```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "McSxJAwcOdZ1"
   },
   "source": [
    "# Basic Python"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "CU48hgo4Owz5"
   },
   "source": [
    "## 1. Split this string"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "s07c7JK7Oqt-"
   },
   "outputs": [],
   "source": [
    "s = \"Hi there Sam!\""
   ]
  },
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "6mGVa3SQYLkb",
  "outputId": "53f8d9e0-0781-4559-e7ef-62a049a1ca2c"
 },
 "outputs": [
  {
   "data": {
    "text/plain": [
     "['Hi', 'there', 'Sam!']"
    ]
   },
   "execution_count": 53,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "a=s.split(' ')\n",
  "a"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "GH1QBn8HP375"
  },
  "source": [
   "## 2. Use .format() to print the following string. \n",
   "\n",
   "### Output should be: The diameter of Earth is 12742 kilometers."
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "id": "_ZHoml3kPqic"
  },
  "outputs": [],
  "source": [
   "planet = \"Earth\"\n",
   "diameter = 12742"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/"
   },
   "id": "HyRyJv6CYPb4",
   "outputId": "2ac7b054-7d26-4a8b-bc41-4f9d9b42dbfc"
  },
  "outputs": [
   {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "The diameter of Earth is 12742 kilometers.\n"
     ]
    }
   ],
   "source": [
    "print('The diameter of {} is {} kilometers.'.format(planet,diameter))"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "KE74ZEwkRExZ"
   },
   "source": [
    "## 3. In this nest dictionary grab the word \"hello\""
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "fcVwbCc1QrQI"
   },
   "outputs": [],
   "source": [
    "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}"
   ]
  },
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 36
  },
  "id": "MvbkMZpXYRaw",
  "outputId": "ddaad2eb-52ef-4772-8e94-b39927b26c6e"
 },
 "outputs": [
  {
   "data": {
    "application/vnd.google.colaboratory.intrinsic+json": {
     "type": "string"
    },
    "text/plain": [
     "'hello'"
    ]
   },
   "execution_count": 9,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "d['k1'][3]['tricky'][3]['target'][3]"
 ]
},
{
   "cell_type": "markdown",
   "metadata": {
    "id": "bw0vVp-9ddjv"
   },
   "source": [
    "# Numpy"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "LLiE_TYrhA1O"
   },
   "outputs": [],
   "source": [
    "import numpy as np"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "wOg8hinbgx30"
   },
   "source": [
    "## 4.1 Create an array of 10 zeros? \n",
    "## 4.2 Create an array of 10 fives?"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "NHrirmgCYXvU",
    "outputId": "f72bb9de-93e1-44fd-ccdb-4564f8d212fa"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])"
      ]
     },
     "execution_count": 22,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "a=np.zeros(10)\n",
    "a"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "e4005lsTYXxx",
    "outputId": "088c3049-209d-405d-d97f-3f081e532d0f"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([5, 5, 5, 5, 5, 5, 5, 5, 5, 5])"
      ]
     },
     "execution_count": 21,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "a=np.ones(10,dtype=int)*5\n",
    "a"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "gZHHDUBvrMX4"
   },
   "source": [
    "## 5. Create an array of all the even integers from 20 to 35"
   ]
  },
  {
```

```
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "oAI2tbU2Yag-",
  "outputId": "a3a20c77-489e-449c-f2fb-46d557e632dc"
 },
 "outputs": [
  {
   "data": {
    "text/plain": [
     "array([20, 22, 24, 26, 28, 30, 32, 34])"
    ]
   },
   "execution_count": 20,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "a=np.arange(20,35,2)\n",
  "a"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "NaOM308NsRpZ"
 },
   "source": [
    "## 6. Create a 3x3 matrix with values ranging from 0 to 8"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "tOlEVH7BYceE",
    "outputId": "0227e5be-c1de-4651-ac3d-39f929c66890"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([[0, 1, 2],\n",
       "       [3, 4, 5],\n",
       "       [6, 7, 8]])"
      ]
     },
     "execution_count": 24,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "np.arange(0,9).reshape(3,3)\n"
   ]
```

```
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "hQ0dnhAQuU_p"
 },
 "source": [
  "## 7. Concatenate a and b \n",
  "## a = np.array([1, 2, 3]), b = np.array([4, 5, 6])"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "rAPSw97aYfE0",
  "outputId": "09ab48d4-aab0-4fa8-a317-9ae7448b96cf"
 },
 "outputs": [
  {
   "data": {
    "text/plain": [
     "array([1, 2, 3, 4, 5, 6])"
    ]
   },
   "execution_count": 29,
   "metadata": {},
   "output_type": "execute_result"
```

```
    }
   ],
