On the training manifesto

Sam Dutton, Developer Advocate at Google, wrote A Manifesto For Teaching Web Development as a way to write down his ideas for explaining HTML, CSS and JavaScript to absolute beginners.

As an instructional designer I'm in agreement with pretty close to 100% of Sam's ideas but would further tailor some of them to make them better. I understand that Sam wrote bullet points but these

> For live classes, consider flipped classroom techniques. Get students to learn at home (with specific goals and requirements) then complete activities with supervision in the classroom. Be upfront about the level of commitment expected. Provide a clear path: how to get help, where to find out more, what to do next.

I like the idea of using a flipped classroom as our primary teaching methodology but I think we also need to be upfront on the following:

Be clear on what a flipped classroom is and what the expectations are I've had experiences where half the class did not complete pre-requisite assignments and it changed the dynamics of the class for everyone: the students who had completed the assignments grew bored when I had to cover the material that was in the assignments and some of the students who did not complete the assignments felt that we were rushing over content that they were not familiar with.

You could state that material in the 'home-review' portion is required knowledge for a face to face or online workshop. This puts the ball back on the students' court in terms of expectations. Students may not need to do the 'home-review' if you know the content but you should at least glance it to make sure they're in the same page with how you're teaching the content.

Either provide the content to review before class or do rigurous vetting of the content before you give to students.

In learning to work with Express (a Node.js based server framework) I've come accross at least 3 different ways to accomplish a task. What will happen if students picked different ways to accomplish a task, for example: setting up a new Express

application or configuring routes? Which way is correct? Why should students learn one way over the other ones? That's for the facilitator (and the Subject Matter Experts she works with) to decide.

It is unlikely but there may be resources that are factually incorrect; this used to be the problem with W3Schools documented in w3fools. If people access inncorrect informtion and that's all they learn

If you provide your handouts and a list of resources in advance students will all be in the same page and will have time to work through the material ahead of the lecture/demonstration/discussion.

> Make examples as simple as possible. Get rid of all redundant content and code.

Make sure that the code examples work either on their own or when added to an existing script.

You should consider that students, and other people who may come upon the code outside the content of teaching a workshop or doing flipped classes, may want to see the code work and have a fully developed example to work with and modify.

> Many people new to coding don't use or understand hierarchical file systems. When you start, make sure everyone is comfortable with files and folders. Many problems for beginning web developers are caused by something 'in the wrong place'.

Provide a sample project with files and folders for people to get familiar with. If your pre-training assessment tells you it's necessary you may want to consider providing a structure that they can become familiar with. It may also make sense to introduce some terminology at this point. A folder is a directory and a relative path is different than an absolute one.

> Beginners make different kinds of mistakes than experienced developers. Be aware of this when you're debugging. Show beginners what to watch out for: missing angle brackets (`<div class="foo"`), missing equals signs (`<div class"foo">`), missing closing tags, copy-pasted curly quotes instead of

> straight quotes.

Leave assumptions at the door I've always thought it better to assume that students don't know the content. This makes it easier to start from scratch and for them to gradually geet aclimated to the content. This point dovetails well with the next one.

> Tailor your content and teaching to your audience. Try to work out how to reach a wide variety of people and capture their interest via problems they want or need to solve.

If possible do a survey of your student population as part of your needs assessment. The needs assessment is part of the analysis phase in the ADDIE Instructional Design process.

There are other questions but for the purposes of this discussion I will only worry about the audience.

Who is the audience? Can we find self-reported skills and comfort levels? In an ideal world I would have a survey for participants before the training begins or they get access to the content if a self-paced instruction.