

# Déroulé Global du MOOC

1. Semaine 1: Démarrage et Découverte de la Syntaxe (Welcome on Board and Syntax Discovery)
2. Semaine 2: Syntaxe Pharo, Blocks et Inspecteur (Pharo Syntax, Blocks and Inspector)
3. Semaine 3: Conception, Méthodes de classes et Collections (Design, Class Methods and Collections)
4. Semaine 4: Héritage et Recherche de Méthode et Développement Web (Inheritance&Lookup and Web Development)
5. Semaine 5: Exception, Debuggage et Réflexion (Exception, Debugging and Reflection)
6. Semaine 6: Repenser la recherche de méthode et points avancés (Re-think lookup and Advanced OO Design)
7. Semaine 7: Conception OO et Tests (More on OO Design and Testing)

## 1.1 Différentes ressources et activités

Ce document résume le déroulé du Mooc semaine par semaine. Nous pensons qu'il est préférable de faire les activités proposer dans l'ordre que nous avons conçu. Maintenant il est fort possible que pour certains lecteurs un autre ordre ne pose pas de difficultés particulières.

Nous proposons plusieurs activités (identifiable par leur nom ou étiquette accolée):

- **[Lecture]** signifie que vous devez suivre la vidéo qui contient le cours décrit dans les diapositives du cours. Tous les PDFs des diapositives

sont fournis. De manière générale tous les morceaux de code présentés sont executables directement dans Pharo donc n'hésitez pas à le faire.

- **[Live]** signifie que vous devez suivre une vidéo qui démontre certains aspects de l'IDE interactif de Pharo. Nous suggérons de répéter les points qui vous semblent opportun.
- **[Quiz]** signifie que vous devez répondre a des séquences de questions. Regardez les cours doit suffir pour trouver les bonnes réponses. Ce sont les résultats des quiz qui vous permettront d'obtenir un certificat de réussite.
- **[Redo]** signifie que vous devez
  - (1) suivre les videos et
  - (2) reproduire ce qui est démontré pas à pas en utilisant votre installation de Pharo.

Durant le premier redo, vous devez vraiment suivre et reproduire ce que vous voyez.

- **[Exercise]** signifie que vous devez suivre un exercice guidé et le reproduire en utilisant Pharo. L'exercice est décrit dans un document. Certains exercices proposent deux versions : une complète et une à trous. Nous vous conseillons de commencer par la version à trous. Nous ne donnerons pas de notes pour ces exercices mais ils sont importants pour vous entrainer.
- **[Mini project]** signifie que vous avez un exercice plus long nommé TinyBlog. Il va durer toutes la durée du Mooc. Chaque semaine nous allons ajoutr des fonctionnalités. Chaque semaine vous aurez acces à une version complète de la solution de la semaine précédente de manière à ce que vous puissiez continuer en cas de problèmes.
- **[Challenge]** signifie que vous avez un petit défi à résoudre en farfouillant dans Pharo.

L'idée est que vous devez vous débrouiller pour trouver des informations et ne pas avoir peur de vous perdre. Vous pouvez poser des questions sur la mailing-list Pharo-users. Vous devez faire le challenge pour vous. Nous n'allons pas le noter car il y a souvent plusieurs façons de résoudre un challenge. Chaque semaine vous aurez la solution du challenge et nous vous encourageons à lire les messages et les classes utilisées dans la solution. De plus les challenges sont souvent marrants et des petits défis. Donc ne boudez pas ce petit plaisir

## 1.2 Semaine 1 : Démarrage et Découverte de la Syntaxe

1. [Lecture] Objectifs (Objectives of this MOOC)

2. [Lecture] Pharo: un système objet immersif (Pharo: an Immersive Object-Oriented System)
3. [Lecture] Vision (Pharo Vision)
4. [Lecture] Modèle objet de Pharo en 5 min (Pharo Object Model in a Nutshell)
5. [Lecture] Syntaxe Pharo en 5 min (Pharo Syntax in a Nutshell)
6. [Lecture] Définition de classe et méthodes (Class and Method Definitions)
7. [Live] Sélectionner et exécuter des expressions (Selecting and executing expressions)
8. [Live] Apprendre Pharo avec Prof Stef (Learn Pharo with Prof Stef)
9. [Live] Colorisation de syntaxe (Syntax Highlighting gives Feedback)
10. [Live] Apprendre les raccourcis claviers (Learn Keyboard Shortcuts)
11. [Redo] Développer un compteur (Coding a Counter)
12. [Exercise] Expressions et Messages (Expressions and Messages)
13. [Exercise] Développer un simple compteur (Developing a Simple Counter)
14. [Challenge] Défi 0: (Challenge 0)

