

```

{
  "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": 3,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "print (s.split())"
      ],
      "metadata": {
        "id": "6mGVa3SQYLkb",
        "colab": {

```

```

        "base_uri": "https://localhost:8080/"
    },
    "outputId": "7acdfdf1-edfb-4933-953c-8b27f0d042d6"
},
"execution_count": 4,
"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": 10,
    "outputs": []
},
{
    "cell_type": "code",
    "source": [
        "print(f'The diameter of {planet} is {diameter} kilometers')"
    ],
    "metadata": {
        "id": "HyRyJv6CYPb4",
        "colab": {
            "base_uri": "https://localhost:8080/"
        }
    },
    "outputId": "f18a50f1-b404-4fdc-ba03-1b970c239c4d"
},
"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": "fcVwbCc1QrQI"
    },
    "execution_count": 8,
    "outputs": []
},
{
    "cell_type": "code",
    "source": [
        "print(d['k1'][3]['tricky'][3]['target'][3])"
    ],
    "metadata": {
        "id": "MvbkMZpXYRaw",
        "colab": {
            "base_uri": "https://localhost:8080/"
        },
        "outputId": "4343bd10-f849-4735-e67b-e6f79a3b1589"
    },
    "execution_count": 12,
    "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",
        "np.zeros(10)"
    ],
    "metadata": {
        "id": "NHrirmgCYXvU",
        "colab": {
            "base_uri": "https://localhost:8080/"
        },
        "outputId": "c0dcb6d2-9dd7-4bf0-d3f1-08573516a4a2"
    },
    "execution_count": 15,
    "outputs": [
        {
            "output_type": "execute_result",
            "data": {
                "text/plain": [
                    "array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])"
                ]
            },
            "metadata": {},
            "execution_count": 15
        }
    ]
}

```

```
]
},
{
    "cell_type": "code",
    "source": [
        "import numpy as np\n",
        "np.ones((10)*5)"
    ],
    "metadata": {
        "id": "e4005lsTYXxx",
        "colab": {
            "base_uri": "https://localhost:8080/"
        },
        "outputId": "c8b89301-5dbb-4e26-f49f-dbd3feb3b63c"
    },
    "execution_count": 16,
    "outputs": [
        {
            "output_type": "execute_result",
            "data": {
                "text/plain": [
                    "array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,\n1., 1., 1., 1.,\n",
                    "       \"      1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,\n1., 1., 1., 1.,\n",
                    "       \"      1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,\n1., 1., 1.]")
                ]
            },
            "metadata": {},
            "execution_count": 16
        }
    ]
},
{
    "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",
        "print (np.arange(20,35,3))"
    ],
    "metadata": {
        "id": "oAI2tbU2Yag-",
        "colab": {
            "base_uri": "https://localhost:8080/"
        },
```

```

    "outputId": "7a13048d-53fb-408f-9b1e-43f82ab82e8b"
  },
  "execution_count": 17,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "[20 23 26 29 32]\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",
    "np.arange(0,9).reshape(3,3)"
  ],
  "metadata": {
    "id": "tOlEVH7BYceE",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "0ce7ef5b-ff2a-4393-9ba6-7ebbea52b60b"
  },
  "execution_count": 18,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "array([[0, 1, 2],\n",
          "       [3, 4, 5],\n",
          "       [6, 7, 8]])"
        ]
      },
      "metadata": {},
      "execution_count": 18
    }
  ]
},
{
  "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",
    "np.concatenate((a,b),axis=0)"
  ],
  "metadata": {
    "id": "rAPSw97aYfE0",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "cf830597-c726-415e-c2c6-5a1965fcc5dd"
  },
  "execution_count": 20,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "array([1, 2, 3, 4, 5, 6])"
        ]
      },
      "metadata": {},
      "execution_count": 20
    }
  ]
},
{
  "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": 37,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "\n",
    "a=np.arange(0,12,2).reshape(3,2)\n",
    "df=pd.DataFrame(a)\n",
    "print(df)\n"
  ],
  "metadata": {
    "id": "xNpI_XXoYhs0",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "outputId": "350f84ad-c35c-41b4-d0a1-c4580fd09648"
},
{
  "execution_count": 39,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "   0   1\n0   0   2\n1   4   6\n2   8  10\n"
      ]
    }
  ]
},
{
  "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 pandas as pd\n",

