

Lab#2

Résolution de problème et gestion du changement



Noureddine Kerzazi

Plan

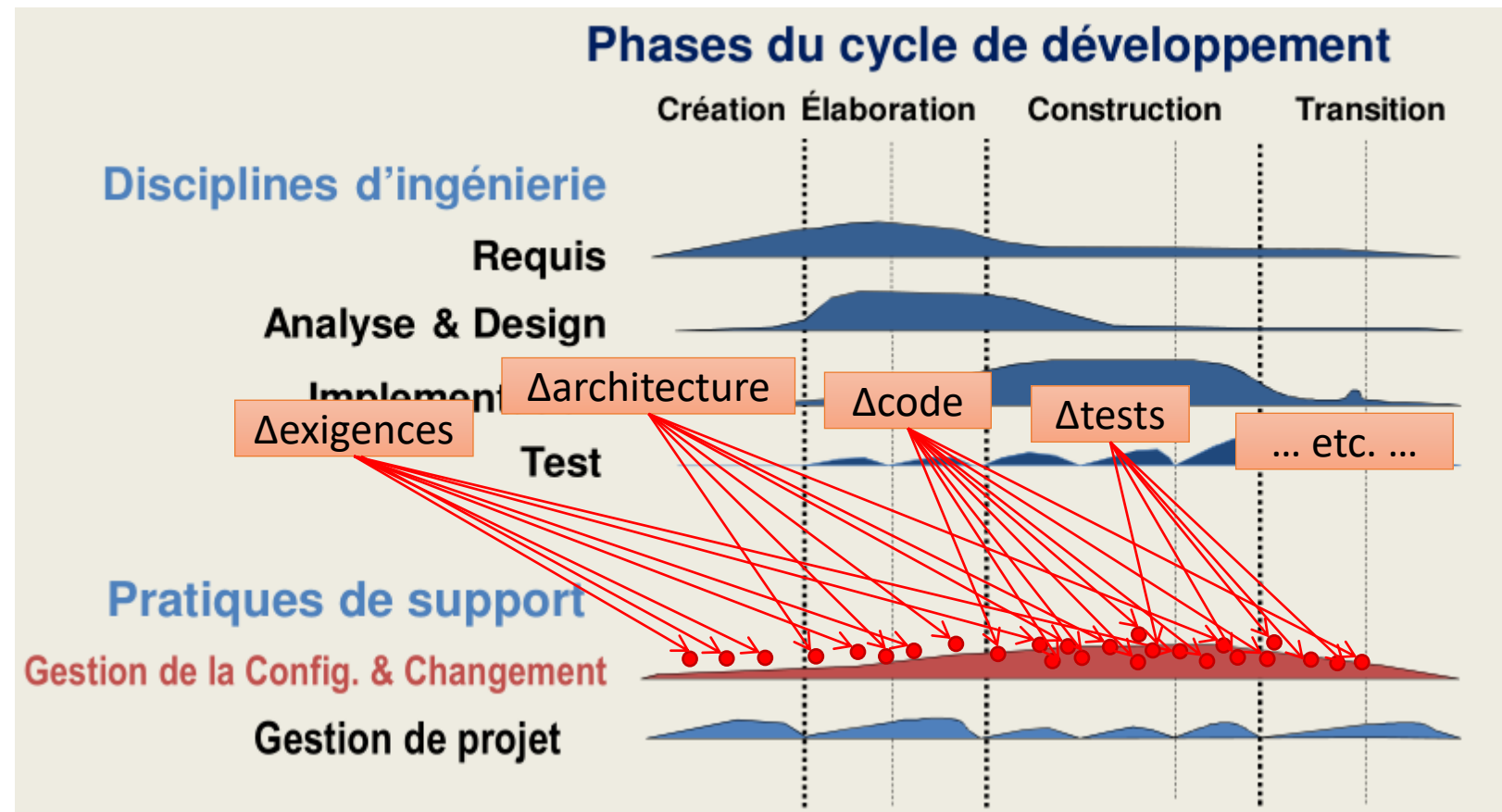
- Retour sur le TD #1
- Qu'est ce que Git, GitHub, Bitbucket ?
- Structure et concepts de base de GitHub
- Démo & exercice
- Résolution de problème : implémentation d'un arbre binaire
- **Devoir #2**

Qu'est ce que la gestion de version?

- **Version Control - Un système de gestion des modifications apportées aux documents et autres fichiers informatiques**
- **Avec quels types de fichiers pouvons-nous l'utiliser?**
 - Code source
 - Documentation
 - Histoires courtes
 - Fichiers binaires (musique et images)
- **Quand utiliser un système de gestion de version ?**
 - Fichiers texte qui évoluent
 - Projets qui ont beaucoup de révisions (modifications)
 - Pour collaborer (plusieurs personnes travaillent en même temps sur le même fichier)

Pourquoi la gestion de configuration et du changement (C&C) ?

- La gestion C&C s'applique sur la totalité des activités de développement : du changements des exigences jusqu'aux changements des tests.



Systemes de contrôle de versions

- Beaucoup de choix pour les systèmes de contrôle de versions
 - CVS
 - SVN
 - Perforce
 - Git
 - Bitbucket (base sur GIT ou Mercurial)
 - Mercurial (Hg)
 - Bazaar
 - TFS

Git, GitHub, BitBucket ?

