Object Internals and Custom Attributes

Robert Smallshire
COFOUNDER – SIXTY NORTH
@robsmallshire rob@sixty-north.com

Summary

Summary

Summary

Internal object representation

Summary

Internal object representation

The `__dict__` dictionary

Summary

Internal object representation

The `__dict__` dictionary

Manipulating `__dict__`

Summary

Internal object representation

The `__dict__` dictionary

Manipulating `__dict__`

Faking attributes with

Summary

Internal object representation

The `__dict__` dictionary

Manipulating `__dict__`

Faking attributes with

- `__getattr__()`

Summary

Internal object representation

The `__dict__` dictionary

Manipulating `__dict__`

Faking attributes with

- `__getattr__()`
- `__getattribute__()`

Summary

Internal object representation

The `__dict__` dictionary

Manipulating `__dict__`

Faking attributes with

- `__getattr__()`
- `__getattribute__()`
- `__setattr__()`

Summary

Internal object representation

The `__dict__` dictionary

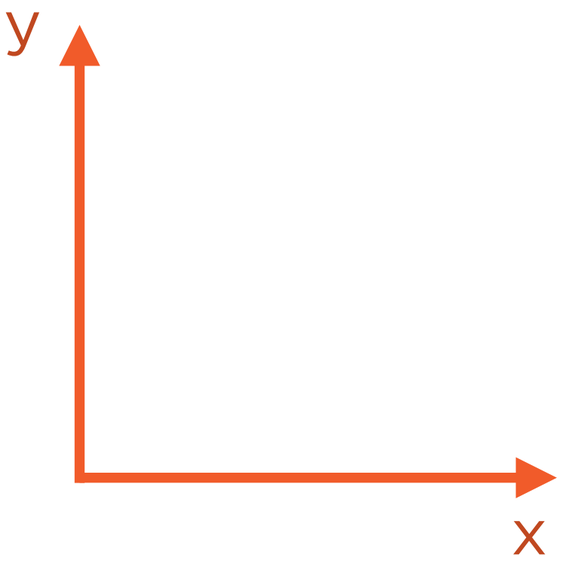
Manipulating `__dict__`

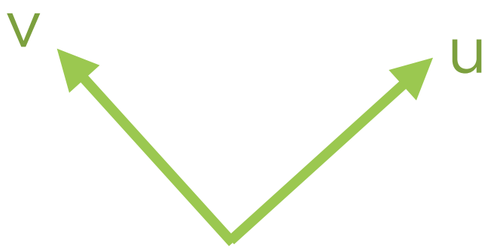
Faking attributes with

- `__getattr__()`
- `__getattribute__()`
- `__setattr__()`
- `__delattr__()`

How are Python Objects Represented?