```



```
"day=pd.date_range(start=\"01-01-2023\",end=\"10-02-2023\")\n",\n"for i in day:\n",\n"    print(i)\n",\n],\n"metadata": {\n    "id": "dgyC0JhVYl4F",\n    "colab": {\n        "base_uri": "https://localhost:8080/"\n    },\n    "outputId": "6201aecd-d0c3-4be6-f03b-55fe416b4948"\n},\n"execution_count": 26,\n"outputs": [\n    {\n        "output_type": "stream",\n        "name": "stdout",\n        "text": [\n            "2023-01-01 00:00:00\\n",\n            "2023-01-02 00:00:00\\n",\n            "2023-01-03 00:00:00\\n",\n            "2023-01-04 00:00:00\\n",\n            "2023-01-05 00:00:00\\n",\n            "2023-01-06 00:00:00\\n",\n            "2023-01-07 00:00:00\\n",\n            "2023-01-08 00:00:00\\n",\n            "2023-01-09 00:00:00\\n",\n            "2023-01-10 00:00:00\\n",\n            "2023-01-11 00:00:00\\n",\n            "2023-01-12 00:00:00\\n",\n            "2023-01-13 00:00:00\\n",\n            "2023-01-14 00:00:00\\n",\n            "2023-01-15 00:00:00\\n",\n            "2023-01-16 00:00:00\\n",\n            "2023-01-17 00:00:00\\n",\n            "2023-01-18 00:00:00\\n",\n            "2023-01-19 00:00:00\\n",\n            "2023-01-20 00:00:00\\n",\n            "2023-01-21 00:00:00\\n",\n            "2023-01-22 00:00:00\\n",\n            "2023-01-23 00:00:00\\n",\n            "2023-01-24 00:00:00\\n",\n            "2023-01-25 00:00:00\\n",\n            "2023-01-26 00:00:00\\n",\n            "2023-01-27 00:00:00\\n",\n            "2023-01-28 00:00:00\\n",\n            "2023-01-29 00:00:00\\n",\n            "2023-01-30 00:00:00\\n",\n            "2023-01-31 00:00:00\\n",\n            "2023-02-01 00:00:00\\n",\n            "2023-02-02 00:00:00\\n",\n            "2023-02-03 00:00:00\\n",\n            "2023-02-04 00:00:00\\n",\n            "2023-02-05 00:00:00\\n",\n            "2023-02-06 00:00:00\\n",\n        ]\n    }\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",  
"2023-02-11 00:00:00\n",  
"2023-02-12 00:00:00\n",  
"2023-02-13 00:00:00\n",  
"2023-02-14 00:00:00\n",  
"2023-02-15 00:00:00\n",  
"2023-02-16 00:00:00\n",  
"2023-02-17 00:00:00\n",  
"2023-02-18 00:00:00\n",  
"2023-02-19 00:00:00\n",  
"2023-02-20 00:00:00\n",  
"2023-02-21 00:00:00\n",  
"2023-02-22 00:00:00\n",  
"2023-02-23 00:00:00\n",  
"2023-02-24 00:00:00\n",  
"2023-02-25 00:00:00\n",  
"2023-02-26 00:00:00\n",  
"2023-02-27 00:00:00\n",  
"2023-02-28 00:00:00\n",  
"2023-03-01 00:00:00\n",  
"2023-03-02 00:00:00\n",  
"2023-03-03 00:00:00\n",  
"2023-03-04 00:00:00\n",  
"2023-03-05 00:00:00\n",  
"2023-03-06 00:00:00\n",  
"2023-03-07 00:00:00\n",  
"2023-03-08 00:00:00\n",  
"2023-03-09 00:00:00\n",  
"2023-03-10 00:00:00\n",  
"2023-03-11 00:00:00\n",  
"2023-03-12 00:00:00\n",  
"2023-03-13 00:00:00\n",  
"2023-03-14 00:00:00\n",  
"2023-03-15 00:00:00\n",  
"2023-03-16 00:00:00\n",  
"2023-03-17 00:00:00\n",  
"2023-03-18 00:00:00\n",  
"2023-03-19 00:00:00\n",  
"2023-03-20 00:00:00\n",  
"2023-03-21 00:00:00\n",  
"2023-03-22 00:00:00\n",  
"2023-03-23 00:00:00\n",  
"2023-03-24 00:00:00\n",  
"2023-03-25 00:00:00\n",  
"2023-03-26 00:00:00\n",  
"2023-03-27 00:00:00\n",  
"2023-03-28 00:00:00\n",  
"2023-03-29 00:00:00\n",  
"2023-03-30 00:00:00\n",  
"2023-03-31 00:00:00\n",  
"2023-04-01 00:00:00\n",

"2023-04-02 00:00:00\n",  
"2023-04-03 00:00:00\n",  
"2023-04-04 00:00:00\n",  
"2023-04-05 00:00:00\n",  
"2023-04-06 00:00:00\n",  
"2023-04-07 00:00:00\n",  
"2023-04-08 00:00:00\n",  
"2023-04-09 00:00:00\n",  
"2023-04-10 00:00:00\n",  
"2023-04-11 00:00:00\n",  
"2023-04-12 00:00:00\n",  
"2023-04-13 00:00:00\n",  
"2023-04-14 00:00:00\n",  
"2023-04-15 00:00:00\n",  
"2023-04-16 00:00:00\n",  
"2023-04-17 00:00:00\n",  
"2023-04-18 00:00:00\n",  
"2023-04-19 00:00:00\n",  
"2023-04-20 00:00:00\n",  
"2023-04-21 00:00:00\n",  
