

Les MOOC - une révolution dans l'enseignement supérieur ?

Franck Meunier - Stéphanie Conrad
Grenoble INP

Mai 2013

Historique

- MOOC pour *Massive Open Online Course* (2008)
- Origine : Georges Siemens de l'université Atabasca (CA) initiateur du connectivisme
→ cMOOCs = connectivistes
- Le buzz : nov 2011 Stanford propose un cours en libre accès sur l'intelligence artificielle → 58 000 inscrits
→ xMOOCs = transmitifs (évt allant jusqu'à la certification)
- Fin 2011 naissance de Coursera (Stanford)
- Avril 2012 : edX (MIT + Harvard...)

Un MOOC c'est quoi ?

- Massif : 1 000 - 50 000 étudiants
- Open : gratuit, ouvert à tous
- On line : disponible sur des plate-formes dédiées
- Course : sessions limitées dans le temps et encadrées
 - courtes vidéos
 - quizz
 - évaluations par les pairs
 - outils sociaux

Un nouveau modèle d'enseignement (US)

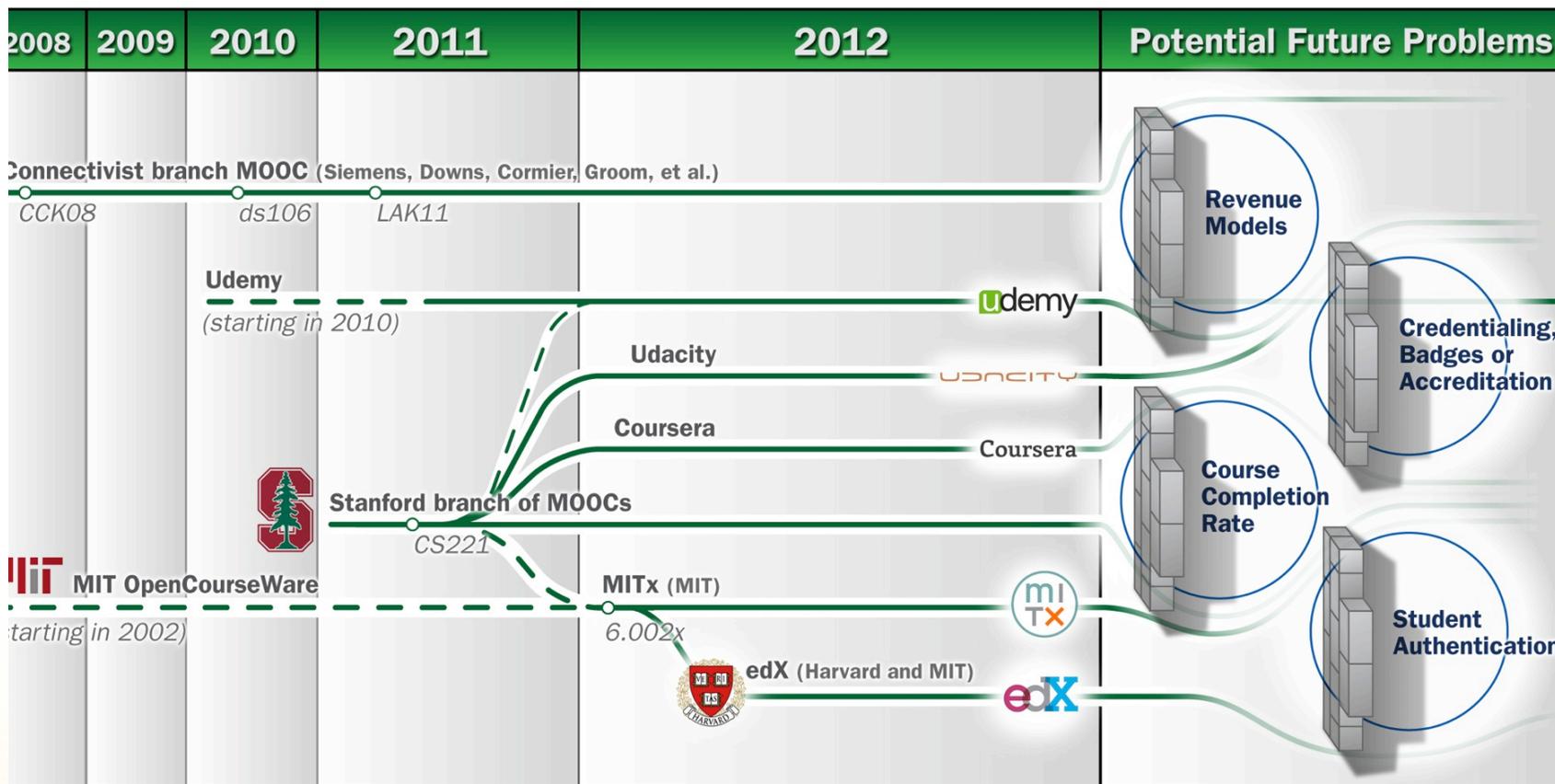
- décision de l'Etat de Washington de numériser et de rendre disponibles gratuitement 81 cours de ses meilleurs universités → devrait permettre aux étudiants d'économiser 40 millions de dollars par an
- premier congrès mondial sur le Open Educational Resources en juillet 2012 → le meilleur moyen de garantir le développement dans le monde dans les prochaines décennies
- cette forme d'enseignement serait la plus efficace (Carnegie Mellon 2005-07)

