

# À propos des continuations

Bernard Tatin

2017

---

**Résumé.** Ici, on s'occupe des continuations tout d'abord avec *Scheme* puis, si possible, avec d'autres langages dont *Standard ML* ou *F#*. Le fil conducteur provient, sauf indication contraire, des articles de *Wikipedia* en anglais ou en français qui concernent ces continuations de la programmation fonctionnelle.

Le choix de **noweb** provient du simple fait que documentation et sources sont conçus en même temps.<sup>a</sup>

---

<sup>a</sup>Document crée le 22/11/2017 à 16:10.



---

## *Contents*

<b>I</b>	<b>introduction</b>	<b>3</b>
I.1	un premier test et quelques définitions	4
<b>II</b>	<b>annexes</b>	<b>5</b>
<b>III</b>	<b>tables et index</b>	<b>6</b>
III.1	table des extraits de code	6
III.2	index des symboles	6
III.3	Définitions	6

# I

---

## *introduction*

Belle image.



Figure 1: *la loire* (r)  
Source: photo de l'auteur

On ne s'en lasse pas.

---

**Note.** Ce qui suit provient pour l'essentiel de l'article de Wikipedia (en anglais): Continuation-passing style.

---

## I.1 un premier test et quelques définitions

Commençons donc par les définitions essentielles:

---

**Définition 1 - CPS.** Le *continuation-passing style* ou **CPS** est un style de programmation où le contrôle est passé explicitement sous forme de continuation.

C'est ce style que nous allons présenter dans les pages qui suivent. En attendant, voyons ce qu'est une continuation :

---

**Définition 2 - continuation.** Une continuation d'un programme est *la suite des instructions qu'il lui reste à exécuter à un moment précis*<sup>a</sup>

---

<sup>a</sup>Cf. Continuation.

## II

---

### *annexes*

# III

## *tables et index*

### III.1

#### *III.1 table des extraits de code*

### III.2

#### *III.2 index des symboles*

### III.3

#### *III.3 Définitions*

CPS, 4

continuation, 4