

Name:- Jigyasu Pant
Reg. No.:-19BEC0760

ASSIGNMENT-1

```
{
  "cells": [
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "fwU2iooz85jt"
      },
      "source": [
        "## Exercises\n",
        "\n",
        "Answer the questions or complete the tasks outlined in bold  
below, use the specific method described if applicable."
      ]
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "SzBQQ_ml85j1"
      },
      "source": [
        "*** What is 7 to the power of 4?*"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "UhvE4PBC85j3",
        "outputId": "a9ad665c-fdb7-421c-e797-334bfd32a667",
        "colab": {
          "base_uri": "https://localhost:8080/"
        }
      },
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "2401\n"
          ]
        }
      ],
      "source": [
        "a=7**4\n",
        "print(a)"
      ]
    },
    {
      "cell_type": "markdown",

```

```

"metadata": {
  "id": "ds8G9S8j85j6"
},
"source": [
  "** Split this string:**\n",
  "\n",
  "    s = \"Hi there Sam!\"\n",
  "    \n",
  "**into a list. **"
]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "collapsed": true,
    "id": "GD_Tls3H85j7"
  },
  "outputs": [],
  "source": [
    "s=\"Hi there Sam\"\n",
    "t=s.split()"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "RRGOKoai85j8",
    "outputId": "53717fad-832f-472e-bbf6-f7f1bdc771ec",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "['Hi', 'there', 'Sam']\n"
      ]
    }
  ],
  "source": [
    "print(t)"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "_bBNOu-785j9"
  },
  "source": [
    "** Given the variables:**\n",

```

```

        "\n",
        "    planet = \"Earth\\\"\\n",
        "    diameter = 12742\\n",
        "\n",
        "*** Use .format() to print the following string: **\\n",
        "\n",
        "    The diameter of Earth is 12742 kilometers."
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
        "collapsed": true,
        "id": "2TrzmDcS85j-"
    },
    "outputs": [],
    "source": [
        "planet = \"Earth\\\"\\n",
        "diameter = 12742"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
        "id": "s_dQ7_xc85j_",
        "outputId": "959fa239-53aa-49f1-c97a-1db23aaca6ee",
        "colab": {
            "base_uri": "https://localhost:8080/"
        }
    },
    "outputs": [
        {
            "output_type": "stream",
            "name": "stdout",
            "text": [
                "The diameter of Earth is 12742\\n"
            ]
        }
    ],
    "source": [
        "s=\"The diameter of {} is {}\".format(planet,diameter)\\n",
        "print(s)"
    ]
},
{
    "cell_type": "markdown",
    "metadata": {
        "id": "QAKtN7Hh85kB"
    },
    "source": [
        "*** Given this nested list, use indexing to grab the word  

        \\\"hello\\\" ***"
    ]
}

```

```

    ]
  },
  {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
      "collapsed": true,
      "id": "-7dzQDyK85kD"
    },
    "outputs": [],
    "source": [
      "lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]"
    ]
  },
  {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
      "id": "6m5C0sTW85kE",
      "outputId": "c3127e99-40f6-41bb-cabe-f1854272addb",
      "colab": {
        "base_uri": "https://localhost:8080/",
        "height": 37
      }
    },
    "outputs": [
      {
        "output_type": "execute_result",
        "data": {
          "text/plain": [
            "'hello'"
          ],
          "application/vnd.google.colaboratory.intrinsic+json": {
            "type": "string"
          }
        },
        "metadata": {},
        "execution_count": 10
      }
    ],
    "source": [
      "lst[3][1][2][0]"
    ]
  },
  {
    "cell_type": "markdown",
    "metadata": {
      "id": "9Ma7M4a185kF"
    },
    "source": [
      "*** Given this nest dictionary grab the word \"hello\". Be prepared, this will be annoying/tricky ***"
    ]
  },
  },

```

```

{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "vrYAxSYN85kG"
  },
  "outputs": [],
  "source": [
    "d =
{'k1':[1,2,3,{ 'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}
]}}]"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "FlILSdm485kH",
    "outputId": "448f2ba6-dd59-4355-9af3-2616e9cd4e6c",
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 37
    }
  },
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "'hello'"
        ],
        "application/vnd.google.colaboratory.intrinsic+json": {
          "type": "string"
        }
      },
      "metadata": {},
      "execution_count": 13
    }
  ],
  "source": [
    "d['k1'][3]['tricky'][3]['target'][3]\n"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "FInV_FKB85kI"
  },
  "source": [
    "*** What is the main difference between a tuple and a list? ***"
  ]
},
{
  "cell_type": "code",

```

```

    "execution_count": null,
    "metadata": {
        "collapsed": true,
        "id": "_VBWf00q85kJ"
    },
    "outputs": [],
    "source": [
        "tuples are immutable objects the lists are mutable"
    ]
},
{
    "cell_type": "markdown",
    "metadata": {
        "id": "zP-j0HZj85kK"
    },
    "source": [
        "*** Create a function that grabs the email website domain from a  

string in the form: **\n",
        "\n",
        "    user@domain.com\n",
        "    \n",
        "***So for example, passing \"user@domain.com\" would return:  

domain.com**"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
        "collapsed": true,
        "id": "unvEAWjk85kL"
    },
    "outputs": [],
    "source": [
        "def fnc1(s):\n",
        "    t=s.split('@');\n",
        "    return(t[1])\n"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
        "id": "Gb9dspLC85kL",
        "outputId": "bb84a8ab-4aa0-4cad-de71-e45821326157",
        "colab": {
            "base_uri": "https://localhost:8080/"
        }
    },
    "outputs": [
        {
            "output_type": "stream",
            "name": "stdout",
            "text": [

