

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` `__var` `__lt__` 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

`__my_var`

↑
double underscore

`__my_var__`

↑ ↑
double underscore

This is a convention to indicate "internal use" or "private" objects

Objects named this way will not get imported by a statement such as :
`from module import *`

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.

`utilities`

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

`db_utils` `dbutils`

Classes CapWords (upper camel case) convention

`BankAccount`

Functions lowercase, words separated by underscores (snake_case)

`open_account`

Variables lowercase, words separated by underscores (snake_case)

`account_id`

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)