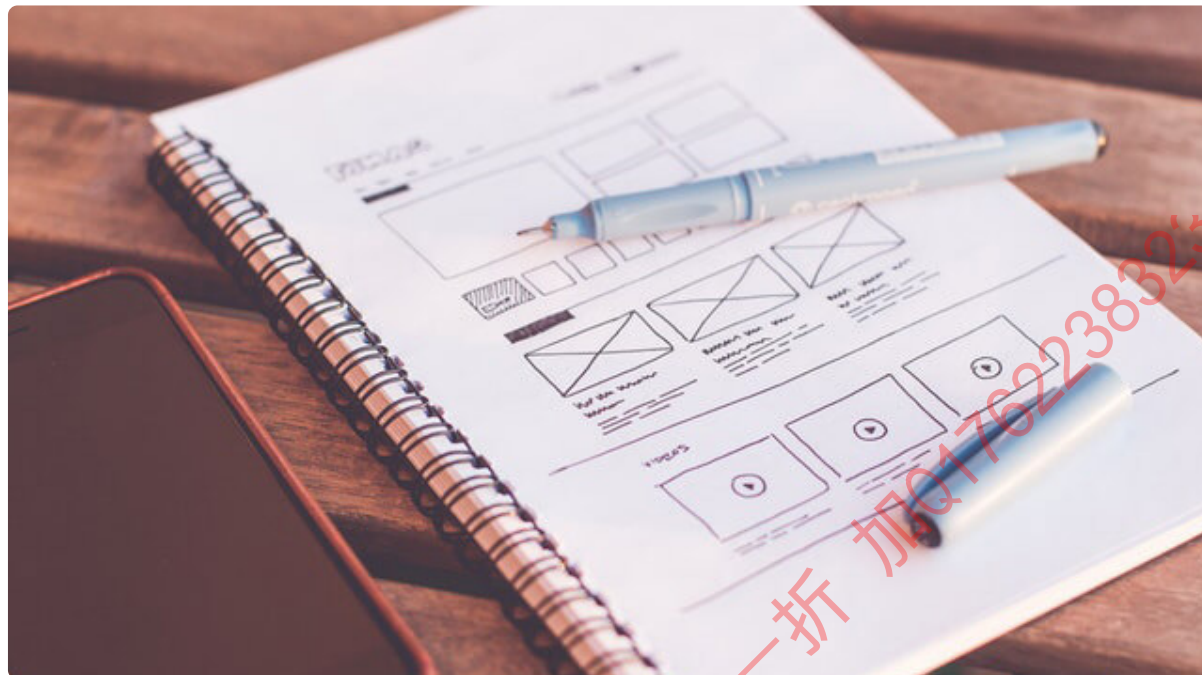


02 什么是 UI 框架？

更新时间：2019-06-28 17:14:37



“读书给人以快乐、给人以光彩、给人以才干。”

——培根”

UI 是什么？

先来说下 UI，这俩字母是 User Interface 的缩写，一般翻译成“用户界面”。UI 最主要的功能就是建立用户和系统后台之间的联系，系统后台通过 UI 把数据转换成可视化的内容展示给用户，同时用户也要通过 UI 把操作指令（包括数据）传给系统后台。

对 UI 不太熟悉的同学一听到这个概念，可能会觉得它的作用就是怎么把一个产品做的漂亮，所以 UI 设计师经常被人叫成美工。而事实上 UI 应该是负责“交互”和“视觉”这两方面的工作，这两部分内容构成了产品的用户体验。

用户体验里最重要的应该是这个产品好不好用，也就是“交互”这部分，这其中包括产品功能是否完善，产品流程是否设计的合理，使用是否方便，操作是否流畅等。在一些大公司里，为了保证产品好用，还会专门设置交互设计师这个职位，专门做交互部分的设计工作。

“视觉”在用户体验中也很重要，但它一定要建立在好用的基础上。如果一个用户在使用产品的功能上出现问题，这个用户很可能就流失了。但如果产品用起来没问题，只是界面不那么漂亮，一般也就是被吐槽一下，不会造成功能缺失所导致的那么严重的后果。在我看来，一个系统的交互舒适程度要比美观更重要。

UI 的贡献者

在一个项目的开发周期中，UI 的形成需要多个角色进行配合。产品经理、UI 设计师、前端开发人员都是用户界面重要的贡献者，甚至后端开发人员在做数据处理时也都需要考虑到 UI 的影响。下面大概说下这几个贡献者的各自分工：

产品经理，项目的主导者。主要考虑的是系统里功能性的部分。产品经理需要了解一个系统面向的用户群体，以及这个群体的特点和使用习惯，然后设计出满足用户需求的产品。在这个阶段产品经理会产出产品的原型，包括功能设计和大致的功能分布。产品的设计决定了产品的实用性。

UI 设计师，用户体验的设计者。当产品经理确定产品功能后，UI 设计师就该进场了。设计师会和产品沟通，依照产品模型再做交互和视觉上的优化，最后依照设计结果产出 UI 设计图。UI 设计图是 UI 设计师的最终产出结果，但这个工作更重要的是在设计上，不仅仅是画设计图。所以，千万别再把人家叫美工了，因为叫错被打了也就忍着吧。

