```
{
  “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”: 1,
   “outputs”: []
  },
  {
   “cell_type”: “code”,
   “source”: [
    “print(s.split())”
   ],
   “metadata”: {
    “id”: “6mGVa3SQYLkb”,
    “colab”: {
     “base_uri”: https://localhost:8080/
    },
    “outputId”: “53f0fcad-8f82-419b-fdaf-aad7aa61caa2”
   },
   “execution_count”: 3,
   “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”: 5,
  “outputs”: []
 },
 {
  “cell_type”: “code”,
  “source”: [
   “print(\”The diameter of {} is {} kilometers\”.format(planet,diameter))”
  ],
  “metadata”: {
   “id”: “HyRyJv6CYPb4”,
   “colab”: {
    “base_uri”: https://localhost:8080/
   },
   “outputId”: “e4f4b3f1-7fa9-446e-e6a9-8a7d764a22e7”
  },
  “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”: “KE74ZEwkRExZ”
      }
    },
    {
      “cell_type”: “code”,
      “source”: [
        “d = {‘k1’:[1,2,3,{‘tricky’:[‘oh’,’man’,’inception’,{‘target’:[1,2,3,’hello’]}]}]}”
      ],
      “metadata”: {
        “id”: “fcVwbCc1QrQl”
      },
      “execution_count”: 7,
      “outputs”: []
    },
    {
      “cell_type”: “code”,
      “source”: [
        “print(d[‘k1’][3][‘tricky’][3][‘target’][3])”
      ],
      “metadata”: {
        “id”: “MvbkMZpXYRaw”,
        “colab”: {
          “base_uri”: https://localhost:8080/
        },
    “outputId”: “66fac63e-081c-4431-e1b0-2f43f8f88ceb”
   },
   “execution_count”: 8,
   “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”: 9,
 “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”: [
  “np.zeros(10)”
 ],
 “metadata”: {
  “id”: “NHrirmgCYXvU”,
  “colab”: {
   “base_uri”: https://localhost:8080/
  },
  “outputId”: “541c58b3-5ae1-4026-e93d-9ab293a335a9”
 },
 “execution_count”: 45,
 “outputs”: [
  {
   “output_type”: “execute_result”,
   “data”: {
    “text/plain”: [
     “array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])”
    ]
   },
   “metadata”: {},
   “execution_count”: 45
  }
 ]
},
{
 “cell_type”: “code”,
 “source”: [
  “np.ones(10)*5”
 ],
 “metadata”: {
  “id”: “e4005lsTYXxx”,
  “colab”: {
   “base_uri”: https://localhost:8080/
  },
  “outputId”: “70176ac7-4a33-40d7-876f-c28d98b9aa18”
 },
 “execution_count”: 46,
 “outputs”: [
  {
   “output_type”: “execute_result”,
   “data”: {
    “text/plain”: [
          “array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])”
        ]
      },
      “metadata”: {},
      “execution_count”: 46
    }
  ]
},
{
  “cell_type”: “markdown”,
  “source”: [
    “## 5. Create an array of all the even integers from 20 to 35”
  ],
  “metadata”: {
    “id”: “gZHHDUBvrMX4”
  }
},
{
  “cell_type”: “code”,
  “source”: [
    “np.array([int(i) for I in range(20,35,2)])”
  ],
  “metadata”: {
    “id”: “oAI2tbU2Yag-“,
    “colab”: {
      “base_uri”: https://localhost:8080/
    },
    “outputId”: “2e755b91-cd23-4d3c-a468-c9d384b26bc2”
  },
 “execution_count”: 44,
 “outputs”: [
  {
   “output_type”: “execute_result”,
   “data”: {
    “text/plain”: [
     “array([20, 22, 24, 26, 28, 30, 32, 34])”
    ]
   },
   “metadata”: {},
   “execution_count”: 44
  }
 ]
},
{
 “cell_type”: “markdown”,
 “source”: [
  “## 6. Create a 3x3 matrix with values ranging from 0 to 8”
 ],
 “metadata”: {
  “id”: “NaOM308NsRpZ”
 }
},
{
 “cell_type”: “code”,
 “source”: [
  “np.arange(0, 9).reshape(3,3)”
 ],
 “metadata”: {
   “id”: “tOlEVH7BYceE”,
   “colab”: {
    “base_uri”: https://localhost:8080/
   },
   “outputId”: “28af9638-641c-410a-b2d4-76aa3856acb5”
  },
  “execution_count”: 51,
  “outputs”: [
   {
    “output_type”: “execute_result”,
    “data”: {
     “text/plain”: [
      “array([[0, 1, 2],\n”,
      “       [3, 4, 5],\n”,