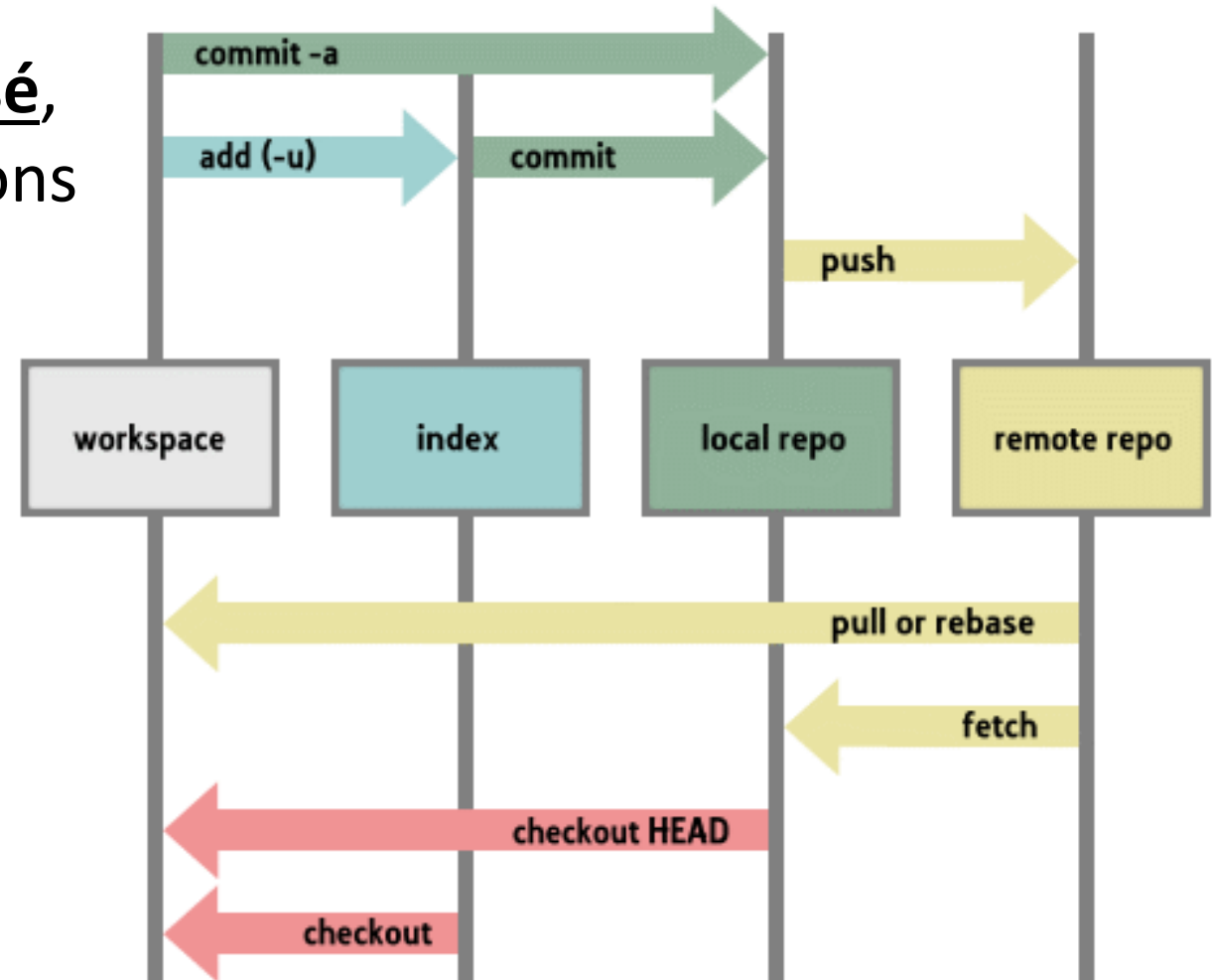


Concepts de base Git et GitHub

- **fork** a project on github.
- **clone** your github fork to your computer
- **create a topic branch** for your own work in your local clone
- **commit** changes to your local repository
- **push** the changes to your github fork
- send a **pull request** back to the original project

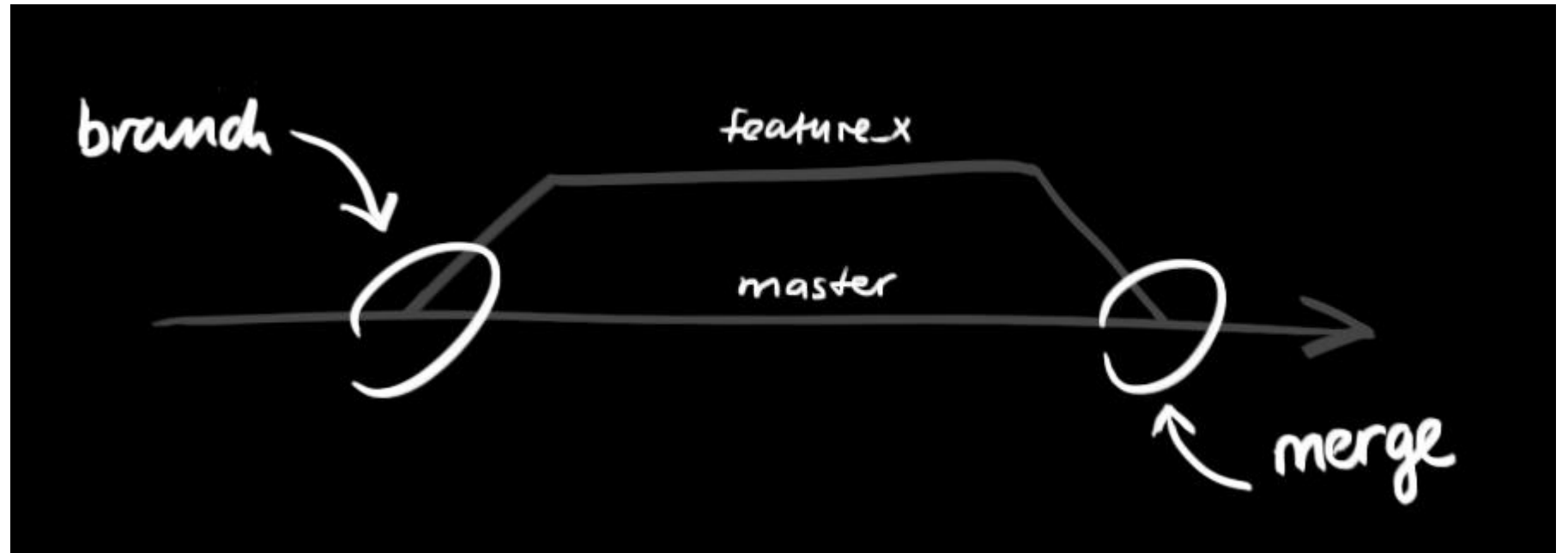
Git un système de gestion de version décentralisé