The convenience of different notations for different coordinate systems





Overriding `__getattr__()`

Customising **Attribute Access**

Customising **Attribute Access**

Fallback

`__getattr__()`

Invoked **after** requested attribute/ property not found by normal lookup

Customising **Attribute Access**

Fallback

`__getattr__()`

Invoked **after** requested attribute/ property not found by normal lookup

Preemptive

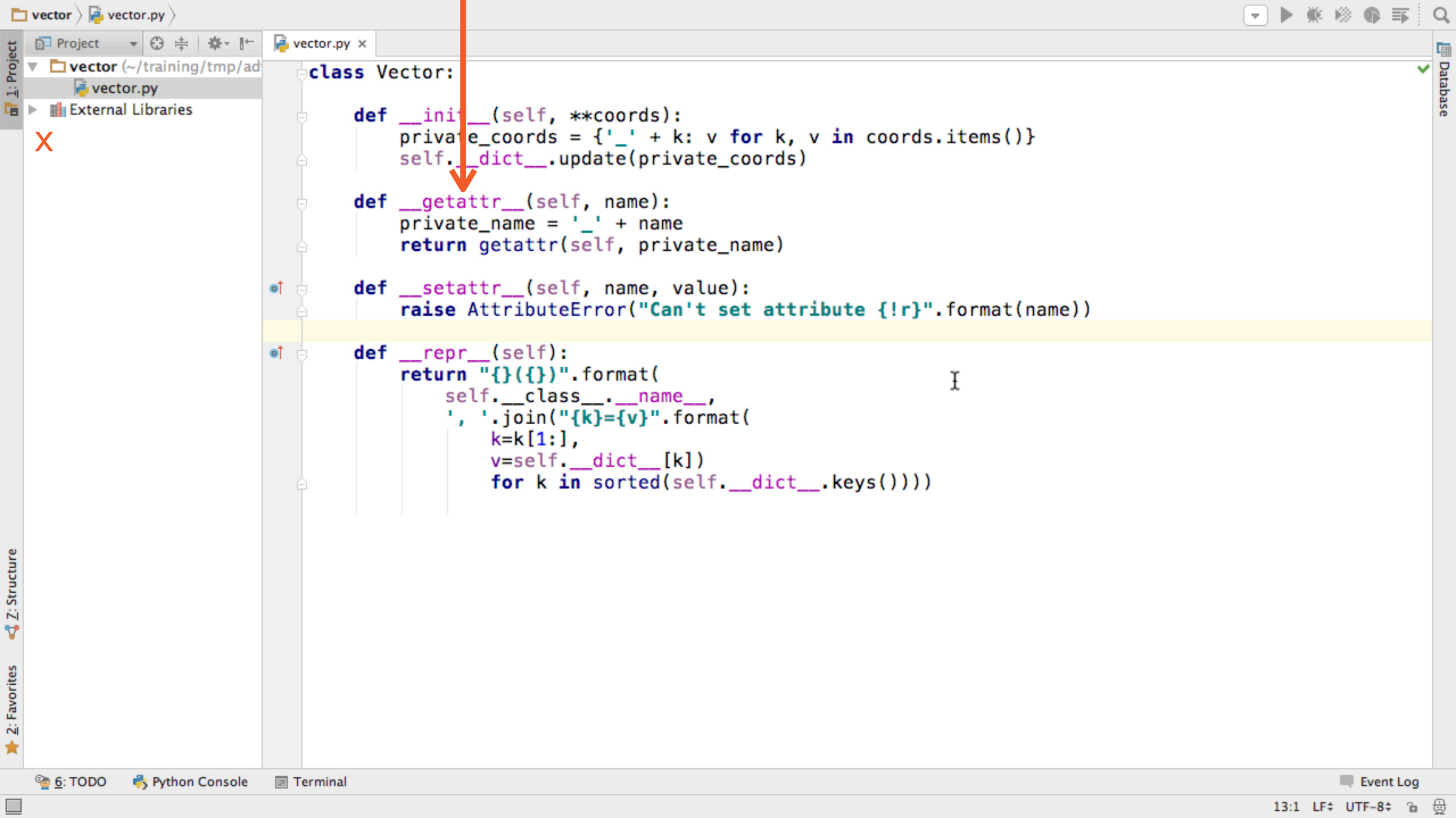
`__getattribute__()`

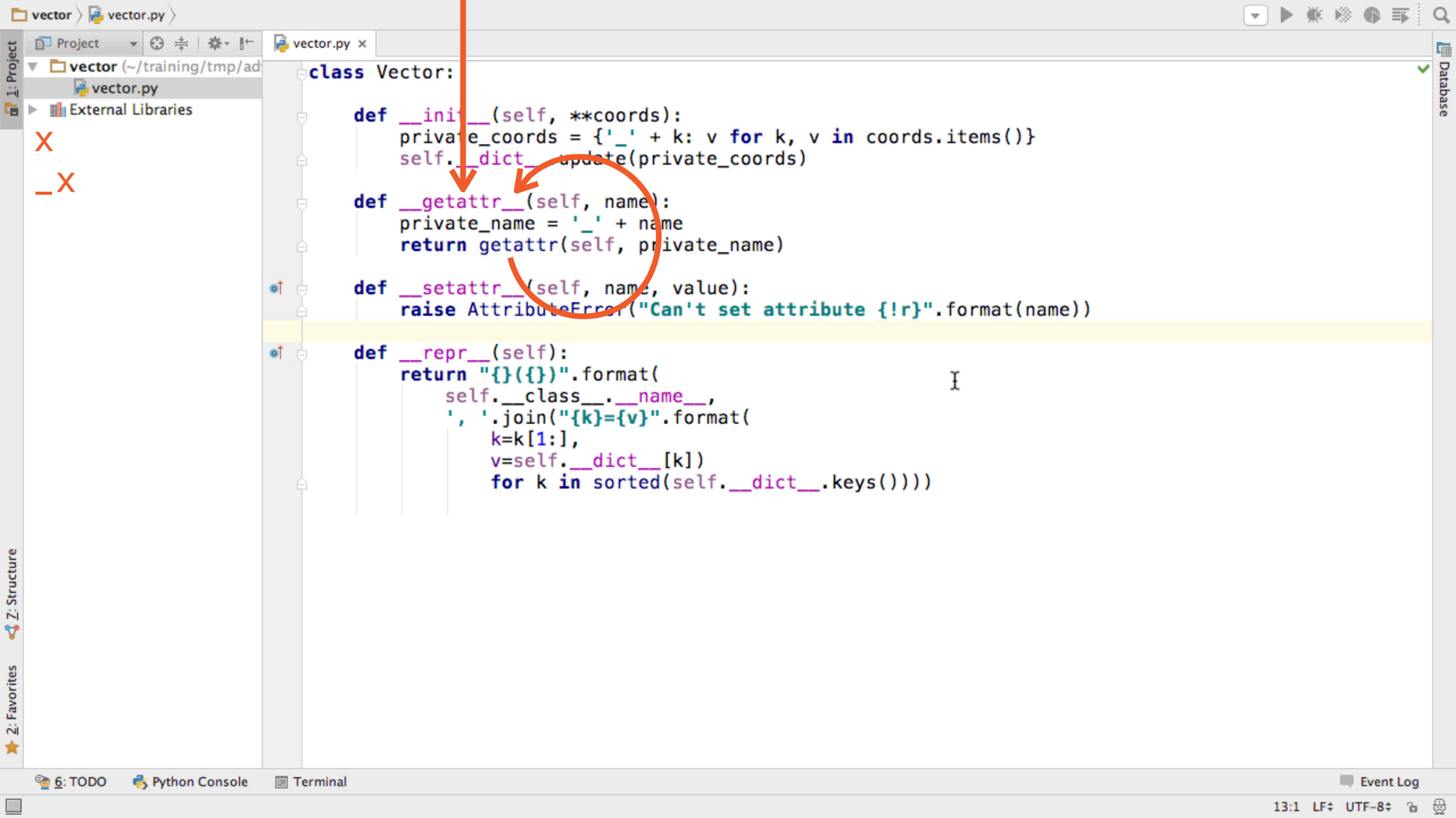
Invoked **instead of** normal lookup

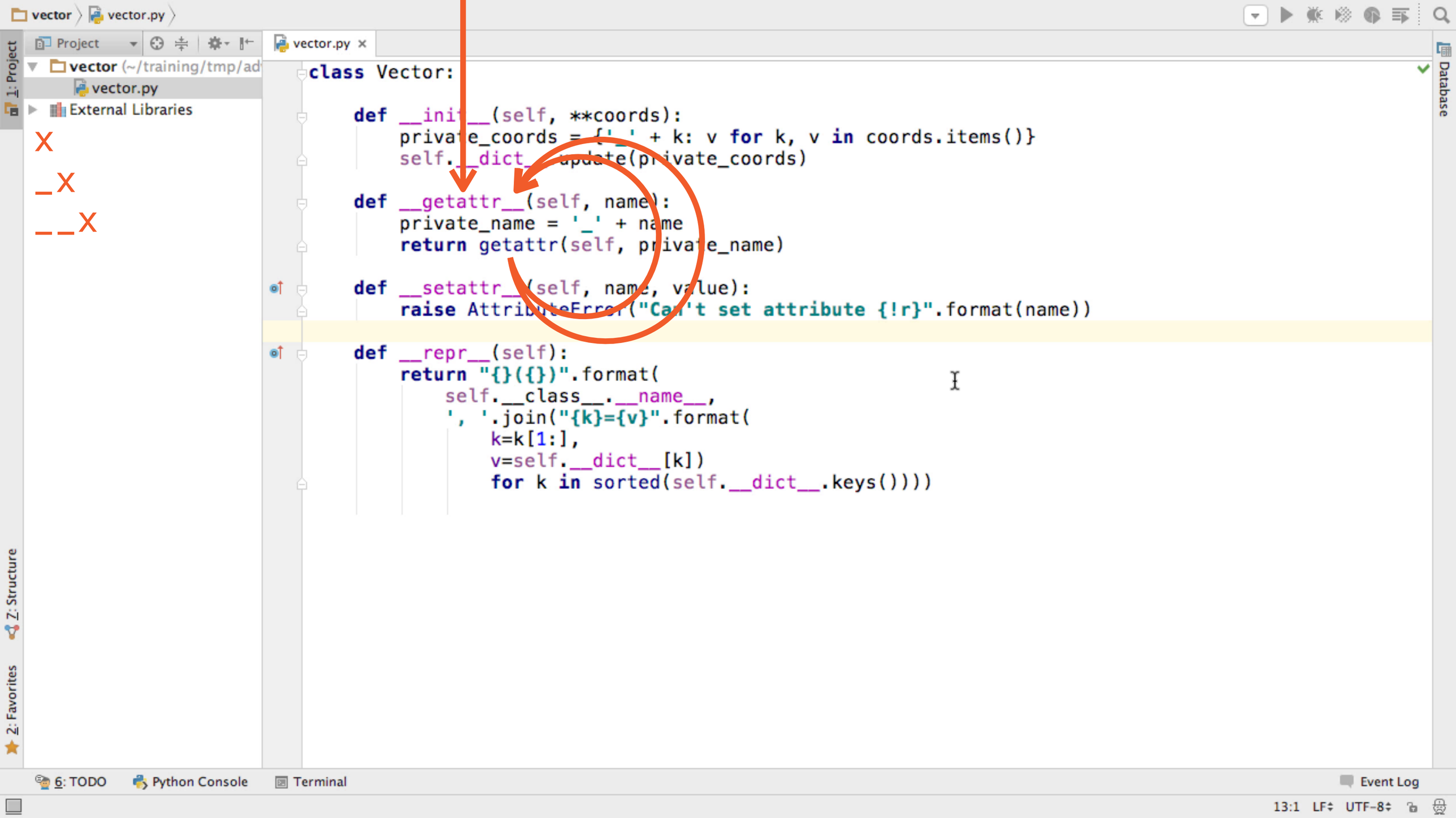
Overriding `__setattr__()`

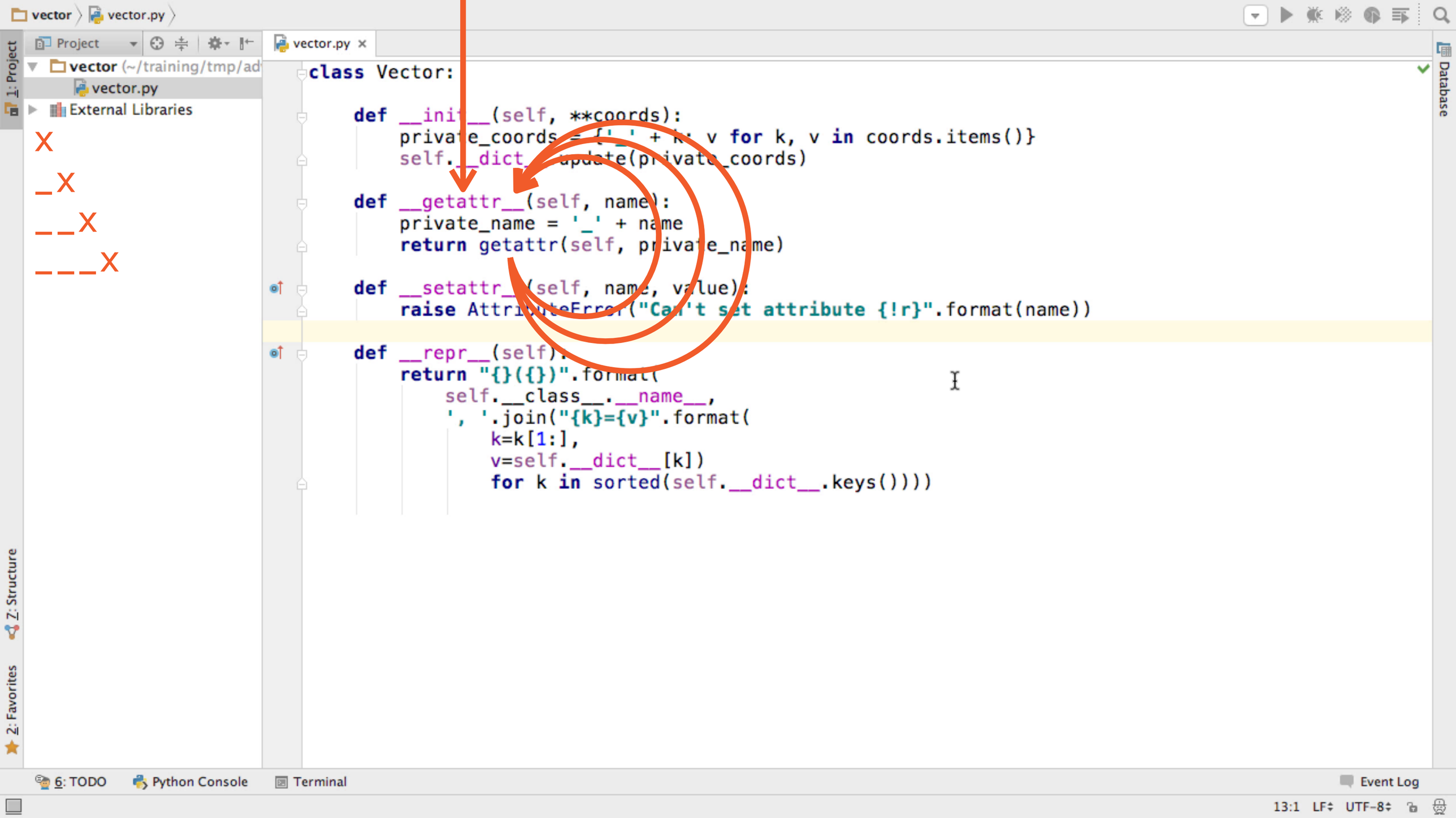
Pitfalls with `__getattr__()`

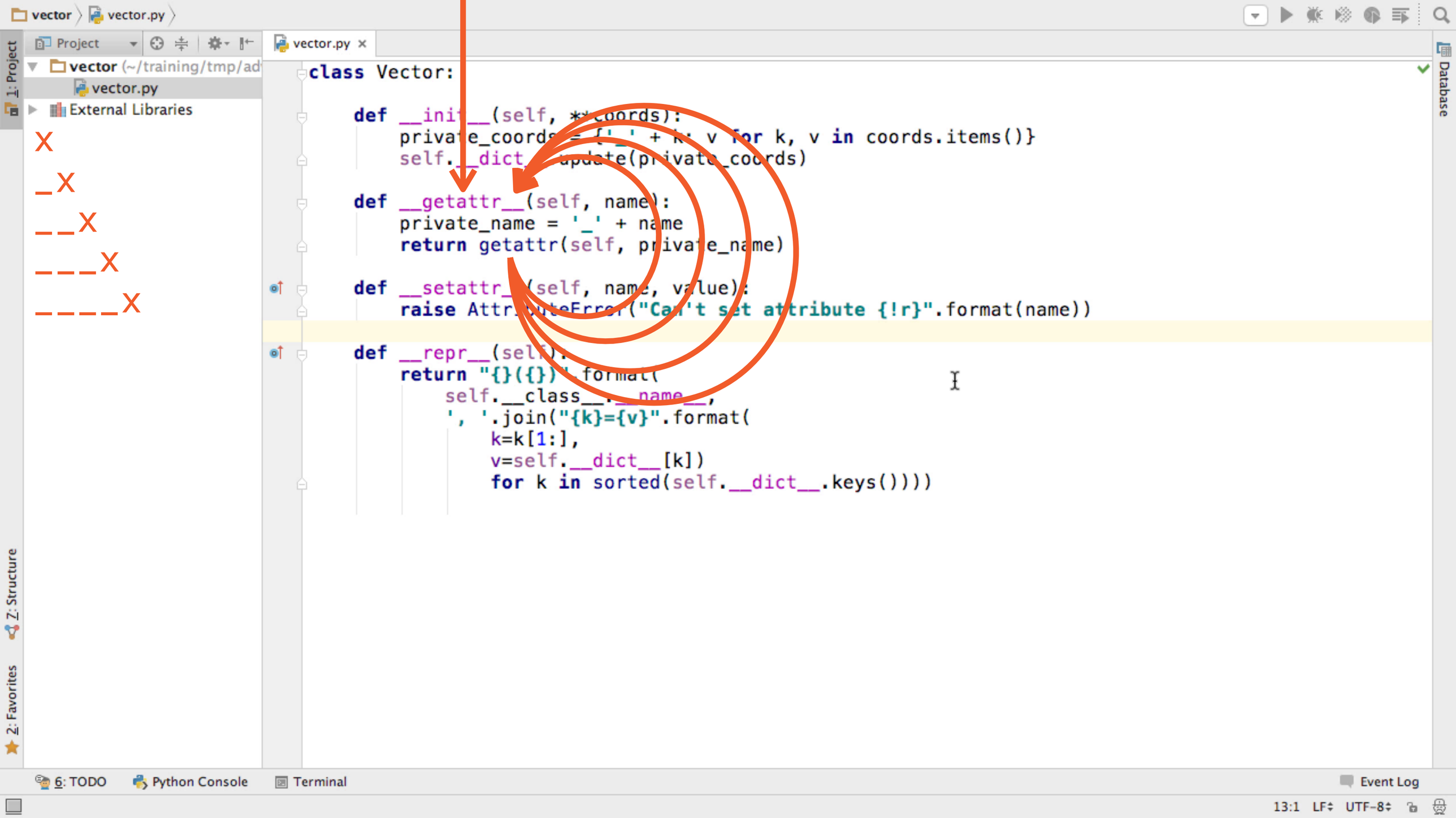

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
1: Project
2: Favorites
7: Structure
Database
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8

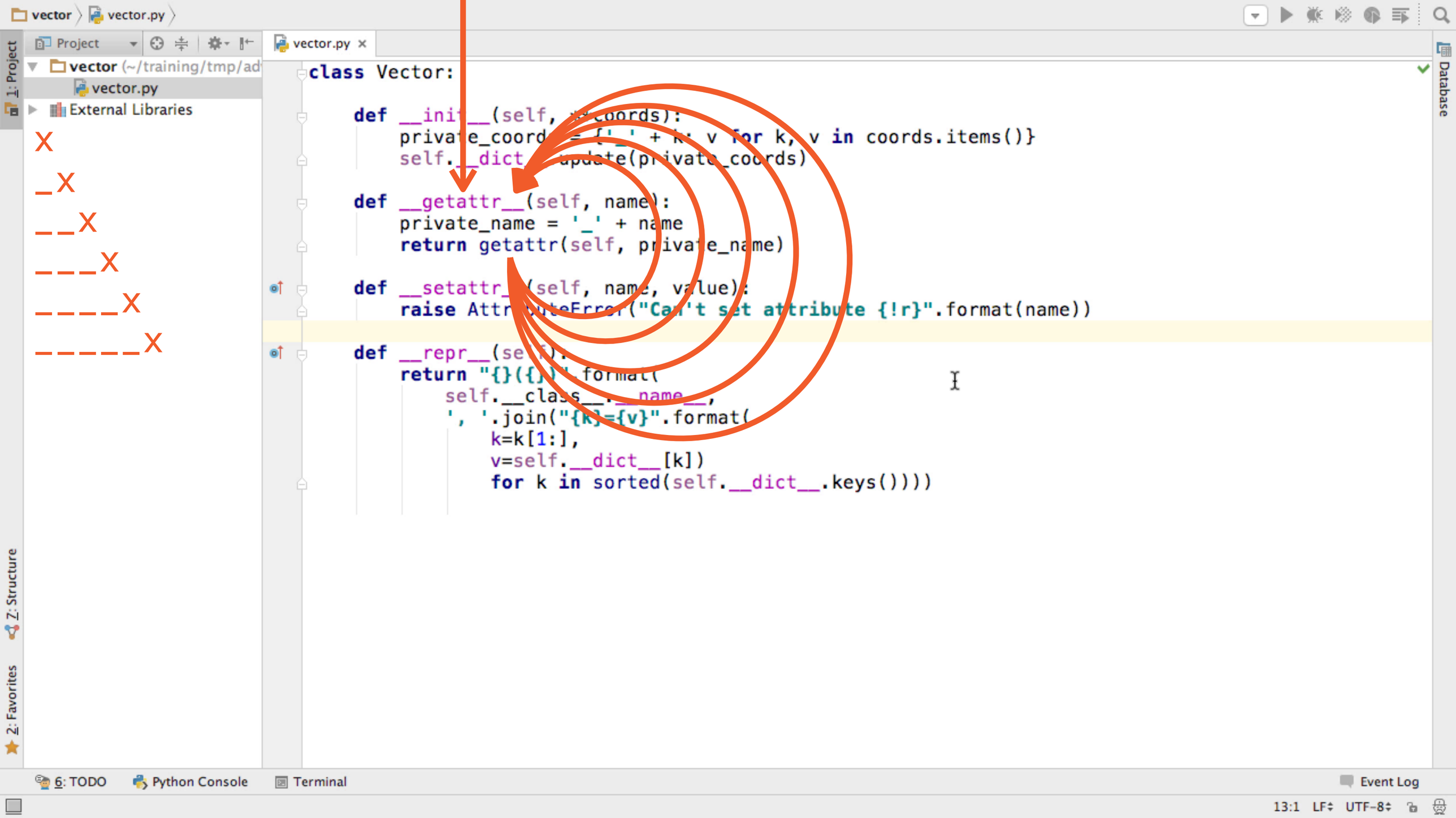

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
X
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
1: Project
2: Favorites
Z: Structure
Database
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8

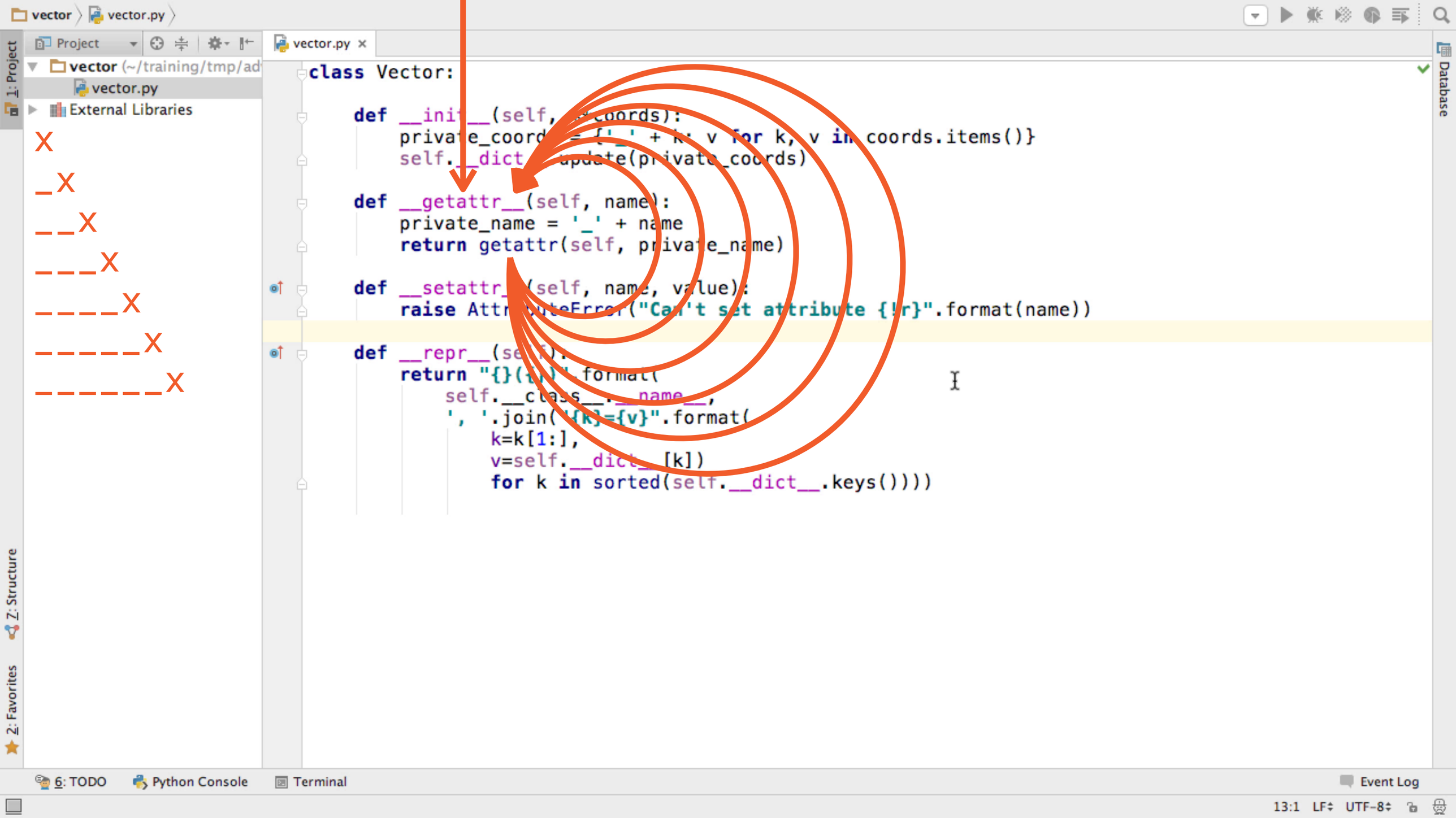

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
X
_X
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
1: Project
2: Favorites
7: Structure
Database
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8

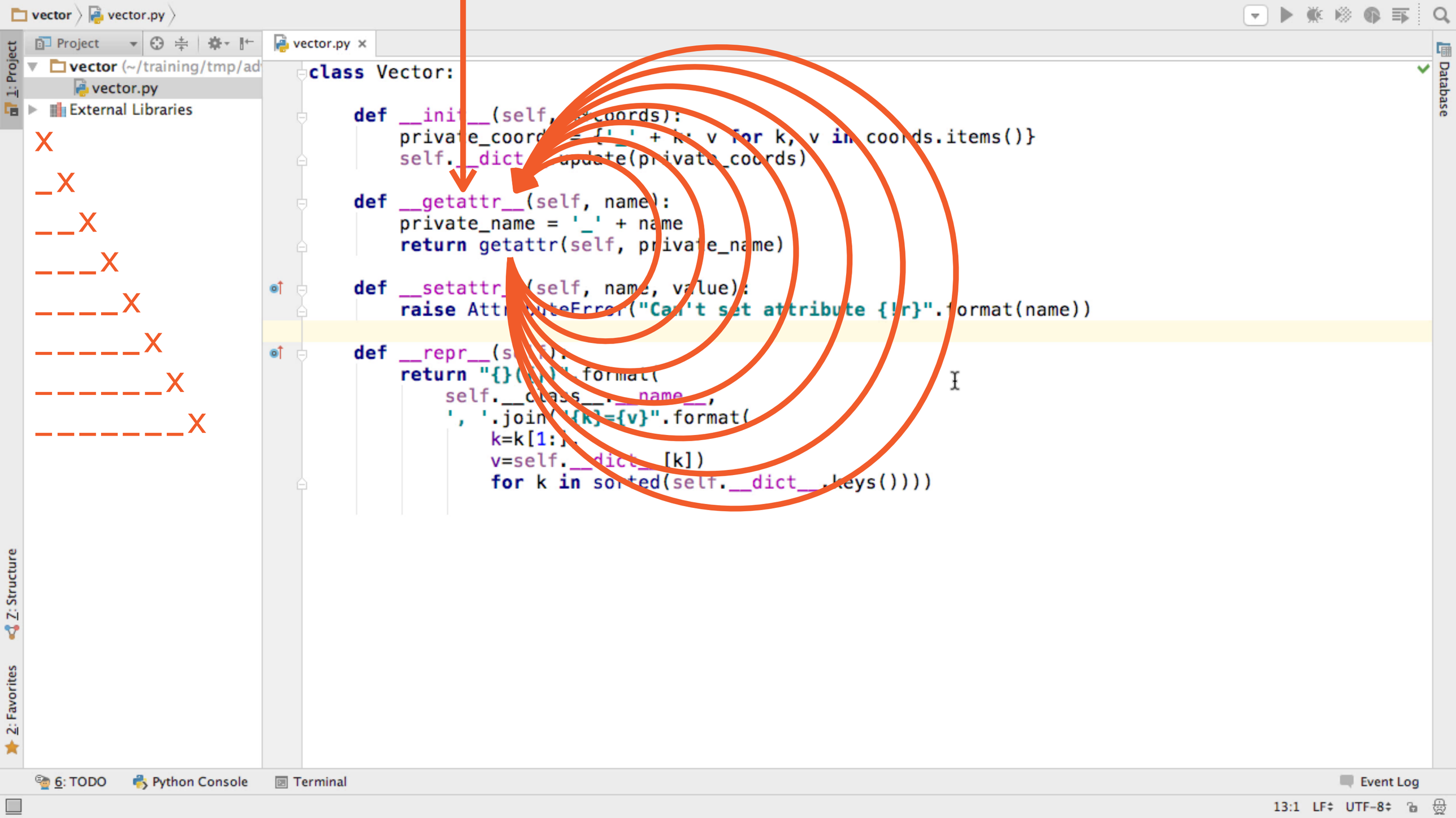

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
X
_X
__X
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
1: Project
2: Favorites
7: Structure
Database
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8

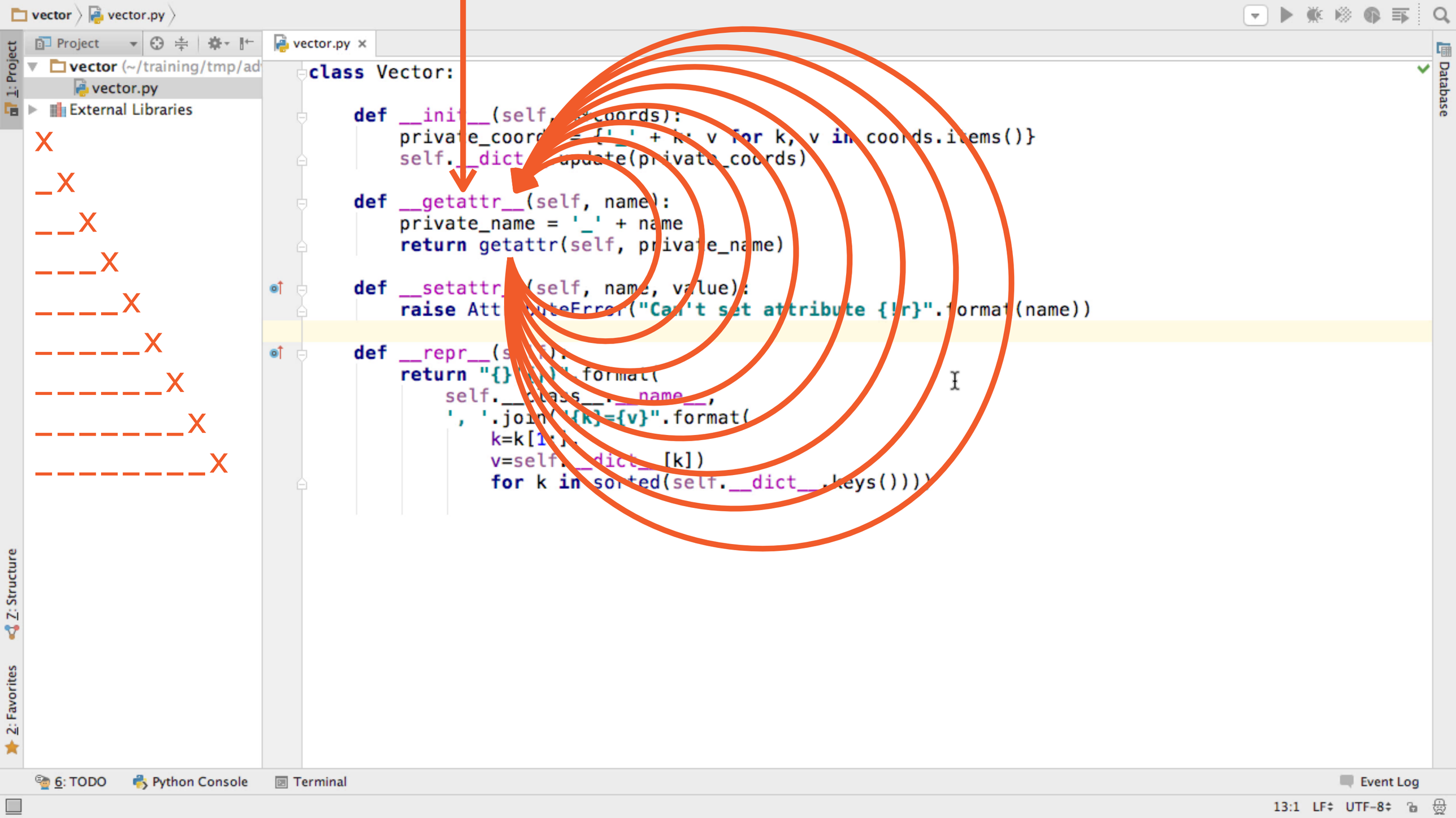

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
X
_X
__X
___X
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8
1: Project
2: Favorites
7: Structure
Database