      “       [6, 7, 8]])”
     ]
    },
    “metadata”: {},
    “execution_count”: 51
   }
  ]
 },
 {
  “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”: [
  “np.concatenate((np.array([1,2,3]),np.array([4,5,6])))”
 ],
 “metadata”: {
  “id”: “rAPSw97aYfE0”,
  “colab”: {
   “base_uri”: https://localhost:8080/
  },
  “outputId”: “8adbd767-a18b-4382-f0cb-a502e6b3dcab”
 },
 “execution_count”: 43,
 “outputs”: [
  {
   “output_type”: “execute_result”,
   “data”: {
    “text/plain”: [
     “array([1, 2, 3, 4, 5, 6])”
    ]
   },
   “metadata”: {},
   “execution_count”: 43
  }
 ]
},
{
  “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”: 15,
  “outputs”: []
},
{
 “cell_type”: “code”,
 “source”: [
  “pd.DataFrame({0:[1]*3,1:[1]*3})”
 ],
 “metadata”: {
  “id”: “xNpI_XXoYhs0”,
  “colab”: {
   “base_uri”: https://localhost:8080/,
   “height”: 143
  },
  “outputId”: “906e1a61-83a6-496f-8381-5d7de3474cf9”
 },
 “execution_count”: 34,
 “outputs”: [
  {
   “output_type”: “execute_result”,
   “data”: {
    “text/plain”: [
     “   0  1\n”,
     “0  1  1\n”,
     “1  1  1\n”,
     “2  1  1”
    ],
    “text/html”: [
     “\n”,
     “  <div id=\”df-9b29a928-12ed-4a1c-9523-f483e2a474dd\”>\n”,
     “    <div class=\”colab-df-container\”>\n”,
     “      <div>\n”,
“<style scoped>\n”,
“    .dataframe tbody tr th:only-of-type {\n”,
“        vertical-align: middle;\n”,
“    }\n”,
“\n”,
“    .dataframe tbody tr th {\n”,
“        vertical-align: top;\n”,
“    }\n”,
“\n”,
“    .dataframe thead th {\n”,
“        text-align: right;\n”,
“    }\n”,
“</style>\n”,
“<table border=\”1\” class=\”dataframe\”>\n”,
“  <thead>\n”,
“    <tr style=\”text-align: right;\”>\n”,
“      <th></th>\n”,
“      <th>0</th>\n”,
“      <th>1</th>\n”,
“    </tr>\n”,
“  </thead>\n”,
“  <tbody>\n”,
“    <tr>\n”,
“      <th>0</th>\n”,
“      <td>1</td>\n”,
“      <td>1</td>\n”,
“    </tr>\n”,
“    <tr>\n”,
“      <th>1</th>\n”,
“    <td>1</td>\n”,
“    <td>1</td>\n”,
“   </tr>\n”,
“   <tr>\n”,
“    <th>2</th>\n”,
“    <td>1</td>\n”,
“    <td>1</td>\n”,
“   </tr>\n”,
“  </tbody>\n”,
“</table>\n”,
“</div>\n”,
“    <button class=\”colab-df-convert\” onclick=\”convertToInteractive(‘df-9b29a928-12ed-
4a1c-9523-f483e2a474dd’)\”\n”,
“            title=\”Convert this dataframe to an interactive table.\”\n”,
“            style=\”display:none;\”>\n”,
“      \n”,
“  <svg xmlns=\http://www.w3.org/2000/svg\ height=\”24px\”viewBox=\”0 0 24 24\”\n”,
“     width=\”24px\”>\n”,
“   <path d=\”M0 0h24v24H0V0z\” fill=\”none\”/>\n”,
“   <path d=\”M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11
1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-
.94-.94-2.06-.94 2.06-2.06.94z\”/><path d=\”M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-
1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2
1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41
20z\”/>\n”,
“  </svg>\n”,
“    </button>\n”,
“    \n”,
“  <style>\n”,
“   .colab-df-container {\n”,
“    display:flex;\n”,
“    flex-wrap:wrap;\n”,
“    gap: 12px;\n”,
“  }\n”,
“\n”,
“  .colab-df-convert {\n”,
“    background-color: #E8F0FE;\n”,
“    border: none;\n”,
“    border-radius: 50%;\n”,
“    cursor: pointer;\n”,
“    display: none;\n”,
“    fill: #1967D2;\n”,
“    height: 32px;\n”,
“    padding: 0 0 0 0;\n”,
“    width: 32px;\n”,
“  }\n”,
“\n”,
“  .colab-df-convert:hover {\n”,
“    background-color: #E2EBFA;\n”,
“    box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n”,
“    fill: #174EA6;\n”,
“  }\n”,
“\n”,
“  [theme=dark] .colab-df-convert {\n”,
“    background-color: #3B4455;\n”,
“    fill: #D2E3FC;\n”,
“  }\n”,
“\n”,