## 1.3 Semaine 2 : Syntaxe Pharo, Blocks et Inspecteur

1. [Lecture] Comprendre les messages (Understanding Messages)
2. [Lecture] Messages pour les programmeurs Java (Messages for Java programmer)
3. [Lecture] Messages : Composition et Précédence (Messages: Composition and Precedence)
4. [Lecture] Comprendre les messages : Séquence et Cascade (Understanding Messages: Sequence and Cascade)
5. [Live] Trouver des méthodes avec le Finder (Finding Methods with Finder)
6. [Live] GTInspector 1 : Inspecter et interagir avec les objets (GTInspector 1: Inspect and Interact with Objects)
7. [Lecture] Introduction aux Blocks (Introduction to Blocks)
8. [Lecture] Boucles (Loops)
9. [Lecture] Booléens et conditions (Booleans and Conditions)
10. [Lecture] Parenthese contre Crochets (Parenthesis vs Square Brackets)

11. [Lecture] Yourself (Yourself)
12. [Lecture] Essence du Dispatch: Un Exemple (The Essence of Dispatch through an Exercise)
13. [Live] Nautilus 1 : Premier contact (Nautilus 1: First Meet with the Code Browser)
14. [Live] Nautilus 2 : Packages, Etiquettes et Groupes (Nautilus 2: Packages, Tags and Groups)
15. [Live] Nautilus 3 : Heritage (Nautilus 3: Inheritance view)
16. [Redo] Coder un compteur dans le debogueur (Coding a Counter in the Debugger)
17. [Exercise] Solution : Expressions et Messages (Solution: Expressions and Messages)
18. [Exercise] Exercice : Résultats d'expressions (Expression Results)
19. [MiniProjet] TinyBlog (Chaptires 1 et 2)
20. [Challenge] Solution : Défi 0 (Solution: Challenge 0)
21. [Challenge] Défi 1 (Challenge 1)

## 1.4 Semaine 3 : Conception, Méthodes de classes et Collections

1. [Lecture] Sélection de méthode: Booléens Pharo comme exemple (Essence of Dispatch: Taking Pharo Booleans as Example)
2. [Lecture] Essence de sélection méthode (Essence of Dispatch)
3. [Lecture] Variables (Variables)
4. [Lecture] Une petite application HTTP: Revisitons la Syntaxe Pharo (A Simple HTTP Application: a Pretext to Revisit Pharo Syntax)
5. [Live] Nautilus 4 : Comment devenir un super Pharoer: comprendre sender/implémenteurs (Nautilus 4: How to Become a Super Pharo Developer: Understanding Sender and Implementors)
6. [Live] Nautilus 5 : Naviguer rapidement à l'intérieur d'une classe (Nautilus 5: Navigating Fast Inside Classes)
7. [Lecture] Méthodes de classe (Class Methods)
8. [Lecture] Un survol des principales collections (An Overview of Essential Collections)
9. [Live] Spotter 1 : Trouver et browser une classe (Spotter 1: Find and Browse a Class)

10. [Live] Spotter 2 : Filtrer les résultats par catégories (Spotter 2: Filter Results by Category)
11. [Lecture] Itérateurs (Iterators)
12. [Lecture] Survol des Streams (Stream Overview)
13. [Lecture] Comprendre Return (Understanding Return)
14. [Redo] Développer un petit langage spécifique (Coding a Little Domain Specific Language)
15. [Exercise] Solution : Résultats d'expressions (Solution: Expression Results)
16. [Exercise] Réécrire des expressions (Rewriting Expressions)
17. [MiniProjet] TinyBlog (Chaptire 3)
18. [Challenge] Solution : Défi 1 (Solution: Challenge 1)
19. [Challenge] Défi 2 (Challenge 2)

## 1.5 **Semaine 4 : Héritage et Recherche de Méthode et Développement Web**

1. [Lecture] Héritage (Inheritance Basics)
2. [Lecture] Héritage et Recherche de Méthode (Inheritance & Lookup: Lookup)
3. [Lecture] Héritage : Super (Inheritance & Lookup: Super)
4. [Lecture] Héritage : doesNotUnderstand: (Inheritance & Lookup: Does-NotUnderstand:)
5. [Lecture] Recherche de Méthode et Métaclasses (Inheritance & Lookup: Lookup in Metaclasses)
6. [Lecture] Méthodes de classes comme mecanisme d'enregistrement (Class Methods at Work)
7. [Lecture] Survol de la Pile Web Pharo (Overview of Pharo Web Stack)
8. [Lecture] Seaside : Un framework web innovant (Seaside: an Innovative Web Application Framework)
9. [Lecture] Seaside : Rendu des composants (Seaside: Rendering Components)
10. [Lecture] Seaside : Un oeil sur MetaDonnees et REST (Seaside: a Glance at MetaData and REST)