Opportunités liées aux MOOCs

- Grands effectifs d'apprenants avec un encadrement « réduit »
- Moteur d'ingénierie pédagogique (classe inversée)
- Durée limitée favorisant l'engagement des étudiants
- Interactions entre les apprenants / les apprenants et l'équipe pédagogique
- Réponse aux problèmes de locaux, d'effectif, de moyens
- Visibilité des enseignants / des établissements

Difficultés / questions

- Certification
- Evaluation entre pairs
- Réaction massive des apprenants en cas de formulation / d'organisation imparfaite
- Risques de problèmes informatiques liés au nombre de connexions
- Uniformisation des enseignements
- Modèle économique à terme ?



Modèle économique

- Vente à certaines entreprises/organismes (recruteurs)
 - de profils utilisateurs
 - de service d'évaluation (évaluation de candidats)
 - de la plateforme adaptée corporate
- Vente aux étudiants
 - de soutien pédagogique
 - cours (à terme certains ne seront pas libres)
 - de certificats / examens
- Bénéfices reversés jusqu'à 15% à l'université

Les acteurs

Diffusion :

- Coursera (issu de Stanford) : 70 établissements, 374 cours, 3,5 millions d'inscrits
- edX (MIT / Harvard + ...)
- Canvas Network



Initiatives francophones :

- ITYPA : Internet Tout Y est Pour Apprendre (cMOOC - 1000 inscrits)
- EPFL, Centrale Lille, Polytechnique

Un exemple : les MOOCs à l'EPFL

- Une volonté politique forte
 - valorisation de leurs enseignements
 - une opportunité pédagogique
 - un levier pour les Relations Internationales
- Des moyens
 - le « Centre pour l'Education Digitale »
 - un « ambassadeur MOOCs »

Exemple de cours : SCALA

Projet de MOOC Grenoble INP

MOOC francophone en hydrologie appliquée

- éprouver le concept
- collaboration avec l'EPFL
- le réseau RESCIF
- une équipe d'enseignants chercheurs impliqués

MOOC francophone en hydrologie appliquée

Hydrologie, langues

Objectifs pédagogiques
 Présenter des problématiques majeures de l'hydrologie appliquée à travers des études de cas. Apporter une aide linguistique à la compréhension des contenus scientifiques en français dans le domaine

Modalités

- 30 séquences vidéos scientifiques
- Exercices auto-corrigés, rendu de travaux
- Forums de discussion, peer assessment

Usages pédagogiques (3 niveaux)

- Sensibilisation aux enjeux de l'eau (tout public)
- Formation initiale L3 / formation continue dans le domaine de l'environnement
- Formation au français de spécialité

TICE MOOC (sessions de 6 semaines)

Apports du projet

- lien direct à l'actualité scientifique
- sujet vital pour la planète et les populations
- approche concrète par études de cas
- collaboration Nord-Sud pertinente dans le cadre du réseau RESCIF
- co-construction entre universités partenaires du réseau
- transfert de l'expertise de l'EPFL dans le domaine des MOOCs
- premier MOOC avec FLE
- formation de communautés d'apprenants sur le thème de l'eau

Contact : florence.michau@grenoble-inp.fr

Grenoble INP

EPFL RESCIF ZIE UCL unit



Liens

- <http://www.rtflash.fr/on-apprend-mieux-en-ligne-que-dans-amphitheatre/article>
- CourseTalk : <http://coursetalk.org/>
- Class Central : <http://www.class-central.com/>
- Blog La révolution MOOC :
<http://blog.educpros.fr/matthieu-cisel/>