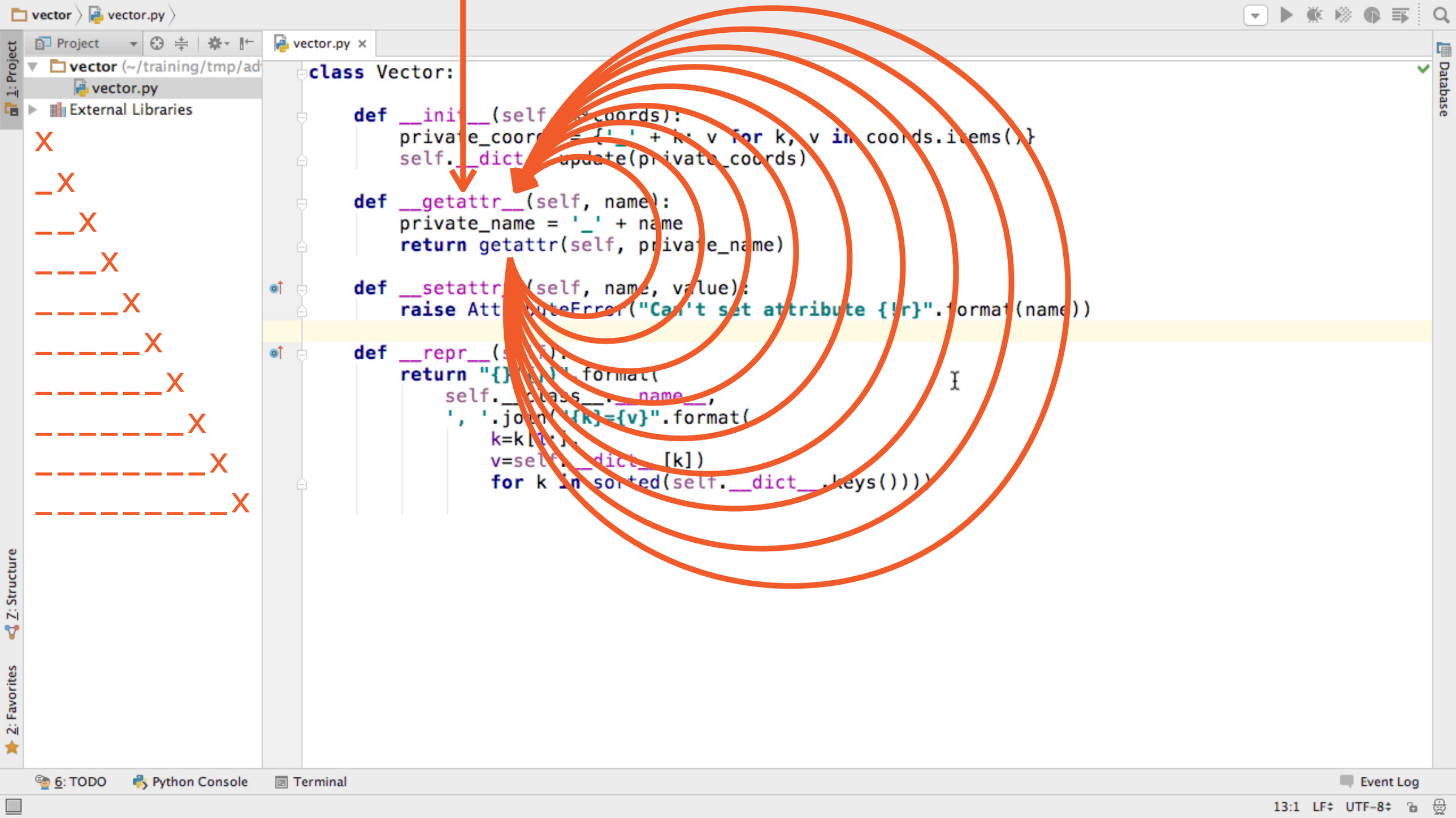

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
X
_X
__X
___X
____X
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8

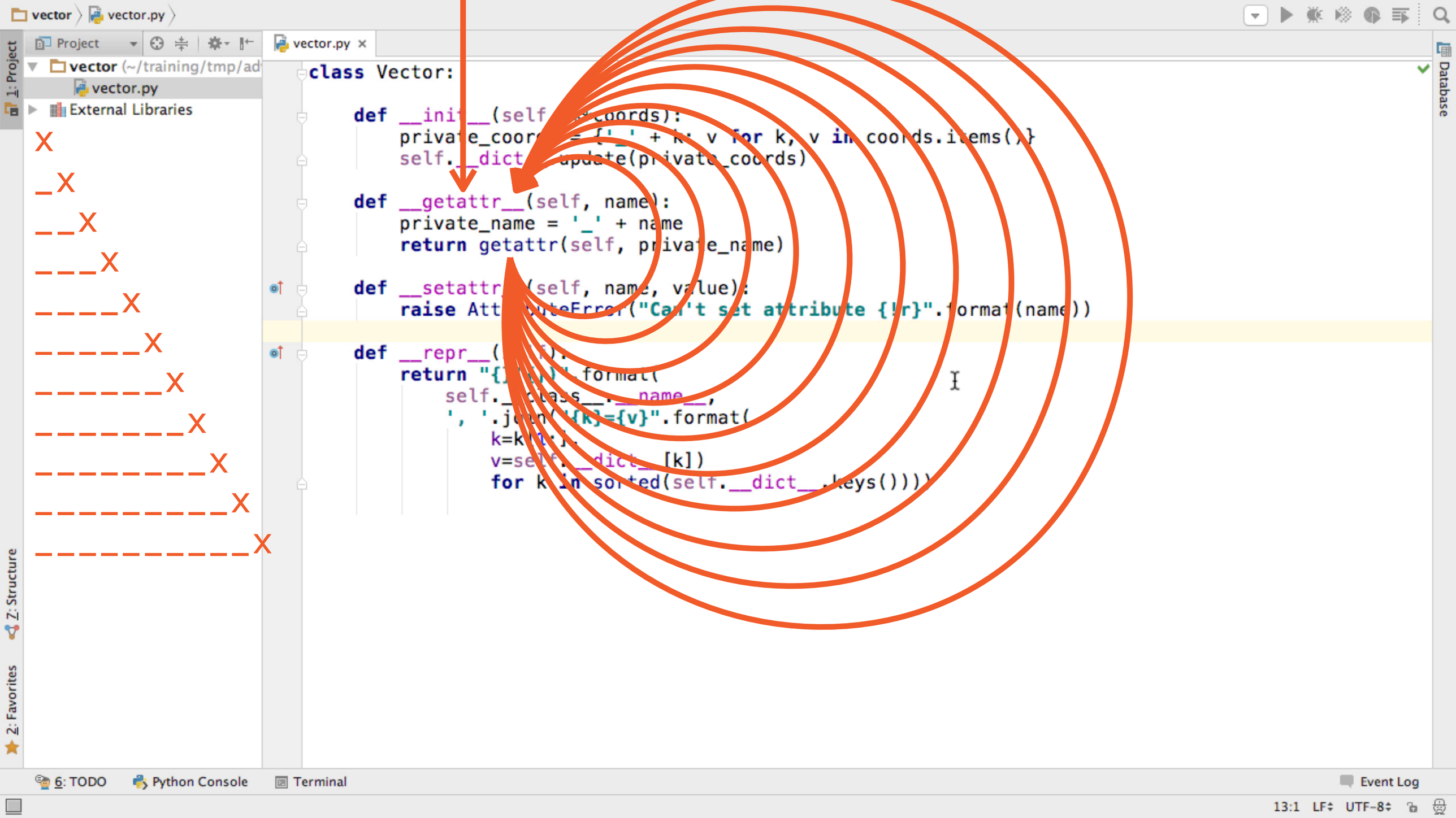

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
X
_X
__X
___X
____X
_____X
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8

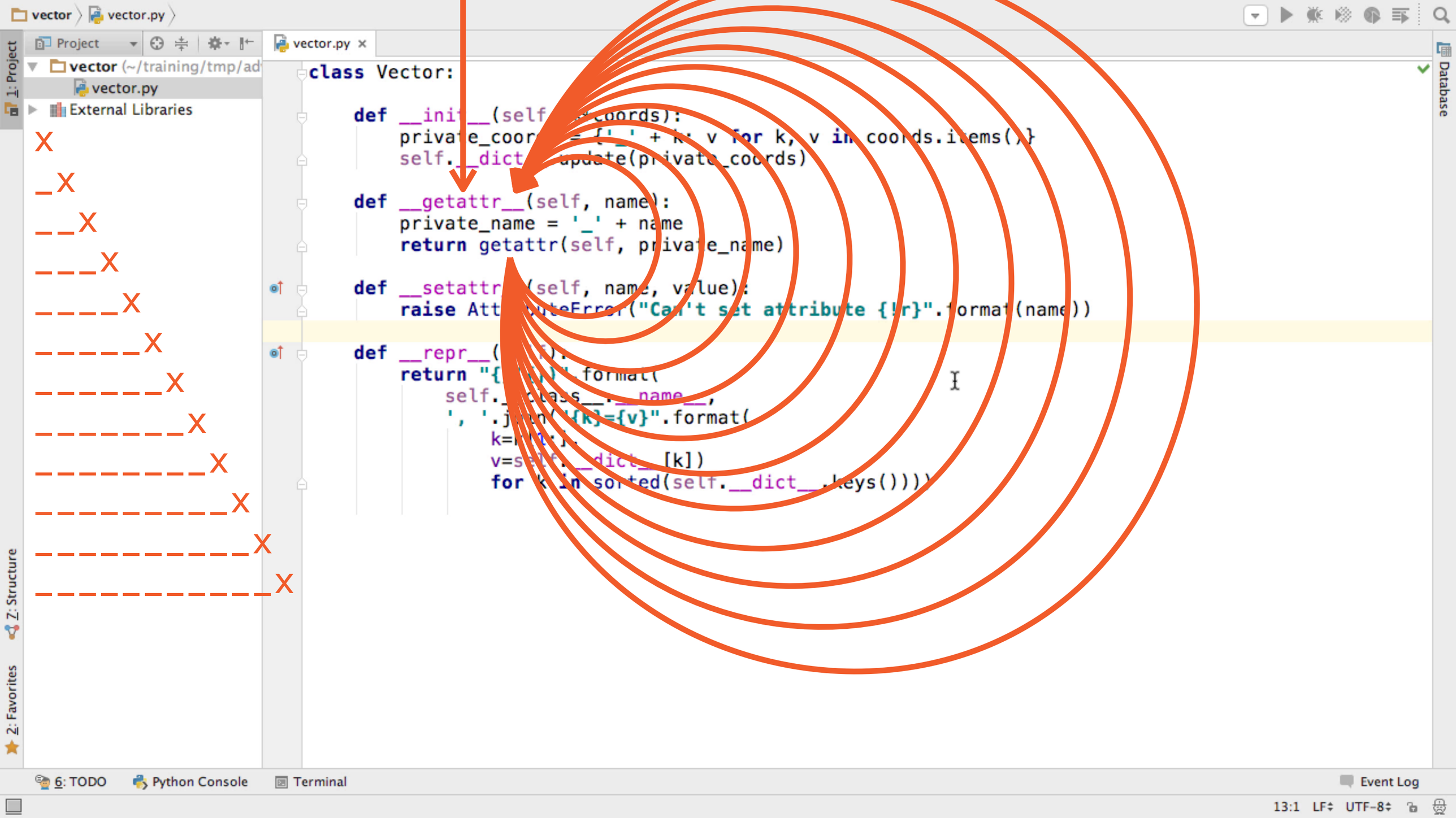

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
X
_X
__X
___X
____X
_____X
______X
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8

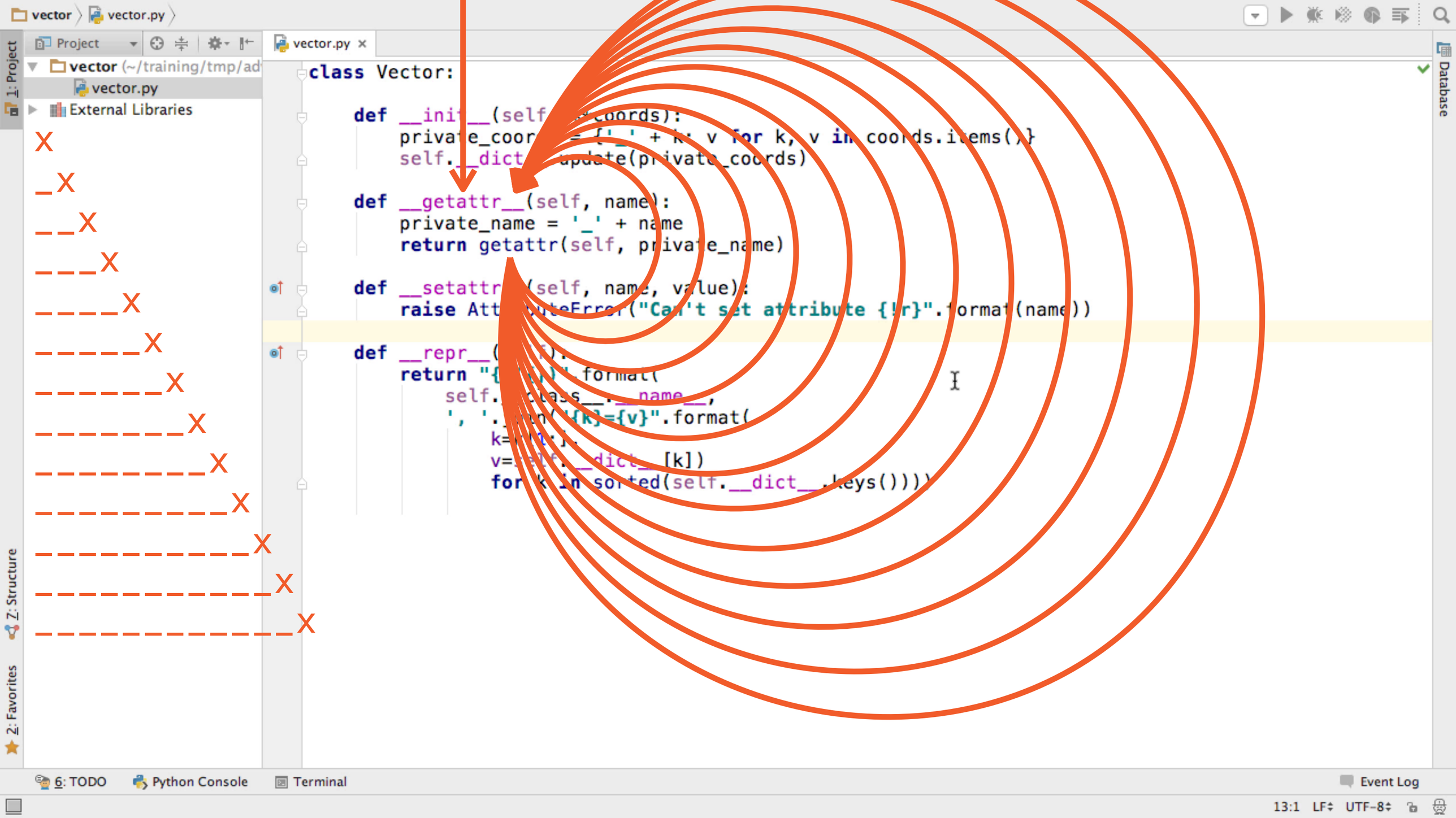

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
X
_X
__X
___X
____X
_____X
______X
_______X
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8

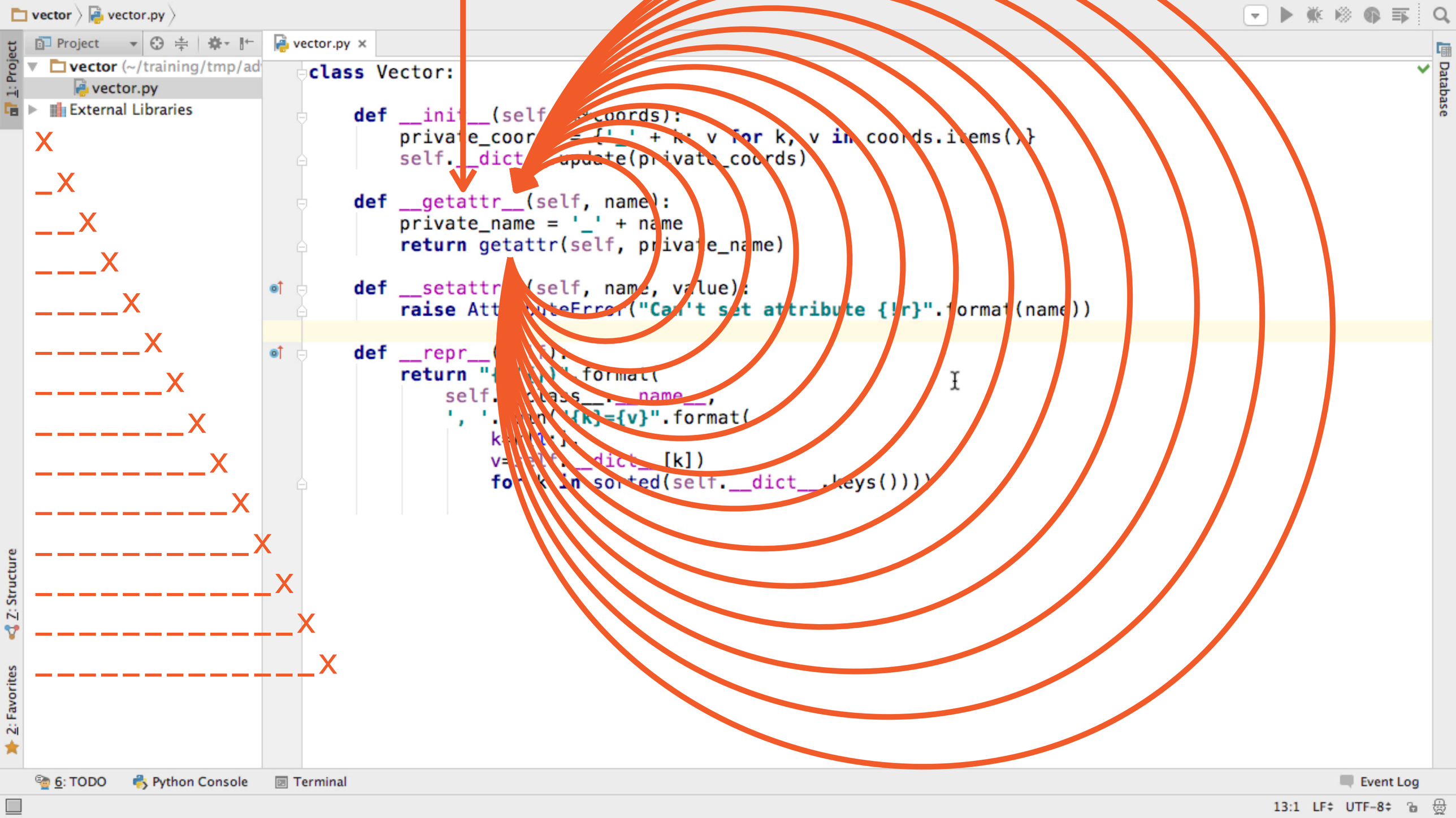

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
X
_X
__X
___X
____X
_____X
______X
_______X
________X
_________X
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
1: Project
2: Favorites
Z: Structure
Database
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8