“  [theme=dark] .colab-df-convert:hover {\n”,
“    background-color: #434B5C;\n”,
"     box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"     filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"     fill: #FFFFFF;\n",
"   }\n",
"  </style>\n",
"\n",
"     <script>\n",
"       const buttonEl =\n",
"         document.querySelector('#df-9b29a928-12ed-4a1c-9523-f483e2a474dd button.colab-df-convert');\n",
"       buttonEl.style.display =\n",
"         google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"       async function convertToInteractive(key) {\n",
"         const element = document.querySelector('#df-9b29a928-12ed-4a1c-9523-f483e2a474dd');\n",
"         const dataTable =\n",
"           await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                    [key], {});\n",
"         if (!dataTable) return;\n",
"\n",
"         const docLinkHtml = 'Like what you see? Visit the ' +\n",
"           '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
"           + ' to learn more about interactive tables.';\n",
"         element.innerHTML = '';\n",
"         dataTable['output_type'] = 'display_data';\n",
"         await google.colab.output.renderOutput(dataTable, element);\n",
"         const docLink = document.createElement('div');\n",
"         docLink.innerHTML = docLinkHtml;\n",
     “          element.appendChild(docLink);\n”,
     “        }\n”,
     “      </script>\n”,
     “    </div>\n”,
     “  </div>\n”,
     “ “
    ]
   },
   “metadata”: {},
   “execution_count”: 34
  }
 ]
},
{
 “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”: [
  “pd.date_range(start=\”01/01/2023\”,end=\”10/02/2023\”)”
 ],
 “metadata”: {
  “id”: “dgyC0JhVYl4F”,
     “colab”: {
      “base_uri”: https://localhost:8080/
     },
     “outputId”: “95780498-cc0e-4d86-aa8a-27e6461a0a2c”
    },
    “execution_count”: 16,
    “outputs”: [
     {
      “output_type”: “execute_result”,
      “data”: {
       “text/plain”: [
        “DatetimeIndex([‘2023-01-01’, ‘2023-01-02’, ‘2023-01-03’, ‘2023-01-04’,\n”,
        “               ‘2023-01-05’, ‘2023-01-06’, ‘2023-01-07’, ‘2023-01-08’,\n”,
        “               ‘2023-01-09’, ‘2023-01-10’,\n”,
        “               …\n”,
        “               ‘2023-09-23’, ‘2023-09-24’, ‘2023-09-25’, ‘2023-09-26’,\n”,
        “               ‘2023-09-27’, ‘2023-09-28’, ‘2023-09-29’, ‘2023-09-30’,\n”,
        “               ‘2023-10-01’, ‘2023-10-02’],\n”,
        “              dtype=’datetime64[ns]’, length=275, freq=’D’)”
       ]
      },
      “metadata”: {},
      “execution_count”: 16
     }
    ]
   },
   {
    “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”: 36,
  “outputs”: []
 },
 {
  “cell_type”: “code”,
  “source”: [
   “pd.DataFrame(lists)”
  ],
  “metadata”: {
   “id”: “knH76sDKYsVX”,
   “colab”: {
```

```
  “base_uri”: https://localhost:8080/,
  “height”: 143
 },
 “outputId”: “44c5d335-c49d-412c-eac8-41c85540fe17”
},
“execution_count”: 37,
“outputs”: [
 {
  “output_type”: “execute_result”,
  “data”: {
   “text/plain”: [
    “   0    1   2\n”,
    “0  1  aaa  22\n”,
    “1  2  bbb  25\n”,
    “2  3  ccc  24”
   ],
   “text/html”: [
    “\n”,
    “  <div id=\”df-6c359152-0dd4-4058-bd46-87b89fa7776c\”>\n”,
    “    <div class=\”colab-df-container\”>\n”,
    “      <div>\n”,
    “<style scoped>\n”,
    “    .dataframe tbody tr th:only-of-type {\n”,
    “        vertical-align: middle;\n”,
    “    }\n”,
    “\n”,
    “    .dataframe tbody tr th {\n”,
    “        vertical-align: top;\n”,
    “    }\n”,
“\n”,
“    .dataframe thead th {\n”,
“        text-align: right;\n”,
“    }\n”,
“</style>\n”,
