Exceptions

## Prévenir les erreurs d'exécution.

Dans notre exemple, introduire une date négative, non entière ou en dehors de l'intervalle valide crée ou doit créer une erreur avec un message d'erreur impliquant souvent les modules. Résoudre cette erreur demanderait donc à l'utilisateur de connaître la solution choisie pour les fonctions de l'interface. Ça ne doit pas être son problème. Une exception interrompt le programme. Les modules doivent donc gérer correctement les exceptions. Il est possible de "lever" raise volontairement une exception et de la commenter. NB on a déjà vu une manière avec assert condition, 'message', mais ici il s'agit de gérer une erreur, pas une contrainte ajoutée.

Quelques erreurs :

NameError : accès à une variable inexistante

IndexError : accès à un indice non valide

KeyError: accès à une clé de dictionnaire non valide

ZeroDivisionError : division par zéro

TypeError : opération appliqué à des valeurs incompatibles

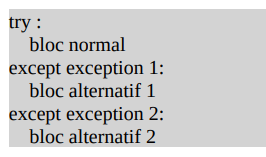
On peut personnaliser ces exceptions, par exemple : if i < 0 : raise IndexError ('Indice négatif')

On peut préciser l'interface pour aider l'utilisateur : ajoute(s, x) : ajoute un élément x à la structure s, si 1 < x < 366, sinon lève une exception ValueError ('La date doit être comprise entre 1 et 366')

# Rattraper une exception : try… except...

Lever un exception conduira à l’interruption du programme si elle se produit. On peut "rattraper" cette situation. C'est utile pour ne pas planter un programme en cas d'erreur de saisie de l'utilisateur. Il faut lui redonner une chance de saisir une nouvelle entrée. Voir fonctions saisirDate( ), saisirDate2( ), saisirDate3( )

Alternatives multiples



Ex 3 Conception collective d'un mini projet : Gestion d'un stock produits et édition de factures avec calcul du montant dû. Plusieurs groupes se partagent l'implémentation des modules (L'assemblage doit fonctionner). Le stock de produits est sauvegardé dans stock.csv : définir les champs utiles. Les factures doivent être sauvegardées en fichier texte, sous le nom facture\_num.txt. num est un numéro entier attribué automatiquement, dans l'ordre croissant. Définir l'interface du programme principal (les fonctions utilisateurs). Penser aux fonctions d'aide, d'affichage... Créer tous les modules utiles.