**Project development phase ( delivery)**

**Splint 3**


```
 1nbformat”: 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”: 2,
  “outputs”: []
},
{
  “cell_type”: “code”,
  “source”: [
    “print(s.split())”
  ],
  “metadata”: {
    “id”: “6mGVa3SQYLkb”,
    “outputId”: “c7eca547-4108-4061-c951-ee5b4c17722c”,
    “colab”: {
      “base_uri”: https://localhost:8080/
    }
```

```
  },
  “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”: 4,
  “outputs”: []
},
{
  “cell_type”: “code”,
  “source”: [
    “print(f\”The diameter of Earth is {diameter} kilometers.\”)”
  ],
  “metadata”: {
    “id”: “HyRyJv6CYPb4”,
    “outputId”: “14daa2e3-a1d3-4a04-d1ef-7d54e04de95b”,
    “colab”: {
      “base_uri”: https://localhost:8080/
    }
  },
  “execution_count”: 7,
  “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”: 8,
 “outputs”: []
},
{
 “cell_type”: “code”,
 “source”: [
  “print(d[‘k1’][3][‘tricky’][3][‘target’][3])”
 ],
 “metadata”: {
  “id”: “MvbkMZpXYRaw”,
  “outputId”: “0b2deb79-1250-4d1b-be1e-560a73b2cb37”,
  “colab”: {
```

```
    “base_uri”: https://localhost:8080/
   }
  },
  “execution_count”: 9,
  “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”: 10,
  “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”: [
   “zeros=np.zeros(10)”
  ],
  “metadata”: {
   “id”: “NHrirmgCYXvU”
  },
  “execution_count”: 11,
  “outputs”: []
},
{
  “cell_type”: “code”,
  “source”: [
```

```
    “fives=np.full(10,5)\n”,
    “print(zeros,fives)”
   ],
   “metadata”: {
    “id”: “e4005lsTYXxx”,
    “outputId”: “0f2ef939-1168-496d-b3fc-6d1e6be7b190”,
    “colab”: {
     “base_uri”: https://localhost:8080/
    }
   },
   “execution_count”: 15,
   “outputs”: [
    {
     “output_type”: “stream”,
     “name”: “stdout”,
     “text”: [
      “[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.] [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”: [
  “arr=[I for I in range(20,35+1) if i%2==0]\n”,
  “arr”
 ],
 “metadata”: {
  “id”: “oAI2tbU2Yag-“,
  “outputId”: “b53dd299-7197-43d0-8d62-4bf0e840c885”,
  “colab”: {
   “base_uri”: https://localhost:8080/
  }
 },
 “execution_count”: 16,
 “outputs”: [
  {
   “output_type”: “execute_result”,
   “data”: {
    “text/plain”: [
     “[20, 22, 24, 26, 28, 30, 32, 34]”
    ]
   },
   “metadata”: {},
   “execution_count”: 16
  }
 ]
},
{
```

```
 “cell_type”: “markdown”,
 “source”: [
  “## 6. Create a 3x3 matrix with values ranging from 0 to 8”
 ],
 “metadata”: {
  “id”: “NaOM308NsRpZ”
 }
},
{
 “cell_type”: “code”,
 “source”: [
  “array=np.arange(0,9).reshape((3,3))\n”,
  “array”
 ],
 “metadata”: {
  “id”: “tOlEVH7BYceE”,
  “outputId”: “a8144bde-e4e9-46ac-f89b-22306e71c7bd”,
  “colab”: {
   “base_uri”: https://localhost:8080/
  }
 },
 “execution_count”: 20,
 “outputs”: [
  {
   “output_type”: “execute_result”,
   “data”: {
    “text/plain”: [
     “array([[0, 1, 2],\n”,
     “       [3, 4, 5],\n”,
```

```
        “       [6, 7, 8]])”
       ]
      },
      “metadata”: {},
      “execution_count”: 20
    }
  ]
},
{
  “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”,
    “c=np.concatenate((a,b))\n”,
    “c”
  ],
  “metadata”: {
    “id”: “rAPSw97aYfE0”,
    “outputId”: “7c0390f6-80f7-4bcf-a20d-0456883124a2”,
```

```
   “colab”: {
    “base_uri”: https://localhost:8080/
   }
  },
  “execution_count”: 27,
  “outputs”: [
   {
    “output_type”: “execute_result”,
    “data”: {
     “text/plain”: [
      “array([1, 2, 3, 4, 5, 6])”
     ]
    },
    “metadata”: {},
    “execution_count”: 27
   }
  ]
 },
 {
  “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”: 28,
  “outputs”: []
},
{
  “cell_type”: “code”,
  “source”: [
    “d={‘name’:[‘raj’,’jhon’,’joe’],’age’:[21,26,28]}\n”,
    “df=pd.DataFrame.from_dict(d)\n”,
    “df”
  ],
  “metadata”: {
    “id”: “xNpI_XXoYhs0”,
    “outputId”: “a7f76b6e-5834-4ec2-b583-bf08743f9a28”,
    “colab”: {
```

```
  “base_uri”: https://localhost:8080/,
  “height”: 143
 }
},
“execution_count”: 34,
“outputs”: [
 {
  “output_type”: “execute_result”,
  “data”: {
   “text/plain”: [
    “   name  age\n”,
    “0   raj   21\n”,
    “1  jhon   26\n”,
    “2   joe   28”
   ],
   “text/html”: [
    “\n”,
    “  <div id=\”df-598e9ee5-fb74-4a51-acc5-099517bc009e\”>\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>name</th>\n”,
“      <th>age</th>\n”,
“    </tr>\n”,
“  </thead>\n”,
“  <tbody>\n”,
“    <tr>\n”,
“      <th>0</th>\n”,
“      <td>raj</td>\n”,
“      <td>21</td>\n”,
“    </tr>\n”,
“    <tr>\n”,
“      <th>1</th>\n”,
“      <td>jhon</td>\n”,
“      <td>26</td>\n”,
“    </tr>\n”,
“    <tr>\n”,
“      <th>2</th>\n”,
“      <td>joe</td>\n”,
“      <td>28</td>\n”,
“    </tr>\n”,
“  </tbody>\n”,
```