À la différence de SVN qui est centralisé,
Git est un système de gestion de versions
distribué (Local repo et Remote repo)



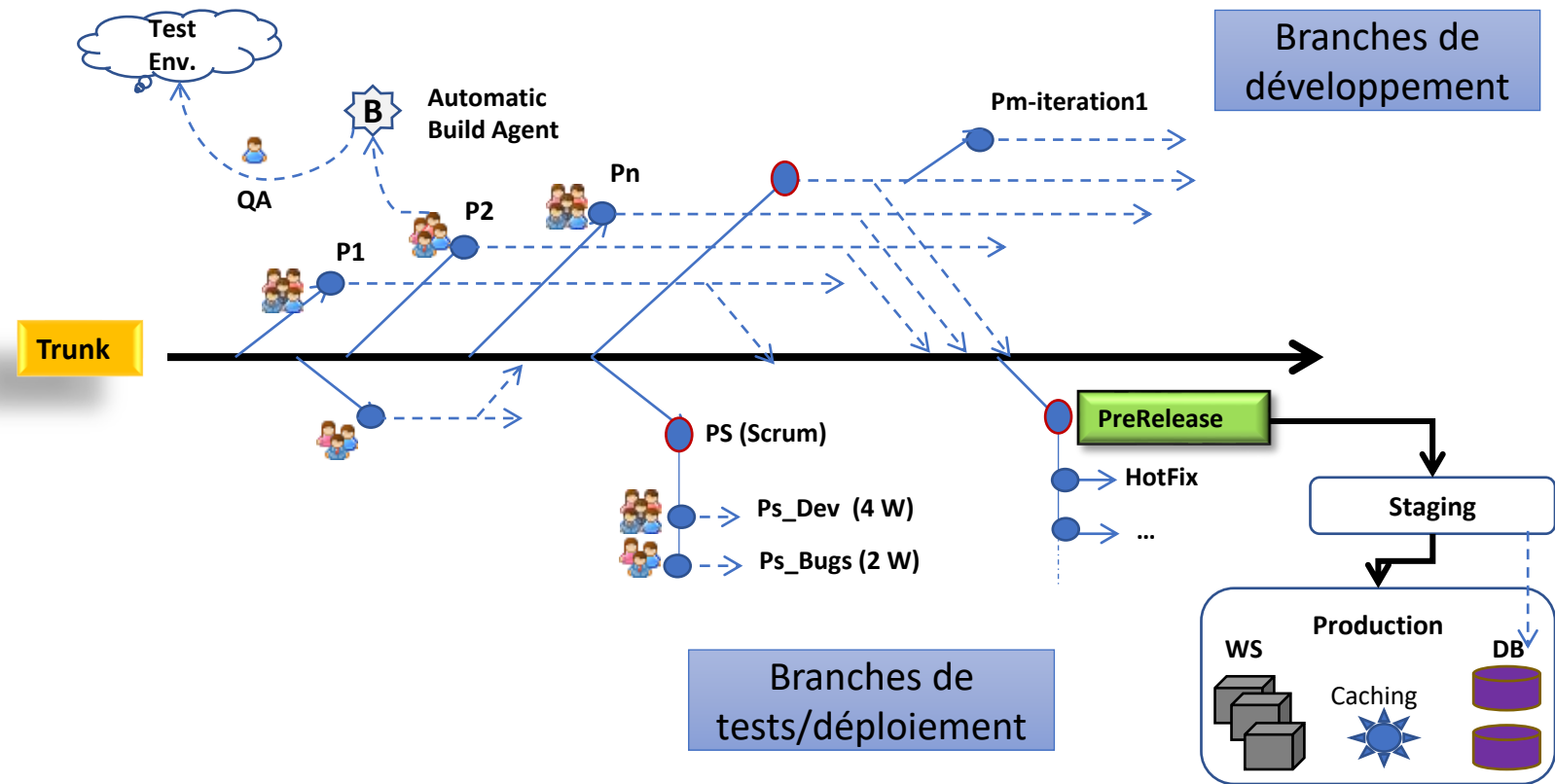
Création de branches

- Isoler le code dans une branche de développement pour ne pas perturber les autres développeurs qui travaillent sur la branche principale(master, truck, main)



Structure de branchement

- Branchement par :
 - Fonctionnalité : Branche créée pour développer la fonctionnalité, mais éliminée lorsqu'elle est en production.
 - Composant : Branches alignées avec l'architecture. La branche principale sert pour l'intégration.
 - Équipe : Chaque équipe travaille sur une branche principale isolée.
 - Livrable : Le « trunk » représente le livrable d'une version.



Résolution de problèmes

Problème

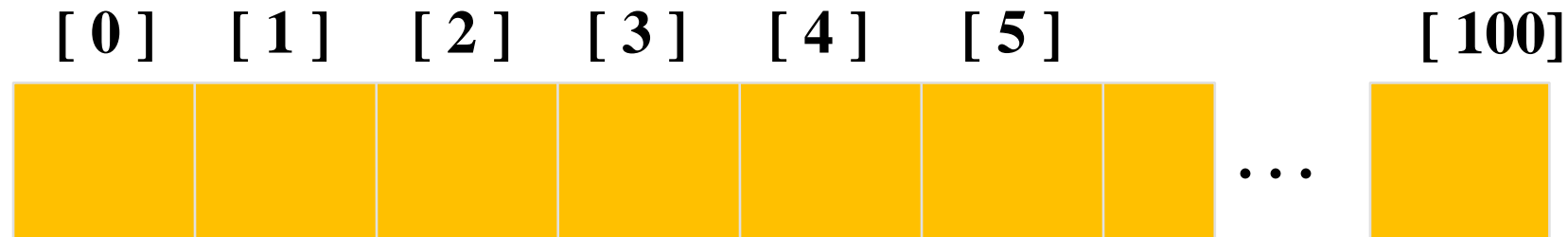
- implémenter un dictionnaire basé sur un tableau

Pour un accès rapide aux données

- Utilisation d'une Table de Hashage

Qu'est ce qu'une table de Hachage ?