"2023-04-22 00:00:00\n",  
"2023-04-23 00:00:00\n",  
"2023-04-24 00:00:00\n",  
"2023-04-25 00:00:00\n",  
"2023-04-26 00:00:00\n",  
"2023-04-27 00:00:00\n",  
"2023-04-28 00:00:00\n",  
"2023-04-29 00:00:00\n",  
"2023-04-30 00:00:00\n",  
"2023-05-01 00:00:00\n",  
"2023-05-02 00:00:00\n",  
"2023-05-03 00:00:00\n",  
"2023-05-04 00:00:00\n",  
"2023-05-05 00:00:00\n",  
"2023-05-06 00:00:00\n",  
"2023-05-07 00:00:00\n",  
"2023-05-08 00:00:00\n",  
"2023-05-09 00:00:00\n",  
"2023-05-10 00:00:00\n",  
"2023-05-11 00:00:00\n",  
"2023-05-12 00:00:00\n",  
"2023-05-13 00:00:00\n",  
"2023-05-14 00:00:00\n",  
"2023-05-15 00:00:00\n",  
"2023-05-16 00:00:00\n",  
"2023-05-17 00:00:00\n",  
"2023-05-18 00:00:00\n",  
"2023-05-19 00:00:00\n",  
"2023-05-20 00:00:00\n",  
"2023-05-21 00:00:00\n",  
"2023-05-22 00:00:00\n",  
"2023-05-23 00:00:00\n",  
"2023-05-24 00:00:00\n",  
"2023-05-25 00:00:00\n",

"2023-05-26 00:00:00\n",  
"2023-05-27 00:00:00\n",  
"2023-05-28 00:00:00\n",  
"2023-05-29 00:00:00\n",  
"2023-05-30 00:00:00\n",  
"2023-05-31 00:00:00\n",  
"2023-06-01 00:00:00\n",  
"2023-06-02 00:00:00\n",  
"2023-06-03 00:00:00\n",  
"2023-06-04 00:00:00\n",  
"2023-06-05 00:00:00\n",  
"2023-06-06 00:00:00\n",  
"2023-06-07 00:00:00\n",  
"2023-06-08 00:00:00\n",  
"2023-06-09 00:00:00\n",  
"2023-06-10 00:00:00\n",  
"2023-06-11 00:00:00\n",  
"2023-06-12 00:00:00\n",  
"2023-06-13 00:00:00\n",  
"2023-06-14 00:00:00\n",  
"2023-06-15 00:00:00\n",  
"2023-06-16 00:00:00\n",  
"2023-06-17 00:00:00\n",  
"2023-06-18 00:00:00\n",  
"2023-06-19 00:00:00\n",  
"2023-06-20 00:00:00\n",  
"2023-06-21 00:00:00\n",  
"2023-06-22 00:00:00\n",  
"2023-06-23 00:00:00\n",  
"2023-06-24 00:00:00\n",  
"2023-06-25 00:00:00\n",  
"2023-06-26 00:00:00\n",  
"2023-06-27 00:00:00\n",  
"2023-06-28 00:00:00\n",  
"2023-06-29 00:00:00\n",  
"2023-06-30 00:00:00\n",  
"2023-07-01 00:00:00\n",  
"2023-07-02 00:00:00\n",  
"2023-07-03 00:00:00\n",  
"2023-07-04 00:00:00\n",  
"2023-07-05 00:00:00\n",  
"2023-07-06 00:00:00\n",  
"2023-07-07 00:00:00\n",  
"2023-07-08 00:00:00\n",  
"2023-07-09 00:00:00\n",  
"2023-07-10 00:00:00\n",  
"2023-07-11 00:00:00\n",  
"2023-07-12 00:00:00\n",  
"2023-07-13 00:00:00\n",  
"2023-07-14 00:00:00\n",  
"2023-07-15 00:00:00\n",  
"2023-07-16 00:00:00\n",  
"2023-07-17 00:00:00\n",  
"2023-07-18 00:00:00\n",

"2023-07-19 00:00:00\n",  
"2023-07-20 00:00:00\n",  
"2023-07-21 00:00:00\n",  
"2023-07-22 00:00:00\n",  
"2023-07-23 00:00:00\n",  
"2023-07-24 00:00:00\n",  
"2023-07-25 00:00:00\n",  
"2023-07-26 00:00:00\n",  
"2023-07-27 00:00:00\n",  
"2023-07-28 00:00:00\n",  
"2023-07-29 00:00:00\n",  
"2023-07-30 00:00:00\n",  
"2023-07-31 00:00:00\n",  
"2023-08-01 00:00:00\n",  
"2023-08-02 00:00:00\n",  
"2023-08-03 00:00:00\n",  
"2023-08-04 00:00:00\n",  
"2023-08-05 00:00:00\n",  
"2023-08-06 00:00:00\n",  
"2023-08-07 00:00:00\n",  
"2023-08-08 00:00:00\n",  
"2023-08-09 00:00:00\n",  
"2023-08-10 00:00:00\n",  
"2023-08-11 00:00:00\n",  
"2023-08-12 00:00:00\n",  
"2023-08-13 00:00:00\n",  
"2023-08-14 00:00:00\n",  
"2023-08-15 00:00:00\n",  
"2023-08-16 00:00:00\n",  
"2023-08-17 00:00:00\n",  
"2023-08-18 00:00:00\n",  
"2023-08-19 00:00:00\n",  
"2023-08-20 00:00:00\n",  
"2023-08-21 00:00:00\n",  
"2023-08-22 00:00:00\n",  
"2023-08-23 00:00:00\n",  
"2023-08-24 00:00:00\n",  
"2023-08-25 00:00:00\n",  
"2023-08-26 00:00:00\n",  
"2023-08-27 00:00:00\n",  
"2023-08-28 00:00:00\n",  
"2023-08-29 00:00:00\n",  
"2023-08-30 00:00:00\n",  
"2023-08-31 00:00:00\n",  
"2023-09-01 00:00:00\n",  
"2023-09-02 00:00:00\n",  
"2023-09-03 00:00:00\n",  
"2023-09-04 00:00:00\n",  
"2023-09-05 00:00:00\n",  
"2023-09-06 00:00:00\n",  
"2023-09-07 00:00:00\n",  
"2023-09-08 00:00:00\n",  
"2023-09-09 00:00:00\n",  
"2023-09-10 00:00:00\n",