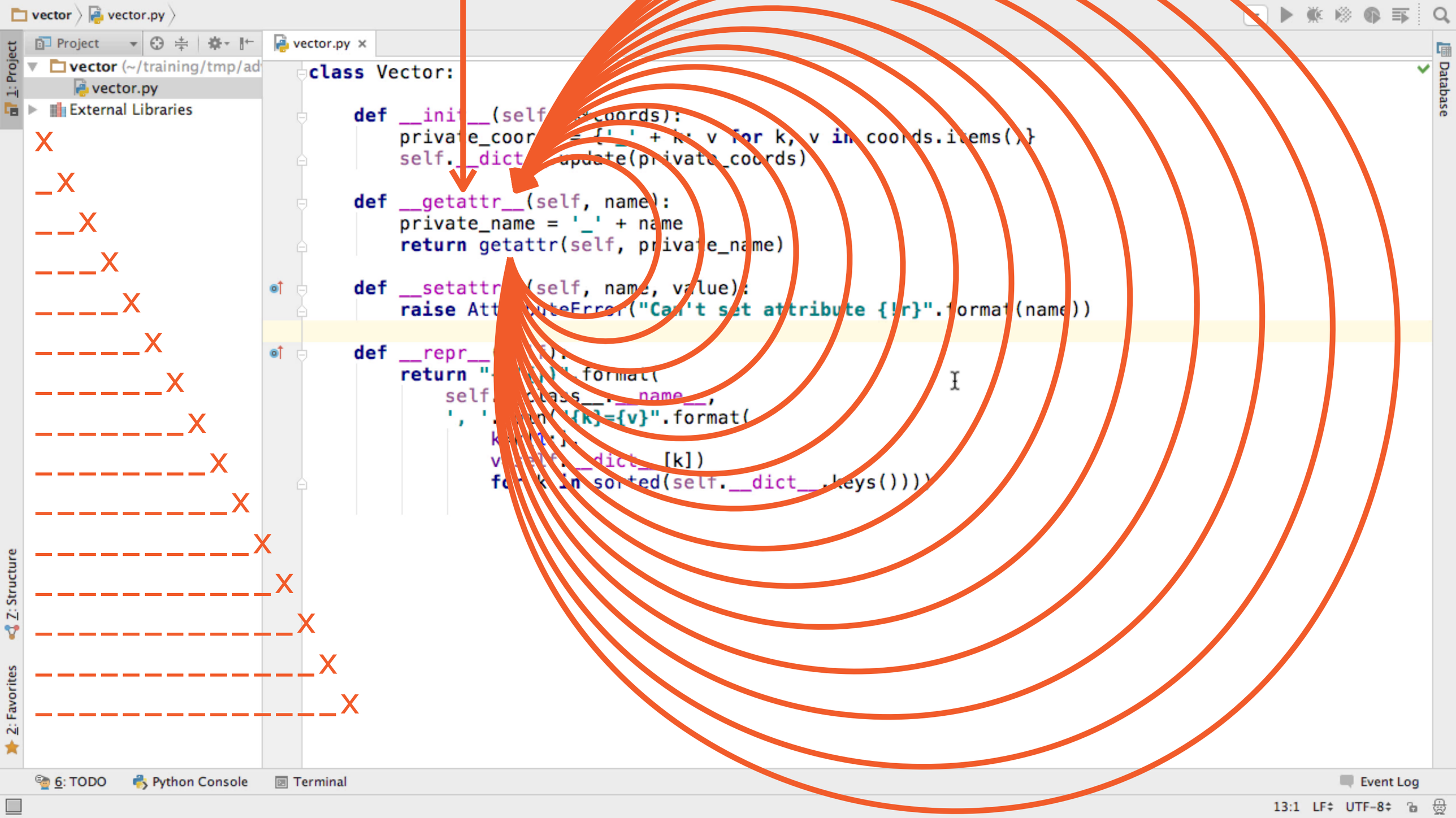

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __getattr__(self, name):
private_name = '_' + name
return getattr(self, private_name)
X
_X
__X
___X
____X
_____X
______X
_______X
________X
_________X
__________X
1: Project
Z: Structure
2: Favorites
Database
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8

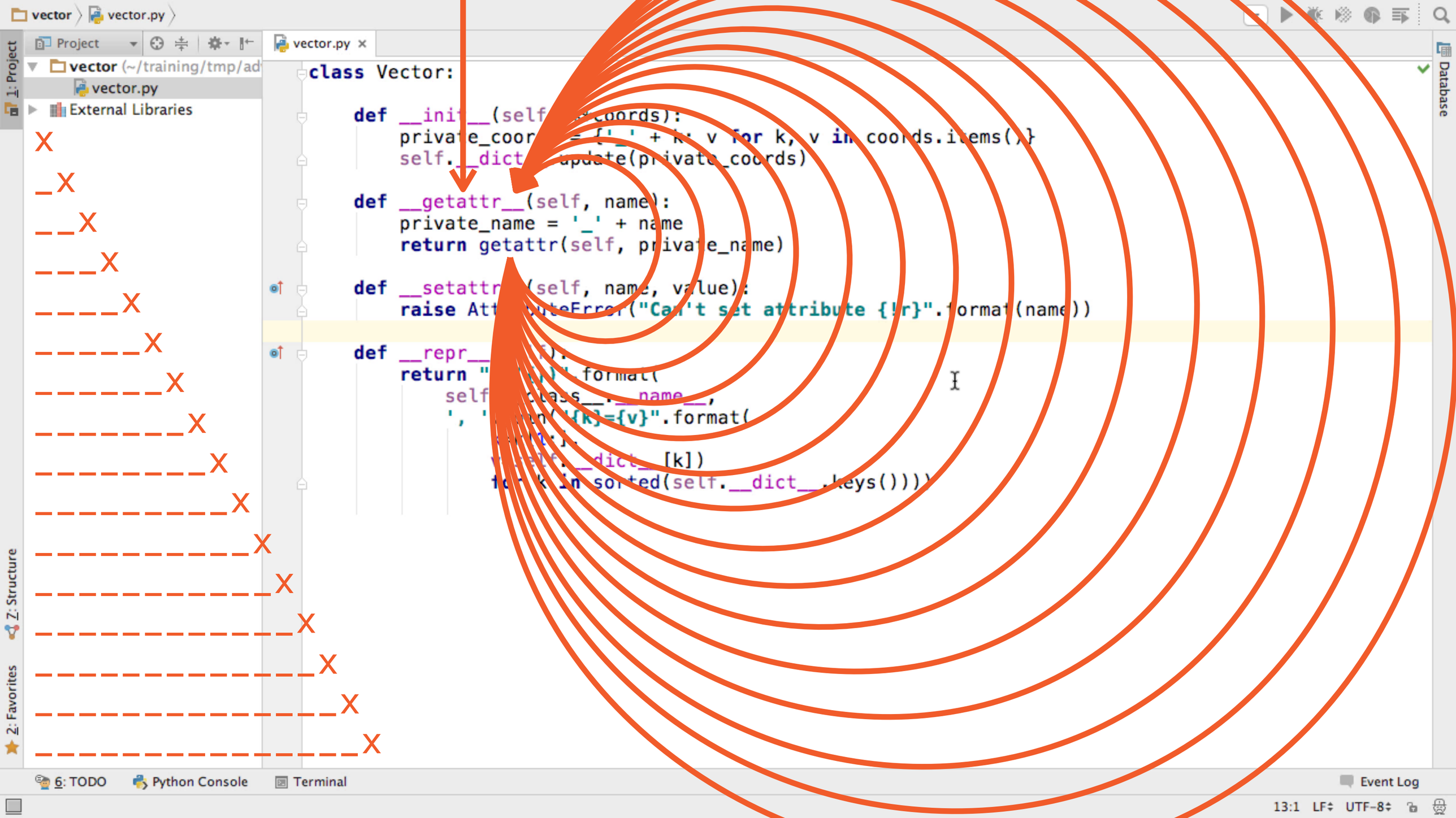

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self, coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
X
_X
__X
___X
____X
_____X
______X
_______X
________X
_________X
__________X
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8
1: Project
2: Favorites
7: Structure
Database

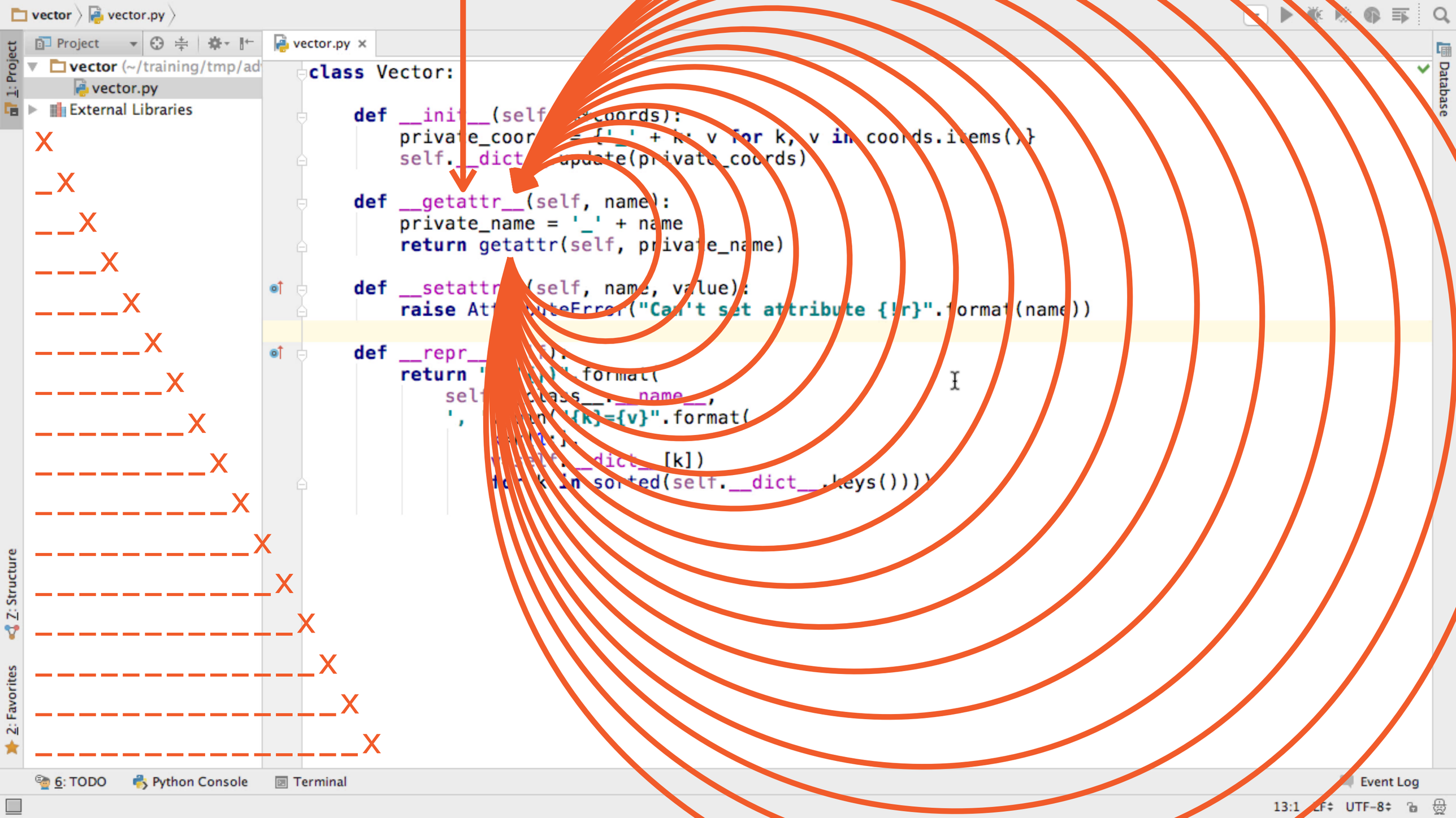

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self, coords):
def __getattr__(self, name):
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(
return "{
self.__class__.__name__,
', '.join("{k}={v}".format(
for k in sorted(self.__dict__.keys())))
X
_X
__X
___X
____X
_____X
______X
_______X
________X
_________X
__________X
___________X
____________X
1: Project
Z: Structure
2: Favorites
Database
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8


vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self, coords):
def __getattr__(self, name):
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(
return "{
self.
', '.
k=
v=
for
X
_X
__X
___X
____X
_____X
______X
_______X
________X
_________X
__________X
___________X
____________X
_____________X
1: Project
2: Favorites
Z: Structure
Database
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8


vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self, coords):
private_coor = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(
return "{}(".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
X
_X
__X
___X
____X
_____X
______X
_______X
________X
_________X
__________X
___________X
____________X
_____________X
______________X
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8
1: Project
2: Favorites
7: Structure
Database


vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
class Vector:
def __init__(self, coords):
def __getattr__(self, name):
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k,
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
X
_X
__X
___X
____X
_____X
______X
_______X
________X
_________X
__________X
___________X
____________X
_____________X
______________X
_______________X
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8


vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
X
_X
__X
___X
____X
_____X
______X
_______X
________X
_________X
__________X
___________X
____________X
_____________X
______________X
_______________X
________________X
6: TODO
Python Console
Terminal
Event Log
13:1
LF
UTF-8

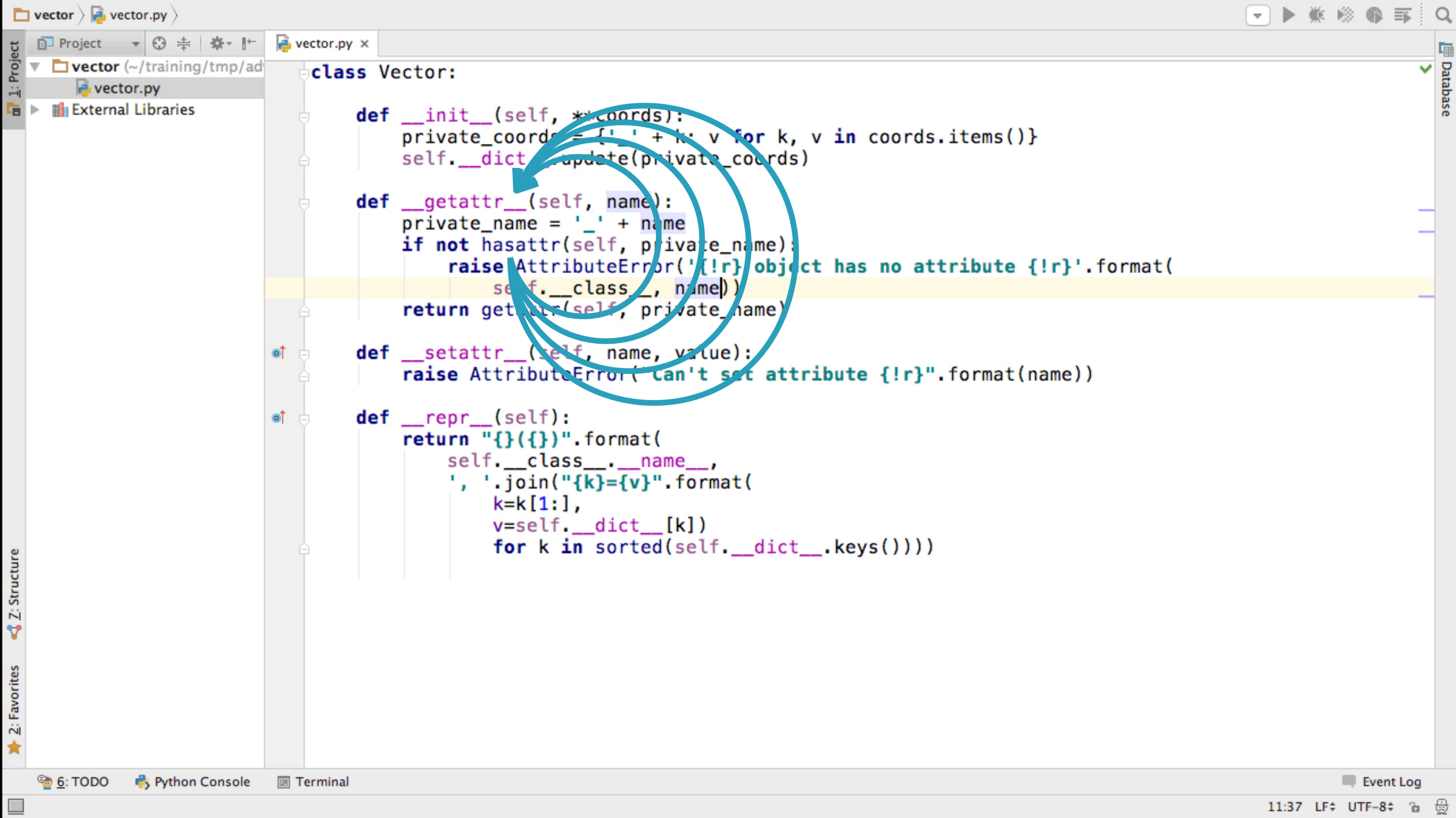

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self
coords):
private_coor
= {'_' + k: v for k, v in coords.items()}
self.__dict
update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
return getattr(self, private_name)
def __setattr
(self, name, value):
raise Att
uteError("Can't set attribute {!r}".format(name))
def __repr__
):
return '
}'.format(
self
class__.__name__,
',
n("{k}={v}".format(
[k])
k in sorted(self.__dict__.keys())))
X
_X
__X
___X
____X
_____X
______X
_______X
________X
_________X
__________X
___________X
____________X
_____________X
______________X
_______________X
________________X
6: TODO
Python Console
Terminal
Event Log
13:1
UTF-8
1: Project
Z: Structure
2: Favorites
Database

