IDENTIFIER NAMES

Rules and Conventions

Identifier names

are case-sensitive my_var are different identifiers my_Var ham Ham

must follow certain rules

should follow certain conventions

```
Must
```

```
start with underscore (_) or letter (a-z A-Z)
followed by any number of underscores (_), letters (a-z A-Z), or digits (0-9)
var my_var index1 index_1 var val are all legal names
```

cannot be reserved words:

None	True	False			
and	or	not			
if	else	elif			
for	while	break	continue	pass	
def	lambda	global	nonlocal	return	yield
del	in	is	assert	class	
try	except	finally	raise		
import	from	with	as		

Conventions

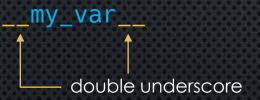
_my_var ____ single underscore This is a <u>convention</u> to indicate "internal use" or "private" objects

Objects named this way will not get imported by a statement such as:

from module import *

_my_var ____ double underscore

Used to "mangle" class attributes – useful in inheritance chains



Used for system-defined names that have a special meaning to the interpreter.

Don't invent them, stick to the ones pre-defined by Python!

__init___
x < y → x.__lt__(y)

Other Naming Conventions

from the PEP 8 Style Guide

Packages short, all-lowercase names. Preferably no underscores.

Modules short, all-lowercase names. Can have underscores. db_utils dbutils

Classes CapWords (upper camel case) convention <u>BankAccount</u>

Functions lowercase, words separated by underscores (snake_case) open_account

Variables lowercase, words separated by underscores (snake_case) account_td

Constants all-uppercase, words separated by underscores MIN_APR

https://www.python.org/dev/peps/pep-0008/ This is a should-read!

A foolish consistency is the hobgoblin of little minds (Emerson)