

DÉVELOPPEURS·EUSES : FACILITEZ-VOUS LA VIE EN (RE)APPRENANT À LIRE.

Mixit 2023 :
Guillaume Saint-Etienne
Anthony Cassaigne



CC BY-SA 4.0 DEED

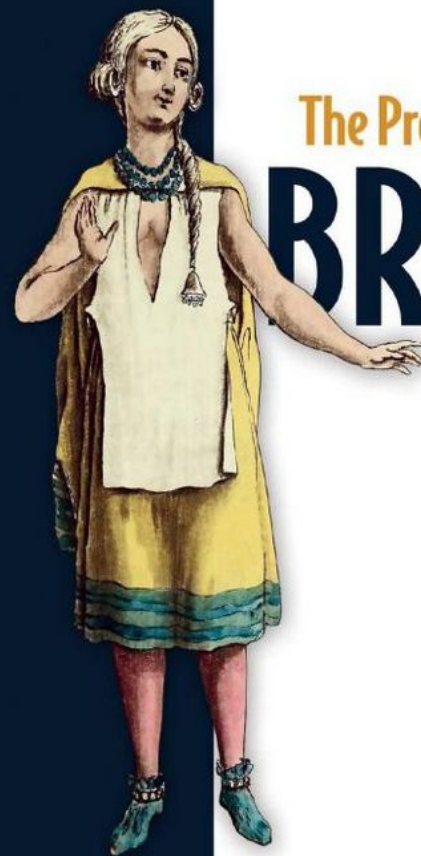
Attribution-ShareAlike 4.0 International

Libre à vous pouvez de ré-utiliser & adapter ce matériel !

FELIENNE HERMANS



<https://www.youtube.com/watch?v=z7w2lKG8zWM>



Copyrighted Material

What every programmer needs to know about cognition

The Programmer's BRAIN

Felienne Hermans

Foreword by Jon Skeet

 MANNING

Copyrighted Material

OBJECTIF DE CET ATELIER



L'objectif de cet atelier est de comprendre l'essence d'un code (d'une partie).

Quelle est la sous-partie du code à laquelle il faut prêter une attention particulière ?

Quelles lignes de code auraient l'impact le plus important ?

Se sont les attendues de cet atelier.

CE N'EST PAS DE LA REVUE DE CODE !

