**Planetary quiz**

You will create a Python program that asks the user three questions about the order of the planets in our solar system. Each question will be of a different type.

Task 1 . Ask for a planet’s position

For the first question, your program should randomly select a planet, display its name, and ask the user for its position in the solar system, with respect to the Sun.

|  |  |
| --- | --- |
| **Example: Correct answer** | |
| Note: This example illustrates how your program should work. The output of your program will depend on the randomly-selected planet and the user’s input, so it will be different each time you execute it. | |
| The program displays a prompt and waits for keyboard input | What is the position of Earth, relative to the Sun? |
| The user types in a reply | 3 |
| The program displays a message that the user’s answer is correct | That is correct.  Earth is planet number 3 from the Sun. |

|  |  |
| --- | --- |
| **Example: Incorrect answer** | |
| Note: This example illustrates how your program should work. The output of your program will depend on the randomly-selected planet and the user’s input, so it will be different each time you execute it. | |
| The program displays a prompt and waits for keyboard input | What is the position of Mars, relative to the Sun? |
| The user types in a reply | 5 |
| The program displays a message that the user’s answer is incorrect, along with the correct answer | That is not correct.  Mars is planet number 4 from the Sun. |

**Checklist**: Tick (✔) the corresponding box if your program:

|  |  |  |
| --- | --- | --- |
| |  | | --- | |  | | Randomly selects a planet for the first question. |
| |  | | --- | |  | | Displays the name of the randomly-selected planet and asks the user for its position in the solar system (see examples). |
| |  | | --- | |  | | Displays a message that informs the user whether or not the answer was correct. |
| |  | | --- | |  | | Displays the correct answer (even when the user’s answer was correct; see examples). |

Task 2 . Ask for a planet’s name

For the second question, your program should randomly select a planet, display its position in the solar system, and ask for the planet’s name.

|  |  |
| --- | --- |
| **Example: Correct answer** | |
| Note: This example illustrates how your program should work. The output of your program will depend on the randomly-selected planet and the user’s input, so it will be different each time you execute it. | |
| The program displays a prompt and waits for keyboard input | What is the name of planet number 3 from the Sun? |
| The user types in a reply | Earth |
| The program displays a message that the user’s answer is correct | That is correct.  Earth is planet number 3 from the Sun. |

|  |  |
| --- | --- |
| **Example: Incorrect answer** | |
| Note: This example illustrates how your program should work. The output of your program will depend on the randomly-selected planet and the user’s input, so it will be different each time you execute it. | |
| The program displays a prompt and waits for keyboard input | What is the name of planet number 4 from the Sun? |
| The user types in a reply | Jupiter |
| The program displays a message that the user’s answer is incorrect, along with the correct answer | That is not correct.  Mars is planet number 4 from the Sun. |

**Checklist**: Tick (✔) the corresponding box if your program:

|  |  |  |
| --- | --- | --- |
| |  | | --- | |  | | Randomly selects a planet for the second question. |
| |  | | --- | |  | | Displays the position of the randomly-selected planet in the solar system and asks for the planet’s name (see examples). |
| |  | | --- | |  | | Displays a message that informs the user whether or not the answer was correct. |
| |  | | --- | |  | | Displays the correct answer (even when the user’s answer was correct; see examples). |

Task 3 . Ask for a planet’s successor

For the third question, your program should randomly select a planet, display its name, and ask for the name of the planet that comes after it.

|  |  |
| --- | --- |
| **Example: Correct answer** | |
| Note: This example illustrates how your program should work. The output of your program will depend on the randomly-selected planet and the user’s input, so it will be different each time you execute it. | |
| The program displays a prompt and waits for keyboard input | The name of the planet that comes after Venus is... |
| The user types in a reply | Earth |
| The program displays a message that the user’s answer is correct | That is correct.  Earth comes after Venus. |

|  |  |
| --- | --- |
| **Example: Incorrect answer** | |
| Note: This example illustrates how your program should work. The output of your program will depend on the randomly-selected planet and the user’s input, so it will be different each time you execute it. | |
| The program displays a prompt and waits for keyboard input | The name of the planet that comes after Earth is... |
| The user types in a reply | Jupiter |
| The program displays a message that the user’s answer is incorrect, along with the correct answer | That is not correct.  Mars comes after Earth. |

**Checklist**: Tick (✔) the corresponding box if your program:

|  |  |  |
| --- | --- | --- |
| |  | | --- | |  | | Randomly selects a planet for the third question, making sure it’s not the last planet in the solar system. |
| |  | | --- | |  | | Displays the name of the randomly-selected planet and asks for the name of the planet that comes after it (see examples). |
| |  | | --- | |  | | Displays a message that informs the user whether or not the answer was correct. |
| |  | | --- | |  | | Displays the correct answer (even when the user’s answer was correct; see examples). |

Task 4 . Display the number of correct answers

After the three questions have been posed, the program should display the number of answers that were correct.

|  |  |
| --- | --- |
| **Example** | |
| Note: This example illustrates how your program should work. The output of your program will depend on the number of correct answers the user provides, so it will be different each time you execute it. | |
| At the end of the game, the program displays the number of correct answers | You got 2 out of 3 answers right. |

**Checklist**: Tick (✔) the corresponding box if your program:

|  |  |  |
| --- | --- | --- |
| |  | | --- | |  | | Displays the number of questions the user has answered correctly (see example). |

Explorer task . A ‘guess my planet’ game

Take a look at [this game](https://ncce.io/py-cities-2) (ncce.io/py-cities-2) where the user is prompted to guess a randomly-selected city. When a guessing attempt fails, the program provides additional hints to the user.

Create a similar game where the user tries to guess a randomly-selected planet, instead of a city. Think of the hints that your program could provide to the user.

Clues . Look here if you need help

**What are the variables I will need?**

Think about the data or the values that your program will need to keep track of. You will certainly need a list of the planets:

|  |  |
| --- | --- |
|  | planets = ["Mercury", "Venus", "Earth",  "Mars", "Jupiter", "Saturn",  "Uranus", "Neptune"] |

For each question, your program will randomly select a planet and you will need variables for the name of the planet, as well as its index or position in the solar system. You will also need one variable (or more) to assign the user’s answer to each question.

**How do I randomly select a planet?**

The simplest way is to use the randint function, from the random module, to randomly select the *index* of the planet. Remember to import the function, at the beginning of the program.

|  |  |
| --- | --- |
|  | index = randint(0, 7) |

The index will need to range from 0 to 7, as there are 8 planets and indices in lists are zero-based. However, the user of the program will count planets and their positions starting at one, so your program will need to take that into account.

For a given index, you can retrieve the corresponding planet from the list of planets:

|  |  |
| --- | --- |
|  | planet = planets[index] |

Resources are updated regularly — the latest version is available at: [ncce.io/tcc](http://ncce.io/tcc).

This resource is licensed under the Open Government Licence, version 3. For more information on this licence, see [ncce.io/ogl](http://ncce.io/ogl).