



- ▶ 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
- ▼ **Semaine 4. Fonctions et portée des variables**

1. Fonctions

Quiz Echéance le janv 25, 2018 at 23:30 UTC

2. Tests if/elif/else et opérateurs booléens

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

3. Boucles while

QUIZ 22 - PORTÉE DES VARIABLES (3/3 points)

Portée des variables (1)

Dans quel cas rencontre-t-on une exception `UnboundLocalError` ?

☐ si on référence une variable globale depuis une fonction

☐ si on affecte dans une fonction une variable déjà définie globalement

☒ si on référence dans une fonction une variable définie en dessous dans la fonction ✓

☐ si on affecte deux fois une variable locale dans une fonction



EXPLANATION

L'exception `UnboundLocalError` ne se produit que si on référence dans une fonction une variable définie en dessous dans la fonction.

Portée des variables (2)

Dans quel cas la variable `var` est locale à la fonction.

Proposition 1

```
var = 'globale'
def f():
    tmp = 'locale'
    return var
```

Proposition 2



Portée des variables - règle LEGB

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

5. Modification de la portée avec global et nonlocal

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

6. Passage d'arguments et appel de fonctions

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

- 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

```
return var
```

Proposition 3

```
var = 'globale'
def f():
    print(var)
    var = 'locale'
    return var
```

Choisissez une ou plusieurs propositions

☐ Proposition 1

☒ Proposition 2 ✓

Rechercher un cours

☐ Proposition 3



EXPLANATION

Dans la proposition 1, la variable `var` n'est jamais affectée dans la fonction, donc elle n'est pas une variable locale à la fonction, mais une variable globale.

Dans la proposition 2, la variable `var` est affectée à la première ligne de la fonction. Elle devient donc locale à la fonction.

Dans la proposition 3, la variable `var` est affectée à la deuxième ligne de la fonction, mais après avoir été référencée à la première ligne. Dans ce cas, Python juge qu'il y a une ambiguïté et va lancer une exception particulière qui n'est lancée que dans ce cas précis : `UnboundLocalError`. Il y a un complément spécifique sur cette erreur qu'il est très important de comprendre.

Portée des variables (3)

Que va afficher : `print(f())` ?



```
def f():  
    def g():  
        return var  
    return g()  
print(f())
```

Choisissez une proposition.

☐ 10

☒ 20 ✓

☐ None

☐ Une exception

EXPLANATION

Suivant la règle LEGB, la valeur de `var` dans `g` est `20`. Comme `g` retourne la valeur de `var` et que `f` retourne la valeur de retour de `g`, `f()` retourne `20`.

Vous avez utilisé 2 essais sur 3

[A propos](#)

[Aide](#)

[Contact](#)

[Conditions générales d'utilisation](#)

[Charte utilisateurs](#)

[Politique de confidentialité](#)

[Mentions légales](#)