Ce n'est pas conduire

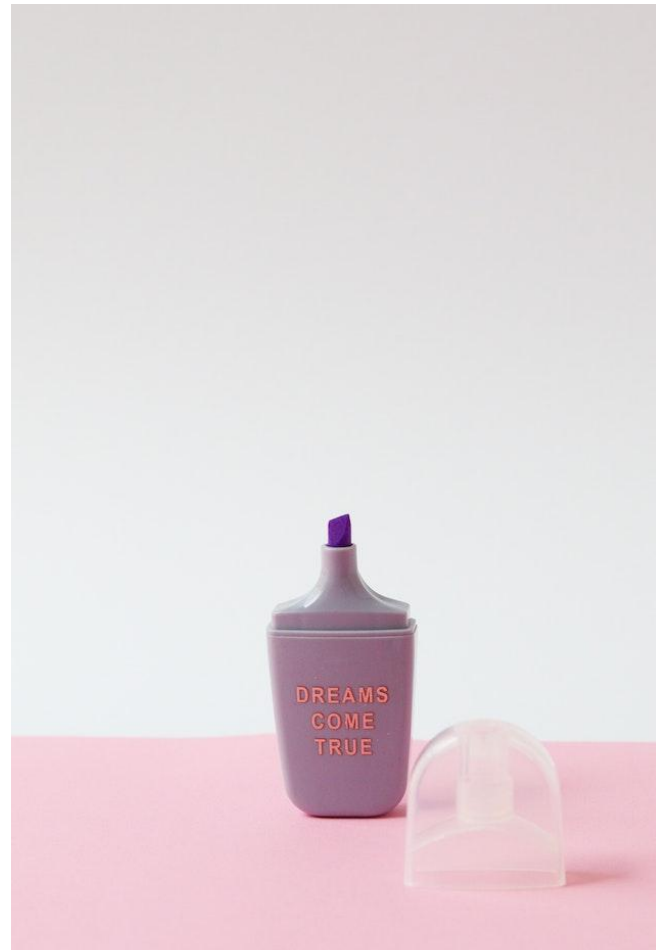
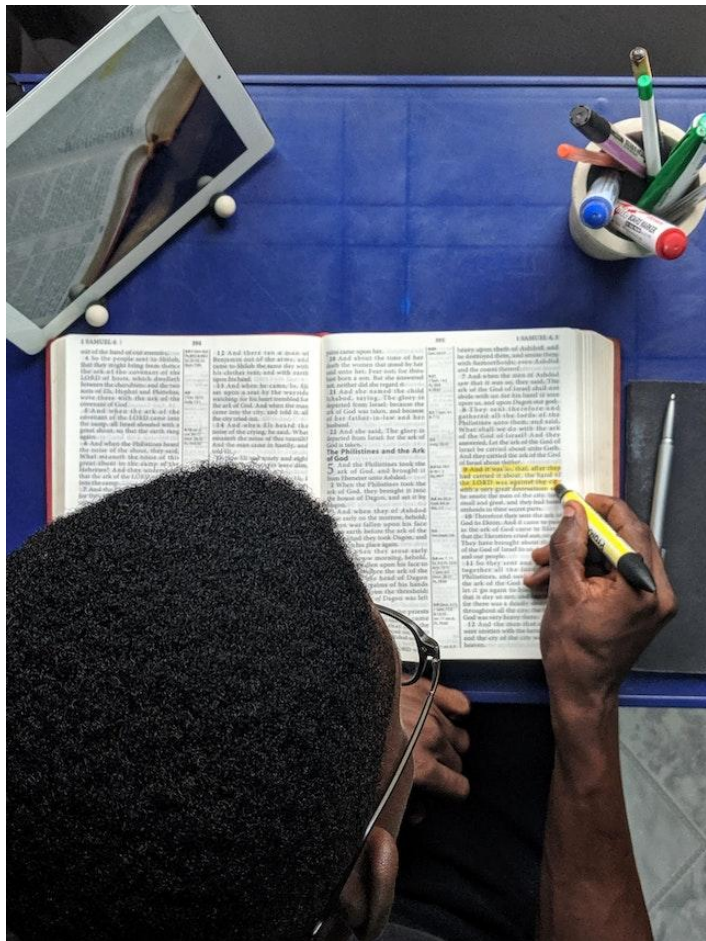
**Workshops, exercises and
techniques to help you
understand your code better**

le,

DÉROULÉ DE L'ATELIER



UNE APPLICATION POUR ANNOTER LE CODE



UNE APPLICATION POUR ANNOTER LE CODE

- <https://annotate.codereading.club/#/>

UN PEU D'ENTRAINEMENT...

- <https://bit.ly/3LVrUei>

Optional, List, Callable

.on(Enum):

VERSUS

Optional, List, Callable

ion(Enum):



UN PEU D'ENTRAINEMENT...

- <https://bit.ly/3LVrUei>

Optional, List, **Callable**

.on(**Enum**):

VERSUS

Optional, List, **Callable**

ion(**Enum**):

Arrow drawing mode: ☐ Freehand ☒ **Jointed**

Annotation brightness: ☐ Light ☒ Medium ☐ Dark

Arrow drawing mode: ☒ **Freehand** ☐ Jointed

Annotation brightness: ☐ Light ☒ Medium ☐ Dark

OK, VOUS AVEZ LES SKILLS... POUR PLUS TARD...

```
8 class KindOfDirection(Enum):
9     CARDINAL = 0
10    ORDINAL = 1
11
12
13 class Direction(Enum):
14     NORTH = 0
15     SOUTH = 1
16     EAST = 2
17     WEST = 3
18     NORTH_WEST = 4
19     NORTH_EAST = 5
20     SOUTH_WEST = 6
21     SOUTH_EAST = 7
22
23 def kind_of_direction(self) -> KindOfDirection:
24     if self in (Direction.NORTH, Direction.SOUTH, Direction.EAST, Direction.WEST):
25         return KindOfDirection.CARDINAL
26     if self in (Direction.NORTH_WEST, Direction.NORTH_EAST, Direction.SOUTH_WEST, Direction.SOUTH_EAST):
27         return KindOfDirection.ORDINAL
```

