In [12]:
```
class f:
    pass
class e:
    pass
class d:
    pass
class c(f,d):
    pass
class b(d,e):
    pass
class a(c,b):
    pass
a.mro()
```

Out[12]:
```
[__main__.a,
 __main__.c,
 __main__.f,
 __main__.b,
 __main__.d,
 __main__.e,
 object]
```

In [10]:
```
class a:
    pass
class b:
    pass
class c(a,b):
    pass
class d(b,a):
    pass
class e(c,d):
    pass
```

```
---------------------------------------------------------------
--------------
TypeError                                 Traceback (most rec
ent call last)
<ipython-input-10-523566d08bf7> in <module>
      7 class d(b,a):
      8     pass
----> 9 class e(c,d):
     10     pass

TypeError: Cannot create a consistent method resolution
order (MRO) for bases a, b
```

In [16]:
```
class A:
    pass
class B:
    pass
class C:
    pass
class X(A,B):
    pass
class Y(C,A,B):
    pass
class Z(A):
    pass
class P(Z,Y,X):
    pass
P.mro()
```

Out[16]:
```
[__main__.P,
 __main__.Z,
 __main__.Y,
 __main__.C,
 __main__.X,
 __main__.A,
 __main__.B,
 object]
```

- pzyxacb
- xab
- ycab
- za
- p+za+ycab+xab
- pzyc+a+ab+xab
- pzycx+a+ab+ab
- pzycxabo

In [18]:
```
class P:
    pass
class Q:
    pass
class R(P,Q):
    pass
class S(Q):
    pass
class T(S,R):
    pass
a=T()
T.__mro__
```

Out[18]:
```
(__main__.T, __main__.S, __main__.R, __main__.P, __main__.Q,
object)
```

- rpq
- sq
- tsr
- t+sq+rpq
- ts+q+rpq
- tsr+q+pq
- tsrp+q+q
- tsrpqo

In [21]:
```
class A: pass
class B: pass
class C: pass
class D:pass
class E:pass
class K1(C,A,B): pass
class K3(A,D): pass
class K2(B,D,E): pass
class Z( K1,K3,K2): pass
```

- k1cab
- k3ad
- k2bde
- z+k1cab+k3ad+k2bde
- zk1ck3+ab+ad+k2bde
- zk1ck3ak2bdeo

In [22]:
```
1 Z.mro()
```

Out[22]:
```
[__main__.Z,
 __main__.K1,
 __main__.C,
 __main__.K3,
 __main__.A,
 __main__.K2,
 __main__.B,
 __main__.D,
 __main__.E,
 object]
```

Comaparison

__eq__ -> ==
__lt__ -> <
__gt__ -> >

In [23]:
```
1  class comp:
2      def __init__(self,d):
3          self.d=d
4      def __eq__(self,other):
5          return self.d == other.d
6      def __it__(self,other):
7          return self.d > other.d
8
9      def __gt__(self,other):
10         return self.d < other.d
11
12
```

In [27]:
```
1 a=comp(10)
2 b=comp(20)
3 print(a == b)
4 print(a > b)
5 print(a < b)
```

```
False
True
False
```

In [ ]:
```
1
```