Richard D. Moores wrote:
> *But could someone give me a clearcut example of a semantic error? And*
> *a definition that delineates semantic errors from syntax errors.*

"Semantics" relates to the *meaning* of words, sentences or programs. In common English, we might say this sentence has a few grammatical errors, but the semantics are clear:

"I getted the milk out off the fridge and putted them into me coffee."

On the other hand, these sentences are grammatically fine but semantically ambiguous:

"Children make nutritious snacks."
"The thief was sentenced to six months in the violin case."
"Cocaine users are turning to ice."
"Police shoot man with crossbow."
"The building workers are refusing to work after fatal accidents."

and of course the classic example of a grammatically valid sentence with no semantic meaning:

"Colourless green ideas sleep furiously."

Often you can guess the meaning of such ambiguous sentences from domain specific knowledge. We know that eating children is generally frowned upon, and so we reject the interpretation of the snacks being made *from* the children rather than *by* the children.

Other times it is much harder to resolve the ambiguity:

"The English history teacher marked the test paper."

Did she, or he, teach English history, or was she English and a history teacher?

Bringing it back to programming in general, and Python specifically, we don't talk about "grammar errors" but "syntax errors" instead. The two are, more or less, analogous. In principle you could design a compiler to try guessing what you probably meant when faced with syntax errors:

```
x, y = alist[1;3]
```

probably is a typo of ; instead of :

but such cases are far too rare to be worth worrying about, and in any case, "Do what I mean" (DWIM) tools are risky and rarely do what you mean.

http://www.catb.org/~esr/jargon/html/D/DWIM.html

A *semantic* error would be something like this:

```
x, y = alist[1:2]
```

when what you actually needed was alist[1:3]. Or:

```
mylist = mylist.sort()
```

(did you really want mylist to be set to None? I don't think so.)

It's rare that the compiler can tell what you want, as opposed to what you asked for, with any accuracy. How could the compiler tell that when you wrote:

```
x = math.tan(1.25)
```

you actually wanted math.atan instead? Or should that be math.tan(1.25*pi)? Or something else?

--
Steven