```
{
  "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": 1,
      "metadata": {
        "id": "UhvE4PBC85j3",
```

```
    "outputId": "ee8bf637-f32e-4e26-f161-e290dd873809",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "2401\n"
      ]
    }
  ],
  "source": [
    "def power4(num):\n",
    "  num=num*num*num*num\n",
    "  print(num)\n",
    "\n",
    "power4(7)"
  ]
},
{
  "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": 2,
  "metadata": {
    "collapsed": true,
    "id": "GD_Tls3H85j7",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "ae675f51-a079-4f34-ef1f-310d94624c38"
  },
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "['Hi', 'there', 'sam!']\n"
      ]
    }
  ],
  "source": [
```

```
    "txt=\"Hi there sam!\"\n",
    "x=txt.split()\n",
    "print(x)"
  ]
},
{
  "cell_type": "code",
  "execution_count": 3,
  "metadata": {
    "id": "RRGOKoai85j8",
    "outputId": "cb127b16-9ed2-440b-efe9-90c36539e1b1",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "['Hi', 'there', 'dad!']\n"
      ]
    }
  ],
  "source": [
    "txt=\"Hi there dad!\"\n",
    "y=txt.split()\n",
    "print(y)"
```

```
  ]
},
{
  "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": 7,
  "metadata": {
    "collapsed": true,
    "id": "2TrzmDcS85j-",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "e77b88e2-fbca-43a7-f7f9-b4e3125f41a4"
```

```
  },
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "The diameter of earth is 12742.00 kilometers\n"
      ]
    }
  ],
  "source": [
    "txt=\"The diameter of earth is {diameter:.2f} kilometers\"\n",
    "print(txt.format(diameter=12742))"
  ]
},
{
  "cell_type": "code",
  "execution_count": 8,
  "metadata": {
    "id": "s_dQ7_xc85j_",
    "outputId": "92c68753-add0-4808-df18-f10718acd91c",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "outputs": [
    {
      "output_type": "stream",
```

```
      "name": "stdout",
      "text": [
        "The diameter of earth is 12742.00 kilometers\n"
      ]
    }
  ],
  "source": [
    "txt=\"The diameter of earth is {diameter:.2f} kilometers\"\n",
    "print(txt.format(diameter=12742))"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "QAKtN7Hh85kB"
  },
  "source": [
    "** Given this nested list, use indexing to grab the word \"hello\" **"
  ]
},
{
  "cell_type": "code",
  "execution_count": 35,
  "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": 47,
  "metadata": {
    "id": "6m5C0sTW85kE",
    "outputId": "0fdc1c75-2105-4fd9-edaa-eaf7b3702b76",
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 36
    }
  },
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "'hello'"
        ],
        "application/vnd.google.colaboratory.intrinsic+json": {
          "type": "string"
        }
      },
      "metadata": {},
      "execution_count": 47
```

```
        }
      ],
      "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": 54,
      "metadata": {
        "id": "vrYAxSYN85kG"
      },
      "outputs": [],
      "source": [
        "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}"
      ]
    },
```

```
{
  "cell_type": "code",
  "execution_count": 53,
  "metadata": {
    "id": "FlILSdm485kH",
    "outputId": "575c40f7-a039-4bfe-c8c1-212747826dcc",
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 36
    }
  },
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "'hello'"
        ],
        "application/vnd.google.colaboratory.intrinsic+json": {
          "type": "string"
        }
      },
      "metadata": {},
      "execution_count": 53
    }
  ],
  "source": [
    "d['k1'][3]['tricky'][3]['target'][3]"
```

```
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "FInV_FKB85kI"
  },
  "source": [
    "** What is the main difference between a tuple and a list? **"
  ]
},
{
  "cell_type": "code",
  "execution_count": 68,
  "metadata": {
    "collapsed": true,
    "id": "_VBWf00q85kJ",
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 53
    },
    "outputId": "849ff4d2-66cd-411d-a38b-afa4d9974e61"
  },
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
```

```
              "'\\n1.List is dynamic , Tuple is static characteristic.\\n2.List is mutable , Tuple is Immutable.\\n3.List consume more memory , Tuple comsumes Less memore compared to Tuple.\\n4.List Example: list[v,i,j,a,y] , Tuple Example: tuple(v,i,j,a,y) '"

            ],

            "application/vnd.google.colaboratory.intrinsic+json": {

              "type": "string"

            }

          },