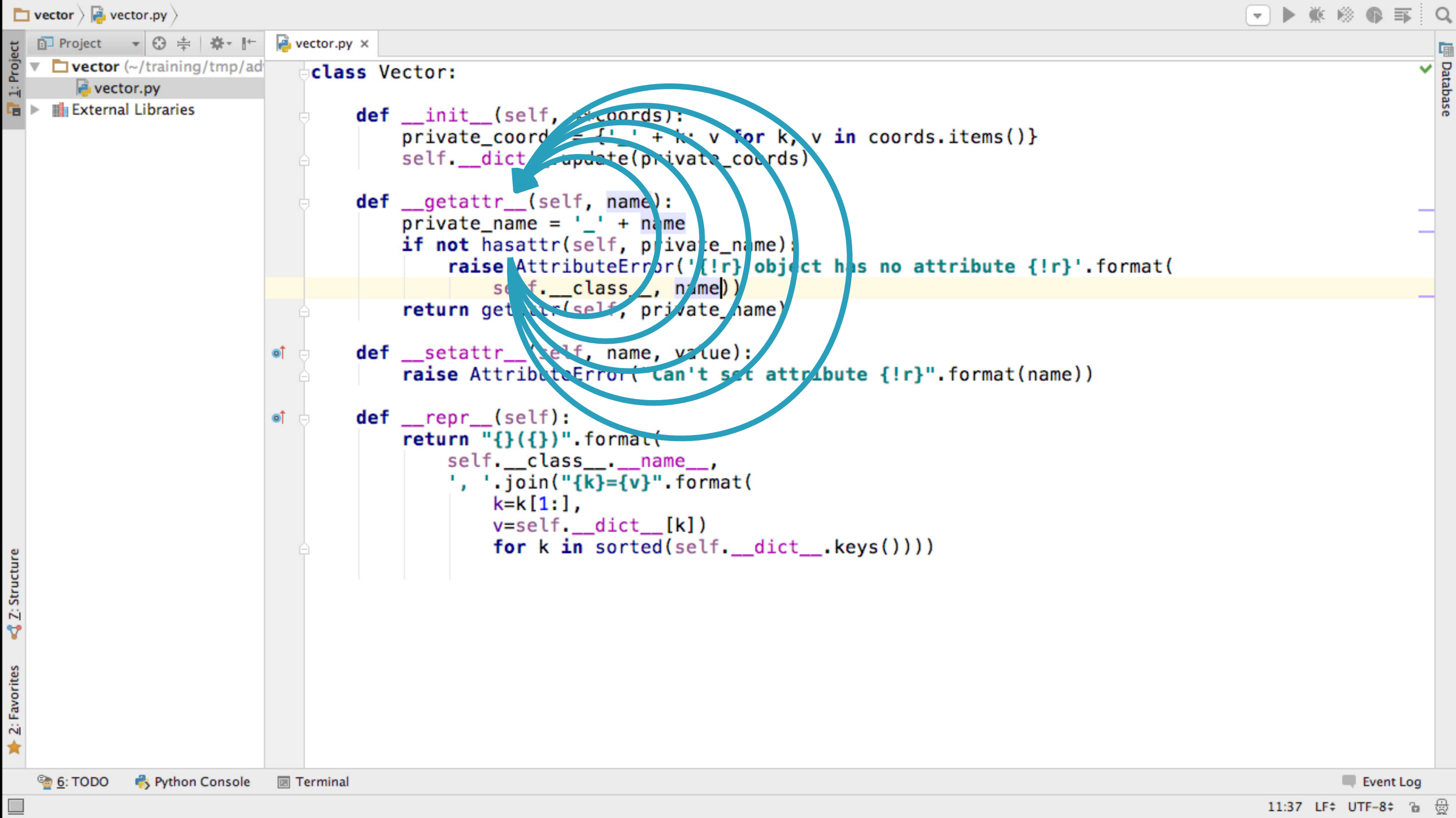

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
1: Project
2: Favorites
7: Structure
Database
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8

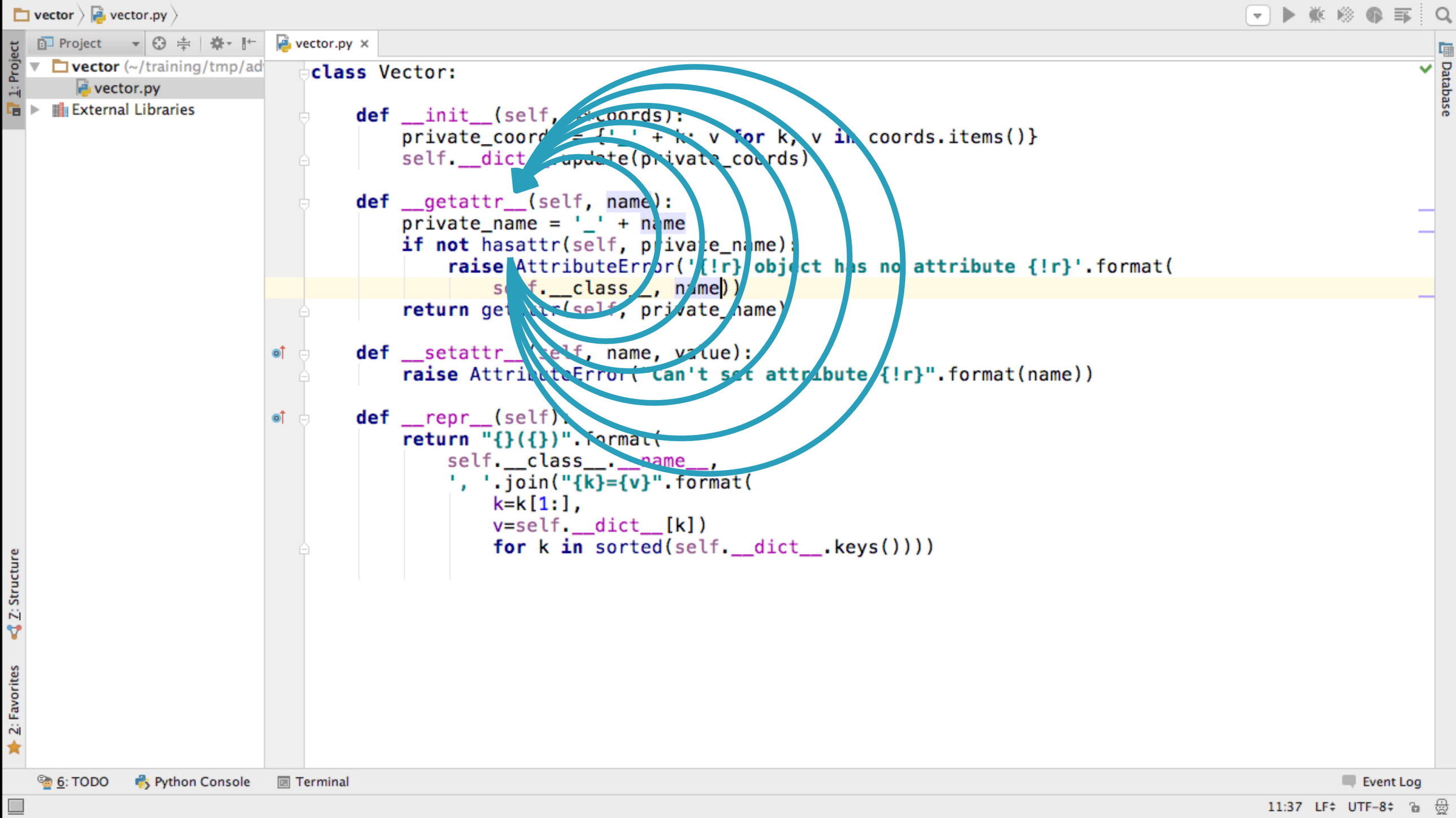

vector
vector.py
Project
vector.py
vector (~/training/tmp/ad
vector.py
External Libraries
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
Database
7: Structure
2: Favorites
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8

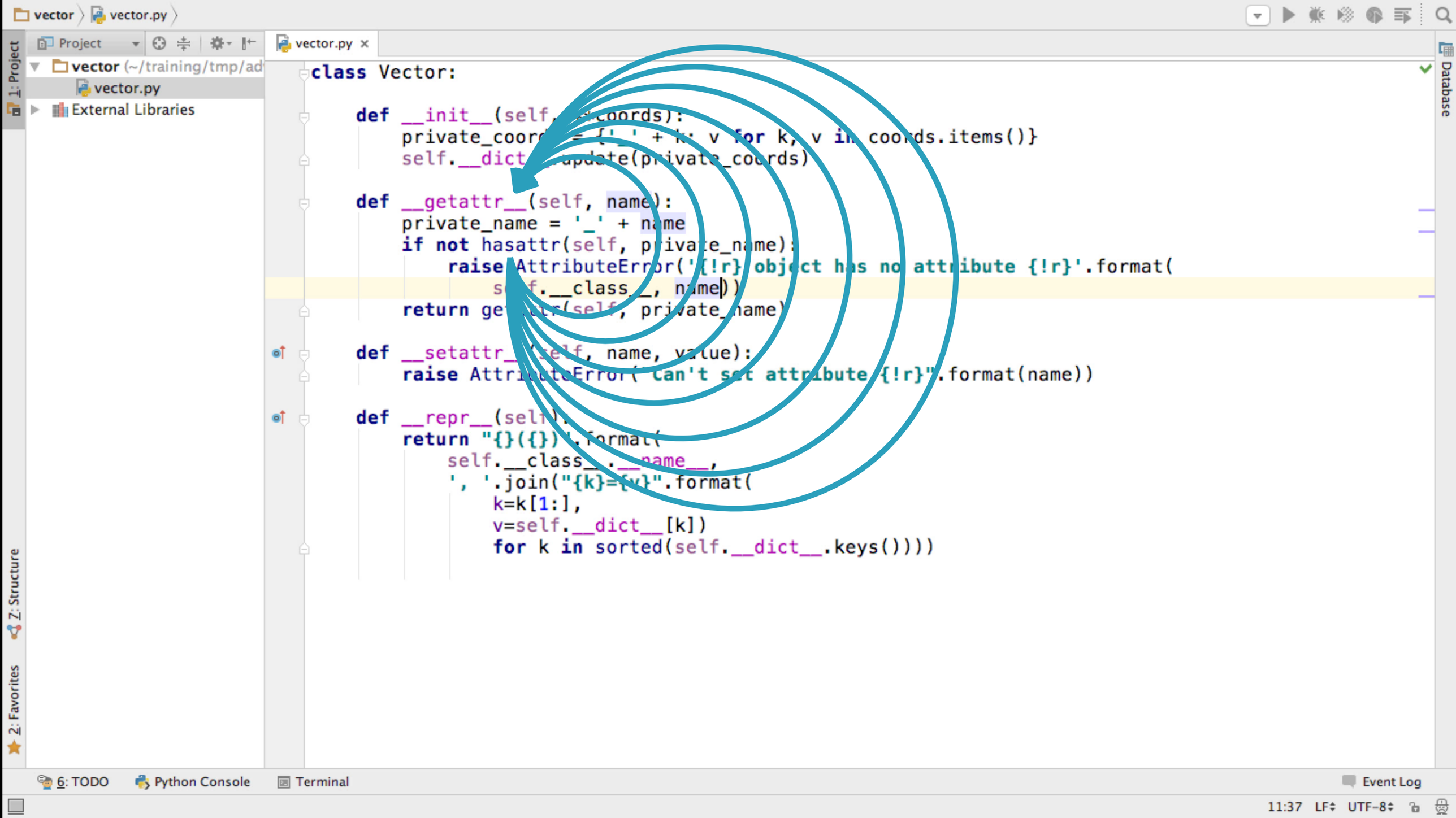

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
1: Project
2: Favorites
7: Structure
Database
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8

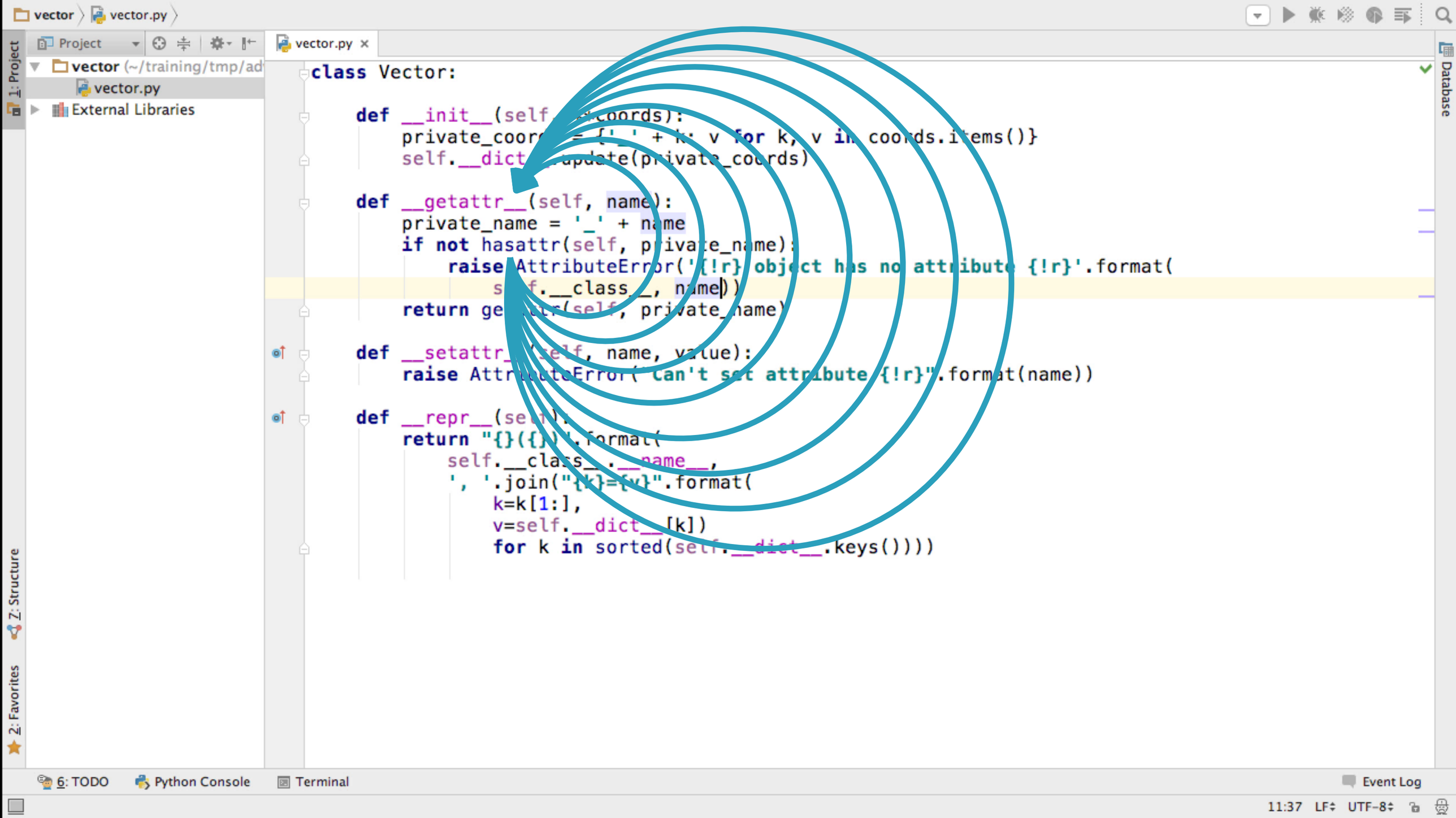

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
1: Project
2: Favorites
7: Structure
Database
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8

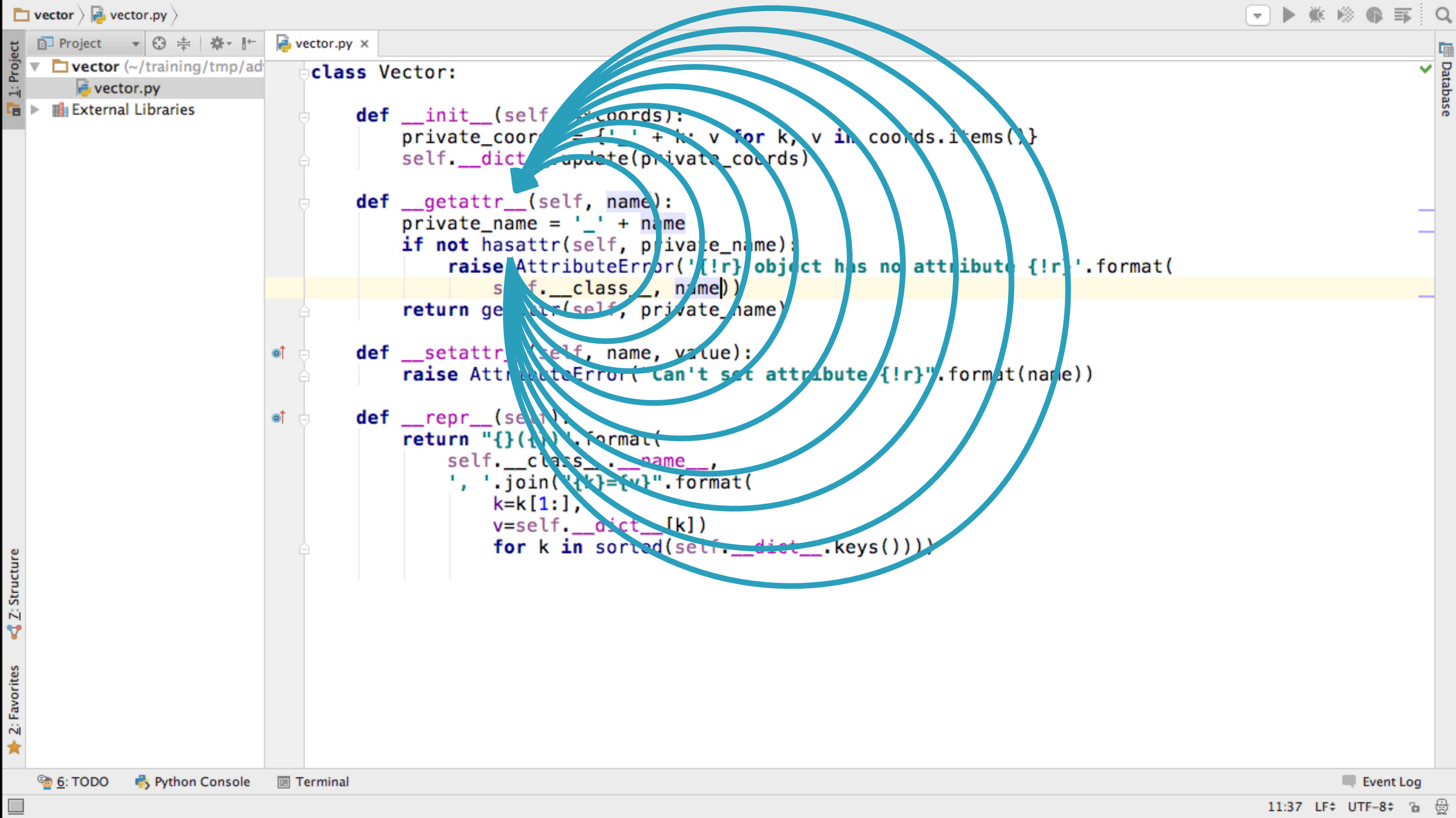

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
1: Project
2: Favorites
7: Structure
Database
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8

