```
{
  "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": null,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "s = \"Hi there Sam!\"\n",
        "string = \"Hi there Sam!\"\n",
        "print(string.split())"
      ],
      "metadata": {
        "id": "6mGVa3SQYLkb",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "efaef93b-b5b6-4060-a78a-68534d5e725e"
      },
      "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",
        "\n",
        "print(\"The diameter of {} is {} kilometers.\".format(planet, diameter))\
n",
        "\n",
        "#or, you can do it like that:\n",
        "\n",
        "print(f\"The diameter of {planet} is {diameter} kilometers.\")"
      ],
      "metadata": {
        "id": "HyRyJv6CYPb4",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "89c5b1ef-71f1-46fa-d1a4-460b6204c0d4"
      },
      "execution_count": null,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "The diameter of Earth is 12742 kilometers.\n",
            "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']}]}]}\n",
        "import numpy as np\n",
        "array=np.arange(20,36,2)\n",
        "print(\"Array of all the even integers from 30 to 70\")\n",
        "print(array)"
      ],
      "metadata": {
        "id": "fcVwbCc1QrQI",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "7735078f-e759-49bb-d68c-c038c58f67a4"
      },
      "execution_count": null,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "Array of all the even integers from 30 to 70\n",
            "[20 22 24 26 28 30 32 34]\n"
          ]
        }
      ]
    },
    {
      "cell_type": "markdown",
      "source": [
        "# Numpy"
      ],
      "metadata": {
        "id": "bw0vVp-9ddjv"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "import numpy as np\n",
        "import numpy as np\n",
        "array=np.arange(20,36,2)\n",
        "print(\"Array of all the even integers from 30 to 70\")\n",
        "print(array)"
      ],
      "metadata": {
        "id": "LLiE_TYrhA1O",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "7568e263-5b87-4632-a97c-3298d77814ee"
      },
      "execution_count": null,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "Array of all the even integers from 30 to 70\n",
            "[20 22 24 26 28 30 32 34]\n"
          ]
        }
      ]
    },
    {
      "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",
        "array=np.zeros(10)\n",
        "print(\"An array of 10 zeros:\")\n",
        "print(array)"
      ],
      "metadata": {
        "id": "NHrirmgCYXvU",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "a5659ae6-e802-4bbe-8977-7d432c734c5d"
      },
      "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": [
    "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": "6f87ee6f-1aa3-4aa0-919c-4cd1e66eec27"
  },
  "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",
    "array=np.arange(20,36,2)\n",
    "print(\"Array of all the even integers from 30 to 70\")\n",
    "print(array)"
  ],
  "metadata": {
    "id": "oAI2tbU2Yag-",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "536ff337-83c5-4854-c8ca-abf5b1821725"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "Array of all the even integers from 30 to 70\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",
    "x =  np.arange(0,9).reshape(3,3)\n",
    "print(x)"
  ],
  "metadata": {
    "id": "tOlEVH7BYceE",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "c1566881-2840-4d8c-e475-3de82cde01fb"
  },
  "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,],[3,4]])\n",
    "b = np.array([[5,6]])\n",
    "np.concatenate((a,b), axis=None)"
  ],
      "metadata": {
        "id": "rAPSw97aYfE0",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "0035bd99-fe82-44b6-b1fd-dded149c21a4"
      },
      "execution_count": null,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "array([1, 2, 3, 4, 5, 6])"
            ]
          },
          "metadata": {},
          "execution_count": 13
        }
      ]
    },
    {
      "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"
      ],
      "metadata": {
        "id": "T5OxJRZ8uvR7"
      },
      "execution_count": null,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "import pandas as pd\n",
        "import pandas as pd\n",
        "\n",
        "data = {\n",
        "  \"calories\": [420, 380, 390],\n",
        "  \"duration\": [50, 40, 45]\n",
        "}\n",
        "\n",
        "#load data into a DataFrame object:\n",
        "df = pd.DataFrame(data)\n",
        "\n",
        "print(df)"
      ],
      "metadata": {
        "id": "xNpI_XXoYhs0",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "b1028b5f-d4a1-4dfc-ed4a-5355d127bef0"
      },
      "execution_count": null,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "   calories  duration\n",
            "0       420        50\n",
            "1       380        40\n",
            "2       390        45\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",
        "\n",
        "  \n",
        "\n",
        "per1 = pd.date_range(start ='01-01-2023',  \n",
        "\n",
        "         end ='02-10-2023') \n",
        "\n",
        "  \n",
        "\n",
        "for val in per1: \n",
        "\n",
        "    print(val)"
      ],
      "metadata": {
        "id": "dgyC0JhVYl4F",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "d94cab11-382e-4665-cedb-b979e9879a2c"
      },
      "execution_count": null,
      "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": "_XMC8aEt0llB"
      },
      "execution_count": null,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "import pandas as pd\n",
        "import numpy as np\n",
        "lists = np.array([[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]])\n",
        "\n",
        "df=pd.DataFrame(lists)\n",
        "\n",
        "print(df)"
      ],
      "metadata": {
        "id": "knH76sDKYsVX",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "592a520e-630d-4fd7-a7fb-54401b18132f"
      },
      "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"
          ]
        }
      ]
    }
  ]
}
```