```

        "2023-09-11 00:00:00\n",
        "2023-09-12 00:00:00\n",
        "2023-09-13 00:00:00\n",
        "2023-09-14 00:00:00\n",
        "2023-09-15 00:00:00\n",
        "2023-09-16 00:00:00\n",
        "2023-09-17 00:00:00\n",
        "2023-09-18 00:00:00\n",
        "2023-09-19 00:00:00\n",
        "2023-09-20 00:00:00\n",
        "2023-09-21 00:00:00\n",
        "2023-09-22 00:00:00\n",
        "2023-09-23 00:00:00\n",
        "2023-09-24 00:00:00\n",
        "2023-09-25 00:00:00\n",
        "2023-09-26 00:00:00\n",
        "2023-09-27 00:00:00\n",
        "2023-09-28 00:00:00\n",
        "2023-09-29 00:00:00\n",
        "2023-09-30 00:00:00\n",
        "2023-10-01 00:00:00\n",
        "2023-10-02 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": 40,
    "outputs": []
},
{
    "cell_type": "code",
    "source": [
        "import pandas as pd\n",

```

```

        "df = pd.DataFrame(lists)\n",
        "df.columns = ['Col_1', 'Col_2', 'Col_3']\n",
        "print(df, \"\\n\\n\")\n",
        "print(df) "
    ],
    "metadata": {
        "id": "knH76sDKYsVX",
        "colab": {
            "base_uri": "https://localhost:8080/"
        },
        "outputId": "5c1c598-009c-44f6-86fc-b5505673536b"
    },
    "execution_count": 49,
    "outputs": [
        {
            "output_type": "stream",
            "name": "stdout",
            "text": [
                "   Col_1 Col_2 Col_3\n",
                "0      1   aaa   22\n",
                "1      2   bbb   25\n",
                "2      3   ccc   24 \n",
                "\n",
                "   Col_1 Col_2 Col_3\n",
                "0      1   aaa   22\n",
                "1      2   bbb   25\n",
                "2      3   ccc   24\n"
            ]
        }
    ]
}

```