    "source": [
        "planet=\"Earth\\\"\\n\",
        "diameter=12742\\n\",
        "print('The diameter of {} is {} kilometers.'.format(planet,diameter
    ],
    "metadata": {
        "id": "HyRyJv6CYPb4",
        "colab": {
            "base_uri": "https://localhost:8080/"
        },
        "outputId": "d6c886a8-b265-4f3d-e192-2c1ab0a11487"
    },
    "execution_count": null,
    "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]"}]}",
    ],
    "metadata": {
      "id": "fcVwbCc1QrQI"
    },
    "execution_count": null,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3]"}]}",
      "print(d['k1'][3][\"tricky\"][3][\"target\"][3])"
    ],
    "metadata": {
      "id": "MvbkmZpXYRaw",
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "outputId": "76ddbf25-2219-40ba-aa57-057fd1552bf9"
    },
    "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_TYrhA10"
    },
    "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",
      "array=np.zeros(10)\n",
      "print(\"An array of 10 zeros:\")\n",
      "print(array)"
    ],
    "metadata": {
      "id": "NHrirmgCYXvU",
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "outputId": "0df15ba0-7f93-4fa5-dae1-8aa708569c3c"
    },
    "execution_count": null,
    "outputs": [
```

```

{
  "output_type": "stream",
  "name": "stdout",
  "text": [
    "An array of 10 zeros:\n",
    "[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]\n"
  ]
}
],
{
  "cell_type": "code",
  "source": [
    "import numpy as np\n",
    "array=np.ones(10)*5\n",
    "print(\"An array of 10 fives:\")\n",
    "print(array)"
  ],
  "metadata": {
    "id": "e40051sTYXxx",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "96b522f9-cdc0-4acc-b5fa-e6a2d96029f1"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "An array of 10 fives:\n",
        "[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",
        "array=np.arange(20,31,2)\n",
        "print(\"Array of all the even integers from 20 to 35\")\n",
        "print(array) "
      ],
      "metadata": {
        "id": "oAI2tbU2Yag-",
        "colab": {
          "base_uri": "https://localhost:8080/"
        }
      },
      "outputId": "6ada6cb5-af1b-46e1-e48e-70f44942679f"
    },
    "execution_count": null,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "Array of all the even integers from 20 to 35\n",
          "[20 22 24 26 28 30]\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",
  "np.arange(0,9).reshape((3,3))"
],
"metadata": {
  "id": "t01EVH7BYceE",
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "outputId": "28e9b3a5-9f43-4e92-9ddc-fb478eb64fcf"
},
"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": 7
  }
],
},
{
  "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": [
    "import numpy as np\n",
    "a=np.array([1,2,3])\n",

```



```

        "b=np.array([4,5,6])\n",
        "c=np.concatenate((a,b))"
    ],
    "metadata": {
        "id": "rAPSw97aYfE0"
    },
    "execution_count": null,
    "outputs": []
},
{
    "cell_type": "markdown",
    "source": [
        "# Pandas"
    ],
    "metadata": {
        "id": "d1PEY9DRwZga"
    }
},
{
    "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": "T50xJRZ8uvR7"
    },
    "execution_count": null,
    "outputs": []
},
{
    "cell_type": "code",
    "source": [
        "import pandas as pd\n",

```

```

import numpy as np\n",
\n",
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine'],\n",
            'score': [12.5, 9, 16.5]}\n",
labels = ['a', 'b', 'c']\n",
\n",
df = pd.DataFrame(exam_data , index=labels)\n",
print("\nFirst three rows of the data frame:\n")\n",
print(df.iloc[:3])"
],
"metadata": {
  "id": "xNpI_XXoYhs0",
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "outputId": "6dfa081f-f322-41f6-a5d6-02c10efb3005"
},
"execution_count": null,
"outputs": [
  {
    "output_type": "stream",
    "name": "stdout",
    "text": [
      "\nFirst three rows of the data frame:\n",
      "      name  score\n",
      "a Anastasia  12.5\n",
      "b      Dima   9.0\n",
      "c Katherine  16.5\n"
    ]
  }
],
},
{
  "cell_type": "markdown",
  "source": [
    "## 9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023"
  ],
  "metadata": {
    "id": "UXSmdNc1yJQD"
  }
},
{

```

```

"cell_type": "code",
"source": [
    "from datetime import date as date_n \n",
    "    \n",
    "def number_of_days(date_1, date_2): \n",
    "    return (date_2 - date_1).days \n",
    "    \n",
    "# Driver program \n",
    "date_1 = date_n(2023, 1,1) \n",
    "date_2 = date_n(2023, 2, 10) \n",
    "print (\n\"Number of Days between the given Dates are: \", number_of_
],
"metadata": {
    "id": "dgyC0JhVY14F",
    "colab": {
        "base_uri": "https://localhost:8080/"
    },
    "outputId": "123e0e54-92df-43b3-80f0-51d6cf33e8c0"
},
"execution_count": null,
"outputs": [
    {
        "output_type": "stream",
        "name": "stdout",
        "text": [
            "Number of Days between the given Dates are: 40 days\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": "_XMC8aEt011B"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "import pandas as pd\n",
    "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]\n",
    "df=pd.DataFrame(lists,columns=['category','name','marks'])\n",
    "print(df)"
  ],
  "metadata": {
    "id": "knH76sDKYsVX",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "outputId": "22887f3e-467d-42af-a5a3-52d5d13773bc"
},
"execution_count": null,
"outputs": [
  {
    "output_type": "stream",
    "name": "stdout",
    "text": [
      "   category name  marks\n",
      "0         1  aaa    22\n",
      "1         2  bbb    25\n",
      "2         3  ccc    24\n"
    ]
  }
]
}
]

```

