

```

{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": [],
      "collapsed_sections": [],
      "toc_visible": true
    },
    "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": [
        "x = s.split()\n",
        "print(x)"
      ],
      "metadata": {

```

```

        "id": "6mGVa3SQYLkb",
        "outputId": "d95a14a0-b1b4-44e8-9fa6-0a96e45a78cb",
        "colab": {
          "base_uri": "https://localhost:8080/"
        }
      },
      "execution_count": 2,
      "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": 3,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "x=\"The diameter of {} is {} kilometers\"\n",
        "print(x.format(planet,diameter))"
      ],
      "metadata": {
        "id": "HyRyJv6CYPb4",
        "outputId": "efc4f9a1-5900-499c-bfb1-ba6529380783",
        "colab": {
          "base_uri": "https://localhost:8080/"
        }
      }
    },
  ],

```

```

    "execution_count": 4,
    "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": 5,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "print(d['k1'][3][\"tricky\"][3]['target'][3])"
    ],
    "metadata": {
      "id": "MvbkMZpXYRaw",
      "outputId": "09190187-f638-4a0f-ebd2-d2f02ef605db",
      "colab": {
        "base_uri": "https://localhost:8080/"
      }
    },
    "execution_count": 6,
    "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": 7,
    "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)\n",
      "print(zeros)"
    ],
    "metadata": {
      "id": "NHrirmgCYXvU",
      "outputId": "4d550339-a446-4fdd-8267-c5e2f4d6ddea",
      "colab": {
        "base_uri": "https://localhost:8080/"
      }
    },
    "execution_count": 8,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]\n"
        ]
      }
    ]
  }

```

```

    ],
    {
        "cell_type": "code",
        "source": [
            "fives=np.ones(10)*5\n",
            "print(fives)"
        ],
        "metadata": {
            "id": "e40051sTYXxx",
            "outputId": "ee46a12c-6507-4d8e-c005-b3e530d7670f",
            "colab": {
                "base_uri": "https://localhost:8080/"
            }
        },
        "execution_count": 9,
        "outputs": [
            {
                "output_type": "stream",
                "name": "stdout",
                "text": [
                    "[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": [
            "array=np.arange(20,36,2)\n",
            "print(array)"
        ],
        "metadata": {
            "id": "oAI2tbU2Yag-",
            "outputId": "e3174acb-0d4b-4f82-e1b5-96bac3a98f4e",
            "colab": {
                "base_uri": "https://localhost:8080/"
            }
        },
        "execution_count": 10,
        "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": [
        "matrix=np.arange(0,9).reshape(3,3)\n",
        "print(matrix)"
    ],
    "metadata": {
        "id": "tO1EVH7BYceE",
        "outputId": "e15944e9-76bb-4923-9d7d-18cd986bce61",
        "colab": {
            "base_uri": "https://localhost:8080/"
        }
    },
    "execution_count": 11,
    "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": "hQ0dnhAQ_u_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",
        "print(c)"
    ],
    "metadata": {
        "id": "rAPSw97aYfE0",
        "outputId": "07482c1e-e16c-4584-e4e3-80def21c83c8",
        "colab": {
            "base_uri": "https://localhost:8080/"
        }
    },
    "execution_count": 12,
    "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": 13,
    "outputs": []
},
{
    "cell_type": "code",
    "source": [

```

```

    "data = {'Name': ['Tom', 'John', 'Krish'], 'Age': [21, 20,
19]}\n",
    "df=pd.DataFrame(data,index=[1,2,3])\n",
    "df  "
],
"metadata": {
    "id": "xNpI_XXoYhs0",
    "outputId": "3f937df4-7714-4c68-8d38-9892eb14b4c9",
    "colab": {
        "base_uri": "https://localhost:8080/",
        "height": 143
    }
},
"execution_count": 14,
"outputs": [
    {
        "output_type": "execute_result",
        "data": {
            "text/plain": [
                "   Name  Age\n1    Tom   21\n2   John   20\n3  Krish   19"
            ],
            "text/html": [
                "\n",
                "  <div id=\"df-de35b4f6-9101-4d53-a895-839ea6ddd640\">\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>1</th>\n",
                "      <td>Tom</td>\n",

