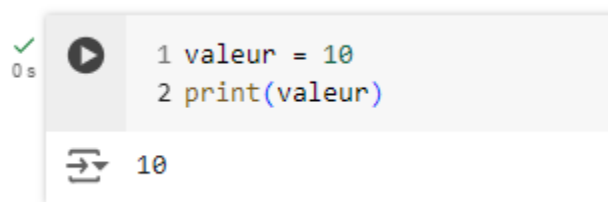


Aide-mémoire pour Google Colab

Relancer la session

Utilisation d'un mot-clé comme nom de variable

Disons qu'on a une cellule toute simple



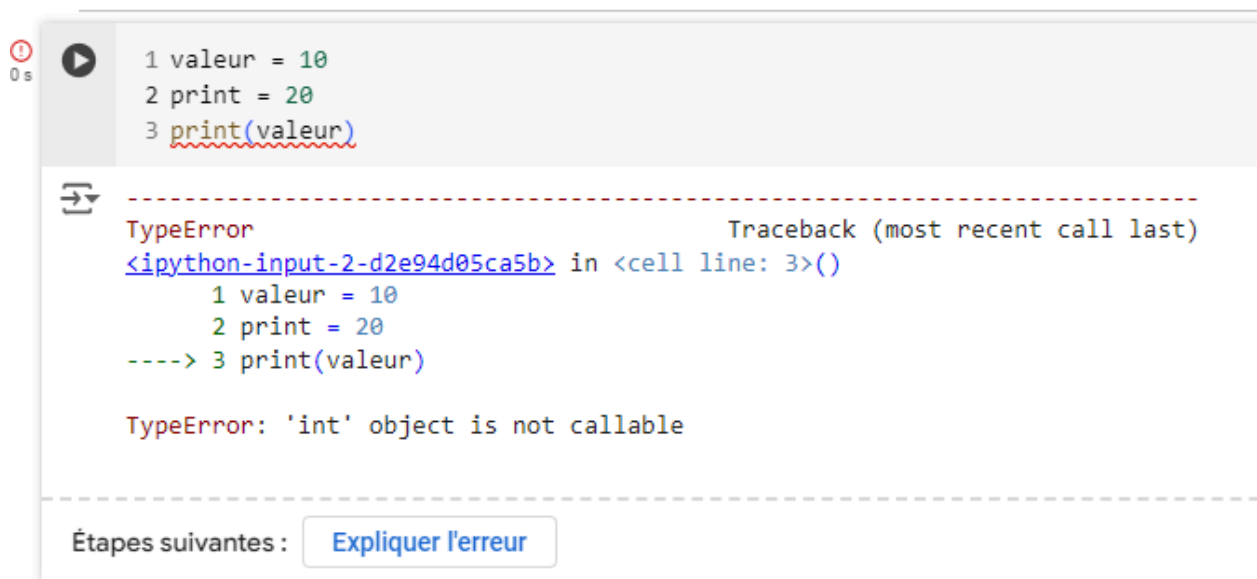
A screenshot of a Google Colab code cell. On the left, there is a green checkmark icon and a play button icon, with '0 s' indicating execution time. The code contains two lines: '1 valeur = 10' and '2 print(valeur)'. Below the code, a variable inspector shows 'valeur' with the value '10'.

```
1 valeur = 10
2 print(valeur)
```

10

L'exécution fonctionne.

Ça arrive qu'on fasse une erreur. Par exemple, sans s'en rendre compte, on utilise un mot-clé comme variable



A screenshot of a Google Colab code cell showing an error. On the left, there is a red error icon and a play button icon, with '0 s' indicating execution time. The code contains three lines: '1 valeur = 10', '2 print = 20', and '3 print(valeur)'. The third line is underlined with a red squiggly line. Below the code, a traceback shows the error: 'TypeError: 'int' object is not callable'. At the bottom, there is a button labeled 'Expliquer l'erreur'.

```
1 valeur = 10
2 print = 20
3 print(valeur)
```

