```
{
  "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": [],
      "metadata": {
        "id": "jFTZZ1THEZUK"
      }
    },
    {
      "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": null,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "string = \"Hi there Sam!\"\n",
        "x=string.split()\n",
        "print(x)"
      ],
```

```
      "metadata": {
        "id": "6mGVa3SQYLkb",
        "outputId": "bd86a886-e9bc-4442-f700-45f1b0382e2b",
        "colab": {
          "base_uri": "https://localhost:8080/"
        }
      },
      "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"
      }
    },
    {
      "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",
        "string = \"The diameter of Earth is 12742 kilometers\"\n",
        "print(string.format(planet,diameter))"
      ],
      "metadata": {
        "id": "HyRyJv6CYPb4",
        "outputId": "6b0c709c-68a3-451e-893c-89b68745d25b",
        "colab": {
          "base_uri": "https://localhost:8080/"
        }
      },
      "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,'hello']}
]}]}"
      ],
      "metadata": {
        "id": "fcVwbCc1QrQI"
      },
      "execution_count": null,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "d =
{'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}
]}]}\n",
        "print(d['k1'][3]['tricky'][3]['target'][3])"
      ],
      "metadata": {
        "id": "MvbkMZpXYRaw",
        "outputId": "270a516a-ba3b-4bde-ef0a-b3a8cb6eb478",
        "colab": {
          "base_uri": "https://localhost:8080/"
        }
      },
      "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_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": [
    "import numpy as np\n",
    "arr=np.zeros(10)\n",
    "print(\"An array of 10 zeros:\")\n",
    "print(arr)"
  ],
  "metadata": {
    "id": "NHrirmgCYXvU",
    "outputId": "c2128482-e57c-4c05-c93a-91c1acc6d08f",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "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": [
    "arr=np.ones(10)*5\n",
    "print(\"An array of 10 fives:\")\n",
    "print(arr)"
  ],
```

```
  "metadata": {
    "id": "e4005lsTYXxx",
    "outputId": "abc9311e-4ba5-4a3c-80fc-95c3888e1d80",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "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",
    "a=np.arange(20,36,2)\n",
    "print(\"Array of all the even integers from 20 to 35\")\n",
    "print(a)"
  ],
  "metadata": {
    "id": "oAI2tbU2Yag-",
    "outputId": "93e21265-d6b8-4626-bb2b-cb7283589c7f",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "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 32 34]\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",
    "m =  np.arange(0, 9).reshape(3,3)\n",
    "print(m)\n"
  ],
  "metadata": {
    "id": "tOlEVH7BYceE",
    "outputId": "b6882632-018e-41e3-f691-65de11016ad0",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "[[0 1 2]\n",
        " [3 4 5]\n",
        " [6 7 8]]\n"
      ]
    }
  ]
},
{
  "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),axis=None)\n",
    "print(c)\n"
  ],
  "metadata": {
    "id": "rAPSw97aYfE0",
    "outputId": "8975517d-fae6-4224-98ab-ffa751f37f94",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "execution_count": null,
  "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": null,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "import pandas as pd\n",
        "  \n",
        "# initialize list of lists\n",
        "data = [['Jack', 'America'], ['peter', 'london'], ['maria',
'korea']]\n",
        "  \n",
        "# Create the pandas DataFrame\n",
        "df = pd.DataFrame(data, columns=['Name', 'Country'])\n",
        "  \n",
        "# print dataframe.\n",
        "df"
      ],
      "metadata": {
        "id": "xNpI_XXoYhs0",
        "outputId": "9075c53d-e177-45f1-bb32-1521c88794d7",
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 143
        }
      },
```

```
      "execution_count": null,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "    Name  Country\n",
              "0   Jack  America\n",
              "1  peter   london\n",
              "2  maria    korea"
            ],
            "text/html": [
              "\n",
              "  <div id=\"df-c75fbd63-9b27-4b8b-894c-317d5eb6bc85\">\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>Country</th>\n",
              "    </tr>\n",
              "  </thead>\n",
              "  <tbody>\n",
              "    <tr>\n",
              "      <th>0</th>\n",
              "      <td>Jack</td>\n",
              "      <td>America</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>1</th>\n",
              "      <td>peter</td>\n",
              "      <td>london</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>2</th>\n",
              "      <td>maria</td>\n",
              "      <td>korea</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-c75fbd63-9b27-4b8b-894c-
317d5eb6bc85')\"\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-c75fbd63-9b27-4b8b-
894c-317d5eb6bc85 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-
c75fbd63-9b27-4b8b-894c-317d5eb6bc85');\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": 10
        }
      ]
    },
    {
      "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": [
        "import datetime\n",
        "import pandas as pd\n",
```

```
        "test_date = datetime.datetime.strptime(\"01-1-2023\", \"%d-%m-
%Y\")\n",
        "K = 41\n",
        " \n",
        "date_generated = pd.date_range(test_date, periods=K)\n",
        "print(date_generated.strftime(\"%d-%m-%Y\"))"
      ],
      "metadata": {
        "id": "dgyC0JhVYl4F",
        "outputId": "bd73f335-6093-42dd-8e8d-32f62e3bf474",
        "colab": {
          "base_uri": "https://localhost:8080/"
        }
      },
      "execution_count": null,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "Index(['01-01-2023', '02-01-2023', '03-01-2023', '04-01-
2023', '05-01-2023',\n",
            "       '06-01-2023', '07-01-2023', '08-01-2023', '09-01-
2023', '10-01-2023',\n",
            "       '11-01-2023', '12-01-2023', '13-01-2023', '14-01-
2023', '15-01-2023',\n",
            "       '16-01-2023', '17-01-2023', '18-01-2023', '19-01-
2023', '20-01-2023',\n",
            "       '21-01-2023', '22-01-2023', '23-01-2023', '24-01-
2023', '25-01-2023',\n",
            "       '26-01-2023', '27-01-2023', '28-01-2023', '29-01-
2023', '30-01-2023',\n",
            "       '31-01-2023', '01-02-2023', '02-02-2023', '03-02-
2023', '04-02-2023',\n",
            "       '05-02-2023', '06-02-2023', '07-02-2023', '08-02-
2023', '09-02-2023',\n",
            "       '10-02-2023'],\n",
            "      dtype='object')\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": [
      "import pandas as pd  \n",
      "lst = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]     \n",
      "df = pd.DataFrame(lst)\n",
      "print(df )"
    ],
    "metadata": {
      "id": "Xs2e7q6l7SBh",
      "outputId": "8bfe9227-0ac5-4177-f89b-0e9cc6f0cf85",
      "colab": {
        "base_uri": "https://localhost:8080/"
      }
    },
    "execution_count": null,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "   0    1   2\n",
          "0  1  aaa  22\n",
          "1  2  bbb  25\n",
          "2  3  ccc  24\n"
        ]
      }
    ]
  }
]
}
```