

CONTRIBUTEURS

Introduction

✓ Étapes de développement

- 1 L'analyse
 - > Quel est le *problème* ?
 - > Que veut l'*utilisateur* ?
 - > Quel est son *budget* ?
 - > Quelles sont les **conséquences** d'une *erreur* ?
- 2 La conception
 - > Comment résoudre le problème ?
 - > Quelles sont les *structures de données* appropriées ?
 - > Quels sont les *algorithmes* nécessaires ?
 - > Quelles sont les *interfaces* requises ?
- 3 La programmation
 - > Implantation de la solution développée aux étapes précédente, en utilisant un ou plusieurs langages de programmation.
- 4 Les tests d'intégration
 - > L'intégration des différents modules en un tout cohérent ;
 - > Les procédures de tests qui permettent d'établir la validité et la fiabilité du logiciel.

□ L'approche hiérarchique pour traiter des données

- > les **programmes** sont composés de **modules** ;
- > les **modules** contiennent des **énoncés** ;
- > les **énoncés** contiennent des **expressions** ;
- > les **expressions** créent et manipulent les **données**.

3 une **valeur** affectée à la variable (e.g. 3.1415) appelée **expression**.

Commentaires Des commentaires sont des lignes du code, qui commencent par #, qui ne sont pas exécutées.

Opérateurs arithmétiques

- + addition ;
- soustraction ;
- * multiplication ;
- / division régulière ;
- // division entière ;
- % reste de la division entière ;
- ** exponentiation.

Fonctions de base

≡ print

Permet d'afficher à la console la valeur d'une ou de plusieurs expressions.

- > permet aussi avec toute sorte d'options de spécifier la façon dont cette ou ces valeurs seront affichées.

≡ input

Permet de lire ce que vous entrez au clavier et retourne le résultat sous la forme d'une chaîne de caractères.

Syntaxe de base

Affectation énoncé (e.g. `pi = 3.1415`) ayant habituellement 3 éléments :

- 1 un nom de variable (e.g. `pi`) appelé **identifieur** ;
- 2 l'opérateur (e.g. `=`) ;