3.3.5 Making requests using Python
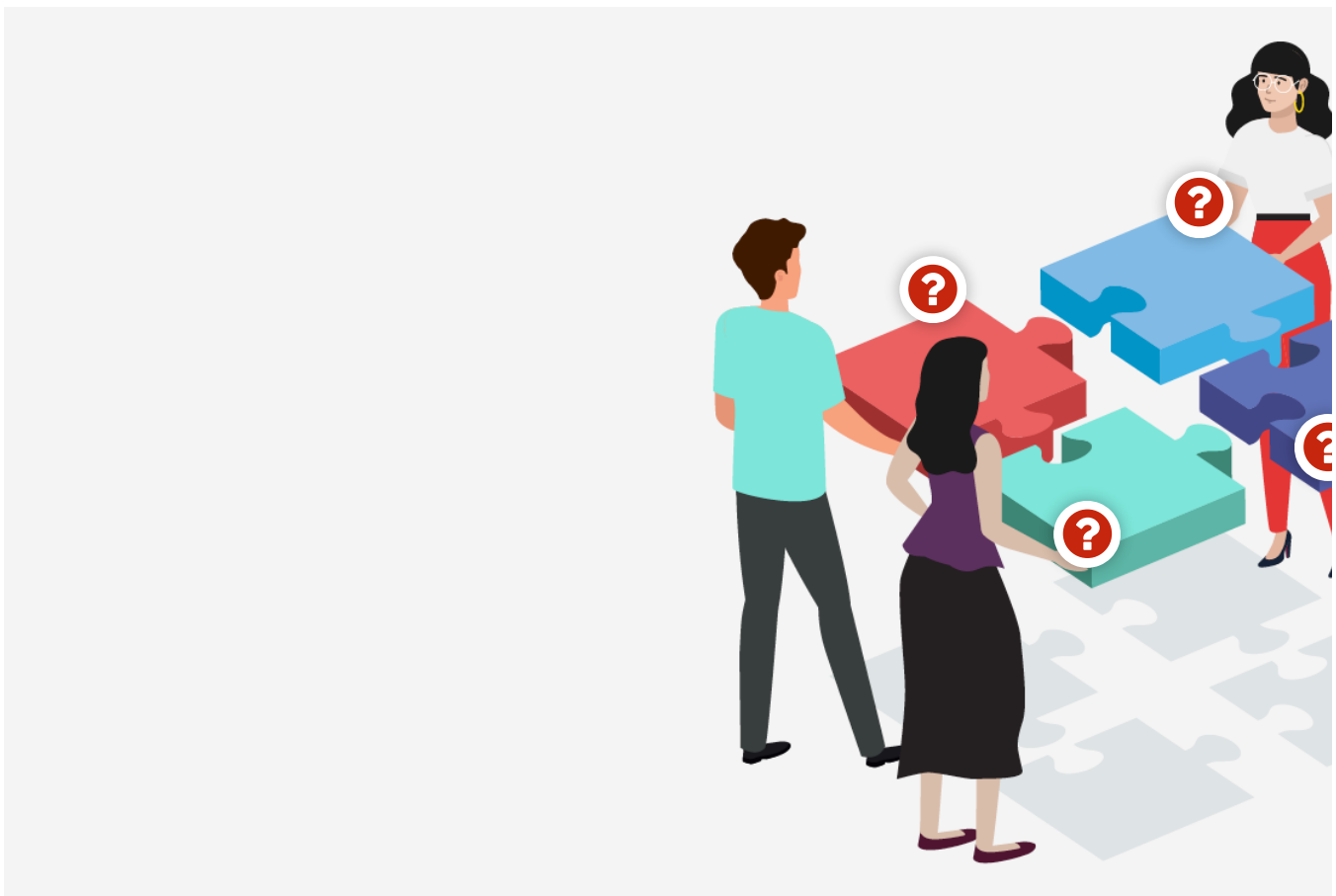
While `GET` requests **retrieve** information, `POST` requests **submit** it to be processed by a server. `POST` requests can create a new resource with the data that is specified in the request body. If you've submitted data in a survey online, you may have unknowingly submitted data using a `POST` request.

The `POST` method requests a dynamic resource (resources that are constantly modified), so it submits large amounts of data as parameters to be processed. The parameter can contain any amount of text or even a file, meaning the `POST` method can be used to upload large text and binary files (Gupta, R 2002).

Although less fundamental, the other HTTP methods are no less useful to your API-efforts. You can explore and deepen your understanding of these methods using **JSONPlaceholder** **(https://jsonplaceholder.typicode.com/)** —a free online REST API that you can use to perform HTTP requests that doesn't affect any data. It is a great way to get data for your application when being developed locally. Now, take a look at the remaining request methods by selecting an icon.

Apply your learning

Head over to **Coding Rooms** ⤴ **(https://app.codingrooms.com/app/org/swinburne-online-NHciF9n/course/cos60016-q1-programming-for-development-2023-piKGheK/b/3.3.5%2BMaking%2Brequests%2Busing%2BPythonNTSKC0G)** to learn how to use GET and POST requests to interact with web servers.

Reflect

Under what circumstances would you use a `POST` request? How is using a `POST` request like hacking? Consider the statement that "When you execute a `POST` or `DELETE` request on an API, you are essentially hacking a resource". Is that an accurate statement, given what you know about `POST` requests?