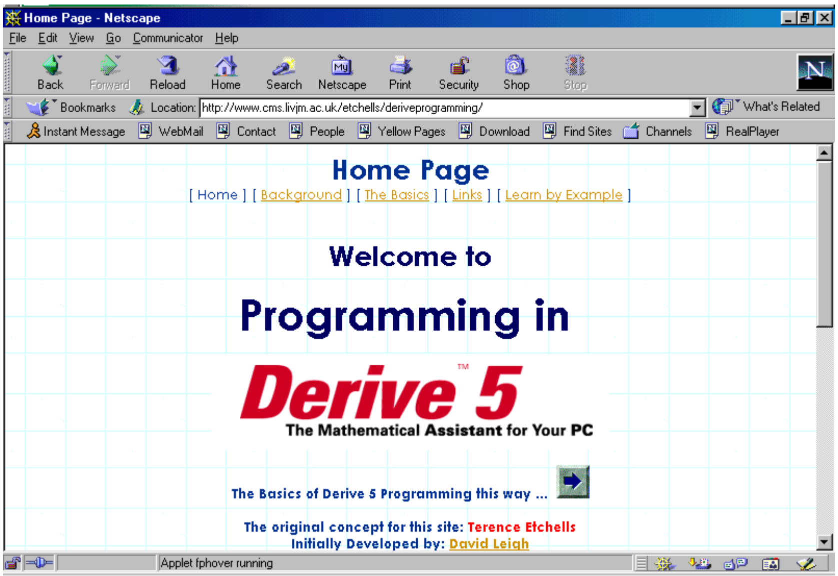
“Welcome to Derive 5 programming”

http://www.cms.livjm.ac.uk/etchells/deriveprogramming/

Hugh C Williams
formerly of:
Nottingham Trent University

hughcwilliams@hotmail.com

When Derive version 5 came out it added some constructs to allow the writing of procedural programs as well as the functional programs it has always supported. Little advice on how to use these facilities is given in the Derive help files or “Introduction to Derive 5” by B. Kutzler and V. Kokol-Voljc, so that although someone used to procedural programming would soon pick up Derive’s version, anyone completely new to programming might struggle.



This web site, written by David Leigh whilst a final year student at Liverpool John Moores University, England, tries to address this problem by providing a basic introduction to procedural programming in Derive 5. The site is now maintained by his project supervisor Terence Etchells.

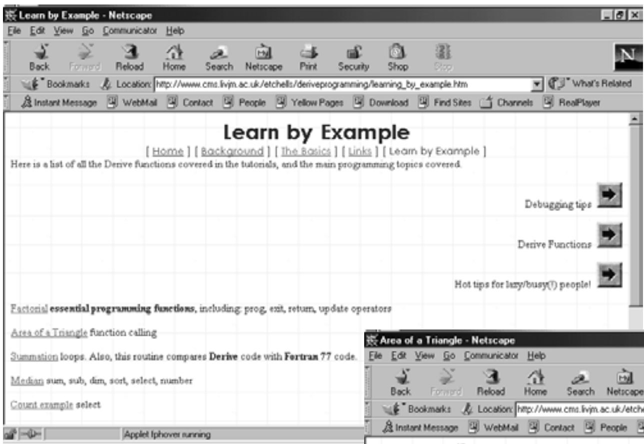
It is an impressive achievement for a final year project although one can see some tidying up, see below, which could usefully be done but was probably precluded by the time constraints of a project.

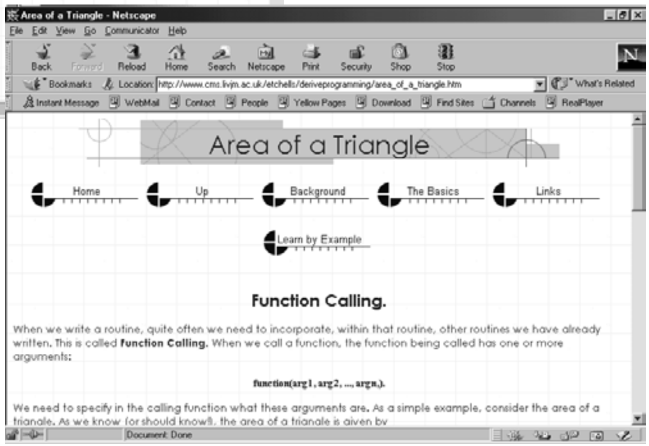
If you are interested in this topic visit the site. You will find it interesting. Do read the author’s own statement of what he is trying before you start by clicking on his name on the title page or on *“Up”* in later links.

After a brief introduction to := , PROG(), LOOP(), RETURN and IF() you are advised to go on to his first sample program, a reworking of the factorial function. Some nine further programs introduce slowly other points about programming and Derive functions which are useful in programming. In the main they are a useful collection. In some cases I would have liked to see a clearer statement of the problem and more explanation as to how the solution was arrived at as the site is trying to teach programming.

The factorial example evolves though several refinements as update operators and Boolean functions are introduced. The final version is a fairly robust and understandable program for factorial.

Probably the weakest element of the site is the ways you can navigate through it. Two if not three ways are used for representing the links. This probably occurred because of experimenting in the project but it would have helped if one style had been used in the final version.





In the early stages of learning programming the student needs to progress through worked examples in a planned sequence. This is contrary to ones natural inclination in a hypertext medium but needs implementing here. It is done in some places using, in effect, Next and Back arrows but not consistently.

Most of the about 10 example programs correctly solve their problems and do so in reasonable style but warning should be given about four cases.

The *"Area of a triangle"* program works correctly but also works with the final *area* omitted. Similarly in *GCD* the first line *z := y* can be omitted.

In the section *"Augmenting"* the program *Gauss2(a, i , c, q)* works by luck not good judgment. David fails to take two bits of advice he gave earlier. One local variable is not defined in the functions arguments *(p)* – here this causes no problems. But *Gauss2* is called as (e.g.) *Gauss(b, d)* wanting *d* to be assigned to *c* not *i* as will normally happen. Why *Gauss2* actually works correctly was a mystery to me until Johann Wiesenbauer kindly explained that as *i* has already been assigned to in the program before *c* is referenced we get *c := d.* Normally this strategy would fail and so should be avoided. Revising the beginning to *Gauss2(a, c, I, p, q)* covers both points. The right hand side of the function definition stays the same.

The program *rout* in the section on identifying convex polygons is wrong. I was suspicious and tried it with further data. On my third attempt I got a concave polygon which was reported as convex. I have not found the exact source of the error. I think it is a poor example of the way to tackle the problem so it would be better to re-program the problem from scratch rather than debugging the existing attempt.

In addition to the example programs there are pages giving: tips; a list of Derive functions useful in programming; links to most of the examples; useful Derive related links and a site map.

In all a useful site which goes a long way towards achieving its objective. With some further development it could become an even more useful resource.

What the Derive community could also do with is a similar site for functional programming in Derive, ideally written by one of the experts in the field!!

Response from provider, Terence Etchells, Liverpool John Moores University

I welcome greatly Hugh Williams' comments and criticisms of this website. It is indeed a 'student project' and is not yet the "finished article", but judging from comments from email comments a useful resource for the Derive user community. I take on board the comments about certain programs and will address these issues forthwith.

In order to make this website a more useful and helpful site I would welcome more examples, criticisms and comments from Derive users. (t.a.etchells@livjm.ac.uk).