

# MRO in Python

Linearizzazione della gerarchia delle classi (grafo per ereditarietà multipla) con algoritmo C3 per stabilire quale metodo chiamare in caso di ambiguità.

Caratteristiche:

- determinismo;
- conservazione dell'ordinamento locale: l'ordine in cui sono specificate le classi da estendere detta la priorità delle implementazioni dei metodi;
- monotonia: le sottoclassi devono precedere le superclassi nell'ordinamento.

Tuttavia,

- esistono gerarchie di classi non linearizzabili (errore a runtime);
- il programmatore deve essere consapevole di quale implementazione del metodo chiamato verrà usata, che per gerarchie complesse non è banale (`C.mro()` stampa l'ordine di linearizzazione).

```
>>> class A: pass
>>> class B(A): pass
>>> class C(A, B): pass
```

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

A deve venire prima di B per la regola di conservazione dell'ordinamento, dopo per quella di monotonia.