

<https://laurentbloch.net/BlogLB/Sources-utiles-pour-debuter-en>



Le langage utilisé pour ces cours :

Sources utiles pour débuter en Scheme

- Zinformatiques - L'informatique : science et industrie - Scheme et les langages Lisp -

Date de mise en ligne : mardi 26 octobre 2004

Copyright © Blog de Laurent Bloch - Tous droits réservés

[Scheme](#)

(voir aussi [ici](#), prononcer *skime*)

est un langage de programmation de la famille [Lisp](#),

elle-même rattachée à la famille plus vaste des langages fonctionnels parmi

lesquels on peut citer [Caml](#) et [Haskell](#). Les versions

récentes de *Scheme* sont dotées en outre de capacités supplémentaires

comme des systèmes d'objets et des fonctions de traitement de textes qui les

rapprochent de langages à objets tels que [Java](#) ou de langages de

script tels que [Perl](#). Cette page est consacrée à mon travail dans ce

domaine, et plus particulièrement au cours que je donne au

[Conservatoire National des Arts et Métiers](#)

dans le cadre d'un [Certificat de Compétence en Bioinformatique](#). Mes autres activités professionnelles sont décrites sur [mon site](#)... professionnel.

Si un simple paragraphe peut convaincre un scientifique ou un ingénieur d'embrasser un nouveau langage, alors ce paragraphe est la [dédicace](#) d'un livre particulièrement beau consacré à *Scheme*,

[Structure and Interpretation of Computer Programs](#).

J'ai enseigné *Scheme* à l'Institut Pasteur (pour un public de

biologistes, donc) pendant plusieurs années et ce cours est devenu un livre intitulé

[Initiation à la programmation avec Scheme](#), qui servira de base à mon enseignement au CNAM. Vous pouvez en lire

quelques extraits ici, avec l'aimable autorisation des Éditions Technip : la [table des matières](#), la [préface](#) que Jacques

Arsac a bien voulu lui consacrer, l'avant-propos qui expose quelques idées à la base de mon enseignement, un

[chapitre consacré aux fondements](#) ainsi que la [bibliographie](#).

J'ai également

écrit [Les systèmes d'exploitation des ordinateurs : histoire, fonctionnement, enjeux](#) qui sert de fil conducteur au

cours d'initiation à l'informatique (B9) et dont les Éditions Vuibert m'ont autorisé à publier le texte intégral sur le [site](#)

[indiqué ici](#).

Vous trouverez sur le site de

[Christian Queinnec](#)

beaucoup de ressources en français, notamment des

[exercices](#) passionnants. Christian Queinnec a consacré plusieurs livres et de nombreux articles à

Lisp et *Scheme*, notamment un [recueil d'exercices](#) en français (avec L. Moreau, D. Ribbens et M. Serrano). Vous consulterez

aussi avec profit le livre (disponible en ligne) qu'il a consacré à [Unix](#) et plus particulièrement au *shell* dont vous aurez tant besoin pour les TP.

Christian Queinnec est aussi l'un des auteurs d'un système

d'enseignement à distance pour *Scheme* disponible sous

forme de [cédéroms télé-chargeables](#) sur le site de l'Université

Pierre et Marie Curie. Ce système très perfectionné décrit par

l'article [Un cédérom pour Scheme : chacun son entraîneur, un entraîneur pour tous](#) comporte un environnement de programmation complet (*DrScheme*), des exercices gradués et un système d'auto-évaluation.

Le logiciel que nous utiliserons pour cet enseignement est un environnement de programmation pour *Scheme* appelé

[Bigloo](#).

C'est un logiciel libre écrit par [Manuel Serrano](#) qui fonctionne sous *Unix* (ou *Linux*) et sous *Windows*. Vous pourrez le télé-charger [ici](#).

Vous aurez aussi besoin de l'éditeur [Xemacs](#), disponible sous Unix/Linux ou Windows-xxx. En fait, sous Unix/Linux, vous pouvez tout aussi bien utiliser [GNU Emacs](#), dont le fonctionnement est très similaire, mais si vous utilisez encore Windows, [Xemacs](#) est plus commode à installer.

Il existe beaucoup de systèmes de programmation pour *Scheme*, si vous voulez en essayer un autre je vous conseille celui-ci :

[DrScheme](#),

qui a l'avantage de fonctionner sous *MacOS 9*. Depuis *MacOS X*, *MacOS* est devenu un véritable système Unix, sur lequel vous pouvez parfaitement installer tout aussi bien [DrScheme](#) que [Bigloo](#) et [GNU Emacs](#).

Dorai Sitaram a lui aussi créé un site d'auto-apprentissage de Scheme (en anglais), [Teach Yourself Scheme in Fixnum Days](#).

Vous trouverez des livres en ligne sur le WWW, en anglais malheureusement le plus souvent. À tout seigneur tout honneur, le merveilleux [Structure and Interpretation of Computer Programs](#) de Harold Abelson et Gerald Jay Sussman, qui a fait découvrir *Scheme* à toute une génération, disponible aussi sous une [autre forme](#).

Signalons le texte en français de [Robert Cori et Pierre Casteran](#) (incidemment, vous trouverez sur ce même site un excellent polycopié d'algorithmique), le remarquable bouquin [Concrete Abstractions – An Introduction to Computer Science Using Scheme](#) de Max Hailperin, Barbara Kaiser et Karl Knight, [How to Design Programs](#) par le groupe à l'origine de [DrScheme](#), [The Scheme Programming Language](#) de R. Kent Dybvig, ainsi que [An Introduction to Scheme and its Implementation](#) par [Paul Wilson](#).

D'autres pages en français : celle de [Danny Dubé](#) à Montréal, celle de [Jean-Paul Roy](#) à Nice, celle de [Marc Chemillier](#) à Caen, l'[article](#) consacré à Scheme par la Wikipédia et l'excellent site du [CENA](#) (Centre d'Études de la Navigation Aérienne).

Scheme est un langage défini par un document, [le R5RS](#). Vous trouverez un maximum de liens et de documents relatifs à *Scheme* [ici](#). Et vous trouverez ici une compilation de références relatives à [l'histoire du langage Lisp](#).

Retour vers [l'accueil du site](#), pour d'autres sujets informatiques.