   "source": [
    "a = np.array([1, 2, 3])\n",
    "b = np.array([4, 5, 6])\n",
    "c=np.concatenate((a,b))\n",
    "c"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "dlPEY9DRwZga"
   },
   "source": [
    "# Pandas"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "ijoYW51zwr87"
   },
   "source": [
    "## 8. Create a dataframe with 3 rows and 2 columns"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "T5OxJRZ8uvR7"
   },
   "outputs": [],
   "source": [
    "import pandas as pd"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 143
    },
    "id": "OX-ng4S0ZNRf",
    "outputId": "09015410-e23f-4c3c-8f1d-bd6bb6a8aa95"
   },
   "outputs": [
    {
     "data": {
      "text/html": [
       "<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>PLACE</th>\n",
"      <th>FAMOUS_PLACE</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>VELLORE</td>\n",
"      <td>FORT</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>CHENNAI</td>\n",
"      <td>MARINA BEACH</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>MADURAI</td>\n",
"      <td>TEMPLE</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
       "</div>"
      ],
      "text/plain": [
       "    PLACE  FAMOUS_PLACE\n",
       "1  VELLORE         FORT\n",
       "2  CHENNAI  MARINA BEACH\n",
       "3  MADURAI       TEMPLE"
      ]
     },
     "execution_count": 3,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "a={'PLACE':['VELLORE','CHENNAI','MADURAI'],'FAMOUS_PLACE':['FORT','MARINA
BEACH','TEMPLE']}\n",
    "b=pd.DataFrame(a,index=(1,2,3))\n",
    "b"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "UXSmdNclyJQD"
   },
   "source": [
    "## 9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023"
   ]
  },
  {
```

```
"cell_type": "code",
"execution_count": null,
"metadata": {
 "colab": {
  "base_uri": "https://localhost:8080/"
 },
 "id": "DYEO_xdXaeoI",
 "outputId": "90a21456-d256-4000-a836-4a110bec22c6"
},
"outputs": [
 {
  "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', '2023-01-11', '2023-01-12',\n",
    "               '2023-01-13', '2023-01-14', '2023-01-15', '2023-01-16',\n",
    "               '2023-01-17', '2023-01-18', '2023-01-19', '2023-01-20',\n",
    "               '2023-01-21', '2023-01-22', '2023-01-23', '2023-01-24',\n",
    "               '2023-01-25', '2023-01-26', '2023-01-27', '2023-01-28',\n",
    "               '2023-01-29', '2023-01-30', '2023-01-31', '2023-02-01',\n",
    "               '2023-02-02', '2023-02-03', '2023-02-04', '2023-02-05',\n",
    "               '2023-02-06', '2023-02-07', '2023-02-08', '2023-02-09',\n",
    "               '2023-02-10'],\n",
    "              dtype='datetime64[ns]', freq='D')"
   ]
  },
  "execution_count": 46,
  "metadata": {},
  "output_type": "execute_result"
 }
```

```
   ],
   "source": [
    "a=pd.date_range('01-01-2023','02-10-2023')\n",
    "a"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "ZizSetD-y5az"
   },
   "source": [
    "## 10. Create 2D list to DataFrame\n",
    "\n",
    "lists = [[1, 'aaa', 22],[2, 'bbb', 25],[3, 'ccc', 24]]"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "_XMC8aEt0llB"
   },
   "outputs": [],
   "source": [
    "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
"metadata": {
 "colab": {
  "base_uri": "https://localhost:8080/",
  "height": 143
 },
 "id": "knH76sDKYsVX",
 "outputId": "8b8759a9-5fa1-4380-8da1-6d237861b154"
},
"outputs": [
 {
  "data": {
   "text/html": [
    "\n",
    "  <div id=\"df-178b37a9-533f-4612-9e81-1224faca903a\">\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>A</th>\n",
"      <th>B</th>\n",
"      <th>C</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>1</td>\n",
"      <td>aaa</td>\n",
"      <td>22</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>2</td>\n",
"      <td>bbb</td>\n",
"      <td>25</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</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-178b37a9-533f-4612-9e81-1224faca903a')\"\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-178b37a9-533f-4612-9e81-1224faca903a 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-178b37a9-533f-4612-9e81-1224faca903a');\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",
      "  "
     ],
     "text/plain": [
      "   A    B   C\n",
      "1  1  aaa  22\n",
      "2  2  bbb  25\n",
      "3  3  ccc  24"
     ]
    },
    "execution_count": 52,
    "metadata": {},
    "output_type": "execute_result"
   }
   ],
   "source": [
    "a=pd.DataFrame(lists,columns=('A','B','C'),index=('1','2','3'))\n",
    "a"
   ]
  }
 ],
 "metadata": {
  "colab": {
   "collapsed_sections": [],
   "provenance": []
  },
  "kernelspec": {
   "display_name": "Python 3 (ipykernel)",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.9.12"
  }
 },
 "nbformat": 4,
  "nbformat_minor": 0
}
```