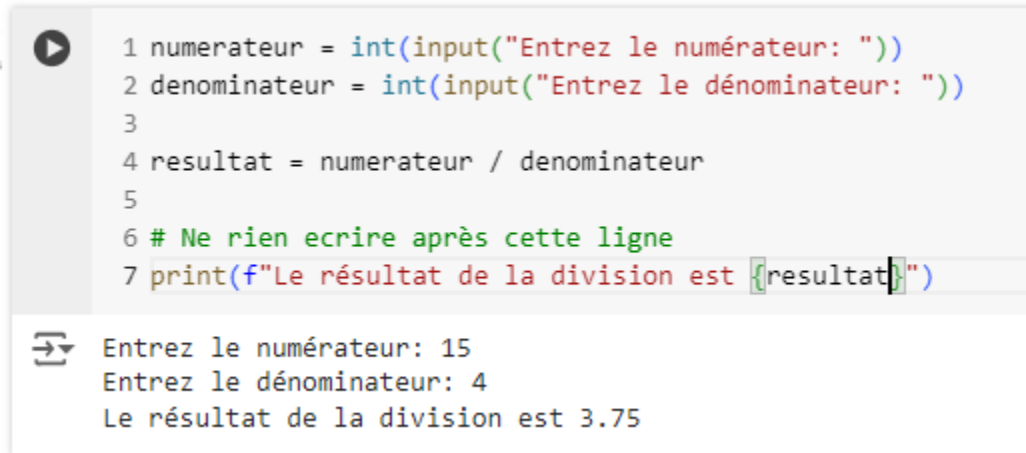


Gestion des erreurs

Disons qu'on a une cellule qui divise deux entiers et affiche le résultat.



```
1 numérateur = int(input("Entrez le numérateur: "))
2 dénominateur = int(input("Entrez le dénominateur: "))
3
4 resultat = numérateur / dénominateur
5
6 # Ne rien écrire après cette ligne
7 print(f"Le résultat de la division est {resultat}")
```

↳ Entrez le numérateur: 15
Entrez le dénominateur: 4
Le résultat de la division est 3.75

Le code est simple. Et ça marche.

Mais que se passe-t-il si on saisit la valeur 0 pour le dénominateur

```
1 numérateur = int(input("Entrez le numérateur: "))
2 dénominateur = int(input("Entrez le dénominateur: "))
3
4 resultat = numérateur / dénominateur
5
6 # Ne rien écrire après cette ligne
7 print(f"Le résultat de la division est {resultat}")
```

➡ Entrez le numérateur: 15
Entrez le dénominateur: 0

```
ZeroDivisionError                                Traceback (most recent call last)
<ipython-input-2-af04017dc5ef> in <cell line: 4>()
      2 dénominateur = int(input("Entrez le dénominateur: "))
      3
----> 4 resultat = numérateur / dénominateur
      5
      6 # Ne rien écrire après cette ligne

ZeroDivisionError: division by zero
```

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

Comme pour un humain, l'opération *diviser par 0* est indéfinie. L'exécution s'arrête avec un message d'erreur. L'ordinateur ne sait pas gérer ce cas-là.

Mais maintenant qu'on connaît l'opération `if`, on peut gérer nous même ce cas.

On se crée une variable de type `bool` qui nous indique si une erreur a été détectée.

✓
4 s



```
1 numérateur = int(input("Entrez le numérateur: "))
2 dénominateur = int(input("Entrez le dénominateur: "))
3
4 # on se crée un variable de type bool (True ou False)
5 # et on l'initialise à False ce qui indique qu'aucune erreur n'a été détectée
6 erreur = False
7 # avant de faire la division, on valide le dénominateur
8 if dénominateur == 0:
9     # si le dénominateur est 0, il est préférable de ne pas exécuter la division
10    erreur = True # On a détecté un cas d'erreur
11 else:
12    resultat = numérateur / dénominateur
13
14 # Ne rien écrire après cette ligne
15 if erreur:
16    print("Division par zéro est impossible")
17 else: # Le else est exécuté que si erreur est False
18    print(f"Le résultat de la division est {resultat}")
```



```
Entrez le numérateur: 15
Entrez le dénominateur: 0
Division par zéro est impossible
```

Si on détecte une situation anormale, la ligne 12 n'est pas exécutée.

Dans le cas normal, l'exécution produit le résultat demandé:

✓
1 s



```
1 numérateur = int(input("Entrez le numérateur: "))
2 dénominateur = int(input("Entrez le dénominateur: "))
3
4 # on se crée un variable de type bool (True ou False)
5 # et on l'initialise à False ce qui indique qu'aucune erreur n'a été détectée
6 erreur = False
7 # avant de faire la division, on valide le dénominateur
8 if dénominateur == 0:
9     # si le dénominateur est 0, il est préférable de ne pas exécuter la division
10    erreur = True # On a détecté un cas d'erreur
11 else:
12    resultat = numérateur / dénominateur
13
14 # Ne rien écrire après cette ligne
15 if erreur:
16    print("Division par zéro est impossible")
17 else: # Le else est exécuté que si erreur est False
18    print(f"Le résultat de la division est {resultat}")
```



```
Entrez le numérateur: 15
Entrez le dénominateur: 4
Le résultat de la division est 3.75
```