前端开发人员，从 UI 设计图到实际产品的实现者。前端开发人员最主要的工作是把产品和 UI 确定的最终设计稿变成可以运行的程序。前端开发人员是一个施工方的角色，但可以从代码的角度对产品经理和 UI 设计师提出有价值的意见。并且在施工中，好的前端开发人员也会注意到用户体验，比如优化加载速度，按钮点击范围，列表滚动的顺滑程度等。

后端开发人员，幕后英雄。我们一般觉得后端开发人员只管处理数据就可以了，和 UI 没什么关系。但当我们确定好一个页面要显示什么内容的时候，还是要找后端开发的同学来帮着我们把这个页面需要的数据整合到一起，这就需要后端人员在做数据结构设计的时候就把这些因素考虑进去。

UI 框架

前面介绍了 UI 的情况，从现在开始，就要进入我们这个课程的主题—UI 框架。我们这个课程主要讲页面切图部分内容，也就是 HTML 和 CSS 的部分。

当我们在做切图时，最常见的是把设计图用 HTML 和 CSS 逐个还原出来。这个过程中我们会发现写了很多冗余的代码。比如 Button，每个页面用到的地方都要写上一堆 Button 的样式，慢慢的为了加快开发，就开始大量复制粘贴，最后就导致整个代码里有几十个很类似的 Button 样式。

页面中其实有很多像 Button 这种会用到很多次的组件，这时候我们会想到把这些具有共性的东西抽离出来，变成通用的组件。这样再写页面的时候，只要引入这些通用的组件，就不用再在页面里重复写这些内容了。经过不断的提炼，就形成了 UI 库或 UI 框架。

使用框架的好处也是显而易见的：

1. **标准化**。UI 设计师应该有统一的设计标准的，当我们制作出一套前端框架的时候，就可以把设计标准转化成开发标准。如果设计师那边出的设计图并没有什么标准，比如同样功能同样位置的按钮，有的用 50px 高度，有的用 55px 高度，这时候前端同学就有义务去和 UI 设计师沟通下，把设计标准确定下来，保证同类组件有统一的尺寸和样式。
2. **提高开发效率**。使用了 UI 框架以后，所有通用的组件的开发量就省下来了，开发人员只需要做每个页面里那些没有共性部分的开发。
3. **方便扩展**。在使用了通用的框架后，我们具有共性的内容都集中在了一起。所以当我们要对产品做样式改造或扩展的时候，只需要对框架进行升级就可以了。不会像原始的那种开发方式中，哪怕改个按钮颜色，也要去每个页面里都改一遍。
4. **提高页面加载速度**。这个优点其实是被动的，当抽离出 UI 组件后，就会少了很多冗余代码，这样总体文件体积会变小。同时抽离出来的通用组件通过做缓存，加载速度还会进一步提高。但说这个优点是被动的，是因为这并不是我们做 UI 框架的初衷，通常不会为了提高加载速度而做一个 UI 框架。

现有移动端 UI 框架

现在市场上已经有很多的 UI 框架，也有很多是免费的，我们可以直接拿来用。下面介绍几个移动端常见的 UI 框架：

- **Bootstrap**，强大的 UI 框架。它同时支持 PC 端和移动端，还可以做到自适应；组件非常全面；稳定性好，不容

易出问题；还提供了基于 jQuery 的 JS 组件库（据说5.x版本后会剔除对jQuery的依赖）。它的缺点就是体积太过庞大，太大而全。

- **WeUI**，微信官方出品的。这个框架设计比较简洁美观，提供了移动端一些基础的组件，体积很小，没有提供 JS 组件。我比较喜欢这个设计，这个课程里一些样式也是参考了这个框架。
- **VUX-UI**，基于 VUE 的 UI 库。它实际是在 WeUI 的基础上做了一些改造，并使用 VUE 实现了一些 JS 组件库。
- **Ant Design Mobile**，基于 React 的移动端 UI 框架。它和 VUX-UI 的定位是一样的，只不过是基于 React。另外这个UI 框架的设计是单独开发的，没有基于 WeUI。
- **Frozen UI**（腾讯）、**Mint UI**（饿了么）、**Cube UI**（滴滴）。这些都是各个大公司内部的 UI 框架，做的成熟了，就做了开源提供给外部的开发者使用。

前面介绍的是一些比较常用的，实际还有很多其他设计的不错的 UI 框架，这里就不细致介绍了。在挑选 UI 框架的时候一定要先看文档，看是否符合自己的需求。再就是尽量用一些大众的框架，可以减少踩坑的机率。

小结

这一节了解了 UI 的概念，制作 UI 的参与者，UI 框架的作用以及一些常用的 UI 框架。希望同学们看完这节能对 UI 和 UI框架相关的内容有个比较具体的印象。最后强调下，不要把 UI 设计师叫成美工！