A diagram consisting of two pink arrows. The first arrow originates from the `CARDINAL` attribute of the `KindOfDirection` class (line 9) and points to the `CARDINAL` attribute of the `KindOfDirection` class (line 25). The second arrow originates from the `CARDINAL` attribute of the `KindOfDirection` class (line 9) and points to the `CARDINAL` attribute of the `KindOfDirection` class (line 25).

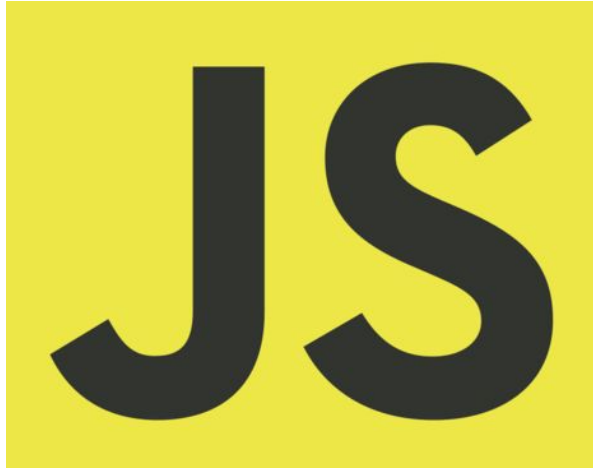
CHOISIR SA BASE DE CODE...



AU MENU

<https://racket-lang.org/>

<https://www.lua.org/>



UN PETIT VOTE

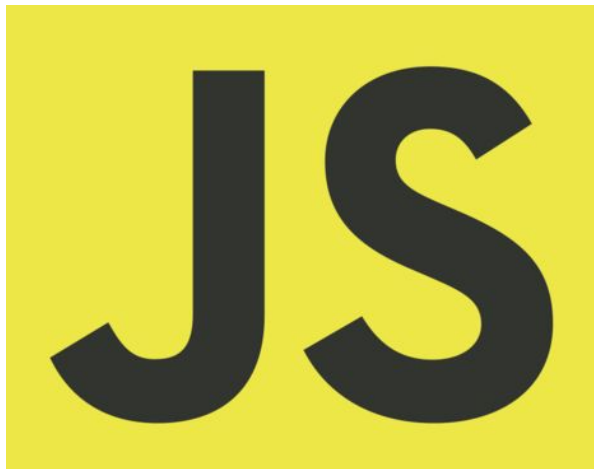
- <https://www.menti.com/alq4w8yy9tjv>
-
- Go to www.menti.com
and use the code

8432 9155



URL JAVASCRIPT

<https://bit.ly/3Qc4Vy4>



EN ROUTE VERS RACKET...

<https://bit.ly/3ZOnmMH>



EN VOYAGE VERS LUA

<https://bit.ly/3rLfoay>




L'ATELIER

PREMIER COUP D'ŒIL

L'objectif de cet étape est de s'entraîner à avoir une première impression du code et à agir en conséquence.

Nous avons tous des instincts et des stratégies différents pour savoir par où commencer lorsque nous sommes confrontés à un nouveau morceau de code.

 première
impression du code



PREMIER COUP D'ŒIL

instincts

stratégies



PREMIER COUP D'ŒIL

Il est important que ce que vous utilisez soit votre réaction immédiate, ne réfléchissez pas trop !

