

```

{
  "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": [
        "s.split()"
      ],
      "metadata": {
        "id": "6mGVa3SQYLkb",
        "colab": {
          "base_uri": "https://localhost:8080/"
        }
      }
    }
  ]
}

```

```

    },
    "outputId": "78c2cdc8-c681-463c-c4f6-cb4cb109ea10"
  },
  "execution_count": 9,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "['Hi', 'there', 'Sam!']"
        ]
      },
      "metadata": {},
      "execution_count": 9
    }
  ]
},
{
  "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": 10,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "print('The diameter of {} is {} kilometers.'.format(planet,diameter))"
  ],
  "metadata": {
    "id": "HyRyJv6CYPb4",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "outputId": "55321550-3b21-422e-b95d-605f28971693"
},
"execution_count": 11,
"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": 3,
    "outputs": []
},
{
    "cell_type": "code",
    "source": [
        "d['k1'][3]['tricky'][3]['target'][3]"
    ],
    "metadata": {
        "id": "MvbkMZpXYRaw",
        "colab": {
            "base_uri": "https://localhost:8080/",
            "height": 36
        },
        "outputId": "03e457bf-f674-4e19-e766-965a3aecc163"
    },
    "execution_count": 18,
    "outputs": [
        {
            "output_type": "execute_result",
            "data": {
                "text/plain": [
                    "'hello'"
                ],
                "application/vnd.google.colaboratory.intrinsic+json": {
                    "type": "string"
                }
            },
            "metadata": {},
            "execution_count": 18
        }
    ]
}

```

```

    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "# Numpy"
  ],
  "metadata": {
    "id": "bw0vVp-9ddjv"
  }
},
{
  "cell_type": "code",
  "source": [
    "import numpy as np"
  ],
  "metadata": {
    "id": "LLiE_TYrhA1O"
  },
  "execution_count": 4,
  "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": [
    "zeroArr = np.zeros(10)\n",
    "print(zeroArr)"
  ],
  "metadata": {
    "id": "NHrirmgCYXvU",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "outputId": "f3b9d57d-98f7-4be2-bac9-1743f1b4885e"
},
"execution_count": 24,
"outputs": [
  {
    "output_type": "stream",
    "name": "stdout",
    "text": [
      "[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]\n"
    ]
  }
]
}

```

```

    ],
    },
    {
      "cell_type": "code",
      "source": [
        "tensArr = np.ones(10)*5\n",
        "print(tensArr)"
      ],
      "metadata": {
        "id": "e4005lsTYXxx",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "4fd52bf8-d0e1-4acf-bee9-6450b8f4b27a"
      },
      "execution_count": 25,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "[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": [
        "array = np.arange(20,36,2)\n",
        "print(array)"
      ],
      "metadata": {
        "id": "oAI2tbU2Yag-",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "a28824bf-5bc1-4137-e809-f043b56b94ec"
      },
      "execution_count": 27,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "[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": [
    "arr = np.arange(0,9).reshape(3,3)\n",
    "print(arr)"
  ],
  "metadata": {
    "id": "tOIEVH7BYceE",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "b18f6017-659a-4d6a-a7d9-219d96ca67f2"
  },
  "execution_count": 33,
  "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": [
    "a = np.array([1, 2, 3])\n",
    "b = np.array([4, 5, 6])\n",
    "np.concatenate((a,b), axis=0)"
  ]
}

```

```

],
"metadata": {
  "id": "rAPSw97aYfE0",
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "outputId": "9e8a758a-f042-4c4a-8f98-38df58579e81"
},
"execution_count": 38,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "array([1, 2, 3, 4, 5, 6])"
      ]
    },
    "metadata": {},
    "execution_count": 38
  }
]
},
{
  "cell_type": "markdown",
  "source": [
    "# Pandas"
  ],
  "metadata": {
    "id": "dIPEY9DRwZga"
  }
},
{
  "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": 5,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "col=['Name','Department']\n",

```

```

"row = [['Sudharsan', 'ECE'], ['Manoj', 'ECE'], ['Shyam', 'ECE']]\n",
"\n",
"df = pd.DataFrame(row, columns=col)\n",
"df"
],
"metadata": {
  "id": "xNpl_XXoYhs0",
  "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 143
  },
  "outputId": "52c49e97-41f9-4fc4-b71a-924421038fa1"
},
"execution_count": 55,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "      Name Department\n",
        "0  Sudharsan      ECE\n",
        "1    Manoj      ECE\n",
        "2    Shyam      ECE"
      ],
      "text/html": [
        "\n",
        "<div id=\"df-fdded36a-fc1e-488b-aa4f-0dee38c3d2ae\">\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>Department</th>\n",
        "</tr>\n",
        "</thead>\n",
        "<tbody>\n",
        "<tr>\n",
        "<th>0</th>\n",
        "<td>Sudharsan</td>\n",
        "<td>ECE</td>\n",

```


[illegible]

```

"      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-fdded36a-fc1e-488b-aa4f-0dee38c3d2ae 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-fdded36a-fc1e-488b-aa4f-0dee38c3d2ae');\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": 55
}
]
},
{

```

```

"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')\n"
    ],
    "metadata": {
        "id": "dgyC0JhVYl4F",
        "colab": {
            "base_uri": "https://localhost:8080/"
        },
        "outputId": "3819f514-5c36-4e5a-b795-22fd6f7ff3cd"
    },
    "execution_count": 51,
    "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', '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')"
                ]
            },
            "metadata": {},
            "execution_count": 51
        }
    ]
},
{
    "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": "_XMC8aEt0IIB"
    },
    "execution_count": 6,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "df = pd.DataFrame(lists, columns=['Col1', 'Col2', 'Col3'])\n",
      "df"
    ],
    "metadata": {
      "id": "knH76sDKYsVX",
      "colab": {
        "base_uri": "https://localhost:8080/",
        "height": 143
      }
    },
    "outputId": "bf0ace66-843b-46b7-836c-e07d75cd8e7c"
  },
  "execution_count": 54,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "   Col1 Col2  Col3\n",
          "0     1  aaa   22\n",
          "1     2  bbb   25\n",
          "2     3  ccc   24"
        ],
        "text/html": [
          "\n",
          "  <div id=\"df-25293c72-8517-4411-9e2f-815095c4390d\">\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",
          "          .dataframe tbody tr th {\n",
          "            vertical-align: top;\n",
          "          }\n",
          "          .dataframe thead th {\n",
          "            text-align: right;\n",

```

[illegible]

```

"      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-25293c72-8517-4411-9e2f-
815095c4390d 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-25293c72-8517-4411-
9e2f-815095c4390d');\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
notebook</a>'\n",
"        + ' to learn more about interactive tables.';\n",

```

```
"      element.innerHTML = ";\n",\n"\n      dataTable['output_type'] = 'display_data';\n",\n"\n      await google.colab.output.renderOutput(dataTable, element);\n",\n"\n      const docLink = document.createElement('div');\n",\n"\n      docLink.innerHTML = docLinkHtml;\n",\n"\n      element.appendChild(docLink);\n",\n"\n    }\n",\n"\n  </script>\n",\n"\n</div>\n",\n"</div>\n",\n""\n\n]\n},\n"metadata": {},\n"execution_count": 54\n}\n}\n}\n}
```