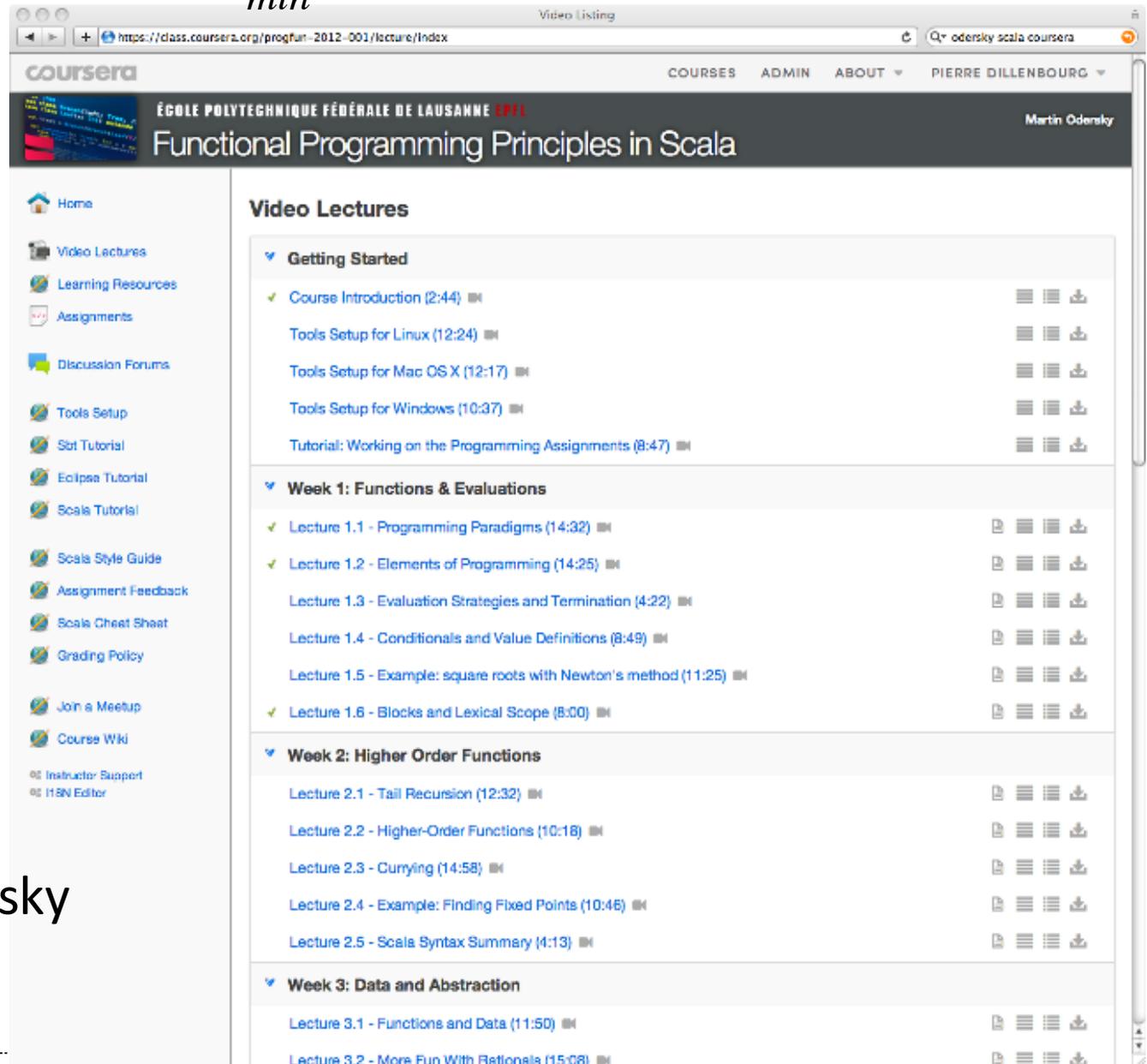
Pedro Pinto, I&C

CRAFT Studio



1. Segmentation du cours

e.g. 7 semaines X 6-8 vidéos X 10-12 min



The screenshot shows a web browser window displaying the Coursera course page for "Functional Programming Principles in Scala" by Martin Odersky at EPFL. The page is titled "Video Listing" and shows a list of video lectures organized into sections. The left sidebar contains navigation links for Home, Video Lectures, Learning Resources, Assignments, Discussion Forums, Tools Setup, Sbt Tutorial, Eclipse Tutorial, Scala Tutorial, Scala Style Guide, Assignment Feedback, Scala Cheat Sheet, Grading Policy, Join a Meetup, and Course Wiki. The main content area lists video lectures under the following sections:

