EXTENDED JAVASCRIPT PROJECTS

Programming projects (or extended programming examples)

There is no single path to learning any programming language, Sure there are some basic concepts you need to get out of the way first, but after that you learn the bits that are relevant to the current problem.

Regardless of which programming language you choose to use, the main stumbling block is usually defining the problem itself, not the programming, or language syntax. These extended projects are intended to give you some practice in outlining a program solution, and to illustrate the importance of the human computer interface (HCI) and how it may influence the design. To build up your knowledge of javascript, each project provides a different path through the Javascript syntax and related functions. All however rely on some basics principles.

> Until you can solve a problem yourself using nothing more than pencil and paper, there is no point attempting a program solution. Your role as programmer, is to *teach* the computer how to solve a problem by giving it an algorithm which it can then blindly follow.The programs you write will therefore be a reflection of, and are limited by, your own knowledge.

The solution to any procedural language problem is based on the the constructs of sequence, selection (decision) and repetition, so before attempting any of these projects, as a first step, you should at least be able

to write simply pseudo code algorithms for each of these. As a second step you need to familiarise yourself with the basic Javascript syntax and be able to convert your pseudo code algorithm into a basic program.

> You will find that apparently simple problems are in fact quite complex, and seemingly complex problems turn out to actually be quite simple. There are of course some classes of problem, which are not amenable to a procedural programming approach.

Each project will outline a problem to be solved, and provide one, or more ideas for how it may be solved; the **program specification**. Your task will be develop a algorithm, using either pseudo code or a graphical technique (e.g. flow chart) which you are more comfortable with. Your solution should also identify any constraints your algorithm imposes and define any necessary user interface. A simple sketch will do which can be implemented using HTML / CSS.

Some of the projects may have additional constraints placed on them, to make them more interesting, and to illustrate**problem decomposition**. That is, take a complex problem and break it down into a number of smaller problems, each of which can be solved independently of one another, then combined to provide a solution to the original problem. There are various structured programming techniques for doing this. i.e. Object Orientated Analysis (OOA) or the Universal Modelling language (UML).

Source : http://www.soslug.org/node/1746