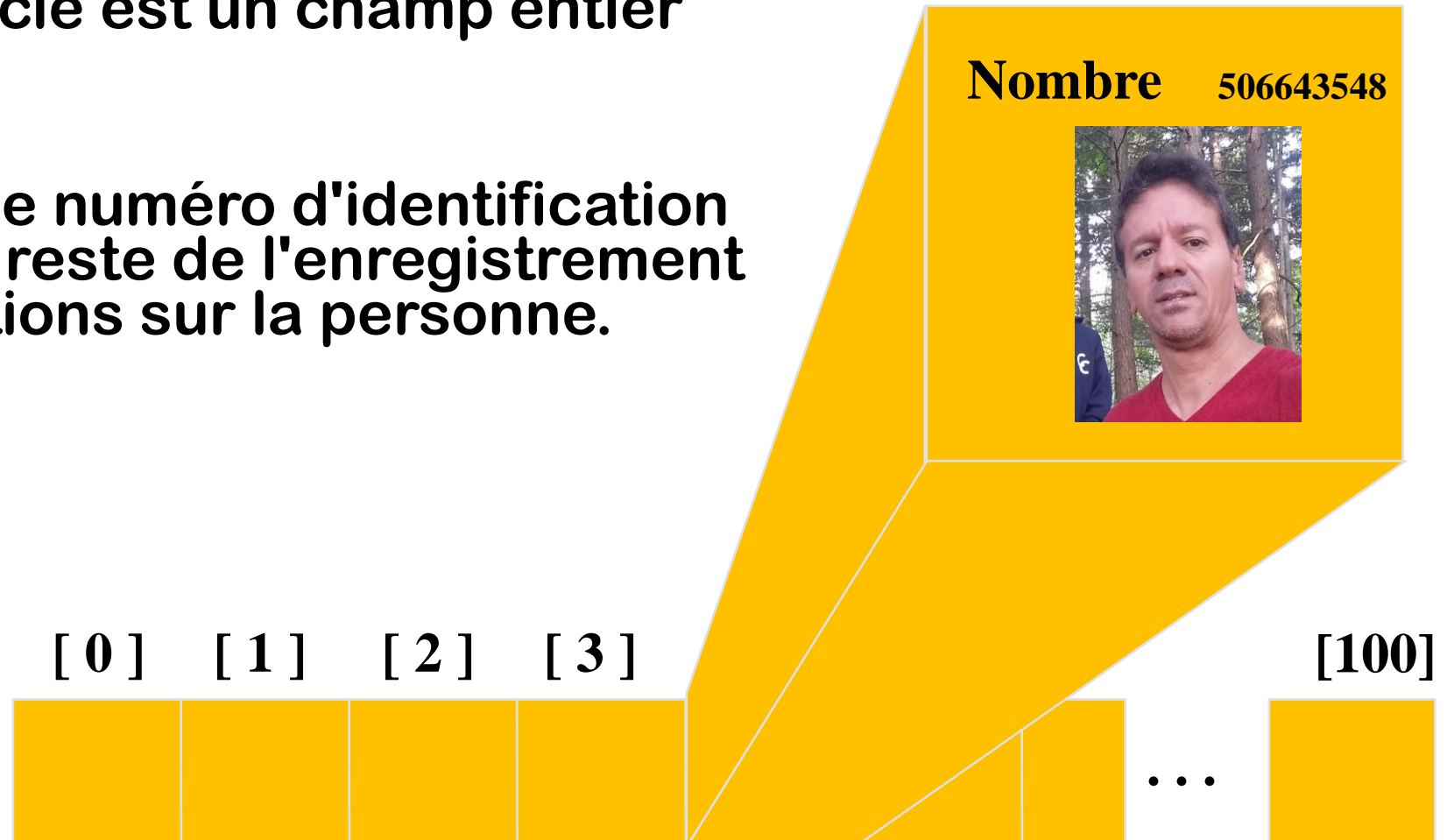
- Le type le plus simple de table de hachage est un tableau d'enregistrements.
- Exemple : soit une liste de personnes de 101 enregistrements



Un tableau d'enregistrement

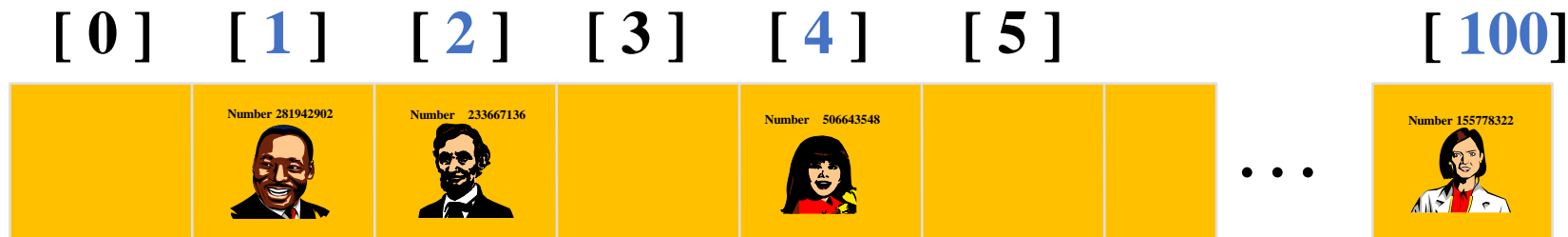
Qu'est ce qu'une table de hachage ?

- Chaque enregistrement a un champ spécial, appelé **clé**.
- Dans cet exemple, la clé est un champ entier long appelé Nombre.
- Le numéro peut être le numéro d'identification d'une personne, et le reste de l'enregistrement contient des informations sur la personne.



