```
{
  "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": 5,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "x = s.split()\n",
        "print(x)"
      ],
```

```
      "metadata": {
        "id": "6mGVa3SQYLkb",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "c1c40d7d-feeb-4746-81a8-3ff0644e738c"
      },
      "execution_count": 7,
      "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": 8,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "print('The diameter of {}is {} kilometer.'.format
(planet,diameter));"
      ],
      "metadata": {
        "id": "HyRyJv6CYPb4",
        "colab": {
          "base_uri": "https://localhost:8080/"
```

```
        },
        "outputId": "fb93e141-29d3-4c07-acd9-f55d371fd428"
      },
      "execution_count": 11,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "The diameter of Earthis 12742 kilometer.\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": 12,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "print(d['k1'][3][\"tricky\"][3]['target'][3])"
      ],
      "metadata": {
        "id": "MvbkMZpXYRaw",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "852a2977-3e38-4e47-ce5f-06e3a12541f8"
      },
      "execution_count": 13,
      "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": 14,
  "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": [
    "array=np.zeros(10)\n",
    "print(\"An array of 10 zeros:\")\n",
    "print(array)"
  ],
  "metadata": {
    "id": "NHrirmgCYXvU",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "8be3d530-c1e2-45b6-bcd6-61d7b3b85c50"
  },
  "execution_count": 15,
```

```
  "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": [
    "array=np.ones(10)*5\n",
    "print(\"An array of 10 fives:\")\n",
    "print(array)"
  ],
  "metadata": {
    "id": "e4005lsTYXxx",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "aa51d1c9-aba0-488d-dac3-d5c4ba2ece67"
  },
  "execution_count": 16,
  "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(20,36,2)\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": "fe3b43de-cace-4565-ad6d-9ba21b4628c3"
   },
   "execution_count": 17,
   "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": [
    "x =  np.arange(0, 9).reshape(3,3)\n",
    "print(x)"
   ],
   "metadata": {
    "id": "tOlEVH7BYceE",
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "52206b1b-c788-4494-9de8-898eec288dd7"
   },
   "execution_count": 18,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "[[0 1 2]\n",
      " [3 4 5]\n",
      " [6 7 8]]\n"
```

```
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "## 7. Concatinate 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",
    "np.concatenate([a, b])"
  ],
  "metadata": {
    "id": "rAPSw97aYfE0",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "c1087411-10bd-43f1-935c-a72bc3460c41"
  },
  "execution_count": 19,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "array([1, 2, 3, 4, 5, 6])"
        ]
      },
      "metadata": {},
      "execution_count": 19
    }
  ]
},
{
  "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": 20,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "import pandas as pd\n",
    "import numpy as np \n",
    "p=np.arange(6).reshape(3,2)\n",
    "label1=['a','b','c']\n",
    "label2=['A','B',]\n",
    "p\n",
    "df = pd.DataFrame(p,index= label1, columns=label2)\n",
    "df"
  ],
  "metadata": {
    "id": "xNpI_XXoYhs0",
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 143
    },
    "outputId": "8107d2fd-a6a1-45ab-abdb-112929d8499a"
  },
  "execution_count": 22,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "   A  B\n",
          "a  0  1\n",
          "b  2  3\n",
          "c  4  5"
        ],
        "text/html": [
          "\n",
```

```
              "  <div id=\"df-c10b90fa-b5f3-428f-8d36-
5955715bb670\">\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",
              "    </tr>\n",
              "  </thead>\n",
              "  <tbody>\n",
              "    <tr>\n",
              "      <th>a</th>\n",
              "      <td>0</td>\n",
              "      <td>1</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>b</th>\n",
