

Learning from the mistakes others have made



CHILDREN CAN BE GREAT AT ASKING QUESTIONS ABOUT THINGS THAT MOST ADULTS TAKE FOR GRANTED, AND LIKE TO CHALLENGE ASSUMPTIONS.

While it's a great sign of a curious and reasoning mind, it can be overwhelming, and you can't really learn (or teach) everything at once. Some things are better to be taken on faith in the short term, until you fully understand the issue.

That brings up a term you'll be hearing quite a bit in this class: "**Best practice**". It's often said that bad programs can be written in any language, and we've found that to be true (at least in every language we've seen). Over time, developers learn that some habits are better than others, in other words that some habits, like [avoiding redundancy and repetition](#), tend to make a program more clear and easier to understand and maintain than other habits, like [using goto statements](#). It could also be about performance, i.e. in a given language doing a task one way may be faster than another.

A carpenter will tell you that if you want to hammer a nail, it's best to do it in as few strikes as possible (e.g. 2 or 3). That may not be obvious to non-handy people like me, but I've been told that's the best way by people with a lot more experience than I, so I'll take it on faith (at least for now).

There are sometimes arguments about which are the best practices. New techniques are discovered, new ideas are born, and sometimes, fashions change. For our purposes and the duration of this course, when we use the term "best practice" you can trust that it is, even though we may not be able to explain it at that point in the course, so you'll want to make it a habit.