



- ▶ Présentation générale
- ▶ QUESTIONNAIRE : VOS ATTENTES ET MOTIVATIONS
- ▶ Semaine 1. Introduction au MOOC et aux outils Python
- ▶ Semaine 2. Notions de base pour écrire son premier programme en Python
- ▼ **Semaine 3. Renforcement des notions de base, références partagées**
  - 1. Les fichiers  
Quiz Échéance le janv 25, 2018 at 23:30 UTC
  - 2. Les tuples  
Quiz Échéance le janv 25, 2018 at 23:30 UTC
  - 3. Tables de hash  
Quiz Échéance le janv 25, 2018 at 23:59 UTC
  - 4. Les dictionnaires  
Quiz Échéance le janv 25, 2018 at 23:59 UTC
  - 5. Les ensembles

## QUIZ 16 - LES EXCEPTIONS (2 points possibles)

### Exceptions (1)

Cochez la ou les affirmations exactes.

☐ Une exception provoque toujours l'arrêt du programme

☒ Une exception non capturée

Rechercher un cours



☐ Une exception non capturée remonte la pile d'exécution ✓

☐ Les exceptions représentent le mécanisme standard de notification des erreurs d'exécution dans un programme ✓

☒ On peut capturer une exception et déclencher une action pour corriger le problème notifié ✓



#### EXPLANATION

Une exception capturée ne va pas provoquer l'arrêt du programme. Il existe également des exceptions (comme `StopIteration`) qui sont automatiquement capturée (par exemple, par une boucle `for`) sans provoquer d'arrêt du programme.

Une exception non capturée n'est jamais perdue, elle remonte la pile d'exécution jusqu'au moment où le programme s'arrête.

Les exceptions sont très efficaces en python et sont utilisées de manière standard pour communiquer à l'intérieur d'un programme.

Avec une instruction `try except`, on peut capturer une exception et faire une action corrective.

### Exceptions (2)



## 6. Les exceptions

Quiz Échéance le janv  
25, 2018 at 23:59 UTC

## 7. Les références

### partagées

Quiz Échéance le janv  
25, 2018 at 23:59 UTC

## 8. Introduction aux classes

Quiz Échéance le janv  
25, 2018 at 23:59 UTC

- ▶ Semaine 4.  
Fonctions et  
portée des  
variables
- ▶ Semaine 5.  
Itération,  
importation et  
espace de  
nommage
- ▶ Semaine 6.  
Conception  
des classes
- ▶ Semaine 7.  
L'écosystème  
data science  
Python
- ▶ Semaine 8.  
Programmation  
asynchrone -  
asyncio

☐ Il vaut mieux capturer `Exception` plutôt que `TypeError`, ainsi  
notre code sera plus robuste.

☒ On peut capturer l'exception n'importe où le long de la pile  
d'exécution. ✓

☐ Il vaut mieux capturer l'exception au plus proche de là où elle est  
produite. ✓



### EXPLANATION

On doit capturer les exceptions qui sont prévisibles et donc, on doit  
éviter de capturer `Exception` qui permet de capturer toutes les  
exceptions d'un coup et qui donc peut cacher une exception que  
nous n'avons pas prévue et pour laquelle on n'a implémenté aucun  
comportement spécifique.

Une exception peut se capturer n'importe où dans la pile  
d'exécution, mais on recommande en général de capturer  
l'exception au plus près de là où elle est produite.

*Vous avez utilisé 3 essais sur 3*

[A propos](#)

[Aide](#)

[Contact](#)



[Politique de confidentialité](#)

[Mentions légales](#)



POWERED BY  
OPENedX