“</table>\n”,

“ \n”,

“ <button class=\”colab-df-convert\” onclick=\”convertToInteractive(‘df-598e9ee5-fb74-4a51-acc5-099517bc009e’)\”\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-598e9ee5-fb74-4a51-acc5-099517bc009e 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-598e9ee5-fb74-4a51-acc5-099517bc009e');\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”: [
   “date=pd.date_range(\”01-01-2023\”,\”10-02-2023\”,freq=\”D\”)”
  ],
  “metadata”: {
   “id”: “dgyC0JhVYl4F”,
   “outputId”: “187fb707-db6f-42fa-d730-c96366dfadae”,
   “colab”: {
    “base_uri”: https://localhost:8080/
   }
  },
  “execution_count”: 35,
  “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": 35
   }
  ]
 },
 {
  "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”: [
    “df=pd.DataFrame(lists,columns=[\”C1\”,\”C2\”,\”C3\”])\n”,
    “df”
  ],
  “metadata”: {
    “id”: “knH76sDKYsVX”,
    “outputId”: “1f47a5ef-d884-493b-85a7-4e462c85d335”,
    “colab”: {
      “base_uri”: https://localhost:8080/,
      “height”: 143
    }
  },
  “execution_count”: 38,
  “outputs”: [
```

```
{
 “output_type”: “execute_result”,
 “data”: {
  “text/plain”: [
   “   C1   C2  C3\n”,
   “0   1  aaa  22\n”,
   “1   2  bbb  25\n”,
   “2   3  ccc  24”
  ],
  “text/html”: [
   “\n”,
   “  <div id=\”df-fef6f28e-9431-4092-82cd-da1a9e44b091\”>\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>C1</th>\n”,
“     <th>C2</th>\n”,
“     <th>C3</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-fef6f28e-9431-4092-82cd-da1a9e44b091’)\”\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-fef6f28e-9431-4092-82cd-da1a9e44b091 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-fef6f28e-9431-4092-82cd-da1a9e44b091’);\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”: 38
```

```
        }
       ]
     },
     {
       “cell_type”: “code”,
       “source”: [],
       “metadata”: {
         “id”: “5xEVQXkjMsDS”
       },
       “execution_count”: null,
       “outputs”: []
     }
   ]
}
```