1. Benefits of planning training and development

Understanding your objectives. Planning affords an opportunity to identify learning objectives and establish how to recognise achievement of them.

Preparing material. Training material needs to be prepared in context, with an understanding of the expected audience and learning objectives.

Time management and pacing. The course should be structured so as to provide fixed points for instructors (to ensure they don't lose track or get lost) and to provide room for adaptation based on learning speeds.

Accommodating varying instructor quality. A course may be given by different instructors with different levels of understanding of the material, or an instructor may be sub-par (e.g. not feeling well) on the day. In these situations, it may be necessary to fall back more on pre-planned material.

Taking environmental factors into account. Facilities available may vary, and the planning process can take into account various different presentation venues and arrange for a “plan B” for situations where something goes wrong.

2. Plan for a training course lasting half a day or more

This plan is for Day 3 of a software development workshop for scientific researchers in the audio and music fields.

Days 1 and 2, which have been planned and taught a number of times already by a third party, cover basic software design, development, and testing, for scientists working in various fields, using the Python language. Day 3 focuses specifically on audio and music processing and is a new course which this author is helping to design.

Learning objectives

By the end of the course, learners will have:

- reviewed the fundamentals of audio representation and processing by computer (including the theory of sampled audio files and audio playback);
- been introduced to standard Python modules for handling audio processing;
- successfully worked through at least one audio processing exercise in which they took an algorithm whose theory they were already acquainted with, and implemented it with standard modules in Python using good practices for style, construction, documentation and testability of their code
- consolidated the material learned during Days 1 and 2 of the workshop, on subjects such as organisation of Python programs and writing unit tests, and obtained an understanding of how to apply these principles in their own audio- and music-related research work.

Day plan

The course is for up to 30 students working on laptops, in pairs and occasionally larger groups. There will be two primary presenters, and 5-6 relatively experienced developers will be present in the room to provide hands-on help.

The day runs from 9am to 4.30pm, to be scheduled as follows:

Time period	**Activity**
0900-0930	Coffee, finding seats, setup

0930-1030	**Introduction** and **worked example**: Basics of audio representation and processing; open an audio file and retrieve its data using Python
1030-1045	**Pair exercise**: Write program to open and read an audio file (file provided)
1045-1100	Coffee and biscuits
1100-1115	Review exercise
1115-1200	**Worked example**: Read audio in blocks, apply a filter, write out to file
1200-1230	**Pair exercise**: Write program to read a music audio file and perform note onset detection (the detection method will be given)
1230-1315	Lunch
1315-1345	Review exercise
1345-1500	**Worked example**: Synthesising and playing a sound **Group exercise**: Make a noise! **Worked example**: Synthetic audio as input for unit tests **Pair exercise**: Unit-test onset detector developed earlier
1500-1520	Coffee and biscuits
1520-1540	Review exercise
1540-1630	Wrap up (and allowance for overrunning)

Some details about the course structure:

- There will be two instructors, but only one will be leading the class at any given time: one will handle the morning and the other the afternoon sessions. The “inactive” instructor will remain on-hand as a helper.
- The entire course has a “live learning” structure. The active instructor is always leading the class, and they have a computer connected to a projector with all introduction and worked-example material coded “live”. Learners may follow along on their own laptops, and obtain assistance from helpers placed around the room at any time.
- The usual rhythm will be: the instructor steps through a worked example “live”; learners follow along with the explanation of what is going on; learners then work on an exercise in pairs or small groups; each exercise, except for a short one mid-afternoon, is followed by a break during which learners can discuss and dissect the exercise material.
- Exercises are modulated so as to pick up from the example material that preceded them, but with slightly different subject material so that learners are not required to have verbatim copied the example material before they can get started on the exercise.
- A reminder sheet will be available at the end of the day, giving pointers to online material reinforcing the work done during the day; however, handouts will not be used during the course itself – in order to ensure learners remain focused on the task in hand.

Adapting to learners' needs

Learners will all have some audio and music background, but will have widely varying levels of experience in audio and music software programming. For a majority of learners, we expect that the first two days of the workshop will have been their first experience with the Python language.