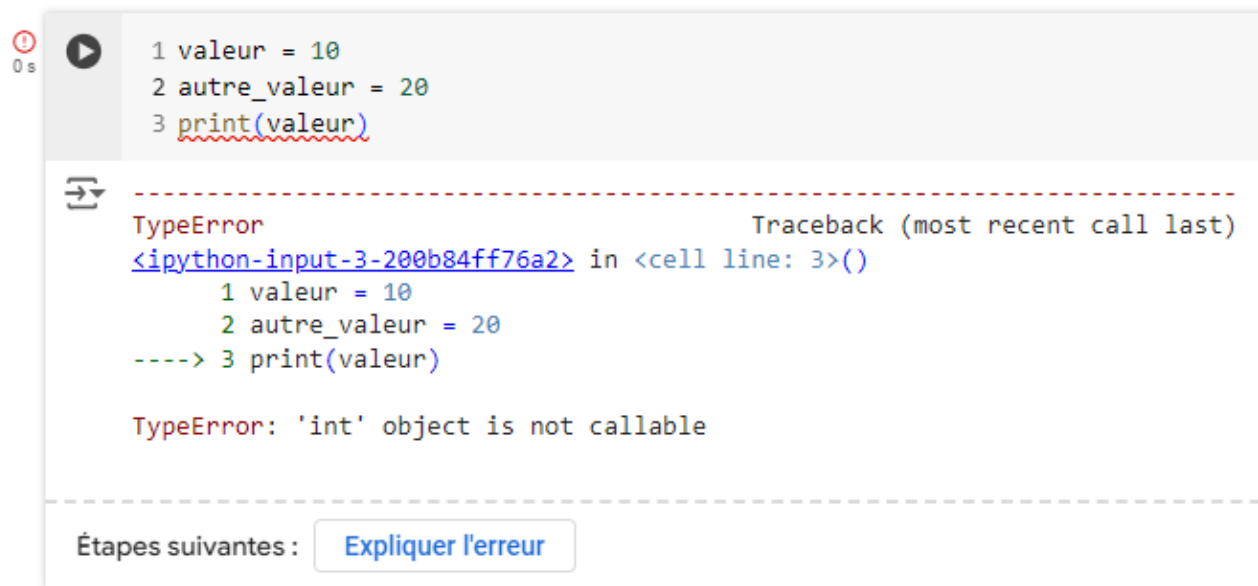
TypeError Traceback (most recent call last)
<ipython-input-2-d2e94d05ca5b> in <cell line: 3>()
 1 valeur = 10
 2 print = 20
----> 3 print(valeur)

TypeError: 'int' object is not callable

Étapes suivantes : [Expliquer l'erreur](#)

Le code ne voit plus la fonction *print* mais voit maintenant la variable *print*.

Même si on corrige l'erreur, ça ne marche plus.



```
1 valeur = 10
2 autre_valeur = 20
3 print(valeur)
```

TypeError Traceback (most recent call last)

<ipython-input-3-200b84ff76a2> in <cell line: 3>()

1 valeur = 10

2 autre_valeur = 20

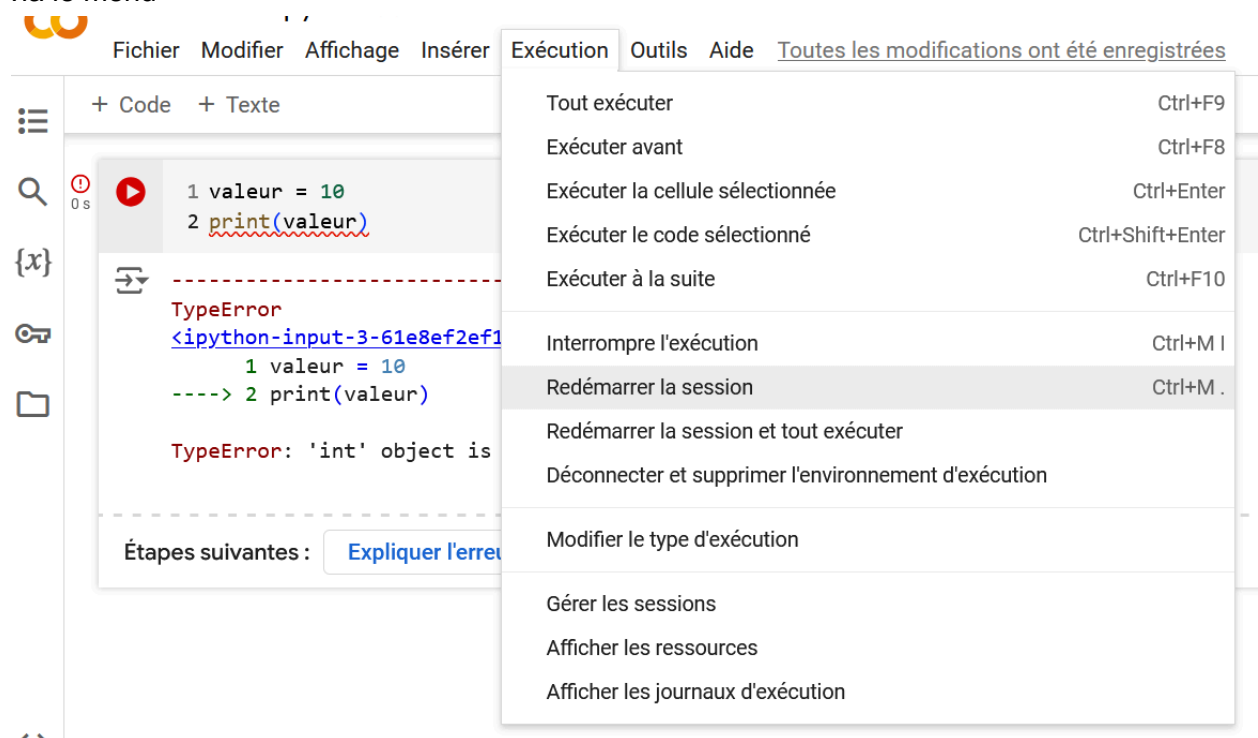
----> 3 print(valeur)