11. [Lecture] Voyage : Base de données objet noSQL (Voyage: NoSql Object Database)
12. [Live] Comment charger du code d'un repository (How to load code from repository)
13. [Live] Comment charger un projet (How to load projects)
14. [Live] Nautilus 6 : Un survol des refactorings (Nautilus 6: An overview of Refactorings)
15. [Live] Nautilus 7 : Utiliser l'assistant de qualité et code critique (Nautilus 7: Using the Quality Assistant and Code Critic)
16. [Exercise] Solution : Réécrire des expressions (Solution: Rewriting Expressions)
17. [MiniProjet] TinyBlog (Chapitres 4 à 7)
18. [Exercise] Construisons une gestion de contacts (Building a Simple Contact Book Application)
19. [Challenge] Solution : Défi 2 (Solution: Challenge 2)
20. [Challenge] Défi 3 (Challenge 3)

## 1.6 Semaine 5 : Exception, Debuggage et Réflexion

1. [Lecture] Seaside: Composing Components
2. [Lecture] Really Understanding Class Methods
3. [Lecture] Erreurs communes (Common Errors)
4. [Lecture] Exceptions: un survol (Powerful Exceptions: an Overview)
5. [Lecture] Debugger avec Pharo (Debugging in Pharo)
6. [Lecture] SUnit: tests unitaire en Pharo (SUnit: Unit Tests in Pharo)
7. [Lecture] Fichiers en Pharo (Files in Pharo)
8. [Lecture] Réflexion : Simple Introspection (Reflection: Basic Introspection())
9. [Lecture] Optimisation en Pharo (Benchmarking in Pharo)
10. [Live] Spotter 3 : réduire les recherches (Spotter 3: Scoping and Reducing Search)
11. [Live] Spotter 6 : Chercher, naviguer et prévisualisation (Spotter 6: Search, Navigate and Preview Files)
12. [Live] Comprendre le débogueur (Learning the Debugger)
13. [Live] Comment trouver un bug (How to find a bug?)

14. [Live] GTInspector 2 : inspecter fichiers et dossiers (GTInspector 2: Inspect Files and Directories)
15. [Exercise] TinyChat (TinyChat)
16. [MiniProjet] TinyBlog (Chapitres 8 et 9)
17. [Challenge] Solution : Défi 3 (Solution: Challenge 3)
18. [Challenge] Défi 4 (Challenge 4)

## 1.7 **Semaine 6 : Repenser la recherche de méthode et points avancés**

1. [Lecture] Avez-vous vraiment compris super ? (Did You Really Understand Super?)
2. [Lecture] Comprendre l'implementation de ifTrue:ifFalse: (Understanding the Implementation of ifTrue:ifFalse:)
3. [Lecture] Dice new et self class new (Dice new vs. self class new)
4. [Lecture] Envoi de messages : planification pour la réutilisation (Message sends are plans for reuse)
5. [Lecture] Crochets et Templates (Hooks and Templates)
6. [Lecture] Architecture d'exécution (Runtime Architecture)
7. [Lecture] Caractères, chaînes et symboles (Characters, Strings and Symbols)
8. [Lecture] Tableaux dynamiques et de littéraux (Dynamic Vs. Literal Arrays)
9. [Live] Comprendre le fichier image et changes (Understanding Images and Changes Files)
10. [Live] Utiliser le Launcher pour gérer vos images (Using the Pharo Launcher to manage your Pharo images)
11. [Live] Obtenir une différence et fusionner du code en Pharo (Diffing and Merging in Pharo)
12. [Live] GTInspector 4 : Construire une view dédiée (GTInspector 4: Build Custom Tab Views for your Objects)
13. [Exercise] TinyBlog : Deployment (TinyBlog: Deployment)
14. [Challenge] Solution : Défi 4 (Solution: Challenge 4)
15. [Challenge] Défi 5 (Challenge 5)

## 1.8 Semaine 7 : Conception OO et Tests

1. [Lecture] Points avancés sur les classes (Advanced Points on Classes)
2. [Lecture] Objets à taille variables (Variable Size Objects)
3. [Lecture] Comprendre les métaclasse (Understanding Metaclasses)
4. [Lecture] Opérations réflexives pour la programmation live (Reflective Operations for Live Programming)
5. [Lecture] DoesNotUnderstand: un point d'entrée précieux (DoesNotUnderstand: a Precious Hook)
6. [Lecture] Réflexion : la pile comme un objet (Reflection: Stack as an Object)
7. [Lecture] Evitons les tests nil (Avoid Null Checks)
8. [Lecture] Future Radieux (A Journey in Pharo: A Bright Future)
9. [Live] Spotter 4 : Le panneau de prévisualisation (Spotter 4: The Preview pane)
10. [Live] Spotter 5 : Rechercher dans l'historique de Playground (Spotter 5: Search in Playground History)
11. [Live] GTInspector : Comprendre l'implémentation interne de Pharo avec l'inspecteur (GTInspector 3: Understand Pharo Internals by Inspection)
12. [Challenge] Solution : Défi 5 (Solution: Challenge 5)