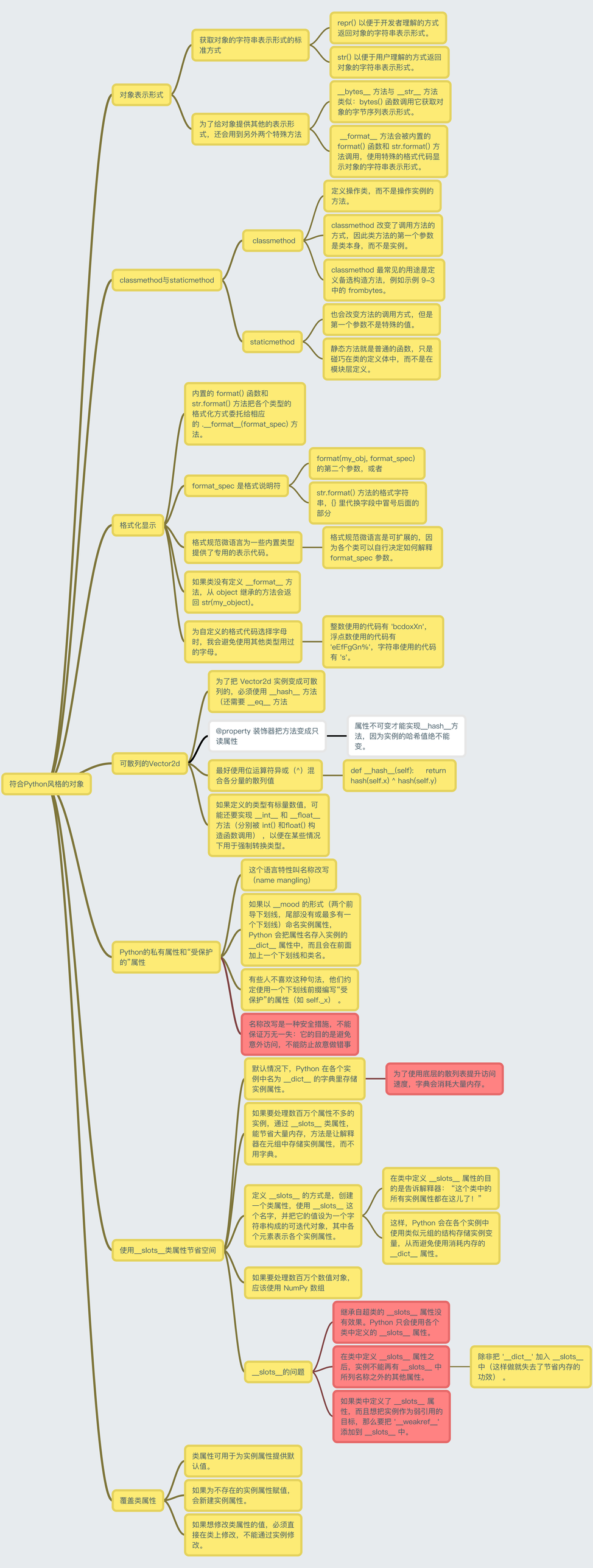
符合Python风格的对象
对象表示形式
获取对象的字符串表示形式的标准方式
repr() 以便于开发者理解的方式返回对象的字符串表示形式。
str() 以便于用户理解的方式返回对象的字符串表示形式。
为了给对象提供其他的表示形式，还会用到另外两个特殊方法
__bytes__ 方法与 __str__ 方法类似：bytes() 函数调用它获取对象的字节序列表示形式。
__format__ 方法会被内置的 format() 函数和 str.format() 方法调用，使用特殊的格式代码显示对象的字符串表示形式。
classmethod与staticmethod
classmethod
定义操作类，而不是操作实例的方法。
classmethod 改变了调用方法的方式，因此类方法的第一个参数是类本身，而不是实例。
classmethod 最常见的用途是定义备选构造方法，例如示例 9-3 中的 frombytes。
staticmethod
也会改变方法的调用方式，但是第一个参数不是特殊的值。
静态方法就是普通的函数，只是碰巧在类的定义体中，而不是在模块层定义。
格式化显示
内置的 format() 函数和 str.format() 方法把各个类型的格式化方式委托给相应的 .__format__(format_spec) 方法。
format_spec 是格式说明符
format(my_obj, format_spec) 的第二个参数，或者
str.format() 方法的格式字符串，{} 里代换字段中冒号后面的部分
格式规范微语言为一些内置类型提供了专用的表示代码。
格式规范微语言是可扩展的，因为各个类可以自行决定如何解释 format_spec 参数。
如果类没有定义 __format__ 方法，从 object 继承的方法会返回 str(my_object)。
为自定义的格式代码选择字母时，我会避免使用其他类型用过的字母。
整数使用的代码有 'bcdoxXn'，浮点数使用的代码有 'eEfFgGn%'，字符串使用的代码有 's'。
可散列的Vector2d
为了把 Vector2d 实例变成可散列的，必须使用 __hash__ 方法（还需要 __eq__ 方法
@property 装饰器把方法变成只读属性
属性不可变才能实现__hash__方法，因为实例的哈希值绝不能变。
最好使用位运算符异或（^）混合各分量的散列值
def __hash__(self): return hash(self.x) ^ hash(self.y)
如果定义的类型有标量数值，可能还要实现 __int__ 和 __float__ 方法（分别被 int() 和float() 构造函数调用），以便在某些情况下用于强制转换类型。
Python的私有属性和“受保护的”属性
这个语言特性叫名称改写（name mangling）
如果以 __mood 的形式（两个前导下划线，尾部没有或最多有一个下划线）命名实例属性，Python 会把属性名存入实例的 __dict__ 属性中，而且会在前面加上一个下划线和类名。
有些人不喜欢这种句法，他们约定使用一个下划线前缀编写“受保护”的属性（如 self._x）。
名称改写是一种安全措施，不能保证万无一失：它的目的是避免意外访问，不能防止故意做错事
使用__slots__类属性节省空间
默认情况下，Python 在各个实例中名为 __dict__ 的字典里存储实例属性。
为了使用底层的散列表提升访问速度，字典会消耗大量内存。
如果要处理数百万个属性不多的实例，通过 __slots__ 类属性，能节省大量内存，方法是让解释器在元组中存储实例属性，而不用字典。
定义 __slots__ 的方式是，创建一个类属性，使用 __slots__ 这个名字，并把它的值设为一个字符串构成的可迭代对象，其中各个元素表示各个实例属性。
在类中定义 __slots__ 属性的目的是告诉解释器：“这个类中的所有实例属性都在这儿了！”
这样，Python 会在各个实例中使用类似元组的结构存储实例变量，从而避免使用消耗内存的 __dict__ 属性。
如果要处理数百万个数值对象，应该使用 NumPy 数组
__slots__的问题
继承自超类的 __slots__ 属性没有效果。Python 只会使用各个类中定义的 __slots__ 属性。
在类中定义 __slots__ 属性之后，实例不能再有 __slots__ 中所列名称之外的其他属性。
除非把 '__dict__' 加入 __slots__ 中（这样做就失去了节省内存的功效）。
如果类中定义了 __slots__ 属性，而且想把实例作为弱引用的目标，那么要把 '__weakref__' 添加到 __slots__ 中。
覆盖类属性
类属性可用于为实例属性提供默认值。
如果为不存在的实例属性赋值，会新建实例属性。
如果想修改类属性的值，必须直接在类上修改，不能通过实例修改。