

```
{
  "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."
        ]
      },
      "output_id": "a05565aa-db43-4716-e87d-41c5c8a6f95e",
      "outputs": [
        {
          "data": {
            "text/plain": "2401"
          },
          "execution_count": 1,
          "metadata": {
            "tags": []
          },
          "output_type": "execute_result"
        }
      ],
      "source": []
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "ds8G9S8j85j6",
        "source": [
          "** Split this string:\n",
          "\n",
          "s = \"Hi there Sam!\"\n",
          "\n",
          "**into a list. **"
        ]
      },
      "output_id": "cc52f0d8-2ed1-4b4d-e956-5bbeb332cdc2",
      "outputs": [
        {
          "data": {
            "text/plain": "['Hi', 'there', 'dad!']"
          },
          "execution_count": 3,
          "metadata": {
            "tags": []
          },
          "output_type": "execute_result"
        }
      ],
      "source": []
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "_bBN0u-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."
        ]
      },
      "output_id": "cc52f0d8-2ed1-4b4d-e956-5bbeb332cdc2",
      "outputs": [
        {
          "data": {
            "text/plain": "The diameter of Earth is 12742 kilometers."
          },
          "execution_count": 4,
          "metadata": {
            "tags": []
          },
          "output_type": "execute_result"
        }
      ],
      "source": []
    }
  ]
}
```

```
{"collapsed":true,"id":"2TrzmDcS85j-  
"}, "outputs":[], "source":[]},  
{"cell_type":"code", "execution_count":null, "metadata":  
{"id":"s_dQ7_xc85j_", "outputId":"4235fdfb-  
5591-4dd9-f9d2-77f311977633"}, "outputs":  
[{"name":"stdout", "output_type":"stream", "text":["The diameter of Earth is 12742  
kilometers.\n"]}], "source":[]},  
{"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":["1st = [1,2,[3,4],[5,  
[100,200,['hello']],23,11],1,7]"]},  
{"cell_type":"code", "execution_count":null, "metadata":  
{"id":"6m5C0sTW85kE", "outputId":"c3417d1c-  
3081-4e24-8489-154cdce1b06b"}, "outputs":  
[{"data":{"text/plain":  
"['hello']"}}, {"execution_count":14, "metadata":  
{"tags":  
[]}, {"output_type":"execute_result"}], "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":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":"4232540d-  
5c2-461d-c78d-24ea62398e08"}, "outputs":  
[{"data":{"text/plain":  
"['hello']"}}, {"execution_count":16, "metadata":  
{"tags":
```

```
[{}], "output_type": "execute_result"}], "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": null, "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": null, "metadata": {"collapsed": true, "id": "unvEAWjk85kL"}, "outputs": [], "source": []}, {"cell_type": "code", "execution_count": null, "metadata": {"id": "Gb9dspLC85kL", "outputId": "4216116b-da08-45a2-9545-d6b13bcefaeb"}, "outputs": [{"data": {"text/plain": ["'domain.com'"]}, "execution_count": 26, "metadata": {"tags": []}, "output_type": "execute_result"}], "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": null, "metadata": {"collapsed": true, "id": "Q4ldLGV785kM"}, "outputs": [], "source": []}, {"cell_type": "code", "execution_count": null, "metadata": {"id": "Q4ldLGV785kM"}, "outputs": [], "source": []}]
```

```
[{}], "output_type": "execute_result"}, {"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 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", "        speed -= 5\n", "    if speed <= 60:\n", "        return \"No ticket\"\n", "    elif speed <= 80:\n", "        return \"Small ticket\"\n", "    else:\n", "        return \"Big ticket\""]}]
```



```

speeding = speed - 5\n", "
else:\n", "          speeding = speed\n", "
\n", "          if speeding > 80:\n", "
return 'Big Ticket'\n", "          elif speeding >
50:\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
": []},
{"cell_type": "code", "execution_count": null, "
metadata":
{"id": "p1AGJ7DM85kR", "outputId": "ca80629f-
5949-4926-8d27-1b61576669ac"}, "outputs":
[{"data": {"text/plain": ["'Small
Ticket'"]}, "execution_count": 5, "metadata":
{"tags":
[]}, "output_type": "execute_result"}], "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 retrieve each employee
salary and calculate total salary
expenditure. "], "metadata":
{"id": "Tie4rC7_kAOC"}},
{"cell_type": "code", "source": [], "metadata":
{"id": "R5-
CdXSKjacN"}, "execution_count": null, "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":

```

```
{  
  "cell_type": "code",  
  "id": "8ugVoEe0k0sk",  
  "execution_count": null,  
  "outputs": [],  
  "metadata": {"colab": {"provenance": []},  
    "kernel_spec": {"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  
}
```