Several aspects of the course structure are intended to make it easier to accommodate learners with varying backgrounds, skills, and learning speeds:

- Learners will be encouraged to sit in groups with others working in a similar subject area or from the same research group, and to discuss their work as they go along.
- There will be a high proportion of hands-on helpers to learners (about 1:5) in the room, and the helpers will be distributed throughout the seated area. Fast learners will be encouraged to act as ad-hoc helpers as well.
- Most exercises will be followed by breaks during which learners will be encouraged to move around and talk to one another, providing opportunities for peer learning.
- Learners will be given sticky notes in two contrasting colours which they can stick on the top of their laptop screen to indicate whether they are keeping pace or whether they need assistance.

Learning evaluation

The course aims to help students who are aware that they wish to improve their software development skills, rather than aiming for a specific qualification. As such it has no marked exercises or specific test component, but the exercises will have a clear goal for completion and learners should find it straightforward to establish their own criteria for success.

Feedback and course evaluation

We can establish how well learners are responding to the material during the course, using some of the adaptation mechanisms listed in the “learners' needs” section above. For example, the use of sticky notes will give us immediate feedback if learners are finding things difficult to follow.

We will also solicit feedback during the wrap-up at the end of the day, asking learners to identify good and bad things about the course, and we will encourage learners to fill in a follow-on questionnaire online.

3. Evaluation of training methods and techniques

Live exposition

Most of the expository material is presented in a hands-on fashion “live”, with the instructor solving a programming problem directly using a computer attached to a projector at the front of the room, and learners optionally following along on their own laptops.

Advantages

- Learners get to see how a program is constructed (in terms of time, as well as in text).
- Because the instructor is required to make an actual working program, the method ensures that vital “auxiliary” techniques such as how to use a text editor and how to build and run a program are not overlooked.
- Learners can seek help as soon as they find something problematic, and instructors can respond directly with new examples.

Disadvantages and risks

- Relatively high bar for technical competence in instructors.
- Instructor must respond to learners and adjust pacing accordingly.
- Stringent technical requirements: there is no fall-back if the projector can't be made to work, for example.

Group and pair exercises

Following each expository segment, learners are presented with a programming task to solve (resulting in a working program on their own computer). They are encouraged to work in pairs, either with two learners at a single computer, or pooling resources informally while producing two solutions. Some exercises focus on groups of three or four learners.

Advantages

- Learners can learn from one another.
- More advanced learners can help others.
- Because learners are encouraged to seat themselves in groups according to their research discipline, they can pool thoughts on how to apply problems in their own disciplines, and retain knowledge better once the course is over.

Disadvantages and risks

- Using real-world exercises requires that everyone present has a functioning laptop with suitable software setup on it. In practice we will try to handle this before the course starts (by providing instructions and help for attendees to ensure they have the proper software beforehand), but some risk remains.
- There is a risk that more passive learners might rely entirely on their neighbours and never actually complete any work. It may be necessary for instructors to give guidance about which individual should lead a task.

4. Evaluation of training resources and visual aids

“Sticky notes” for instant feedback

Learners will be given sticky notes in two contrasting colours which they can stick on the top of their laptop screen – or wave in the air – to indicate whether they are keeping pace or whether they need assistance. This method has been used in Software Carpentry workshops to date and found to be effective.

Advantages

- Provides an immediate way to seek assistance without disrupting the class. Helpers placed around the room can respond to individual learners near to them.
- Can be used for quick shows of hands – show one colour if you're fine with this, another if you're not – in order to gauge the response to material and adjust course pacing.

Disadvantages and risks

- Instructors must not overlook other ways of gauging response: not all learners will be prepared to report their own difficulties, even using this relatively unintrusive method.
- Notes need to be chosen so as to be unambiguous under various types of colourblindness.

Reminder sheet

We will provide a printed sheet with reminders of techniques covered, online resources etc for attendees to take away with them at the end of the course.

5. Health & safety, equal opportunities, regulatory requirements

Attendees will be asked on registration of specific requirements with regard to access, and we will work with the venue to ensure such requirements are met.

The course requires that all attendees have laptop computers, which means providing power to all desks. If this is not built-in, we will need to prepare the room in advance with taped-down extension cables as appropriate.

An area must be provided for attendees to leave bags and spare equipment to avoid blocking access. The venue will have standard access and fire escape requirements, which must be communicated to attendees appropriately. We will also need to take care to comply with room capacity and other venue regulations.

Some learning materials use colour cues, and we need to take care to choose colours appropriately for colourblindness.

The plan includes morning and afternoon coffee breaks and a lunch break. Attendees will be encouraged to get up, move around, and leave the building during breaks.