```



```

"      <td>21</td>\n",
"    </tr>\n",
"  <tr>\n",
"    <th>2</th>\n",
"    <td>John</td>\n",
"    <td>20</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>3</th>\n",
"    <td>Krish</td>\n",
"    <td>19</td>\n",
"  </tr>\n",
" </tbody>\n",
"</table>\n",
"</div>\n",
"  <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-de35b4f6-9101-4d53-a895-
839ea6ddd640')\">\n",
"    title=\"Convert this dataframe to an
interactive table.\",
"    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-.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-.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-
2.05 0 2.83L4 21.41c.39.39.95 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-de35b4f6-9101-4d53-
a895-839ea6ddd640 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-
de35b4f6-9101-4d53-a895-839ea6ddd640');\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": 14
}
]
},
{
    "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": [
        "period = pd.date_range(start ='01-01-2023', end ='02-10-
2023')\n",
        "for val in period:\n",
        "    print(val)"
    ],
    "metadata": {
        "id": "dgyC0JhVYl4F",
        "outputId": "5332ef48-e635-4938-93f7-e58bbb80bd39",
        "colab": {
            "base_uri": "https://localhost:8080/"
        }
    },
    "execution_count": 16,
    "outputs": [
        {
            "output_type": "stream",
            "name": "stdout",
            "text": [
                "2023-01-01 00:00:00\n",
                "2023-01-02 00:00:00\n",
                "2023-01-03 00:00:00\n",
                "2023-01-04 00:00:00\n",
                "2023-01-05 00:00:00\n",
                "2023-01-06 00:00:00\n",
                "2023-01-07 00:00:00\n",
                "2023-01-08 00:00:00\n",
                "2023-01-09 00:00:00\n",

```

```

        "2023-01-10 00:00:00\n",
        "2023-01-11 00:00:00\n",
        "2023-01-12 00:00:00\n",
        "2023-01-13 00:00:00\n",
        "2023-01-14 00:00:00\n",
        "2023-01-15 00:00:00\n",
        "2023-01-16 00:00:00\n",
        "2023-01-17 00:00:00\n",
        "2023-01-18 00:00:00\n",
        "2023-01-19 00:00:00\n",
        "2023-01-20 00:00:00\n",
        "2023-01-21 00:00:00\n",
        "2023-01-22 00:00:00\n",
        "2023-01-23 00:00:00\n",
        "2023-01-24 00:00:00\n",
        "2023-01-25 00:00:00\n",
        "2023-01-26 00:00:00\n",
        "2023-01-27 00:00:00\n",
        "2023-01-28 00:00:00\n",
        "2023-01-29 00:00:00\n",
        "2023-01-30 00:00:00\n",
        "2023-01-31 00:00:00\n",
        "2023-02-01 00:00:00\n",
        "2023-02-02 00:00:00\n",
        "2023-02-03 00:00:00\n",
        "2023-02-04 00:00:00\n",
        "2023-02-05 00:00:00\n",
        "2023-02-06 00:00:00\n",
        "2023-02-07 00:00:00\n",
        "2023-02-08 00:00:00\n",
        "2023-02-09 00:00:00\n",
        "2023-02-10 00:00:00\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": "_XMC8aEt011B"
    },
    "execution_count": 17,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "df = pd.DataFrame(lists, columns =['1-digits', 'letters','2-
digits']) \n",
      "print(df) "
    ],
    "metadata": {
      "id": "knH76sDKYsVX",
      "outputId": "cc493e27-45c1-4c29-8634-e0fc16b99a6e",
      "colab": {
        "base_uri": "https://localhost:8080/"
      }
    },
    "execution_count": 18,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "   1-digits letters  2-digits\n",
          "0           1    aaa      22\n",
          "1           2    bbb      25\n",
          "2           3    ccc      24\n"
        ]
      }
    ]
  }
]
}

```