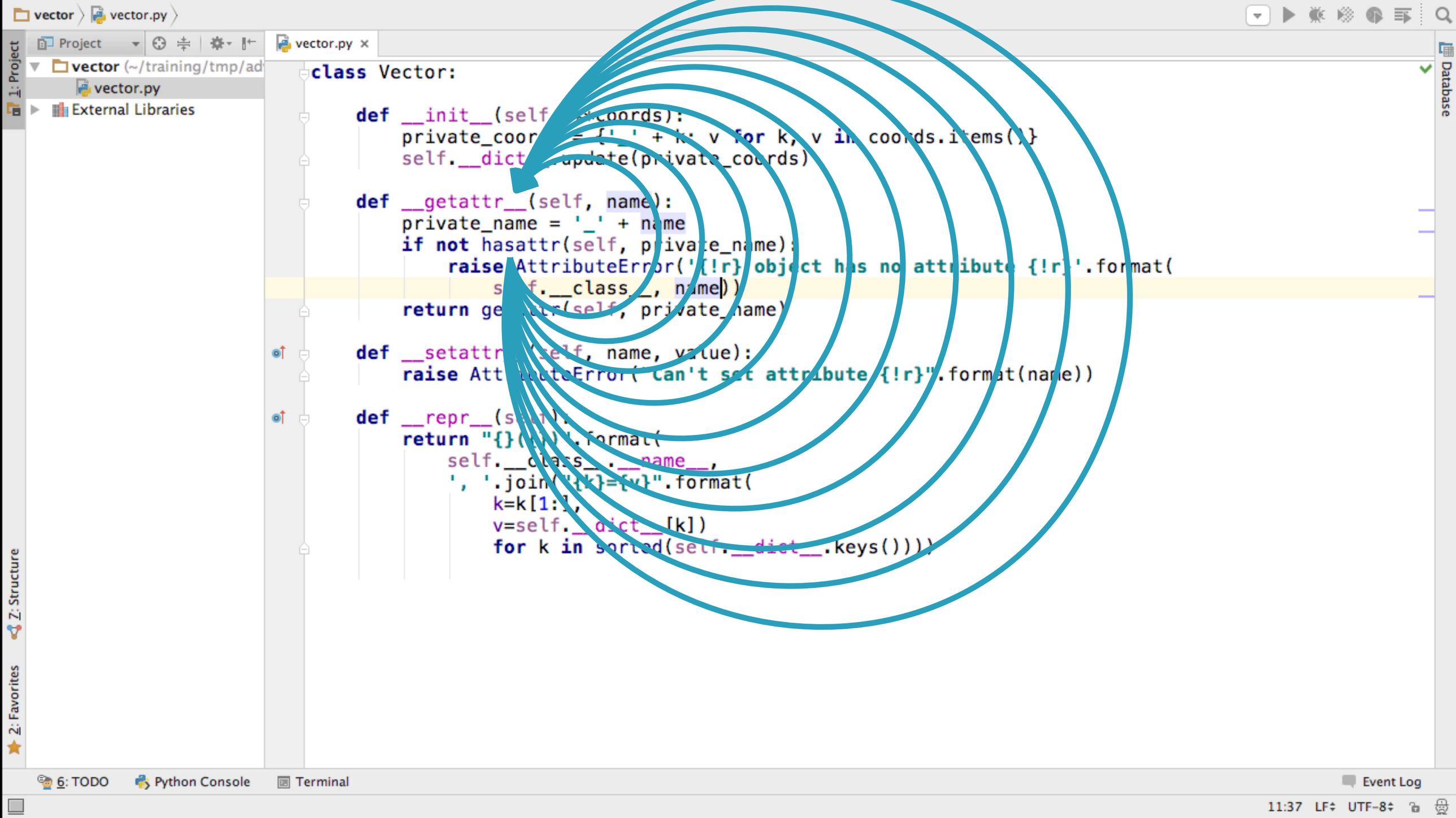

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self, **coords):
private_coord = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
Database
1: Project
7: Structure
2: Favorites
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8

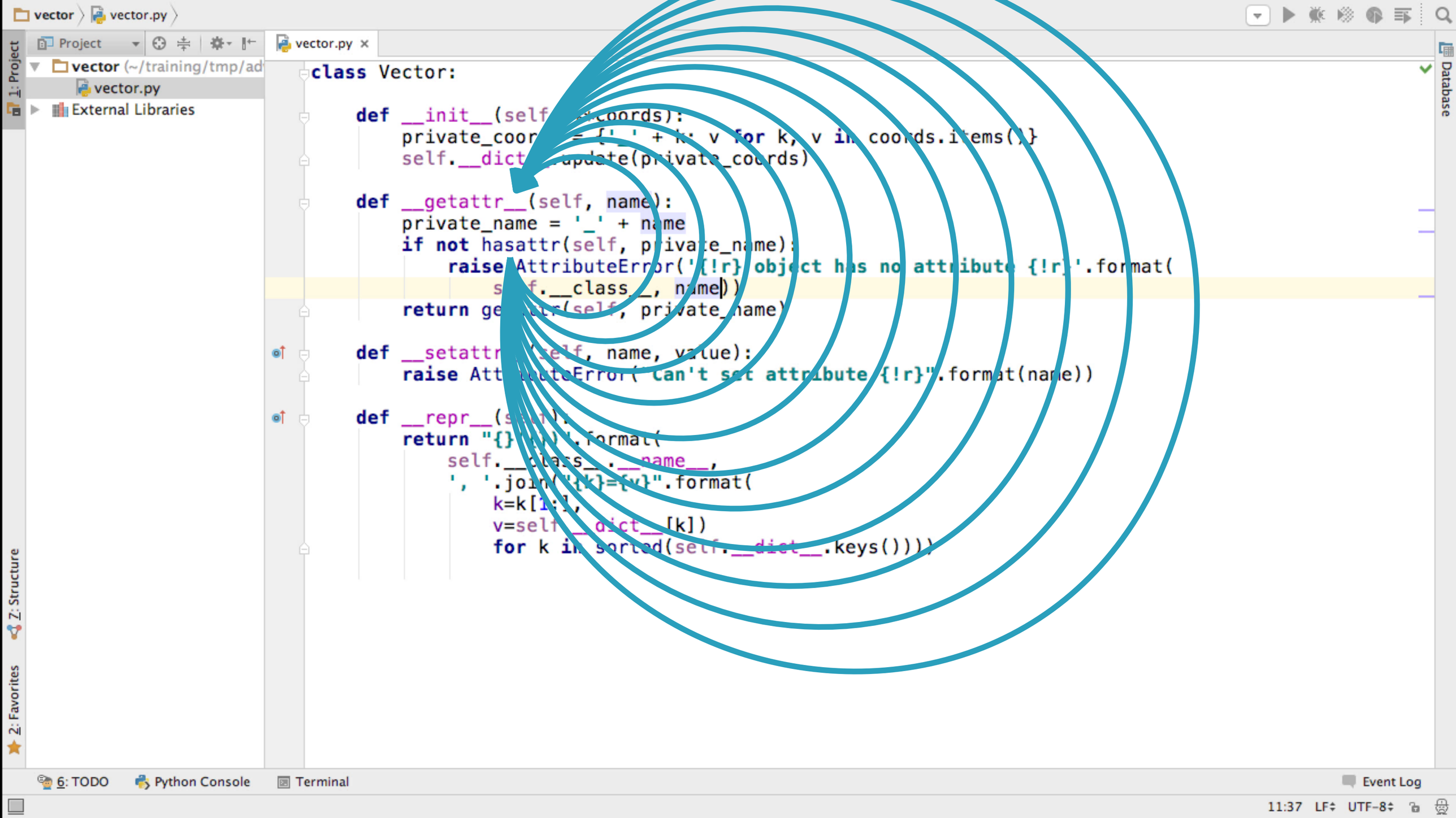

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
Database
7: Structure
2: Favorites
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8

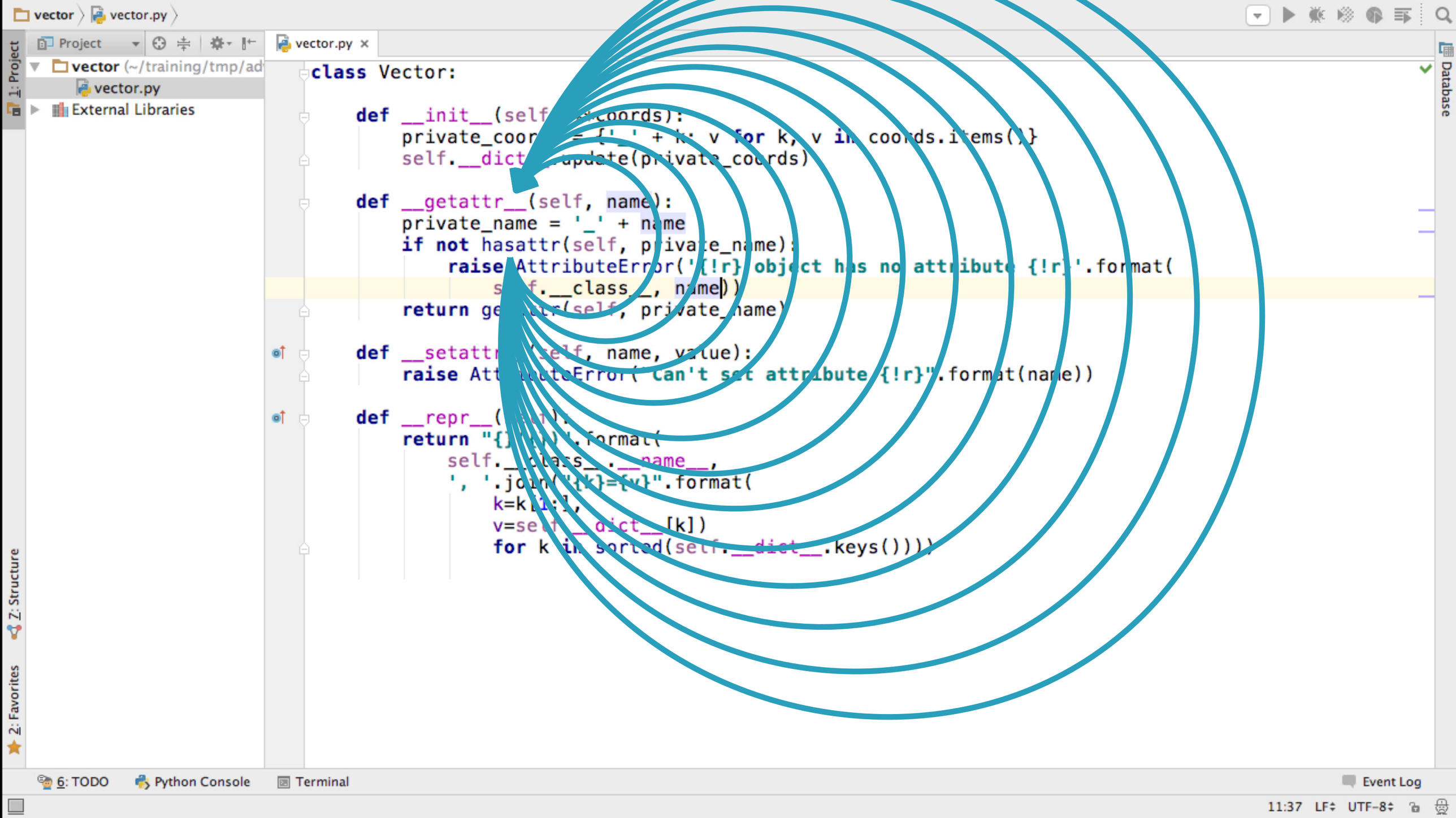

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self, **coords):
private_coords = {'_' + k: v for k, v in coords.items()}
self.__dict__.update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return getattr(self, private_name)
def __setattr__(self, name, value):
raise AttributeError("Can't set attribute {!r}".format(name))
def __repr__(self):
return "{}({})".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
1: Project
2: Favorites
7: Structure
Database
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8

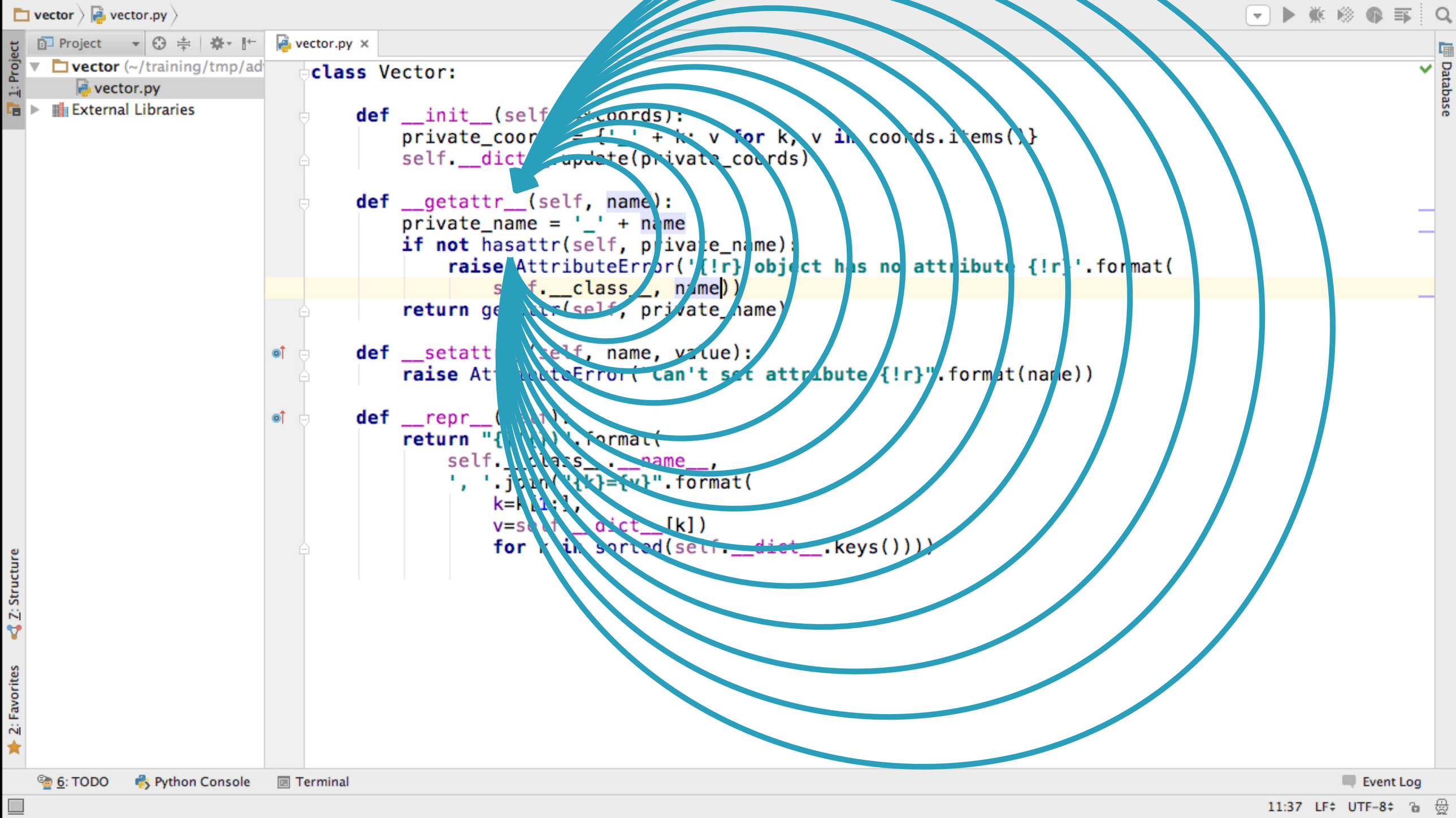

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self
coords):
private_coor
= {'_' + k: v for k, v in coords.items()}
self.__dict
update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
s
f.__class__, name))
return ge
r(self, private_name)
def __setattr
self, name, value):
raise Attr
teError("Can't set attribute {!r}".format(name))
def __repr__(se
):
return "{}({}".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
Database
7: Structure
2: Favorites
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8

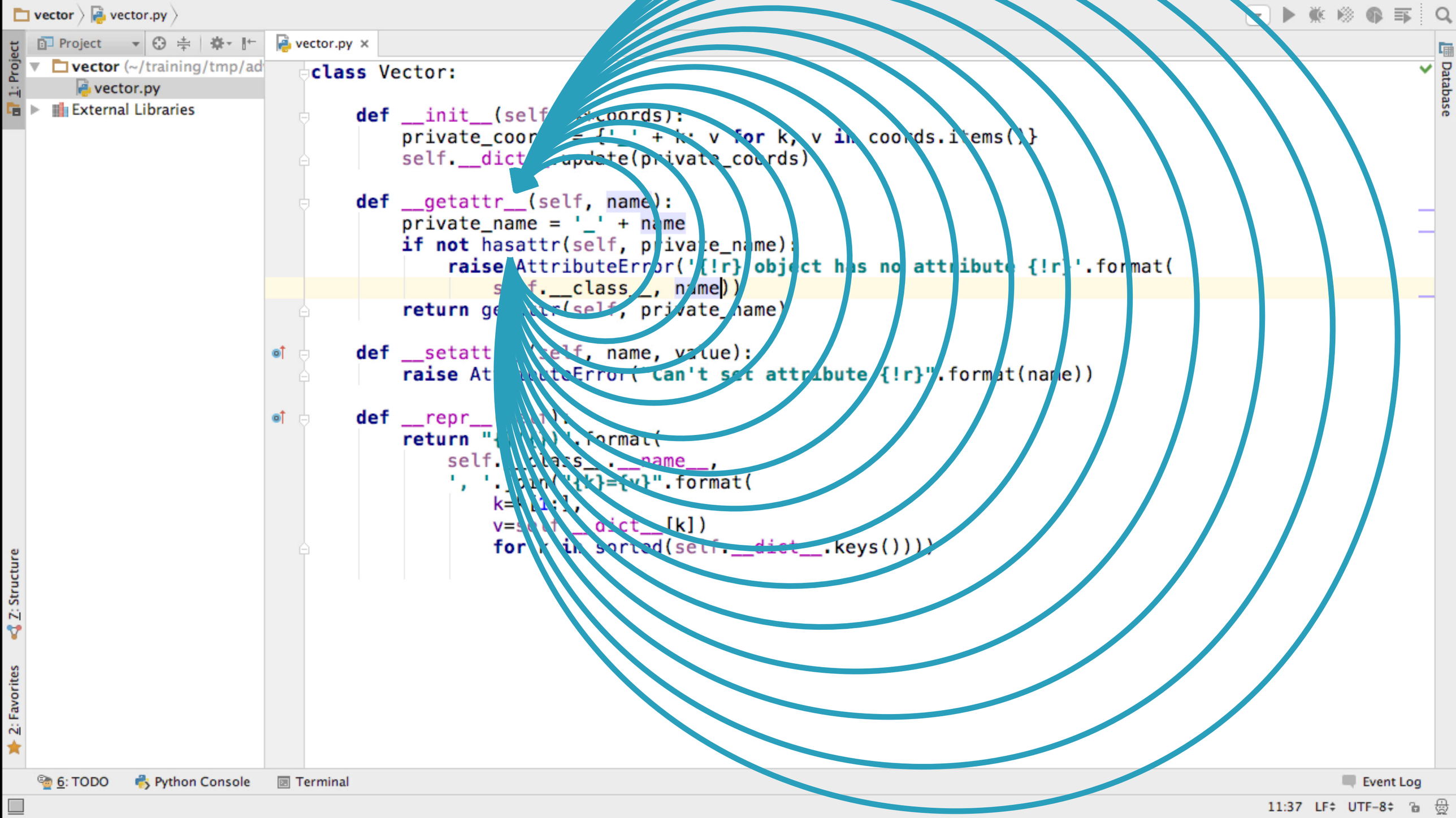

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self
coords):
private_coor
= {'_' + k: v for k, v in coords.items()}
self.__dict
update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return ge
r(self, private_name)
def __setattr
self, name, value):
raise Attr
teError("Can't set attribute {!r}".format(name))
def __repr__(se
return "{}(
.format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
Database
7: Structure
2: Favorites
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8