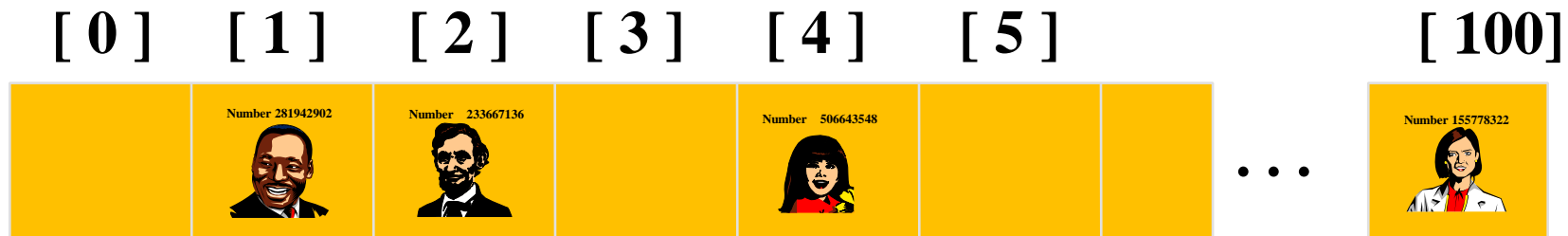
Qu'est ce qu'une table de hachage ?

- Lorsqu'une table de hachage est utilisée, certaines cases du tableau contiennent des enregistrements (1, 2, 4) et d'autres sont "vides« (0, 3, 5).



Insérer un nouvel enregistrement

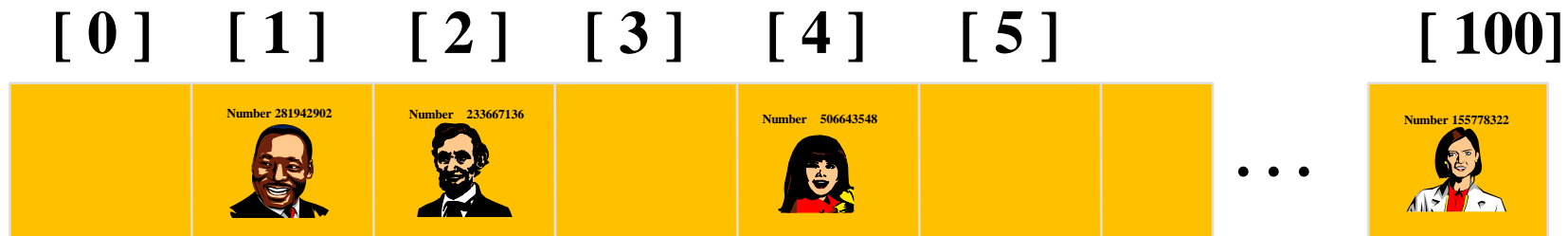
- Pour insérer un nouvel enregistrement, la **clé** doit être **convertie** en **index** de tableau.
- L'index est appelé la **valeur de hachage** de



Insérer un nouvel enregistrement

- La manière typique de créer une valeur de hachage :

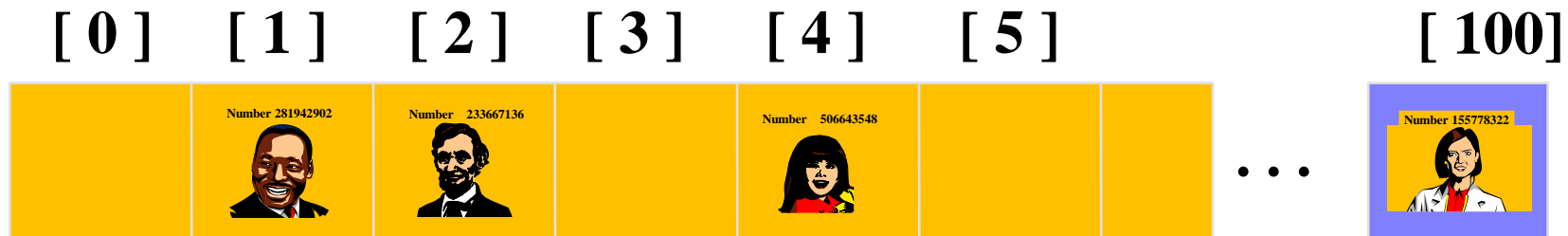
Fonction qui calcul le modulo de « Nombre »
 $= (580625685 \bmod 101) ?$