          "metadata": {},

          "execution_count": 68

        }

      ],

      "source": [

        "1.List is dynamic , Tuple is static characteristic.\n",

        "2.List is mutable , Tuple is Immutable.\n",

        "3.List consume more memory , Tuple comsumes Less memore compared to Tuple.\n",

        "4.List Example: list[v,i,j,a,y] , Tuple Example: tuple(v,i,j,a,y)"

      ]

    },

    {

  "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",
        "> Indented block\n",
        "\n",
        "\n",
        "\n",
        "    user@domain.com\n",
        "    \n",
        "**So for example, passing \"user@domain.com\" would return: domain.com** *italicized text*"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 57,
      "metadata": {
        "collapsed": true,
        "id": "unvEAwjk85kL"
      },
      "outputs": [],
      "source": [
        "def domainGet(email):\n",
        "    return email.split('@')[-1]"
      ]
    },
    {
      "cell_type": "code",
```

```
      "execution_count": 58,
      "metadata": {
        "id": "Gb9dspLC85kL",
        "outputId": "49b79ab8-972f-4d7b-9043-abb7c15c42bd",
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 36
        }
      },
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "'domain.com'"
            ],
            "application/vnd.google.colaboratory.intrinsic+json":
{
              "type": "string"
            }
          },
          "metadata": {},
          "execution_count": 58
        }
      ],
      "source": [
        "domainGet('user@domain.com')"
      ]
```

```
    },
    {
      "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": 59,
      "metadata": {
        "collapsed": true,
        "id": "Q4ldLGV785kM"
      },
      "outputs": [],
      "source": [
        "def findDog(st):\n",
        "    return 'dog' in st.lower().split()"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 60,
```

```
  "metadata": {
    "id": "EqH6b7yv85kN",
    "outputId": "586f13c3-f517-445c-e69d-6d71289b59f3",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "True"
        ]
      },
      "metadata": {},
      "execution_count": 60
    }
  ],
  "source": [
    "findDog('Is there a dog here?')"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "AyHQFALC85kO"
  },
```

```
      "source": [
        "** Create a function that counts the number of times the word \"dog\" occurs in a string. Again ignore edge cases. **"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 61,
      "metadata": {
        "id": "6hdc169585kO"
      },
      "outputs": [],
      "source": [
        "def countDog(st):\n",
        "    count = 0\n",
        "    for word in st.lower().split():\n",
        "        if word == 'dog':\n",
        "            count += 1\n",
        "    return count"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 62,
      "metadata": {
        "id": "igzsvHb385kO",
        "outputId": "c6738daf-cbda-4cac-b950-d335d01302ad",
        "colab": {
```

```
            "base_uri": "https://localhost:8080/"
          }
        },
        "outputs": [
          {
            "output_type": "execute_result",
            "data": {
              "text/plain": [
                "2"
              ]
            },
            "metadata": {},
            "execution_count": 62
          }
        ],
        "source": [
          "countDog('This dog runs faster than the other dog
dude!')"
        ]
      },
      {
        "cell_type": "markdown",
        "metadata": {
          "id": "3n7jJt4k85kP"
        },
        "source": [
          "### 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": 63,
      "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": "BU_UZcyk85kS",
    "outputId": "699de8ef-a18c-436b-fdd9-60dc44979906"
  },
  "outputs": [
    {
      "data": {
        "text/plain": [
          "'Big Ticket'"
        ]
      },
      "execution_count": 6,
      "metadata": {
        "tags": []
      },
      "output_type": "execute_result"
    }
```

```
    ],
    "source": [
      "caught_speeding(81,False)"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": 64,
    "metadata": {
      "id": "p1AGJ7DM85kR",
      "outputId": "06a9e18c-3d2b-48c5-967e-aada701f74e0",
      "colab": {
        "base_uri": "https://localhost:8080/",
        "height": 36
      }
    },
    "outputs": [
      {
        "output_type": "execute_result",
        "data": {
          "text/plain": [
            "'Small Ticket'"
          ],