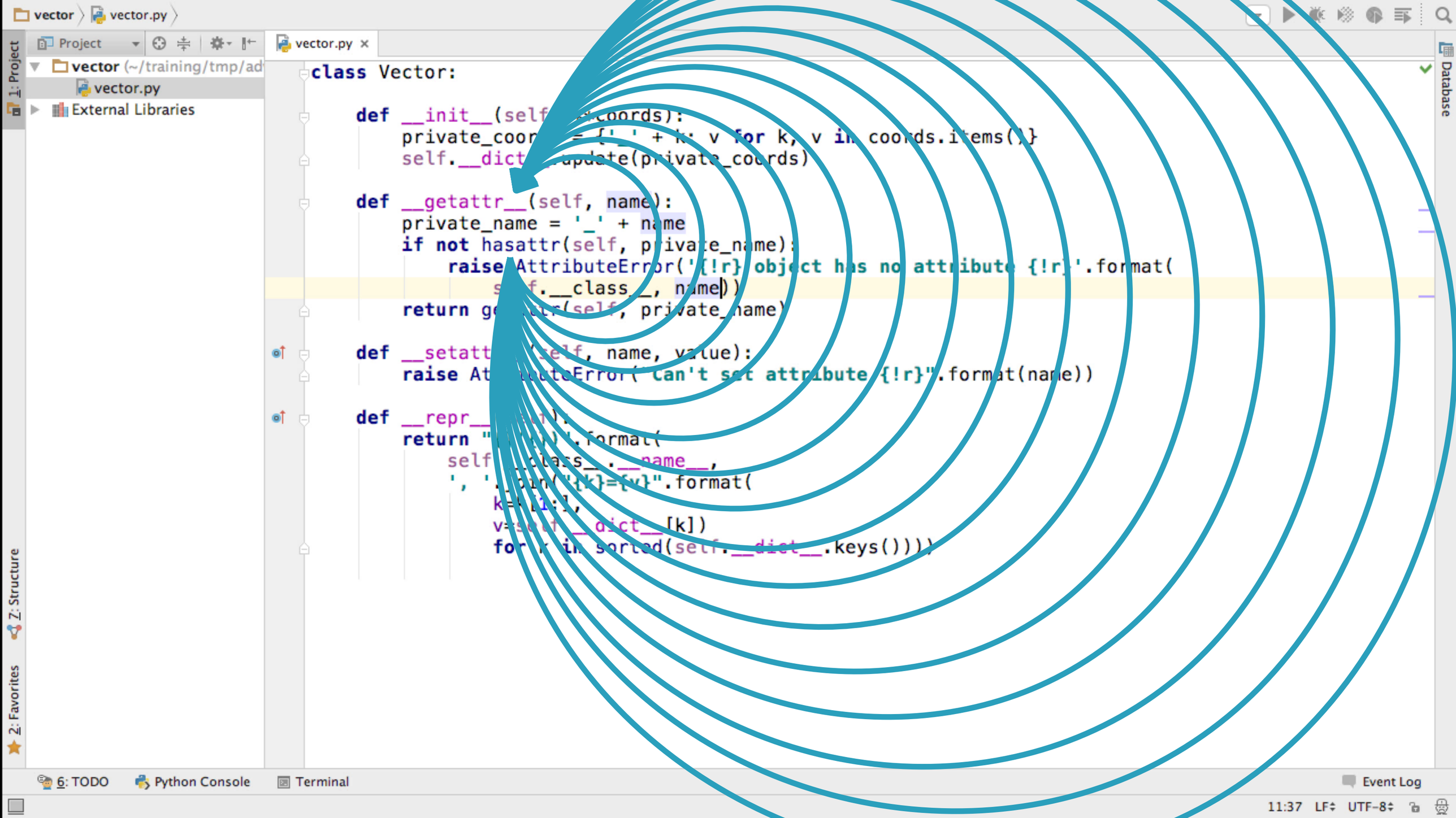

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self
coords):
private_coor
= {'_' + k: v for k, v in coords.items()}
self.__dict
update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return ge
r(self, private_name)
def __setattr
self, name, value):
raise Att
teError("Can't set attribute {!r}".format(name))
def __repr__(s
):
return "{}(
).format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8
1: Project
2: Favorites
7: Structure
Database

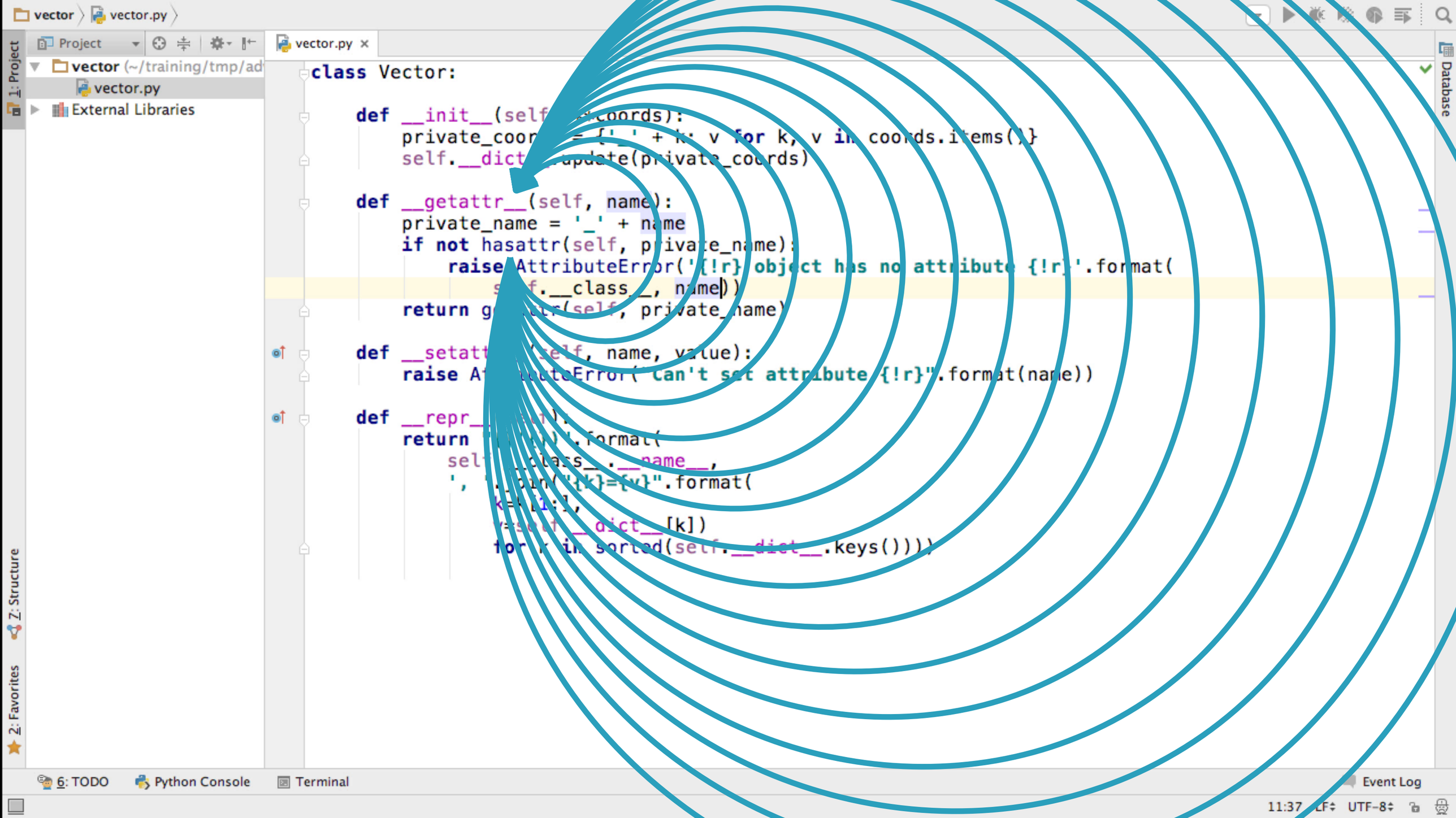

vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self
coords):
private_coor
= {'_' + k: v for k, v in coords.items()}
self.__dict
update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return ge
r(self, private_name)
def __setattr
self, name, value):
raise Att
teError("Can't set attribute {!r}".format(name))
def __repr__(s
):
return "{}
}".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
Database
1: Project
7: Structure
2: Favorites
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8


vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self
coords):
private_coor
= {'_' + k: v for k, v in coords.items()}
self.__dict
update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return ge
r(self, private_name)
def __setattr
(self, name, value):
raise Att
uteError("Can't set attribute {!r}".format(name))
def __repr__(
):
return "{}
}".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
Database
2: Favorites
7: Structure
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8


vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self
coords):
private_coor
= {'_' + k: v for k, v in coords.items()}
self.__dict
update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return ge
r(self, private_name)
def __setatt
(self, name, value):
raise At
uteError("Can't set attribute {!r}".format(name))
def __repr__(
):
return "{
}".format(
self.__class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
Database
2: Favorites
7: Structure
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8


vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self
coords):
private_coor
= {'_' + k: v for k, v in coords.items()}
self.__dict
update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return ge
r(self, private_name)
def __setatt
(self, name, value):
raise At
uteError("Can't set attribute {!r}".format(name))
def __repr__
):
return "{
}".format(
self.
class__.__name__,
', '.
in("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for
in sorted(self.__dict__.keys())))
1: Project
2: Favorites
7: Structure
Database
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8


vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self
coords):
private_coor
= {'_' + k: v for k, v in coords.items()}
self.__dict
update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return ge
r(self, private_name)
def __setatt
(self, name, value):
raise At
uteError("Can't set attribute {!r}".format(name))
def __repr__
):
return "
}'.format(
self
class__.__name__,
', '.
in("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for
in sorted(self.__dict__.keys())))
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8
1: Project
2: Favorites
7: Structure
Database


vector
vector.py
Project
vector (~/training/tmp/ad
vector.py
External Libraries
vector.py
class Vector:
def __init__(self
coords):
private_coor
= {'_' + k: v for k, v in coords.items()}
self.__dict
update(private_coords)
def __getattr__(self, name):
private_name = '_' + name
if not hasattr(self, private_name):
raise AttributeError('{!r} object has no attribute {!r}'.format(
self.__class__, name))
return g
r(self, private_name)
def __setatt
(self, name, value):
raise A
tteError("Can't set attribute {!r}".format(name))
def __repr__
):
return
}'.format(
sel
class__.__name__,
', '.join("{k}={v}".format(
k=k[1:],
v=self.__dict__[k])
for k in sorted(self.__dict__.keys())))
6: TODO
Python Console
Terminal
Event Log
11:37
LF
UTF-8
1: Project
2: Favorites
7: Structure
Database

Overriding `__delattr__()`

Customising Attribute Storage

Direct Versus Indirect Access to `__dict__`

Using `vars()` to Access `__dict__`

Unpythonic?

Pythonic

Using `vars()` to Access `__dict__`

Unpythonic?

```
obj.__dict__
```

Pythonic

Using `vars()` to Access `__dict__`

Unpythonic?

```
obj.__dict__
```

Pythonic

```
vars(obj)
```

Using `vars()` to Access `__dict__`

Unpythonic?

```
obj.__dict__
```

Pythonic

```
vars(obj)
```

```
len(collection)
```

Using `vars()` to Access `__dict__`

Unpythonic?

```
obj.__dict__
```

```
collection.__len__()
```

Pythonic

```
vars(obj)
```

```
len(collection)
```

Using `vars()` to Access `__dict__`

Unpythonic?

```
obj.__dict__
```

```
collection.__len__()
```

```
self.__dict__['color'] = [red, green, blue]
```

Pythonic

```
vars(obj)
```

```
len(collection)
```

Using `vars()` to Access `__dict__`

Unpythonic?

```
obj.__dict__
```

```
collection.__len__()
```

```
self.__dict__['color'] = [red, green, blue]
```

Pythonic

```
vars(obj)
```

```
len(collection)
```

```
vars(self)['color'] = [red, green, blue]
```

Overriding `__getattribute__`

Customising **Attribute Access**

Customising **Attribute Access**

Fallback

`__getattr__()`

Invoked **after** requested attribute/ property not found by normal lookup

Customising **Attribute Access**

Fallback

`__getattr__()`

Invoked **after** requested attribute/ property not found by normal lookup

Preemptive

`__getattribute__()`

Invoked **instead of** normal lookup

Attribute Lookup for Special Methods

Where Are Methods Stored?

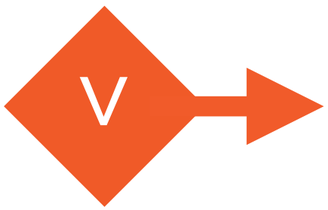
Where are the Methods?

Where are the Methods?

object
reference

Where are the Methods?

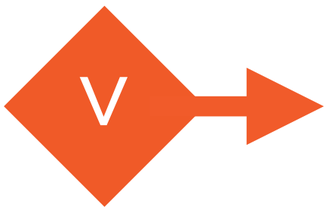
object
reference



Where are the Methods?

object reference

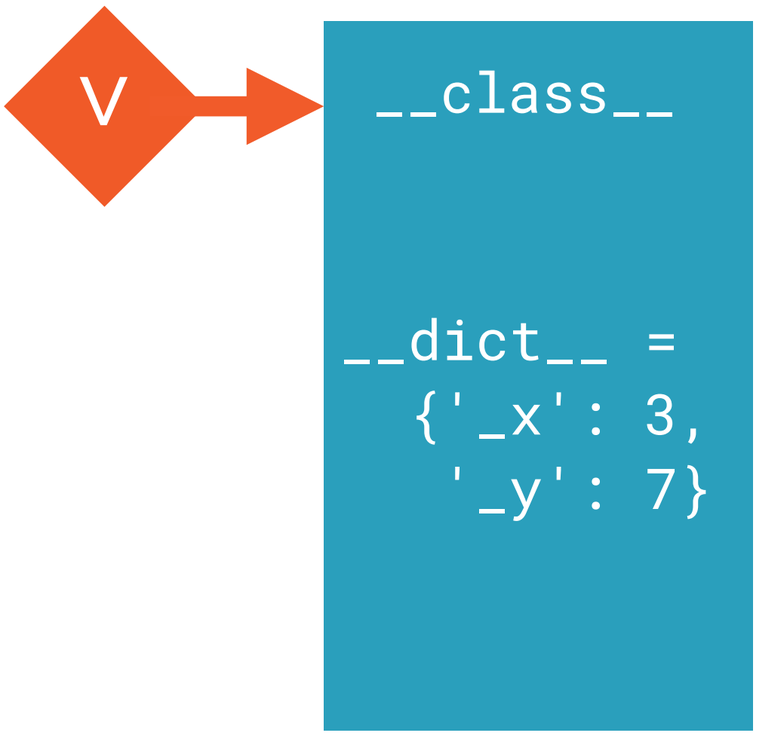
Vector instance



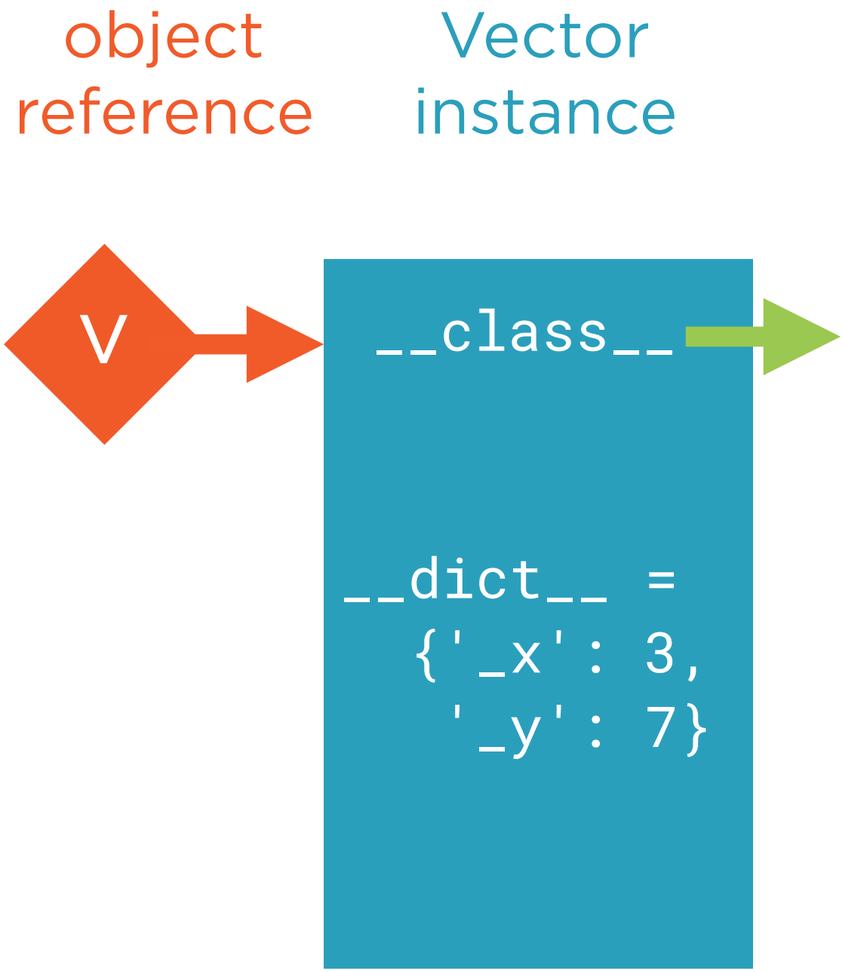
Where are the Methods?

object reference

Vector instance



Where are the Methods?