- Getting Started**
 - Course Introduction (2:44)
 - Tools Setup for Linux (12:24)
 - Tools Setup for Mac OS X (12:17)
 - Tools Setup for Windows (10:37)
 - Tutorial: Working on the Programming Assignments (8:47)
- Week 1: Functions & Evaluations**
 - Lecture 1.1 - Programming Paradigms (14:32)
 - Lecture 1.2 - Elements of Programming (14:25)
 - Lecture 1.3 - Evaluation Strategies and Termination (4:22)
 - Lecture 1.4 - Conditionals and Value Definitions (8:49)
 - Lecture 1.5 - Example: square roots with Newton's method (11:25)
 - Lecture 1.6 - Blocks and Lexical Scope (8:00)
- Week 2: Higher Order Functions**
 - Lecture 2.1 - Tail Recursion (12:32)
 - Lecture 2.2 - Higher-Order Functions (10:18)
 - Lecture 2.3 - Currying (14:58)
 - Lecture 2.4 - Example: Finding Fixed Points (10:46)
 - Lecture 2.5 - Scala Syntax Summary (4:13)
- Week 3: Data and Abstraction**
 - Lecture 3.1 - Functions and Data (11:50)
 - Lecture 3.2 - More Fun With Rationals (15:08)

Cours SCALA
Prof. Martin Odersky
EPFL



https://class.coursera.org/progfun-2012-001/lecture/download.mp4?lecture_id=4



Multiple Parameter Lists

The definition of functions that return functions is so useful in functional programming that there is a special syntax for it in Scala.

For example, the following definition with the nested `sumF` function, but

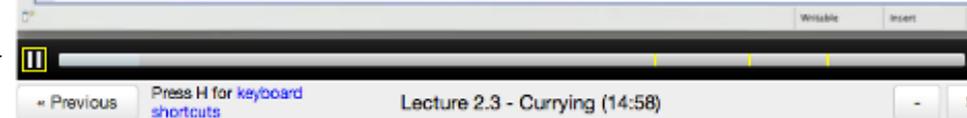
```
def sum(f: Int => Int)(a: Int,
  if (a > b) 0 else f(a) + sum
```

Sum

```
object exercise {
  def product(f: Int => Int)(a: Int, b: Int): Int = map
    if (a > b) 1
    else f(a) * product(f)(a + 1, b)
    product(x => x * x)(3, 4)
    > product: (f: Int => Int)Int
    > res0: Int = 144

  def fact(n: Int) = product(x => x)(1, n)
    > fact: (n: Int)Int
    > res1: Int = 120

  def mapReduce(f: Int => Int, combine: (Int, Int) => Int, zero: Int)(a: Int, b: Int): Int =
    if (a > b) zero
    else combine(f(a), mapReduce(f, combine, zero)(a+1, b))
    > mapReduce: (f: Int => Int)Int
    // Int)Int
}
```



3. L'apprenant choisi la vitesse

is called *currying*,
1900-1982), a

nfinkel and Frege,



- * La vidéo peut être accélérée jusqu'à 1.5 fois
- * Lorsque l'apprenant revient sur les transparents
- * Lorsqu'il y a des parties plus faciles ou déjà connues

4. QCMs pour soutenir l'attention

Write a function `and` such that for all argument expressions `x` and `y`:

- `and(x,y) == x && y`

Please give your answer on one single line. (do not use `&&` or `||` in your implementation)

Submit

Skip



- * Des balises signalent la présence de QCM dans la vidéo
- * Les apprenants se préparent à répondre
- * Similaire à l'utilisation de clickers en classe

5. Structure temporelle explicite

Week 2: Higher Order Functions

Functional Sets

[View Instructions](#)

Due Date:

Thu 4 Oct 2012 11:59:00 PM PDT

Hard Deadline:

Mon 8 Oct 2012 11:59:00 PM PDT

Part	Name	Last Submission	Score	Feedback
1 / 1	Functional Sets	-	- / 10	View
Total Score			0 / 10	