Insérer un nouvel enregistrement

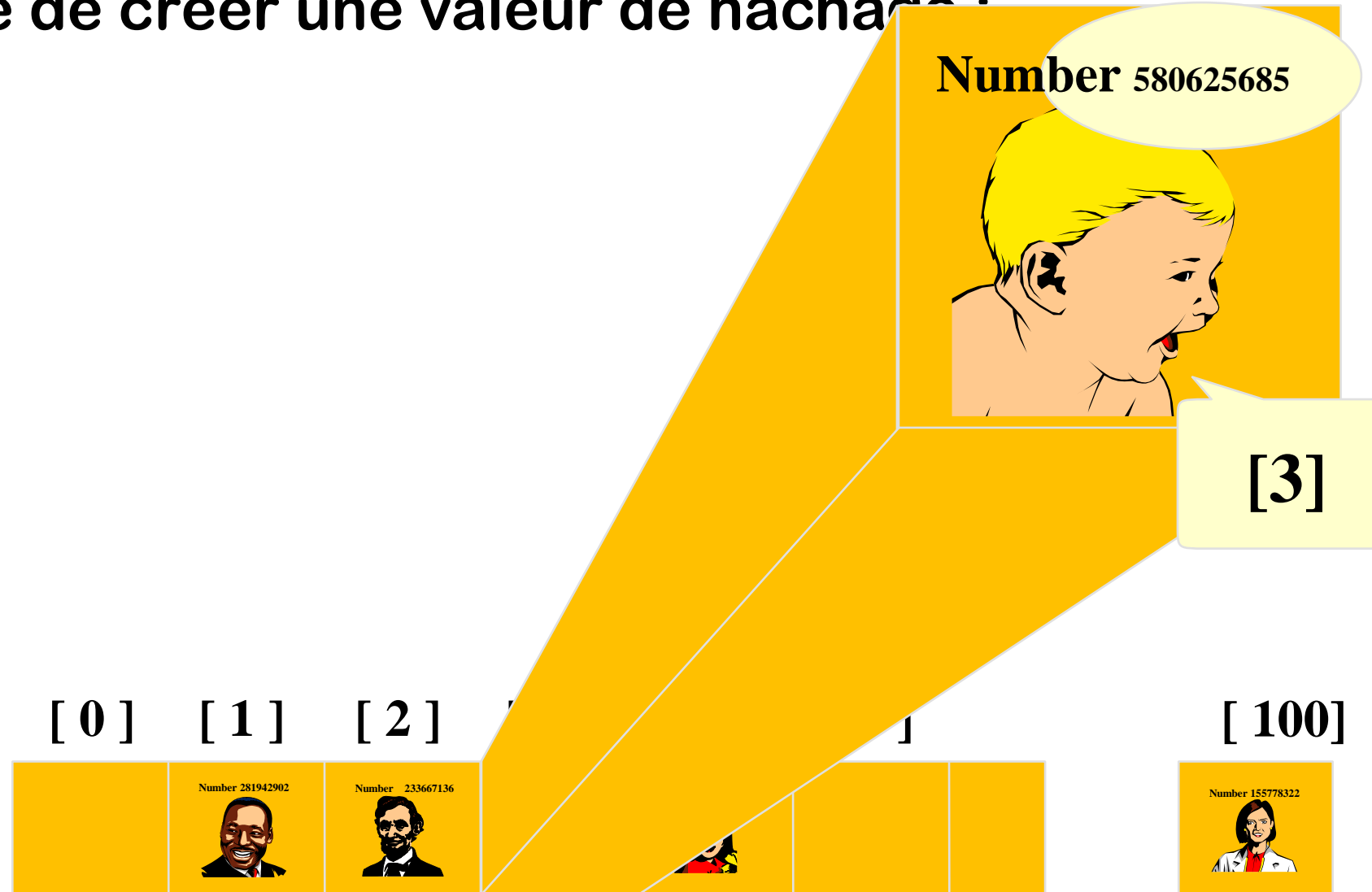
- Une façon typique pour hacher une valeur :

Combien le modulo de $(580625685 \% 101)$?



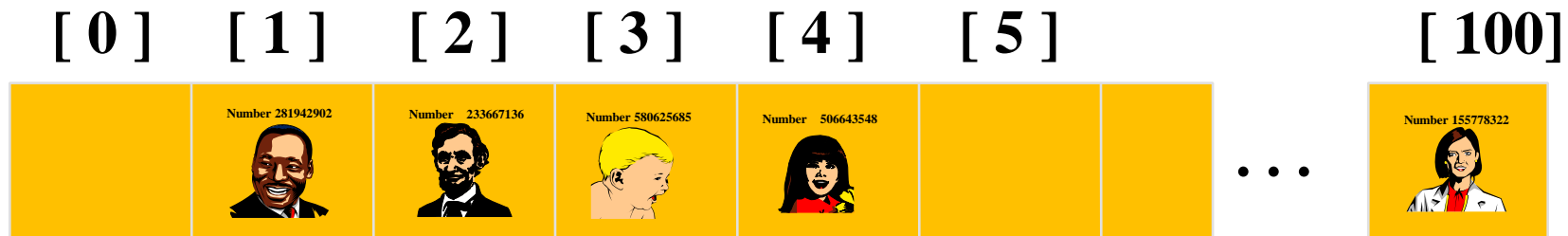
Insérer un nouvel enregistrement

- La manière typique de créer une valeur de hachage :



Insérer un nouvel enregistrement

- La valeur de hachage est utilisée pour placer le nouvel enregistrement dans le tableau (ici index 3).

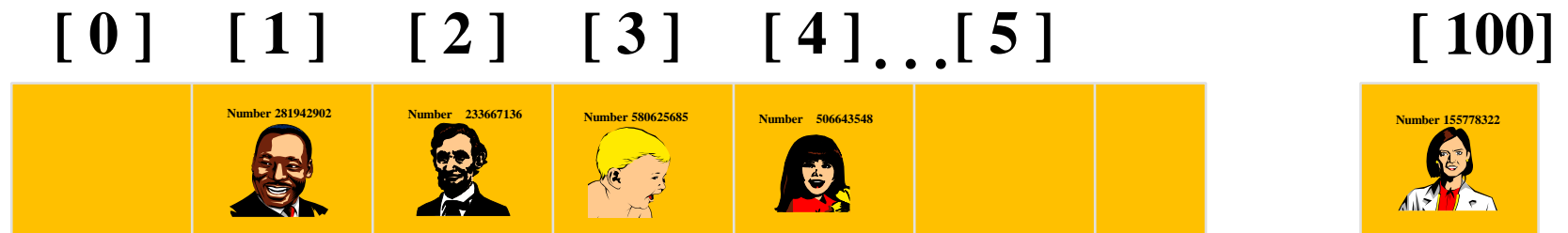


Collisions

- Voici un autre nouvel enregistrement à insérer, mais la valeur de hachage est 2 (place déjà occupée).



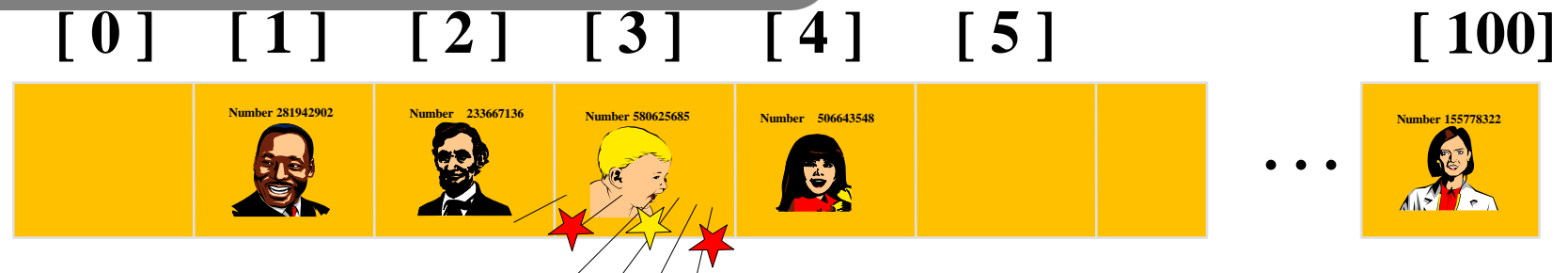
Ma valeur
de hash
est [2].