TypeError: 'int' object is not callable

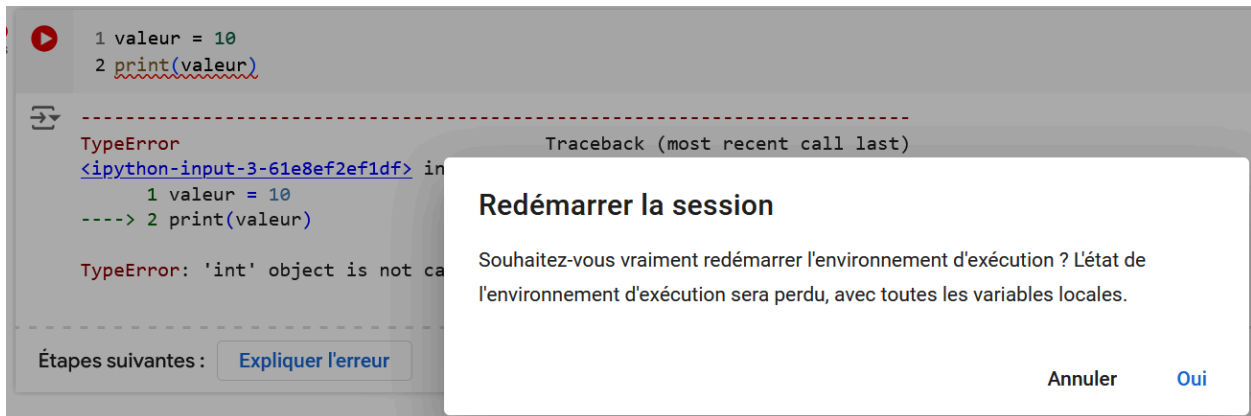
Étapes suivantes : [Expliquer l'erreur](#)

Effacer la cellule ne règle pas le problème.

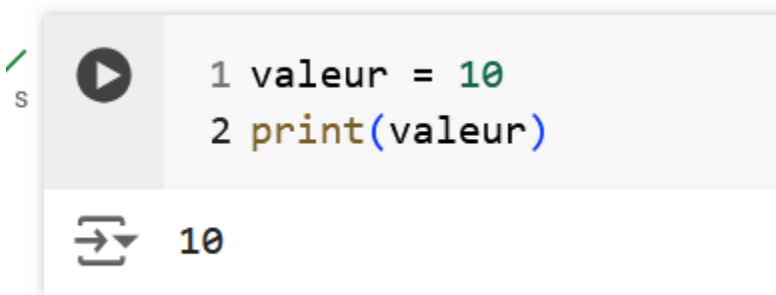
On peut fermer l'onglet et le ré-ouvrir mais une solution plus rapide est "Redémarrer la session" via le menu



ou en tapant Ctrl-M . (on appuie sur Ctrl-M puis sur le point). Une fenêtre apparaît qui nous demande de confirmer l'action



Après avoir cliqué sur “oui

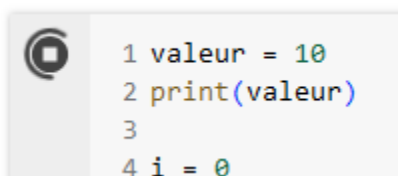


Attention:

Cette action donne le même résultat que de fermer le notebook et de le réouvrir. Il faut donc ré-exécuter les cellules Import et Fonctions.