              "      <td>2</td>\n",
              "      <td>3</td>\n",
              "    </tr>\n",
              "    <tr>\n",
              "      <th>c</th>\n",
              "      <td>4</td>\n",
              "      <td>5</td>\n",
              "    </tr>\n",
              "  </tbody>\n",
              "</table>\n",
              "</div>\n",
              "      <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-c10b90fa-b5f3-428f-8d36-
5955715bb670')\"\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-c10b90fa-b5f3-
428f-8d36-5955715bb670 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-
c10b90fa-b5f3-428f-8d36-5955715bb670');\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": 22
        }
      ]
    },
    {
      "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=\"2023-01-01\",end=\"2023-02-
10\").to_pydatetime().tolist()"
      ],
      "metadata": {
        "id": "dgyC0JhVYl4F",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "8e0b06e2-cc0e-4cb9-e806-0e67cd0e6ece"
      },
      "execution_count": 25,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "[datetime.datetime(2023, 1, 1, 0, 0),\n",
              " datetime.datetime(2023, 1, 2, 0, 0),\n",
              " datetime.datetime(2023, 1, 3, 0, 0),\n",
              " datetime.datetime(2023, 1, 4, 0, 0),\n",
              " datetime.datetime(2023, 1, 5, 0, 0),\n",
              " datetime.datetime(2023, 1, 6, 0, 0),\n",
              " datetime.datetime(2023, 1, 7, 0, 0),\n",
              " datetime.datetime(2023, 1, 8, 0, 0),\n",
              " datetime.datetime(2023, 1, 9, 0, 0),\n",
              " datetime.datetime(2023, 1, 10, 0, 0),\n",
              " datetime.datetime(2023, 1, 11, 0, 0),\n",
              " datetime.datetime(2023, 1, 12, 0, 0),\n",
              " datetime.datetime(2023, 1, 13, 0, 0),\n",
              " datetime.datetime(2023, 1, 14, 0, 0),\n",
              " datetime.datetime(2023, 1, 15, 0, 0),\n",
              " datetime.datetime(2023, 1, 16, 0, 0),\n",
              " datetime.datetime(2023, 1, 17, 0, 0),\n",
              " datetime.datetime(2023, 1, 18, 0, 0),\n",
              " datetime.datetime(2023, 1, 19, 0, 0),\n",
              " datetime.datetime(2023, 1, 20, 0, 0),\n",
              " datetime.datetime(2023, 1, 21, 0, 0),\n",
              " datetime.datetime(2023, 1, 22, 0, 0),\n",
              " datetime.datetime(2023, 1, 23, 0, 0),\n",
              " datetime.datetime(2023, 1, 24, 0, 0),\n",
              " datetime.datetime(2023, 1, 25, 0, 0),\n",
```

```
          " datetime.datetime(2023, 1, 26, 0, 0),\n",
          " datetime.datetime(2023, 1, 27, 0, 0),\n",
          " datetime.datetime(2023, 1, 28, 0, 0),\n",
          " datetime.datetime(2023, 1, 29, 0, 0),\n",
          " datetime.datetime(2023, 1, 30, 0, 0),\n",
          " datetime.datetime(2023, 1, 31, 0, 0),\n",
          " datetime.datetime(2023, 2, 1, 0, 0),\n",
          " datetime.datetime(2023, 2, 2, 0, 0),\n",
          " datetime.datetime(2023, 2, 3, 0, 0),\n",
          " datetime.datetime(2023, 2, 4, 0, 0),\n",
          " datetime.datetime(2023, 2, 5, 0, 0),\n",
          " datetime.datetime(2023, 2, 6, 0, 0),\n",
          " datetime.datetime(2023, 2, 7, 0, 0),\n",
          " datetime.datetime(2023, 2, 8, 0, 0),\n",
          " datetime.datetime(2023, 2, 9, 0, 0),\n",
          " datetime.datetime(2023, 2, 10, 0, 0)]"
        ]
      },
      "metadata": {},
      "execution_count": 25
    }
  ]
},
{
  "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": 23,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]\n",
    "df = pd.DataFrame(lists, columns =['A', 'B','C']) \n",
```

```
        "print(df )"
      ],
      "metadata": {
        "id": "knH76sDKYsVX",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "7317c33f-679e-4844-f794-5bd8f6118b04"
      },
      "execution_count": 24,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "   A    B   C\n",
            "0  1  aaa  22\n",
            "1  2  bbb  25\n",
            "2  3  ccc  24\n"
          ]
        }
      ]
    }
  ]
}
```