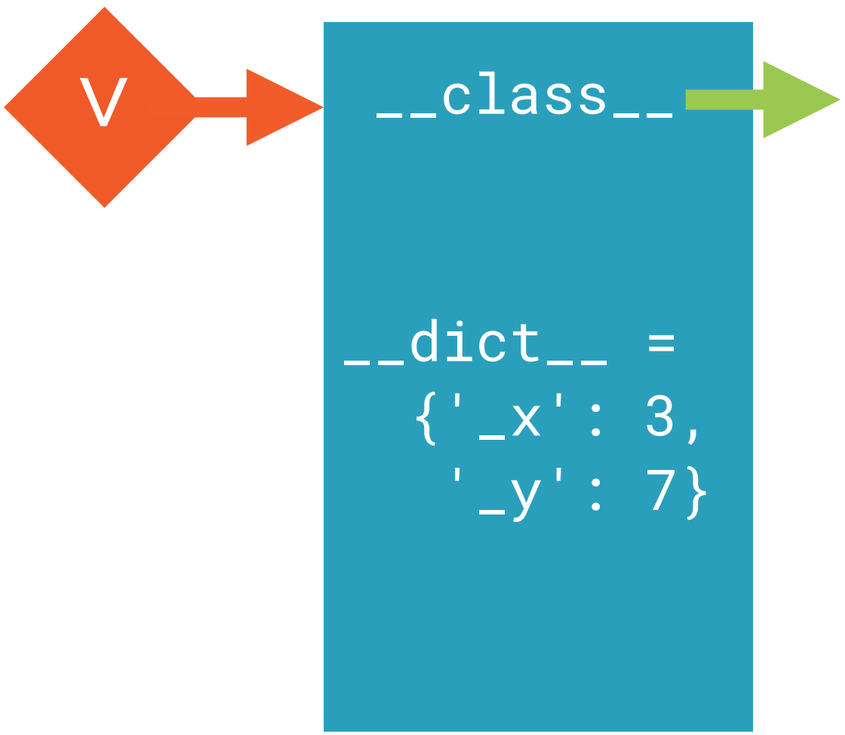


Where are the Methods?

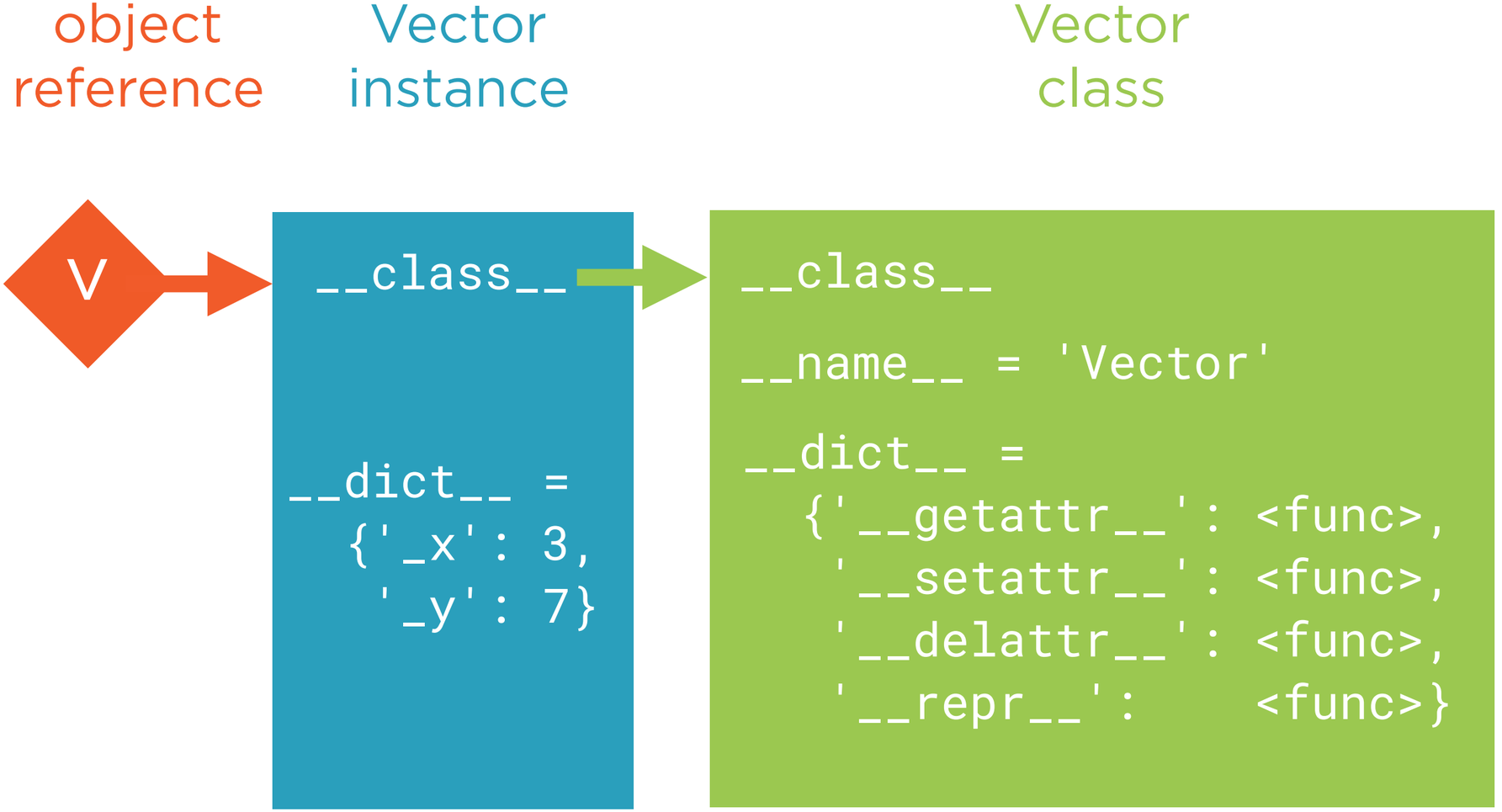
object reference

Vector instance

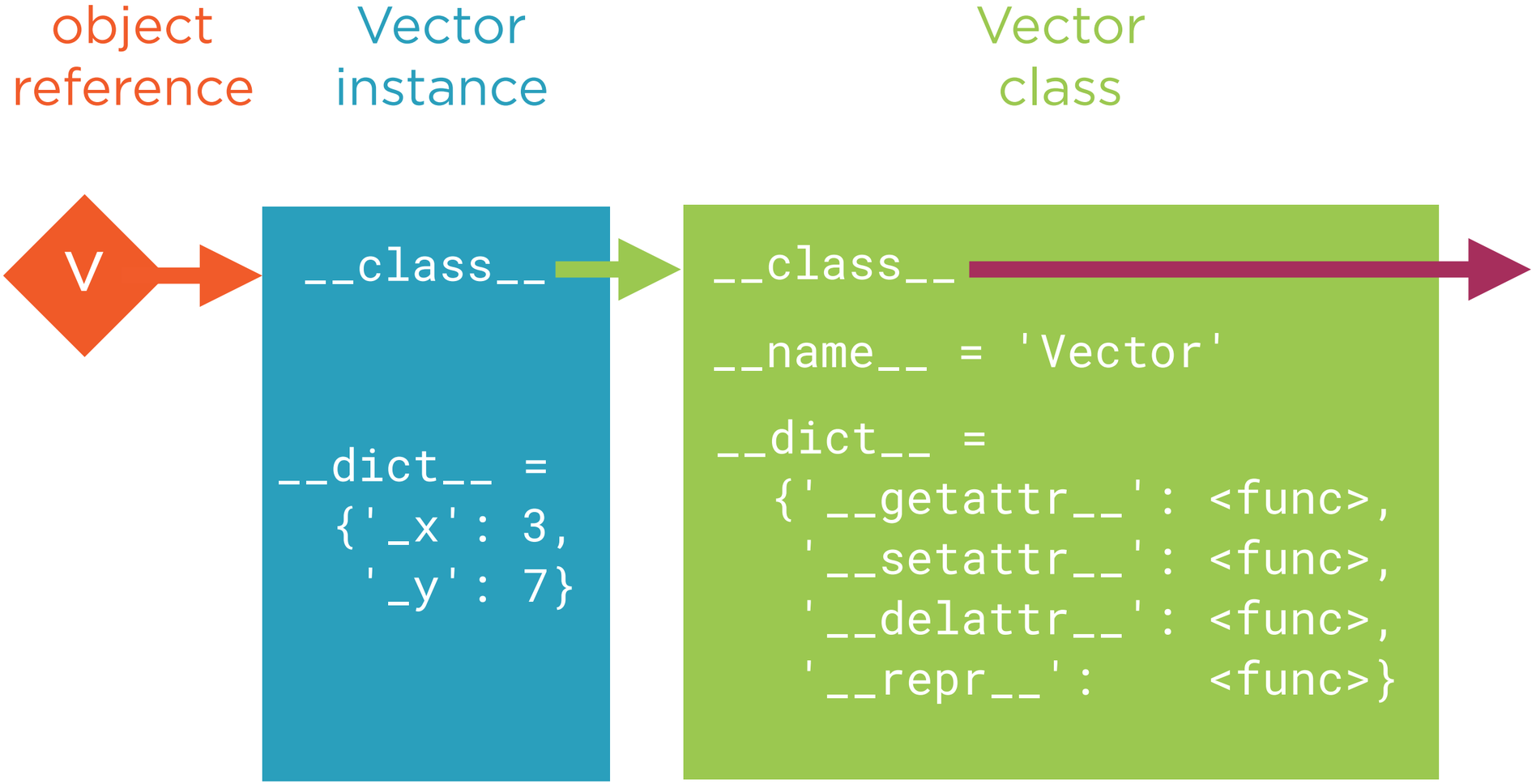
Vector class



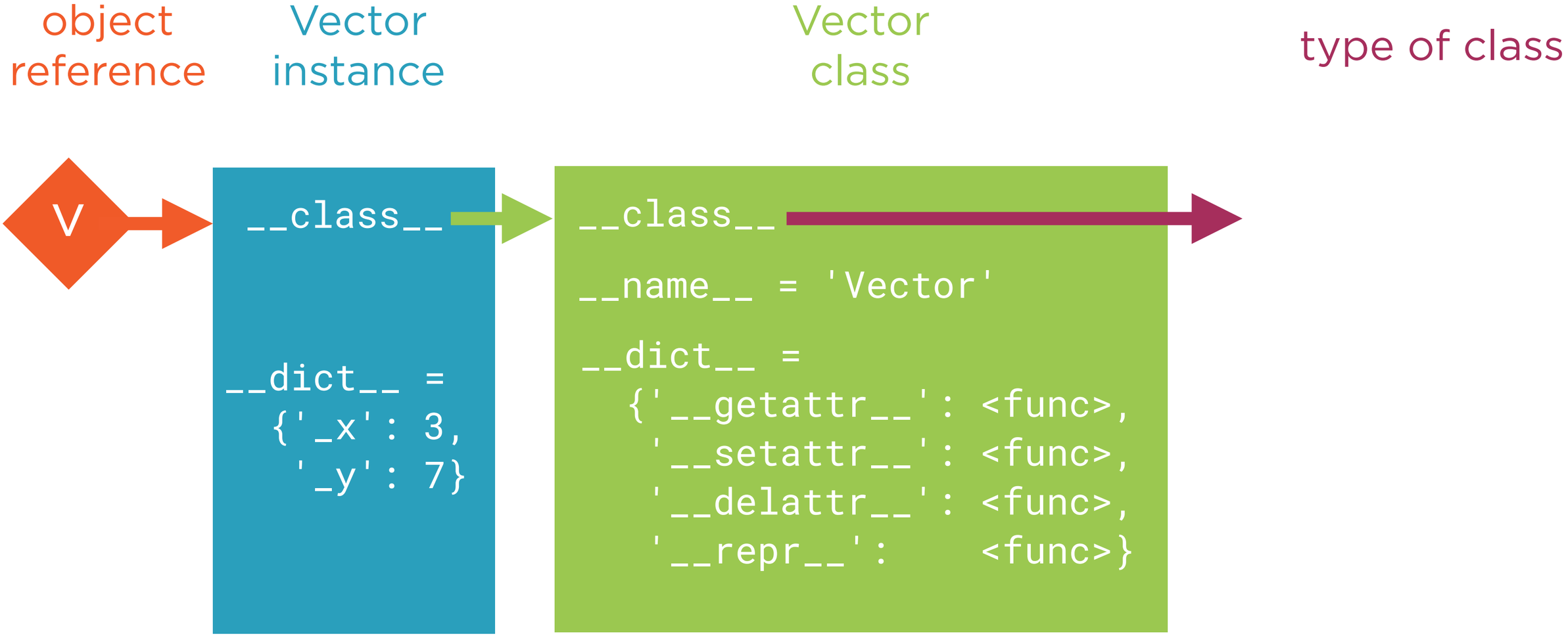
Where are the Methods?



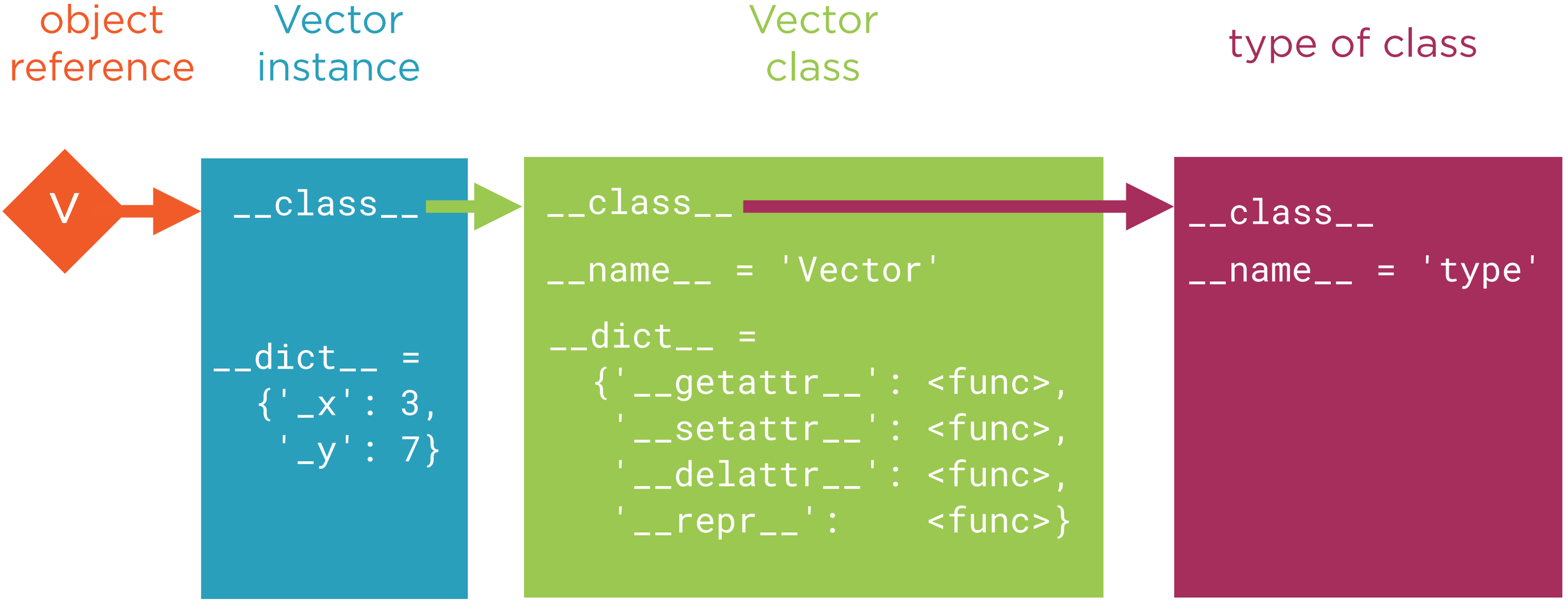
Where are the Methods?



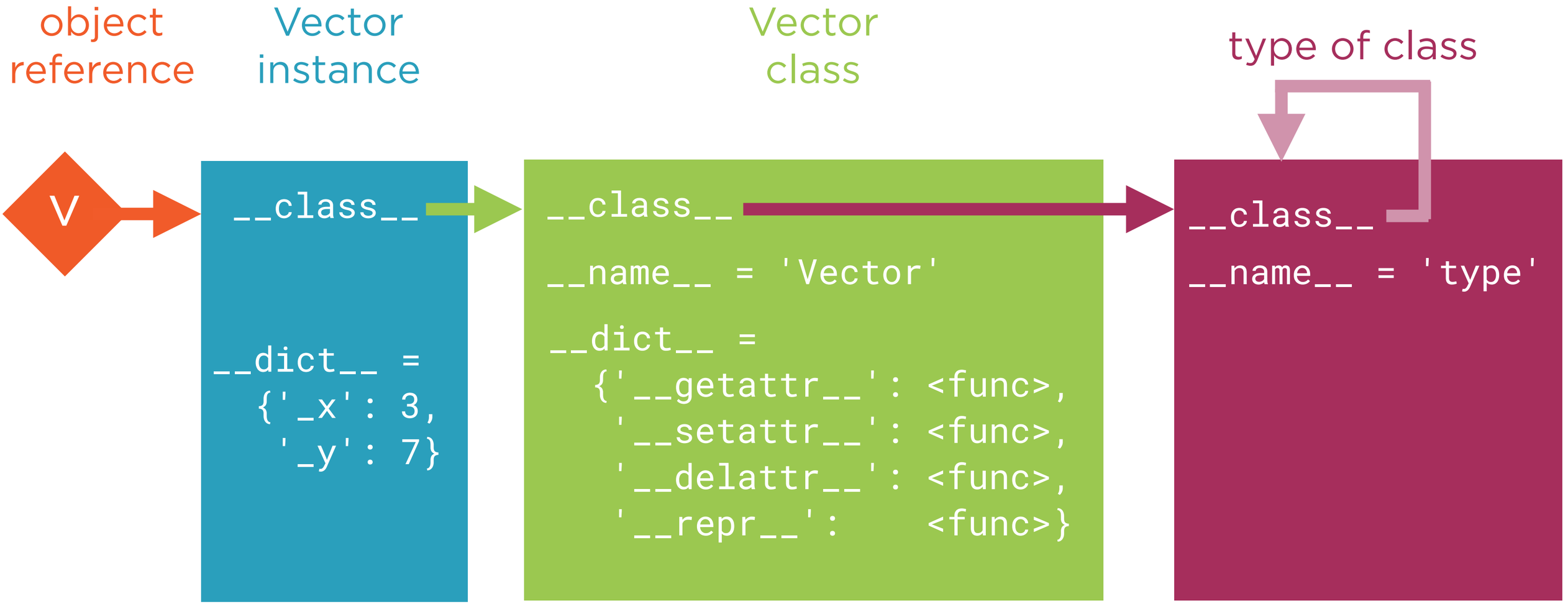
Where are the Methods?



Where are the Methods?



Where are the Methods?




100 GB

Trading Size for Dynamism with Slots

Summary

Summary

Summary

Instance attributes stored in `__dict__`

Summary

Instance attributes stored in `__dict__`

`__dict__` can be manipulated directly

Summary

Instance attributes stored in `__dict__`

`__dict__` can be manipulated directly

Fallback lookup with `__getattr__`

Summary

Instance attributes stored in `__dict__`

`__dict__` can be manipulated directly

Fallback lookup with `__getattr__`

Attribute assignment with `__setattr__`

Summary

Summary

Summary

`hasattr()` **built-in calls** `__getattr__`

Summary

`hasattr()` **built-in calls** `__getattr__`

Intercept with `__getattribute__`

Summary

`hasattr()` **built-in calls** `__getattr__`

Intercept with `__getattribute__`

Methods in `__class__.__dict__`