          "application/vnd.google.colaboratory.intrinsic+json":
{
            "type": "string"
          }
        },
```

```
      "metadata": {},
      "execution_count": 64
    }
   ],
   "source": [
    "caught_speeding(81,True)"
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "Create an employee list with basic salary values(at least 5 values for 5 employees)  and using a for loop retreive each employee salary and calculate total salary expenditure. "
   ],
   "metadata": {
    "id": "Tie4rC7_kAOC"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "def employee_details(name,emp_id,ctc,designation,exp):\n",
    "  print(\"\\n Name:-\",name)\n",
    "  print(\"Emp_ID:-\",emp_id)\n",
    "  print(\"CTC:-\",ctc)\n",
    "  print(\"Designation:-\",designation)\n",
    "  print(\"Exprience:-\",exp)\n",
```

```
        "employee_details(\"
Vijay\",421319104045,\"6.6LPA\",\"BE-CSE IV Year\",\"5
years\")\n",

        "employee_details(\"Ajay\",421319104046,\"6.5LPA\",\"BE-
CSE IV Year\",\"4 years\")\n",

        "employee_details(\"Ajith\",421319104047,\"6.4LPA\",\"BE-
CSE IV Year\",\"3 years\")\n",


"employee_details(\"Madhavan\",421319104048,\"6.3LPA\",\"BE-CSE
IV Year\",\"2 years\")\n",

        "employee_details(\"John\",421319104049,\"6.2LPA\",\"BE-
CSE IV Year\",\"1 years\")\n",

        "Sal_exp=[6.6+6.5+6.4+6.3+6.2]\n",

        "Salary=0\n",

        "for i in Sal_exp:\n",

        "  Salary=Salary+i\n",

        "  print(\"\\n Total Salary Expenditure is:-\", Salary,
\"LPA\")"

      ],

      "metadata": {

        "id": "R5-CdXSKjacN",

        "colab": {

          "base_uri": "https://localhost:8080/"

        },

        "outputId": "f140813a-fc90-4637-df6a-e098704bdcee"

      },

      "execution_count": 65,

      "outputs": [

        {

          "output_type": "stream",
```

```
"name": "stdout",
"text": [
  "\n",
  " Name:-  Vijay\n",
  "Emp_ID:- 421319104045\n",
  "CTC:- 6.6LPA\n",
  "Designation:- BE-CSE IV Year\n",
  "Exprience:- 5 years\n",
  "\n",
  " Name:- Ajay\n",
  "Emp_ID:- 421319104046\n",
  "CTC:- 6.5LPA\n",
  "Designation:- BE-CSE IV Year\n",
  "Exprience:- 4 years\n",
  "\n",
  " Name:- Ajith\n",
  "Emp_ID:- 421319104047\n",
  "CTC:- 6.4LPA\n",
  "Designation:- BE-CSE IV Year\n",
  "Exprience:- 3 years\n",
  "\n",
  " Name:- Madhavan\n",
  "Emp_ID:- 421319104048\n",
  "CTC:- 6.3LPA\n",
  "Designation:- BE-CSE IV Year\n",
  "Exprience:- 2 years\n",
  "\n",
  " Name:- John\n",
```

```
            "Emp_ID:- 421319104049\n",
            "CTC:- 6.2LPA\n",
            "Designation:- BE-CSE IV Year\n",
            "Exprience:- 1 years\n",
            "\n",
            " Total Salary Expenditure is:- 32.0 LPA\n"
          ]
        }
      ]
    },
    {
      "cell_type": "markdown",
      "source": [
        "Create two dictionaries in Python:\n",
        "\n",
        "First one to contain fields as Empid,  Empname, Basicpay\n",
        "\n",
        "Second dictionary to contain fields as DeptName, DeptId.\n",
        "\n",
        "Combine both dictionaries. "
      ],
      "metadata": {
        "id": "-L1aiFqRkF5s"
      }
    },
    {
```

```
      "cell_type": "code",
      "source": [
"dict1={\"EmpID\":5,\"EmpName\":'Vijay',\"BasicPay\":'5.5CTC'}\n",
        "dict2={\"DeptName\":'CSE',\"DeptID\":104}\n",
        "print(dict1)\n",
        "print(dict2)"
      ],
      "metadata": {
        "id": "8ugVoEe0kOsk",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "4a7f3ed6-4b02-44dd-9086-56afaf536df9"
      },
      "execution_count": 66,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "{'EmpID': 5, 'EmpName': 'Vijay', 'BasicPay':
'5.5CTC'}\n",
            "{'DeptName': 'CSE', 'DeptID': 104}\n"
          ]
        }
      ]
    },
```

```
    {
      "cell_type": "code",
      "source": [
        "def Merge(dict1, dict2):\n",
        "  return(dict2.update(dict1))\n",

"dict1={\"EmpID\":5,\"EmpName\":'Vijay',\"BasicPay\":'5.5CTC'}\n"
,
        "dict2={\"DeptName\":'CSE',\"DeptID\":104}\n",
        "print(Merge(dict1,dict2))\n",
        "print(dict2)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "9hURfBDJkaCZ",
        "outputId": "3ebdb317-ec7a-496e-fbf9-ee2b1dc346cd"
      },
      "execution_count": 67,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "None\n",
            "{'DeptName': 'CSE', 'DeptID': 104, 'EmpID': 5,
'EmpName': 'Vijay', 'BasicPay': '5.5CTC'}\n"
          ]
```

```
      }
    ]
  }
],
"metadata": {
  "colab": {
    "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":

}
```