1. 文件命名
2. 类文件与类名一致
3. 非类文件采用小驼峰命名法
4. 变量命名
5. 整体采用小驼峰命名法
6. 复杂或作用域内具有同义名的时候，采用by、for、with、to等连接词组
7. 无意义临时变量采用简单命名即可，如循环中使用i~n范围的作为标记
8. 有意义临时变量采用tmp前缀+词组命名
9. 数字与标记命名
10. 通过定义常量进行命名，define或const
11. 除非涉及到变量和函数，否则优先使用const进行定义
12. 全部字母大写，以下划线分词
13. 根据实际应用场景采用语义化、可读性强的命名
14. 同一个数字可多次定义为不同常量，具体由场景语义决定
15. 类命名
16. 类名在项目中唯一
17. 整体采用小驼峰命名法
18. 以类特性缩写作为后缀，如普通类Cls、特性类Tra、抽象类Abs、库类Lib
19. 应用目录/controller/下的类采用appCon作为固定前缀

应用目录/public/下的类采用appPub作为固定前缀

应用目录/extend/下的类采用appExt作为固定前缀

1. z/core/下的类采用zCore作为固定前缀

z/model/下的类采用zMod作为固定前缀

z/public/下的类采用zPub作为固定前缀

z/extend/下的类采用zExt作为固定前缀

1. model/read/下的类采用modRead作为固定前缀

model/write/下的类采用modWrite作为固定前缀

1. resource/controller/下的类采用resCon作为固定前缀
2. component/模块名/组件名/下的类采用com+组件名首字母大写作为固定后缀
3. 类属性命名
4. 遵从变量命名
5. 类方法和普通函数命名
6. 整体采用驼峰命名法
7. 尽可能2-3个单词完成一个命名，如：动词+名词、动词+形容词+名词
8. 复杂命名使用by、for、with、to等连接词组
9. 类方法和普通函数的参数命名
10. 大体上类似于匈牙利命名法
11. 以类型缩写为变量名前缀，包括并不限于：
12. 混合型（主要指字符串和数组混合）x
13. 无固定类型n
14. 路径path
15. 对象obj
16. 数组arr
17. 字符串str
18. 整数int
19. 浮点float
20. 布尔bool
21. 时间time
22. 日期date
23. 时间戳times
24. 年y
25. 月m
26. 日d
27. 时h
28. 分i
29. 秒s
30. 数据库命名
31. 统一以小写命名，以下划线分词
32. 普通单一库采用语义化、可读性强的词组命名
33. 普通分库时，统一前缀，并以标志性词组作为名字部分，如地域、功能、模块等
34. 在分库基础上进一步细分，在原有命名规范下，增加标记后缀，如年份、区号等
35. 数据表命名
36. 统一以小写命名，以下划线分词
37. 使用统一的前缀
38. 以数据类型命名，如订单、用户、sku、spu等
39. 分表中原始结构表以base作为后缀
40. 分表中数据表从原始结构表中复制生成（手动或由程序执行），以分表策略的缩写和标记作为后缀名，包括并不限于：
41. 年度分表：x\_name\_y2017、x\_name\_y2018
42. 上下半年分表：x\_name\_y2017\_h1、x\_name\_y2017\_h2
43. 季度分表：x\_name\_y2017\_s1、x\_name\_y2017\_s2、…3、…4
44. 地区分表：x\_name\_hd、x\_name\_hn、x\_name\_hb
45. 城市分表：x\_name\_gz、x\_name\_bj、x\_name\_sh
46. 手机网段分表：x\_name\_ns135、x\_name\_ns137
47. 字段命名
48. 统一以小写命名，以下划线分词
49. 命名语义化，且可读性强
50. 避免使用关键词