Interrompre l'exécution

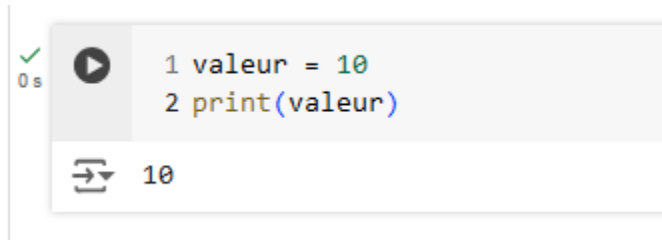
Il arrive, des fois, que l'exécution ne se complète pas.



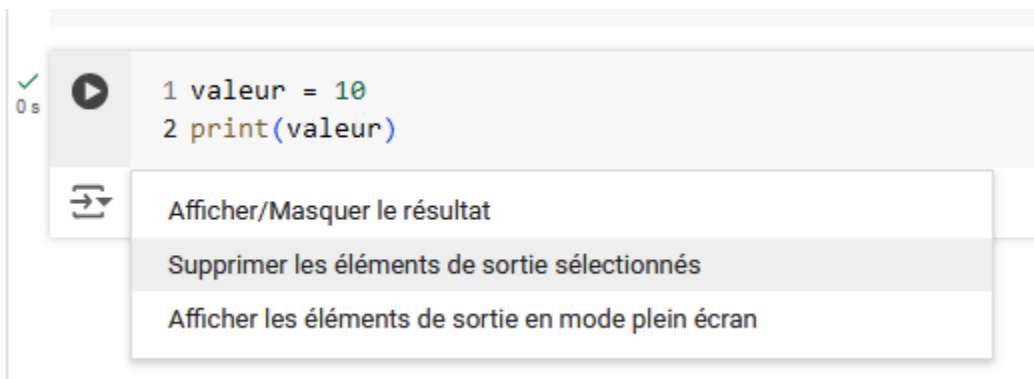
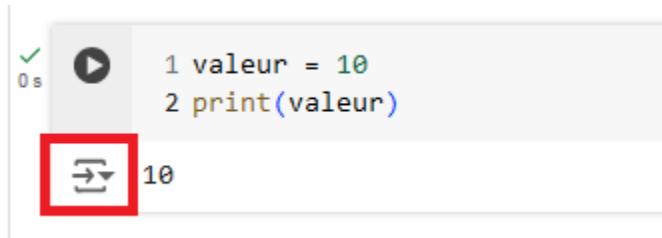
Si c'est le cas, vous pouvez l'arrêter en choisissant le menu Exécution/Interrompre l'exécution.

Supprimer les résultats

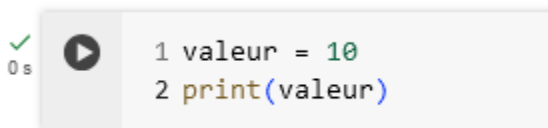
Une fois qu'on exécute une cellule, il y a habituellement une sortie qui est affichée.



Vous pouvez masquer la sortie ou même l'effacer



Plus de sortie



Ça peut être pratique.

Activer/Désactiver des lignes de codes

En informatique, on dit qu'on "commente" une ligne de code en insérant un # au début de la ligne.

```
[2] 1 valeur = 10
     2 # valeur = 20
     3 print(valeur)
```

Cela désactive la ligne 2 dans cet exemple.

Quand on écrit du code, il arrive, pendant nos tests, qu'on essaie plusieurs algorithmes. Pour rapidement basculer entre les algorithmes, on veut "commenter" et décommenter (ré-activer) des lignes de code. Pour le faire rapidement, on sélectionne les lignes qu'on veut commenter et on appuie sur Ctrl+K, Ctrl+C

Avant

```
▶ 1 valeur = 10
   2 resultat = calculer_algo1(valeur)
   3 resultat = calculer_algo2(valeur)
   4 print(valeur)
```

On appuie sur Ctrl-K, Ctrl-C et on obtient

```
▶ 1 valeur = 10
   2 resultat = calculer_algo1(valeur)
   3 # resultat = calculer_algo2(valeur)
   4 print(valeur)
```

On peut utiliser Ctrl-K, Ctrl-U pour "décommenter" des lignes. Dans l'exemple ici, on n'a sélectionné qu'une seule ligne mais vous pouvez en sélectionner plusieurs en même temps.

Il y a aussi d'autres raccourcis. Vous allez dans Outils/Raccourcis clavier



ge Insérer Exécution Outils Aide

Palette de commandes Ctrl+Shift+P

Paramètres

Raccourcis clavier Ctrl+M H

Diff. entre notebooks

pour les trouver