“<table border=\”1\” class=\”dataframe\”>\n”,
“  <thead>\n”,
“    <tr style=\”text-align: right;\”>\n”,
“      <th></th>\n”,
“      <th>0</th>\n”,
“      <th>1</th>\n”,
“      <th>2</th>\n”,
“    </tr>\n”,
“  </thead>\n”,
“  <tbody>\n”,
“    <tr>\n”,
“      <th>0</th>\n”,
“      <td>1</td>\n”,
“      <td>aaa</td>\n”,
“      <td>22</td>\n”,
“    </tr>\n”,
“    <tr>\n”,
“      <th>1</th>\n”,
“      <td>2</td>\n”,
“      <td>bbb</td>\n”,
“      <td>25</td>\n”,
“    </tr>\n”,
“    <tr>\n”,
“      <th>2</th>\n”,
“      <td>3</td>\n”,
“      <td>ccc</td>\n”,
“      <td>24</td>\n”,
“    </tr>\n”,
“  </tbody>\n”,
“</table>\n”,
“</div>\n”,
“      <button class=\”colab-df-convert\” onclick=\”convertToInteractive(‘df-6c359152-0dd4-4058-bd46-87b89fa7776c’)\”\n”,
“              title=\”Convert this dataframe to an interactive table.\”\n”,
“              style=\”display:none;\”>\n”,
“        \n”,
“  <svg xmlns=\http://www.w3.org/2000/svg\ height=\”24px\”viewBox=\”0 0 24 24\”\n”,
“       width=\”24px\”>\n”,
“    <path d=\”M0 0h24v24H0V0z\” fill=\”none\”/>\n”,
“    <path d=\”M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\”/><path d=\”M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\”/>\n”,
“  </svg>\n”,
“      </button>\n”,
“      \n”,
“  <style>\n”,
“    .colab-df-container {\n”,
“      display:flex;\n”,
“      flex-wrap:wrap;\n”,
“      gap: 12px;\n”,
“    }\n”,
“\n”,
“   .colab-df-convert {\n”,
“     background-color: #E8F0FE;\n”,
“     border: none;\n”,
“     border-radius: 50%;\n”,
“     cursor: pointer;\n”,
“     display: none;\n”,
“     fill: #1967D2;\n”,
“     height: 32px;\n”,
“     padding: 0 0 0 0;\n”,
“     width: 32px;\n”,
“   }\n”,
“\n”,
“   .colab-df-convert:hover {\n”,
“     background-color: #E2EBFA;\n”,
“     box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n”,
“     fill: #174EA6;\n”,
“   }\n”,
“\n”,
“   [theme=dark] .colab-df-convert {\n”,
“     background-color: #3B4455;\n”,
“     fill: #D2E3FC;\n”,
“   }\n”,
“\n”,
“   [theme=dark] .colab-df-convert:hover {\n”,
“     background-color: #434B5C;\n”,
“     box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n”,
“     filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n”,
“     fill: #FFFFFF;\n”,
“   }\n”,
“ </style>\n”,
“\n”,
“     <script>\n”,
“       const buttonEl =\n”,
“         document.querySelector(‘#df-6c359152-0dd4-4058-bd46-87b89fa7776c button.colab-df-convert’);\n”,
“       buttonEl.style.display =\n”,
“         google.colab.kernel.accessAllowed ? ‘block’ : ‘none’;\n”,
“\n”,
“       async function convertToInteractive(key) {\n”,
“         const element = document.querySelector(‘#df-6c359152-0dd4-4058-bd46-87b89fa7776c’);\n”,
“         const dataTable =\n”,
“           await google.colab.kernel.invokeFunction(‘convertToInteractive’,\n”,
“                                                     [key], {});\n”,
“         if (!dataTable) return;\n”,
“\n”,
“         const docLinkHtml = ‘Like what you see? Visit the ‘ +\n”,
“           ‘<a target=\”_blank\” href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>’\n”,
“           + ‘ to learn more about interactive tables.’;\n”,
“         element.innerHTML = ‘’;\n”,
“         dataTable[‘output_type’] = ‘display_data’;\n”,
“         await google.colab.output.renderOutput(dataTable, element);\n”,
“         const docLink = document.createElement(‘div’);\n”,
“         docLink.innerHTML = docLinkHtml;\n”,
“         element.appendChild(docLink);\n”,
“       }\n”,
“     </script>\n”,
“   </div>\n”,
       “  </div>\n”,
       “  “
      ]
     },
     “metadata”: {},
     “execution_count”: 37
    }
   ]
  }
 ]
}
```