Collisions

- C'est ce qu'on appelle une **collision**, car il existe déjà un autre enregistrement valide à [2].

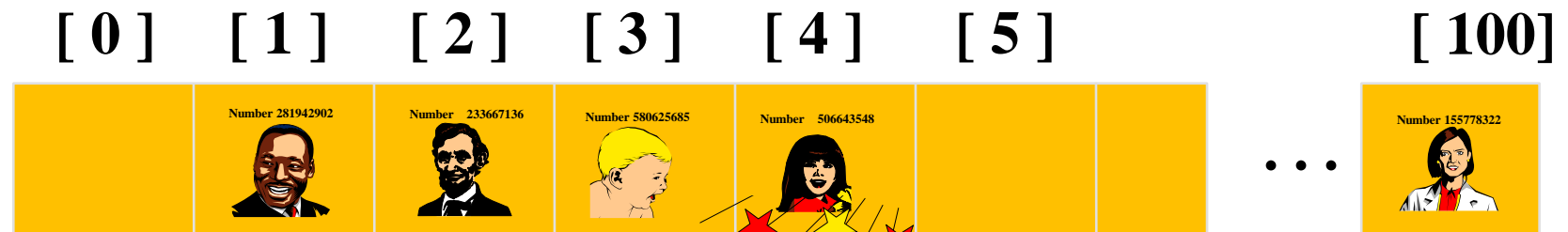
Quand une collision se produit, on peut avancer jusqu'à ce qu'on trouve un point vide.



Collisions

- C'est ce qu'on appelle une **collision**, car il existe déjà un autre enregistrement valide à [2].

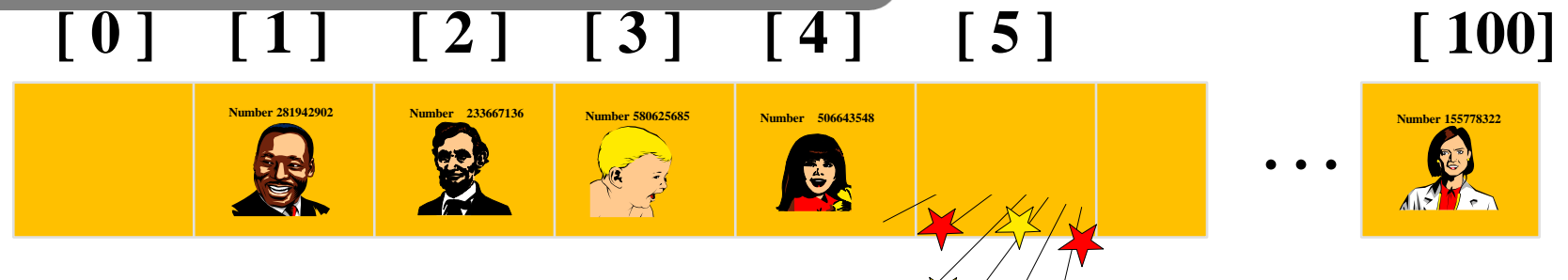
Quand une collision se produit, on peut avancer jusqu'à ce qu'on trouve un point vide.



Collisions

- C'est ce qu'on appelle une **collision**, car il existe déjà un autre enregistrement valide à [2].

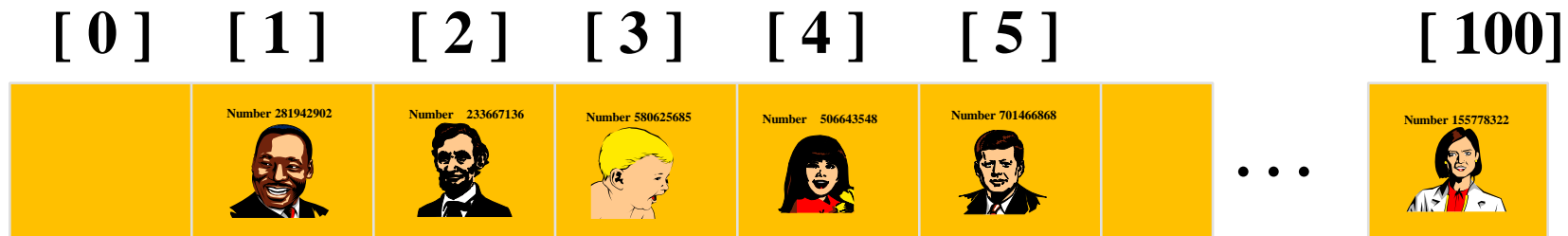
Quand une collision se produit, on peut avancer jusqu'à ce qu'on trouve un point vide.



