```
{
 "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"
   },
   "outputs": [],
   "source": []
  },
  {
   "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": []
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "RRGOKoai85j8"
   },
   "outputs": [],
   "source": []
  },
  {
   "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": []
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "s_dQ7_xc85j_"
   },
   "outputs": [],
   "source": []
  },
  {
   "cell_type": "markdown",
   "metadata": {
        "id": "QAKtN7Hh85kB"
      },
      "source": [
        "** Given this nested list, use indexing to grab the word \"hello\" **"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 1,
      "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": 1,
      "metadata": {
        "id": "6m5C0sTW85kE"
      },
      "outputs": [],
      "source": []
    },
    {
      "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": 2,
      "metadata": {
        "id": "vrYAxSYN85kG"
      },
      "outputs":                 [],
  "source": [
        "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 2,
      "metadata": {
```

```
    "id": "FlILSdm485kH"
   },
   "outputs": [],
   "source": [
    "\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": 2,
   "metadata": {
    "collapsed": true,
    "id": "_VBWf00q85kJ"
   },
   "outputs": [],
   "source": []
  },
  {
   "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": 2,
   "metadata": {
    "collapsed": true,
    "id": "unvEAwjk85kL"
   },
   "outputs": [],
   "source": []
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "metadata": {
    "id": "Gb9dspLC85kL"
   },
   "outputs": [],
   "source": []
  },
  {
   "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": 2,
   "metadata": {
    "collapsed": true,
    "id": "Q4ldLGV785kM"
   },
   "outputs": [],
   "source": []
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "metadata": {
    "id": "EqH6b7yv85kN"
   },
   "outputs": [],
   "source": []
  },
  {
   "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 cas
es. **"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "metadata": {
    "id": "6hdc169585kO"
```

```
      },
      "outputs": [],
      "source": []
    },
    {
      "cell_type": "code",
      "execution_count": 2,
      "metadata": {
        "id": "igzsvHb385kO"
      },
      "outputs": [],
      "source": []
    },
    {
      "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 i
s 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": 4,
      "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": 4,
   "metadata": {
    "id": "BU_UZcyk85kS"
   },
   "outputs": [],
   "source": []
  },
  {
   "cell_type": "code",
   "execution_count": 4,
   "metadata": {
    "id": "p1AGJ7DM85kR"
   },
   "outputs": [],
   "source": []
  },
  {
   "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": [],
   "metadata": {
    "id": "R5-CdXSKjacN"
   },
   "execution_count": 4,
   "outputs": []
  },
  {
   "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": [],
      "metadata": {
        "id": "8ugVoEe0kOsk"
      },
      "execution_count": 4,
      "outputs": []
    }
  ],
  "metadata": {
    "colab": {
      "provenance": [],
      "toc_visible": true
    },
    "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
}
```