

Assignment - 1

Assignment Date	09 September 2022
Student Name	KABINI M
Student Roll no	61771931022
Maximum marks	2 marks

```
{
  "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",
        "x=s.split()\n",
        "x\n"
    ],
    "metadata": {
        "id": "6mGVa3SQYLkb",
        "outputId": "099d254b-dbc3-4eec-e77f-2b75d03c434b",
        "colab": {
            "base_uri": "https://localhost:8080/"
        }
    },
    "execution_count": null,
    "outputs": [
        {
            "output_type": "execute_result",
            "data": {
                "text/plain": [
                    "['Hi', 'there', 'Sam!']"
                ]
            },
            "metadata": {},
            "execution_count": 1
        }
    ]
},
{
    "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",
      "s=\\\"The diameter of {} is {} kilometers.\\\"\\n",
      "print(s.format(planet,diameter))"
    ],
    "metadata": {
      "id": "HyRyJv6CYPb4",
      "outputId": "3f6df589-e440-4bd0-9803-f5c17aba43fc",
      "colab": {
        "base_uri": "https://localhost:8080/"
      }
    },
    "execution_count": null,
    "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"
    },
    "execution_count": null,
    "outputs": [
      {
        "cell_type": "code",
        "source": [
          "d =
          {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}",
          "metadata": {
            "id": "fcVwbCc1QrQI"
          },
          "execution_count": null,
          "outputs": []
        ],
        "cell_type": "code",
        "source": [

```

```

        "d =
{'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}
]]}}\n",
        "print(d['k1'][3]['tricky'][3]['target'][3])\n"
    ],
    "metadata": {
        "id": "MvbkMZpXYRaw",
        "outputId": "4c6bb527-a031-4dcf-f8b0-1692c447211d",
        "colab": {
            "base_uri": "https://localhost:8080/"
        }
    },
    "execution_count": null,
    "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",
    "a=np.zeros(10)\n",
    "print(a)\n"
],
"metadata": {
    "id": "NHrirmgCYXvU",
    "outputId": "93985222-77b6-4e73-f884-a8cc3a54be15",
    "colab": {
        "base_uri": "https://localhost:8080/"
    }
},
"execution_count": null,
"outputs": [
    {
        "output_type": "stream",
        "name": "stdout",
        "text": [
            "[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]\n"
        ]
    }
],
},
{
    "cell_type": "code",
    "source": [
        "import numpy as np\n",
        "b=np.ones(10)*5\n",
        "print(b) "
    ],
    "metadata": {
        "id": "e40051sTYXxx",
        "outputId": "4ee3a1ef-34b0-4ee1-8ee0-4a8f703dddb3",
        "colab": {
            "base_uri": "https://localhost:8080/"
        }
    },
    "execution_count": null,
    "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": [
      "import numpy as np\n",
      "a=np.arange(20,35,2)\n",
      "print(a) "
    ],
    "metadata": {
      "id": "oAI2tbU2Yag-",
      "outputId": "d37c2172-7bae-4f43-9a64-9b26d7f06a30",
      "colab": {
        "base_uri": "https://localhost:8080/"
      }
    },
    "execution_count": null,
    "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": [
      "import numpy as np\n",
      "a=np.arange(0,9).reshape(3,3)\n",
      "print(a) "
    ],
    "metadata": {
      "id": "tO1EVH7BYceE",
      "outputId": "bb251802-6c55-4718-f09b-cfadbb7cf5f2",
      "colab": {
        "base_uri": "https://localhost:8080/"
      }
    },
    "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])\n",
    "b=np.array([4,5,6])\n",
    "x=np.concatenate((a,b),axis=None)\n",
    "print(x)"
  ],
  "metadata": {
    "id": "rAPSw97aYfE0",
    "outputId": "71d50026-5138-44db-c5ec-42d43721efbb",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "execution_count": null,
  "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": null,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "import pandas as pd\n",
        "data={ 'Name': ['Binushya', 'Aruna', 'Deepa'], 'Age': ['08', '06', '09']} \n",
        "a=pd.DataFrame(data) \n",
        "print(a) "
      ],
      "metadata": {
        "id": "xNpI_XXoYhs0",
        "outputId": "681201eb-39c1-4353-912d-c64abf9b4879",
        "colab": {
          "base_uri": "https://localhost:8080/"
        }
      },
      "execution_count": null,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "      Name Age\n",
            "0  Binushya  08\n",
            "1    Aruna  06\n",
            "2    Deepa  09\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 datetime\n",
        "import pandas as pd\n",
        "test =
datetime.datetime.strptime(\"01/01/2023\", \"%d/%m/%Y\")\n",
        "k=41\n",
        "dg = pd.date_range(test, periods=k)\n",
        "print(dg.strftime(\"%d/%m/%Y\"))"
    ],
    "metadata": {
        "id": "dgyC0JhVYl4F",
        "outputId": "6de61446-6fc5-4eb3-922d-b310ccef29ef",
        "colab": {
            "base_uri": "https://localhost:8080/"
        }
    },
    "execution_count": null,
    "outputs": [
        {
            "output_type": "stream",
            "name": "stdout",
            "text": [
                "Index(['01/01/2023', '02/01/2023', '03/01/2023',
'04/01/2023', '05/01/2023',\n",
                "      '06/01/2023', '07/01/2023', '08/01/2023',
'09/01/2023', '10/01/2023',\n",
                "      '11/01/2023', '12/01/2023', '13/01/2023',
'14/01/2023', '15/01/2023',\n",
                "      '16/01/2023', '17/01/2023', '18/01/2023',
'19/01/2023', '20/01/2023',\n",
                "      '21/01/2023', '22/01/2023', '23/01/2023',
'24/01/2023', '25/01/2023',\n",
                "      '26/01/2023', '27/01/2023', '28/01/2023',
'29/01/2023', '30/01/2023',\n",
                "      '31/01/2023', '01/02/2023', '02/02/2023',
'03/02/2023', '04/02/2023',\n",
                "      '05/02/2023', '06/02/2023', '07/02/2023',
'08/02/2023', '09/02/2023',\n",
                "      '10/02/2023'],\n",
                "      dtype='object')\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": "_XMC8aEt01lB"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "import pandas as pd\n",
    "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]\n",
    "a = pd.DataFrame(lists, columns=['No', 'Letter', 'Numbers'])\n",
    "print(a)"
  ],
  "metadata": {
    "id": "knH76sDKYsVX",
    "outputId": "856d93fb-5563-4acb-af59-2a71d560790f",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "   No Letter  Numbers\n",
        "0    1   aaa      22\n",
        "1    2   bbb      25\n",
        "2    3   ccc      24"
      ]
    }
  ]
}
]

```