Collisions

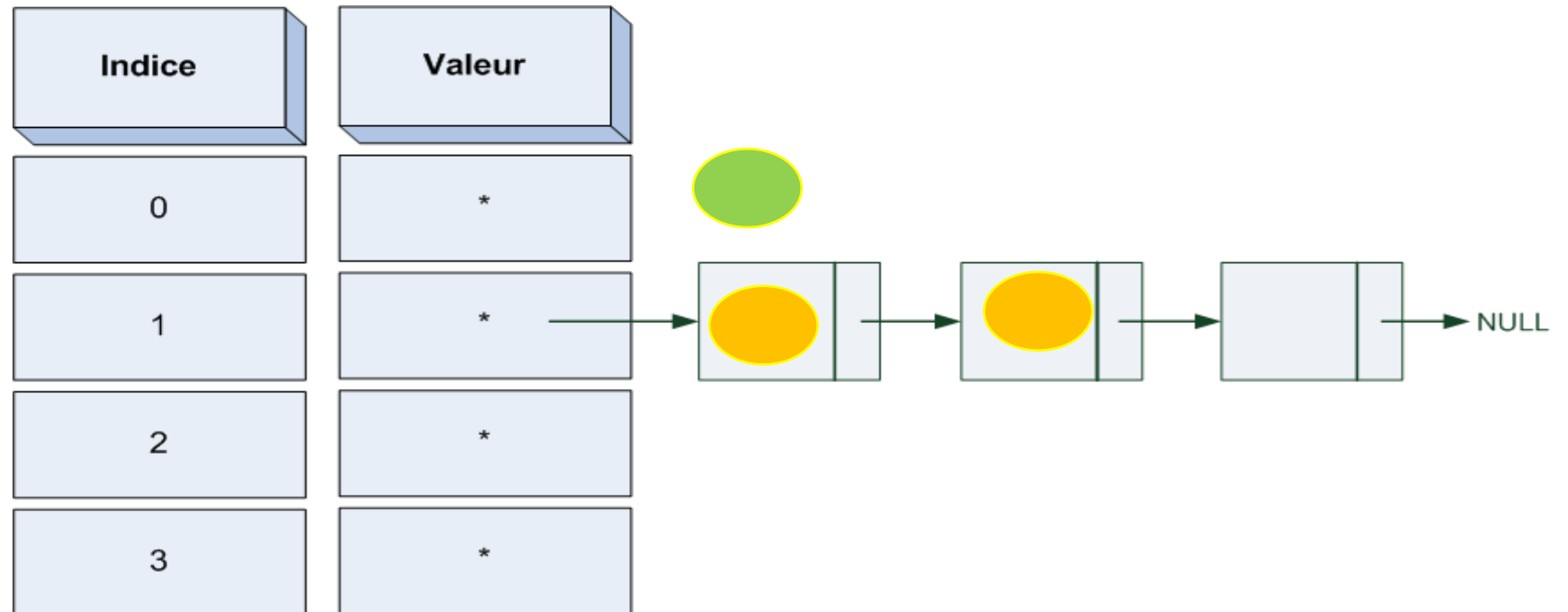
- C'est ce qu'on appelle une **collision**, car il existe déjà un autre enregistrement valide à [2].

Le nouveau enregistrement
va dans la case vide.



Résolution des collisions : le chaînage

- Une autre solution consiste à créer une liste chaînée à l'emplacement de la collision.
- Vous avez deux données (ou plus) à stocker dans la même case ? Utilisez une liste chaînée et créez un pointeur vers cette liste depuis le tableau



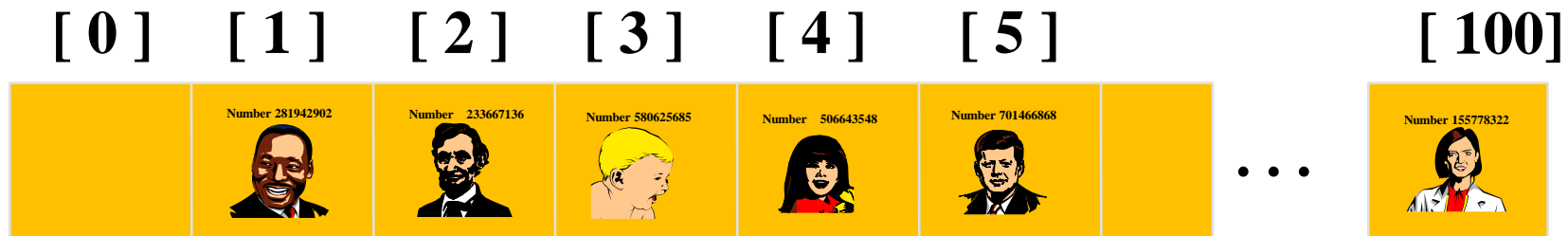
Rechercher une Clé

- Calculer la valeur de hachage.
- Vérifier cet emplacement du tableau pour la clé.

Nombre 701466868

Ma valeur
de hach est
[2].

Ce n'est pas moi



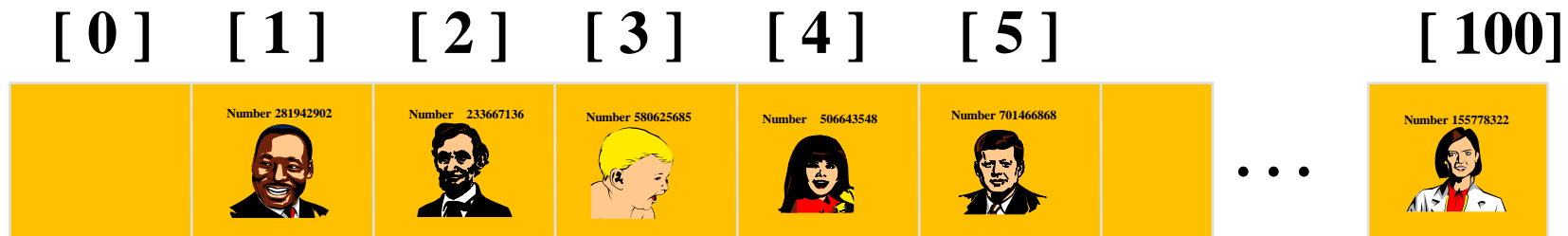
Rechercher une Clé

- Continuer d'avancer jusqu'à ce que vous trouviez la clé, ou vous atteignez un endroit vide.

Nombre 701466868

Ma valeur
de Hach [2].

Ce n'ai pas moi



Devoir #2

- Un problème d'un entrevue chez Google.
- Soit une fonction qui prend comme paramètre une chaîne de caractères et retourne le premier caractère récurrent dans la chaîne.
- "ABCA" → 'A'
- "BCABA" → 'B'
- "ABC" → NULL

Merci

Questions 