Le cours est rythmé par les échéances des travaux à rendre



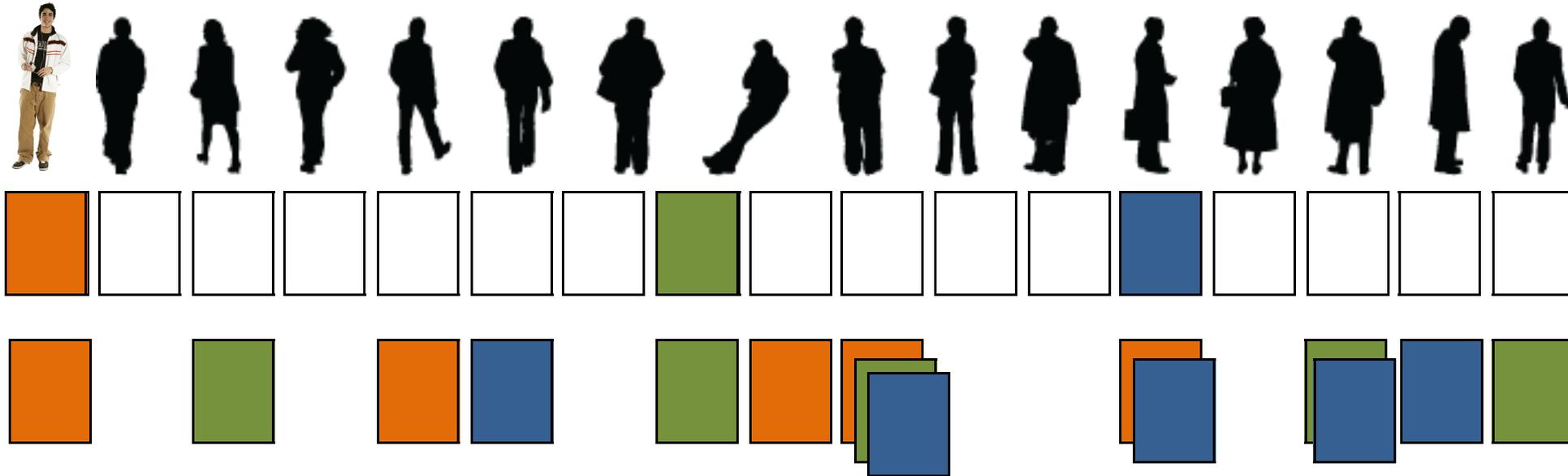
Vidéos, QCMs

Travaux à rendre

**e.g. le cours SCALA demande
5-7 heures de travail
hebdomadaire**

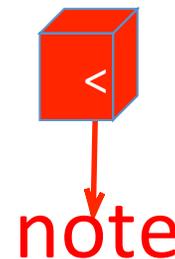
<http://www.chipestimate.com/blogs/IPInsider/?p=458>

7. Evaluation par les pairs



Evaluation par les pairs

- * Lorsqu'une évaluation automatique est impossible
- * Option: entraîner les étudiants à évaluer
- * **Avantage: évaluer est une activité**



8. Forums massifs

Forum Threads

Start New Thread

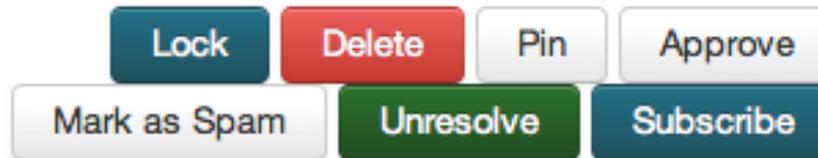
Last Updated

Top Threads

Last Created

Subscribed

Thread Title / Original Poster	Last Post	Votes	Posts	Views
 Functional Programming Principles in Scala 2? Blazej Kolad 	Maxim Matvienko 13 hours ago	260	11	2k
 Lecture 6.5: Which came first for or map? João Bertolino de Sousa Neto   <code>for</code> <code>map</code>	Tobias Schlatter 5 hours ago	0	2	34
 Logging Recursive Functions with the Y Combinator Igor Baltiyskiy (Student)  <code>course</code> <code>lazy-evaluation</code> <code>logging</code> <code>fixed-point</code> <code>combinator</code>	Igor Baltiyskiy 9 hours ago	1	1	63
  Parentheses, precedence and method chaining Frantisek Kocun 	Frantisek Kocun 18 hours ago	2	5	87
Lecture 6.5: Pouring class not visible inside the worksheet Dario Iacampo   <code>Lecture_6_5</code>	Phill Springett 16 hours ago	0	4	60
Differences between Stream.Empty and Stream.empty ? Sergio Castro  <code>Stream</code>	Ivano Pagano 22 hours ago	1	3	96
 Lecture 7.2 streamRange(1,10).take(3) with different result Anonymous  <code>Stream</code> <code>streamRange</code>	Anonymous 18 hours ago	0	1	13
  Lazy evaluation of parameters vs call by name (+ lazy evaluation syntax sugar) Jordi Pradel (Student) 	Jordi Pradel 20 hours ago	0	3	53



Comment faire face à des milliers de messages ?