- Notez tout de suite la première chose qui attire votre attention.
- Ensuite, notez la deuxième chose qui attire votre attention.
- Prenez le reste de la minute pour réfléchir à la raison pour laquelle vous avez remarqué ces choses en premier.

GO !!! PREMIER COUP D'ŒIL



1 MINUTE



DÉBRIEF

DÉBRIEF DU 1ER COUP D'ŒIL



5-7 MINUTES

- Quels détails vous ont marqués ?
- Quelles lignes, quels faits ou quels concepts ont été retenus ?
- Quels types de connaissances avez-vous utilisées ?
 - langage
 - domaine métier
 - framework
- Selon vous, quelles connaissances pourraient être nécessaires pour mieux comprendre ce code ?

CONNAISSANCE DE LA SYNTAXE



5 MINUTES

- Regardez le code et examinez les éléments syntaxiques. Connaissez-vous la signification de tous les éléments ?
- Le rôle de chaque bloc du code est-il clair pour vous (fonction, condition, répétition, etc.) ?
- Reconnaissez-vous tous les opérateurs ?
- Prenez le reste du temps pour réfléchir aux autres éléments qui ne vous sont pas familiers.

DÉBRIEF

DÉBRIEF: CONNAISSANCE DE LA SYNTAXE



5 À 8 MINUTES

- Y a-t-il des concepts qui n'étaient pas familiers ?
- Si oui, quelqu'un peut-il aider ?
Y a-t-il d'autres membres du groupe qui en connaissent la signification ?
- Pourquoi les constructions syntaxiques ne sont-elles pas familières ?
Lesquelles ?
- Sont-elles idéosyncrasiques (inhabituelle ou spécifique à ce langage) de ce langage ou de cette base de code ?

STRUCTURE DU CODE



10 - 12 MINUTES

- Pour les **variables**
 - Entourer en rouge les **variables** et tracer un trait **rouge** entre sa définition et son usage
- Pour les **méthodes** & **fonctions**
 - Entourer en **bleu** les méthode et fonctions puis tracer un trait **bleu** entre sa définition et son usage
- Pour les **instances** de **classes**
 - Entourer en **vert** les instances et la définition des classes puis tracer un trait **vert** entre sa définition et ses instances

L'APPLICATION POUR ANNOTER LE CODE

- <https://annotate.codereading.club/#/>

STRUCTURE DU CODE

```
14 @objc
15 public class PhoneNumberValidator: NSObject {
16
17     @objc
18     public func isValidForRegistration(phoneNumber: PhoneNumber) -> Bool {
19         guard let countryCode = phoneNumber.getCountryCode() else {
20             return false
21         }
22
23         guard let validatedCountryCode = ValidatedPhoneCountryCodes(rawValue: countryCode.uintValue) else {
24             // no extra validation for this country
25             return true
26         }
27
28         switch validatedCountryCode {
29             case .brazil:
30                 return isValidForBrazilRegistration(phoneNumber: phoneNumber)
31             case .unitedStates:
32                 return isValidForUnitedStatesRegistration(phoneNumber: phoneNumber)
33             }
34     }
35
36     let validBrazilPhoneNumberRegex = try! NSRegularExpression(pattern: "^\\+55\\d{2}9?\\d{8}$", options: [])
37     private func isValidForBrazilRegistration(phoneNumber: PhoneNumber) -> Bool {
38         let e164 = phoneNumber.toE164()
39         return validBrazilPhoneNumberRegex.hasMatch(input: e164)
40     }
41
42     let validUnitedStatesPhoneNumberRegex = try! NSRegularExpression(pattern: "^\\+1\\d{10}$", options: [])
43     private func isValidForUnitedStatesRegistration(phoneNumber: PhoneNumber) -> Bool {
44         let e164 = phoneNumber.toE164()
45         return validUnitedStatesPhoneNumberRegex.hasMatch(input: e164)
46     }
47 }
48
```

The diagram illustrates the structure of the code with arrows indicating the flow of data and function calls:

- A pink arrow points from `validatedCountryCode` (line 28) to the `switch` statement (line 28).
- A blue arrow points from `isValidForBrazilRegistration` (line 30) to the `isValidForBrazilRegistration` function definition (line 37).
- A blue arrow points from `isValidForUnitedStatesRegistration` (line 32) to the `isValidForUnitedStatesRegistration` function definition (line 43).
- A pink arrow points from `e164` (line 38) to the `validBrazilPhoneNumberRegex` (line 36).
- A pink arrow points from `e164` (line 44) to the `validUnitedStatesPhoneNumberRegex` (line 42).

DÉBRIEF

DÉBRIEF: STRUCTURE DU CODE



10 MINUTES

- Quelqu'un a-t-il eu des difficultés à déterminer ce qui constituait une variable, une fonction ou une classe ?
- Quels sont les patterns apparents uniquement à partir des couleurs et des liens ?
- Comment les données circulent-elles dans le code ?
- Quelles parties du code semblent mériter plus d'attention ?

L'OBJECTIF

L'objectif de ces exercices est de commencer à réfléchir aux lignes du code qui en définissent l'essence, qui ont l'impact le plus important ou auxquelles il faut prêter une attention particulière.



CONTENU



L'objectif de ces exercices est de commencer à réfléchir aux lignes du code qui en définissent l'essence, qui ont l'impact le plus important ou auxquelles il faut prêter une attention particulière.

CONTENU



Commencez par examiner les noms des choses.

- Le nom des variables
- Le nom des fonctions et méthodes
- Le nom des classes

CONTENU, EN GROUPE AVEC UN SCRIBE



5-7 MINUTE

- Nous sommes le scribe !
- Vous nous donnez les noms.
- Nous co-construisons cette liste de noms

DÉBRIEF

DÉBRIEF DU CONTENU



5 MINUTES

- Que peut-on apprendre de ces noms ?
- Quels éléments sont liés les uns aux autres, d'après les noms uniquement ?
 - Grouper les éléments qui sont similaires ou proches
 - Regrouper en sous domaines, si cela est pertinent
- Y a-t-il des noms qui sont ambigus lorsqu'on les regarde sans contexte ?

IDENTIFIEZ LES LIGNES LES PLUS IMPORTANTES



8 MINUTES

- indépendamment, identifiez les 5 lignes que vous considérez comme les plus importantes

DÉBRIEF

PARTAGEZ VOS LIGNES LES PLUS IMPORTANTES



10 MINUTES

<https://lead.nwp.org/knowledgebase/save-the-last-word-for-me-protocol/>

Les lignes les plus souvent citées:

- 10 +
- 41 +
- 43 +
- 49 +
- 103 +
- 111 +
- 139 +
- 151 +++++
- 160 +++++

Les lignes moins citées:

- 3
- 7
- 109
- 105
- 106
- 108
- ===
- 160 à 163
- 131
- 147

RÉSUMÉ



5 MINUTES

L'OBJECTIF DE CETTE ÉTAPE EST DE RÉFLÉCHIR
À LA FONCTION PRINCIPALE DE CE CODE.

- individuellement, essayez d'écrire l'essence du code en quelques phrases

DÉBRIEF

RÉSUMÉ, DISCUSSION DE GROUPE



5 MINUTES

- Sujets les plus souvent cités au sein des résumés versus ceux moins cités
- Quelle stratégie avez vous utilisez pour réaliser ce résumé (noms des méthodes, documentation, noms des variables, connaissance à priori du système...) ?

