

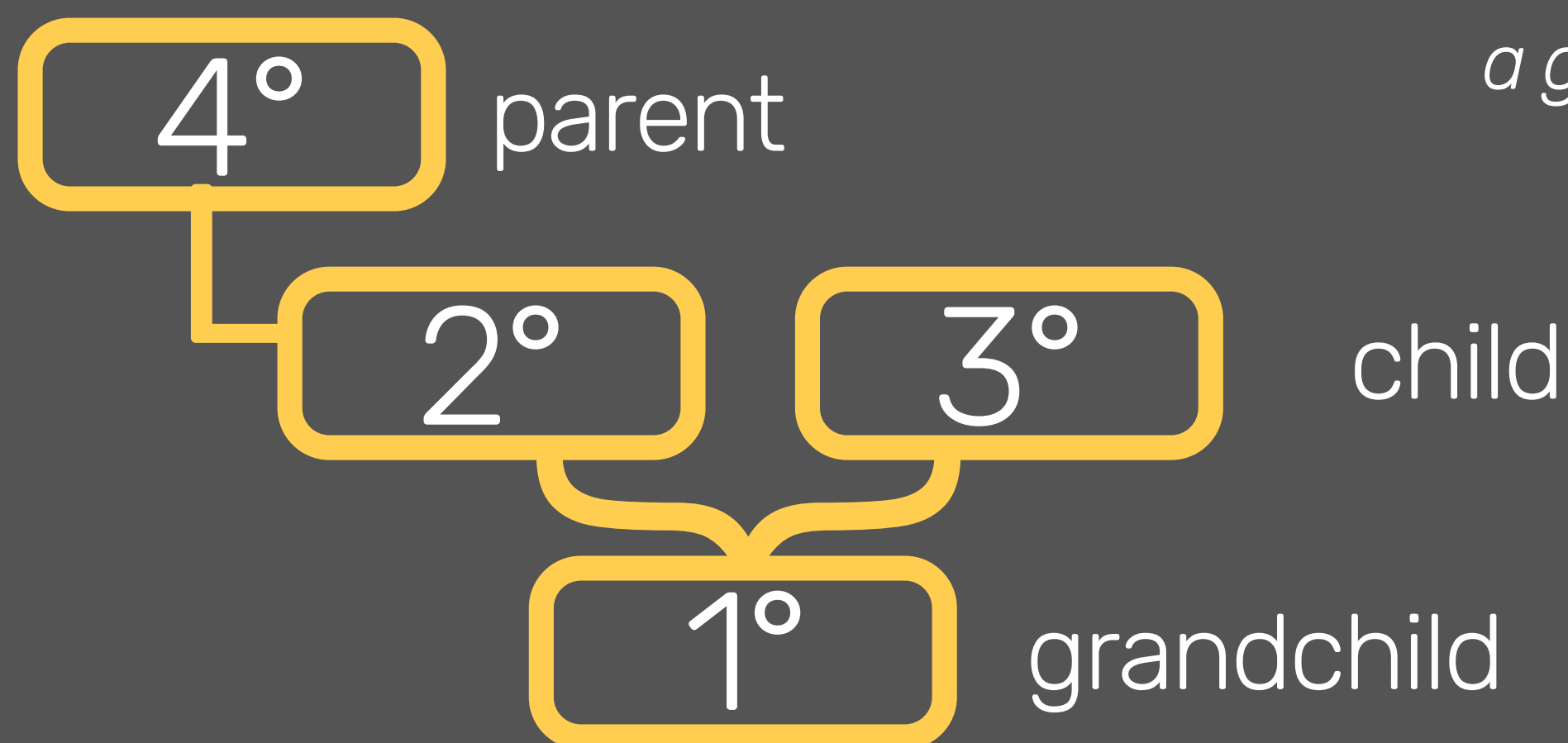
extended inheritance

"Child" classes that inherit from parent classes can create **new** methods or **override** those of the "parent" class. Likewise, a "child" class can inherit from one or more classes, and in turn pass on inheritance to "grandchild" classes.

If several superclasses have the **same** attributes or methods, the subclass can only inherit from *one* of them. In these cases, Python will give priority to the class that is furthest to the *left*.

In the same way, if the same method is inherited from the "parent" and "child" class, the "grandchild" class will have preference for the one closest to it (following our scheme, it will take it from the "child" class).

search order of
a method:



*a given method will be searched first
in the class itself, and if not found,
super classes will be explored*

*...we can continue to extend the
inheritance diagram with the same logic*

Class.__mro__

returns method resolution order

super().__init__(arg1, arg2,...)

inherit attributes from superclasses compactly