- * Des outils de gestion de forum
 - * Tagging
 - * Vote par les apprenants (vote-up / vote-down)
 - * Filtres intelligents
- * Pré-sélection des messages par les assistants



A propos des Meetups Everywhere Coursera

13 086 COURSERIANS 1 449 VILLES

Learning for life.
Education for everyone.

Coursera Meetups are a great way to meet your fellow Courserians, swap stories, share ideas, form study groups, and have a great time.

Find a meetup near you, or start a new one. These events are for you, by you, so meet wherever you like. You choose the topic, the time and the venue.

Groupes d'étude en face à face

- * Travailler ensemble et s'entraider
- * Socialiser

Classes and Substitutions (2)

Answer: The expression `new C(v1, ..., vm).f(w1, ..., wn)` is rewritten to:

$$[w_1/y_1, \dots, w_n/y_n][v_1/x_1, \dots, v_m/x_m][\text{new } C(v_1, \dots, v_m)/\text{this}] b$$

There are three substitutions at work here:

- ▶ the substitution of the formal parameters y_1, \dots, y_n of the function f by the arguments w_1, \dots, w_n ,
- ▶ the substitution of the formal parameters x_1, \dots, x_m of the class C by the class arguments v_1, \dots, v_m ,
- ▶ the substitution of the self reference *this* by the value of the object `new C(v1, ..., vn)`.

*class C (x₁, ..., x_m) {
 def f (y₁, ..., y_n) = b*

here, then it is re-written not using one substitution as for plain functions, but

« Previous Press H for keyboard shortcuts Lecture 3.3 - Evaluation and Operators (16:25) - Speed: 1.25x + Next »

Classes and Substitutions (2)

Answer: The expression `new C(v1, ..., vm).f(w1, ..., wn)` is rewritten to:

$$[w_1/y_1, \dots, w_n/y_n][v_1/x_1, \dots, v_m/x_m][\text{new } C(v_1, \dots, v_m)/\text{this}] b$$

There are three substitutions at work here:

- ▶ the substitution of the formal parameters y_1, \dots, y_n of the function f by the arguments w_1, \dots, w_n ,
- ▶ the substitution of the formal parameters x_1, \dots, x_m of the class C by the class arguments v_1, \dots, v_m ,
- ▶ the substitution of the self reference *this* by the value of the object `new C(v1, ..., vn)`.

Traduction

class C (x₁, ..., x_m) {
 def f (y₁, ..., y_n) =

ici, alors c'est réécrit en utilisant pas une substitution, comme pour les fonctions simples, mais



« Previous Press H for keyboard shortcuts Lecture 3.3 - Evaluation and Operators (16:25) Speed: 1.25x Next »

Hydrologie, langues MOOC francophone en hydrologie appliquée

Objectifs pédagogiques

Présenter des problématiques majeures de l'hydrologie appliquée à travers des études de cas. Apporter une aide linguistique à la compréhension des contenus scientifiques en français dans le domaine

Modalités

- 30 séquences vidéos scientifiques
- Exercices auto-corrigés, rendu de travaux
- Forums de discussion, peer assesment

Usages pédagogiques (3 niveaux)

- Sensibilisation aux enjeux de l'eau (tout public)
- Formation initiale L3 / formation continue dans le domaine de l'environnement
- Formation au français de spécialité

TICE

MOOC (sessions de 6 semaines)



Apports du projet

- lien direct à l'actualité scientifique
- sujet vital pour la planète et les populations
- approche concrète par études de cas
- collaboration Nord-Sud pertinente dans le cadre du réseau RESCIF
- co-construction entre universités partenaires du réseau
- transfert de l'expertise de l'EPFL dans le domaine des MOOCs
- premier MOOC avec FLE
- formation de communautés d'apprenants sur le thème de l'eau



Contact : florence.michau@grenoble-inp.fr



RESCIF
Réseau d'excellence
des sciences de l'ingénieur
de la Francophonie

