Timothy Jelinek

The feature I would like to remove from Python is the eval() function. The eval() function is commonly used to evaluate expressions from an input. When using a string input, the function parses it, compiles it to bytecode, then evaluates it. If you input a compiled code object, then the function enacts the evaluation step. I would remove it because of the security risk behind using it. It is dangerous to use eval() because when using it, you don't know what code will be executed, which can be a security risk if you encounter someone with malicious intent who has access to your application. Programmers normally avoid this function because of its danger; however, you can use it if you don't use an untrusted input. It is tough to know what inputs you can trust being used in it. An example of using eval() irresponsibly is using it to build an online service that evaluates Python expressions. The application runs on your server, which has all your files. A user could input a line of code into the application to delete all your files. A potential problem when removing eval() may be that it will break any premade code that depends on it. This will lead to legacy code being unable to be used. Overall, the eval() function could be helpful if it were not for the security risks. I can see how it could dynamically evaluate Python expressions from different input forms.

Source:

"Python eval(): Evaluate Expressions Dynamically" Ramos, https://realpython.com/python-eval-function/#understanding-pythons-eval. Accessed 29 October 2023.