40 个改变编程技能的小技巧

1、将大块代码分解成小函数

2、今日事今日毕，如果没毕，就留到明天。

如果下班之前还没有解决的问题，那么你需要做的，就是**关闭电脑**，把它留到明天。

中途不要再想着问题了！

3、YAGNI 原则

「You aren't gonna need it！」

你自以为有用的功能，实际上是用不到的。除了要求的核心功能，其他功能一概不要部署。

这一原则的**核心思想**是，尽可能快、尽可能简单的将软件运行起来。

4、不必全知全能，但基础一定要扎实

比如学习一些基础知识，SOLID 原则，如何写干净的代码等等。

5、KISS 原则

「Keep it simple，stupid.」or「Keep it stupid simple.」，一种程序设计原则。

大多数系统往往「最简单」，运行效率最高，但实际操作起来并不简单。

6、别想太多

7、被问题/Bug 卡住时，walk away！

不过还是要记得回来。

当你走在去上班、去厕所、去散步的时候，也许就能想到解决方法。

尤其是在与客户、同事生气时，甚至关乎你工作去留的时候，效率会更高。

8、学会写测试代码 TDD

TDD 是一个软件开发过程，它依赖于重复一个很短的开发周期:写一个测试，运行所有的测试，看看新的测试是否失败，写一些代码，运行测试，重构代码，重复。

9、先分解问题再开始写代码

不要不知道怎么做就开始写代码。

10、代码不要死记硬背

要理解逻辑。

11、学好用好 Stack Overflow

如果你复制粘贴一个 Stack Overflow 解决方案，请一定要确保已经理解了它。

12、不要「光学不练」

如果你想学点什么，就去练习，光学是不够的。

13、与小伙伴互相审查代码

研究别人的代码，让别人时常研究你的代码。

互帮互助，共同进步。

14、Don’t Reinvent The Wheel

「不要重新发明轮子。」

充分利用已有的经验和成果，避免不必要的投入和浪费。

15、你的代码是最好的文档

16、懂得如何搜索

对于这一点，你需要有经验以及读很多书，才知道要找什么东西。

17、写代码时要之后维护考虑

你的代码将来需要你自己或者别人来维护。

所以，写代码的时候要考虑到读者，而不是想成为最聪明的人，让它读起来就像在读一个故事。

18、复制粘贴

用谷歌、百度解决错误的最好方式就是「复制粘贴」。

19、不要放弃

到最后，不管用什么方式，问题肯定会解决。

20、休息、休息再休息

解决问题的最好方法是有一个安稳的心态。

21、学习软件设计模式

设计模式是软件设计中常见问题的解决方案。每一种模式就像一个蓝图，你可以自定义来解决代码中常见的设计问题。（不要重复发明轮子。）

22、使用集成工具

尽可能实现自动化。

23、Do code katas.

「Code kata」是编程中的一种练习，可以帮助程序员通过练习和重复来提高他们的技能。

24、依赖注入是一个要求

编程到一个接口，而不是 implementation。

所谓依赖注入，就是组件之间的依赖关系由容器在应用系统运行期来决定，也就是由容器动态地将某种依赖关系的目标对象实例注入到应用系统中的各个关联的组件之中。

25、重构-测试-重构

重构是一种对现有代码进行重组的技术，在不改变其外部行为的前提下，改变和改进其内部结构。

26、及时寻求帮助

不要浪费时间。

27、Practice makes perfect.

熟能生巧。

28、不必太在意评论

虽然有时评论可以帮到你，但不要太过在意。他们可能已经过时了。

29、了解你的开发环境

了解你的开发环境，并 invest 一个足够强大的环境，如，IntelliJ。

30、重复使用组件

31、考虑相关限制

在开发网络应用时，要考虑到移动优先以及相关的功率和带宽限制。

32、不要过早优化或重构

更重要的是尽快拥有一个最低限度可行的产品。

33、不要投机取巧

千万不要为了节省几分钟的时间而选择效率低下的捷径方式。

「Every time you code, give your best！」

34、遵循规定的标准

35、用户不是技术人员

当你开发你的 UI 时，需要考虑到这一点。

36、坚持使用 Github 或 bitbucket

可以进行小规模、频繁的 **git 提交**。

37、记录所有关键部分

记录系统日志比调试代码更好。

38、风格保持一致

如果你使用一种风格，请总是使用相同的风格。

如果你和更多的人一起工作，对所有的团队都使用相同的风格。

39、Don’t stop learning

但比起学习新语言或框架，更要注重学习软件开发的基础知识。

40、patience and love

最后，对你正在做的事情保有足够的耐心和热爱。