MIND MAP



Photo by RODNAE

MERCI

@guillaume_agile



https://twitter.com/guillaume_agile

@acassaigne@piaille.fr

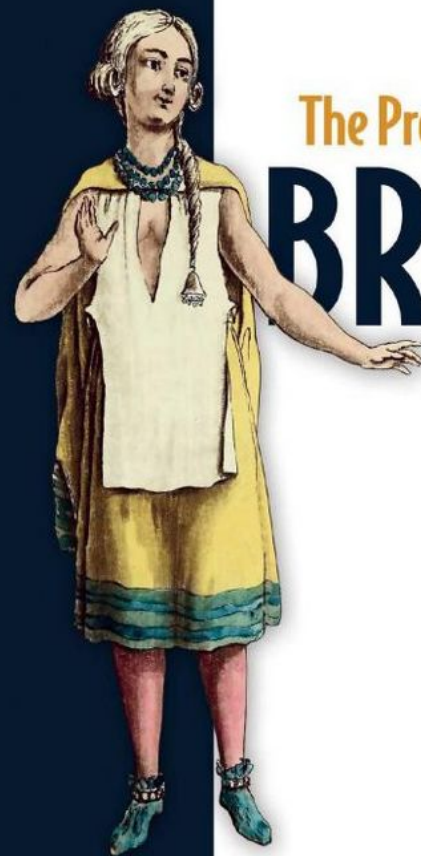


<https://piaille.fr/@acassaigne>

FELIENNE HERMANS

<https://codereading.club/>

<https://www.felienne.com/archives/6472>



Copyrighted Material

What every programmer needs to know about cognition

The Programmer's **BRAIN**

Felienne Hermans

Foreword by Jon Skeet

 MANNING

Copyrighted Material

RESOURCES

<https://github.com/CodeReadingClubs/Resources/blob/trunk/exercises.md>

RESOURCES

Starter Kit :

<https://github.com/CodeReadingClubs/Resources/tree/trunk>

D'autres exemples de code pour des ateliers :

<https://github.com/neontribe/code-reading-club/tree/trunk/lola-club>

RESOURCES AVANCÉES

D'autres exemples de code pour des ateliers expérimentaux :

<https://github.com/neontribe/code-reading-club/tree/trunk>

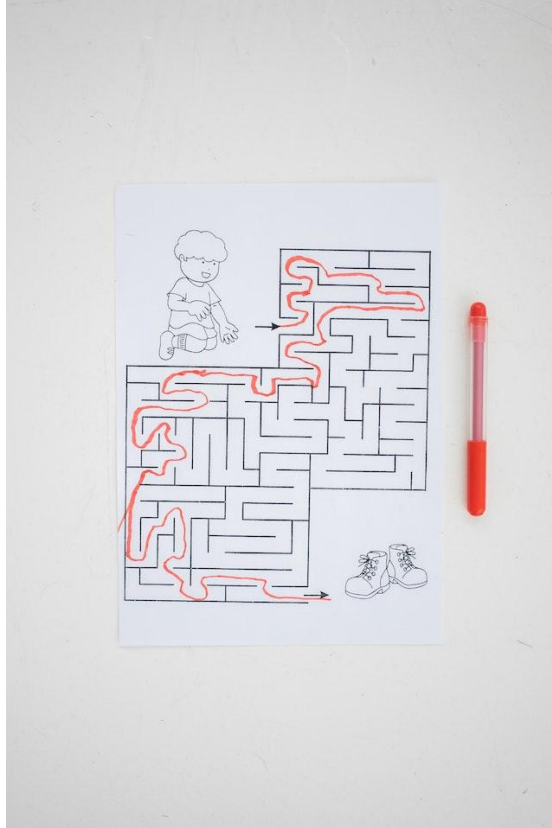
The Expectation

Participants should have some professional coding experience. Apart from that, the sessions should work for a diverse group (in terms of cognitive, formal training, experience, skills, style etc). The sessions use a variety of languages and styles of code. The key benefit should be increased ability to understand code when reading and a more sympathetic style of writing code to be read by others.

D'autres exemples de code pour des ateliers :

<https://github.com/neontribe/code-reading-club/tree/trunk/lola-club>

L'OBJECTIF



L'objectif de ces exercices est de commencer à réfléchir aux lignes du code qui en **définissent l'essence**, qui ont **l'impact le plus important** ou auxquelles il faut prêter une attention particulière.