```

```

        "domain.com\n"
    ]
}
],
"source": [
    "a=fnc1('user@domain.com')\n",
    "print(a)"
]
},
{
    "cell_type": "markdown",
    "metadata": {
        "id": "gYydb-y085kM"
    },
    "source": [
        "*** Create a basic function that returns True if the word 'dog'
is contained in the input string. Don't worry about edge cases like a
punctuation being attached to the word dog, but do account for
capitalization. ***"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
        "collapsed": true,
        "id": "Q4ldLGV785kM"
    },
    "outputs": [],
    "source": [
        "def fnc2(s):\n",
        "    s=s.lower()\n",
        "    t=s.split(' ')\n",
        "    s1='dog'\n",
        "    if(s1 in t):\n",
        "        return True\n",
        "    else:\n",
        "        return False"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
        "id": "EqH6b7yv85kN",
        "outputId": "f37db3dc-097b-4d53-a4aa-90ca870ddfed",
        "colab": {
            "base_uri": "https://localhost:8080/"
        }
    },
    "outputs": [
        {
            "output_type": "stream",
            "name": "stdout",

```

```

        "text": [
            "True\n"
        ]
    },
    "source": [
        "print(fnc2('the sentence has word DOG IN IT'))"
    ]
},
{
    "cell_type": "markdown",
    "metadata": {
        "id": "AyHQFALC85kO"
    },
    "source": [
        "*** Create a function that counts the number of times the word\n\"dog\" occurs in a string. Again ignore edge cases. ***"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
        "id": "6hdc169585kO"
    },
    "outputs": [],
    "source": [
        "def fnc3(s):\n",
        "    s=s.lower()\n",
        "    t=s.split()\n",
        "    c=t.count('dog')\n",
        "    return c"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
        "id": "igzsvHb385kO",
        "outputId": "f0709b09-1330-4455-db62-9a8192bbabd1",
        "colab": {
            "base_uri": "https://localhost:8080/"
        }
    },
    "outputs": [
        {
            "output_type": "stream",
            "name": "stdout",
            "text": [
                "2\n"
            ]
        }
    ],
    "source": [

```



```

    "print(fnc3('ajhd dog hajbdjh dog ajhbd'))"
]
},
{
    "cell_type": "markdown",
    "metadata": {
        "id": "3n7jJt4k85kP"
    },
    "source": [
        "### Final Problem\n",
        "***You are driving a little too fast, and a police officer stops  

you. Write a function\n",
        "    to return one of 3 possible results: \"No ticket\", \"Small  

ticket\", or \"Big Ticket\". \n",
        "    If your speed is 60 or less, the result is \"No Ticket\". If  

speed is between 61 \n",
        "    and 80 inclusive, the result is \"Small Ticket\". If speed is  

81 or more, the result is \"Big Ticket\". Unless it is your birthday  

(encoded as a boolean value in the parameters of the function) -- on your  

birthday, your speed can be 5 higher in all \n",
        "    cases. ***"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
        "collapsed": true,
        "id": "nvXMkvWk85kQ"
    },
    "outputs": [],
    "source": [
        "def caught_speeding(speed, is_birthday):\n",
        "    \n",
        "    if is_birthday:\n",
        "        speeding = speed - 5\n",
        "    else:\n",
        "        speeding = speed\n",
        "    \n",
        "    if speeding > 80:\n",
        "        return 'Big Ticket'\n",
        "    elif speeding > 60:\n",
        "        return 'Small Ticket'\n",
        "    else:\n",
        "        return 'No Ticket'"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
        "id": "p1AGJ7DM85kR",
        "outputId": "7f8ec0ee-5217-4b81-b295-c3410557da4c",
        "colab": {

```

```

        "base_uri": "https://localhost:8080/",
        "height": 37
    },
    "outputs": [
        {
            "output_type": "execute_result",
            "data": {
                "text/plain": [
                    "'Small Ticket'"
                ],
                "application/vnd.google.colaboratory.intrinsic+json": {
                    "type": "string"
                }
            },
            "metadata": {},
            "execution_count": 23
        },
        {
            "source": [
                "caught_speeding(70,True)"
            ]
        },
        {
            "cell_type": "code",
            "execution_count": null,
            "metadata": {
                "id": "BU_UZcyk85kS",
                "outputId": "6b9ba526-780f-4bfe-84d1-38dfbeb75840",
                "colab": {
                    "base_uri": "https://localhost:8080/",
                    "height": 37
                }
            },
            "outputs": [
                {
                    "output_type": "execute_result",
                    "data": {
                        "text/plain": [
                            "'Big Ticket'"
                        ],
                        "application/vnd.google.colaboratory.intrinsic+json": {
                            "type": "string"
                        }
                    },
                    "metadata": {},
                    "execution_count": 24
                },
                {
                    "source": [
                        "caught_speeding(90,False)"
                    ]
                }
            ],
            "cell_type": "code"
        }
    ]
}

```

```

    "cell_type": "markdown",
    "metadata": {
        "id": "QL7sY6NR85kT"
    },
    "source": [
        "# Great job!"
    ]
}
],
"metadata": {
    "colab": {
        "name": "Assignment-1_python_exercise (1).ipynb",
        "provenance": [],
        "collapsed_sections": []
    },
    "kernelspec": {
        "display_name": "Python 3",
        "language": "python",
        "name": "python3"
    },
    "language_info": {
        "codemirror_mode": {
            "name": "ipython",
            "version": 3
        },
        "file_extension": ".py",
        "mimetype": "text/x-python",
        "name": "python",
        "nbconvert_exporter": "python",
        "pygments_lexer": "ipython3",
        "version": "3.8.5"
    }
},
"nbformat": 4,
"nbformat_minor": 0
}

```