

The statements which use quotes in our paper come from the responses to the open-ended questions in our interview.

We paste a part of useful responses here:

1. "Duck typing is a widely-used Python idiom, and I think most Python developers would consider it a feature. Related to this, Python tends to encourage `try`-ing operations and catching the exceptions, over writing code that seeks to avoid exceptions in the first place."

referred in page 8

2. "Dynamic attribute access is really error prone, a class (type) should never have to visit an attribute via reflection, the only valid reason is when you write meta code."

referred in page 7

3. "I'm a functional programming aficionado, so some of the things I do in Python is not pythonic at all, however I think it reduces many of the downsides of python (immutability, higher order functions, composability etcetc)"

referred in page 6

4. "We also had an even harder restriction on inconsistent assignment types: A thing cannot be two things at once so never use the same name for something else (even if it's the same type), if you have done something with it it's not the same thing and therefore should have the same name."

referred in page 8

5. "Python lacks proper support for sum types and matching the 'enum' so it can be a bit error prone having the 'if statements' similar to Inconsistent Variable Types to handle different types in the arguments. However if you have the type contract explicitly in the documentation it's should be fairly straight forward to check that all cases are covered in code review (and if it's not the code is to complex and you should go back to the drawing board)"

referred in page 9

6. "These are all downsides of dynamic typing languages that prove that strong testing is required to validate the program. Test, test, test. Keep test coverage high."

referred in page 8

7. "One of the things is always thinking about when doing Python is the type of variables and have a compact and efficient syntax to convey the even more complex types in the documentation of a method. At my last company we defined sum types with the `|` operator, and list/array with `[T]`

in the documentation. We didn't have a hard requirement on documentation but we had a hard requirement on defining the types of arguments and return values of a function.”

referred in page 9