

Informatique Pour Tous : Fiche sur les normes d'écritures PEP8

72 caractères par ligne au maximum !

Description	Code valide	Code invalide
Positionnement des espaces	<code>spam(ham[1], {eggs: 2})</code>	<code>spam(·ham[·1·], {·eggs·:·2·})</code>
Pas d'espace après une fonction	<code>spam(1)</code> <code>dct['key'] := lst[index]</code>	<code>spam·(1)</code> <code>dct·[·'key'·] :=·lst·[index]</code>
Même nombre d'espaces avant et après un opérateur	<code>i := i + 1</code> <code>submitted += 1</code> <code>x = x * 2 - 1</code> <code>hypot2 = x * x + y * y</code> <code>c = (a + b) * (a - b)</code>	<code>i=i+1</code> <code>submitted+=1</code> <code>x:=x*2-1</code> <code>hypot2:=x*x+y*y</code> <code>c:=(a+b)*(a-b)</code>
Définition des fonctions	<code>def f(x): return 2*x</code>	<code>f:=lambda x: 2*x</code>
Retour à la ligne dans une expression	<code>income = (gross_wages ·····+ taxable_interest ·····+ (div - qual_div) ·····- ira_deduction ·····- student_interest)</code>	<code>income = (gross_wages + ·····taxable_interest + ·····(div - qual_div) - ·····ira_deduction - ·····student_interest)</code>
Séparation des périodes importantes	<code>for i in range(10): ···for j in range(10): ·····print(i, j) ···#Commentaire ou nouvelle ligne ···#pour séparer les boucles. ···print("Nouveau I")</code>	<code>for i in range(10): ···for j in range(10): ·····print(i, j) ···print("Nouveau I")</code>
Indentation	<code>A = [a, b, c, d, ···f, g, h, i, ···] ou A = [a, b, c, d, ···f, g, h, i, ·]</code>	<code>A = [a, b, c, d, f, g, h, i] A = [a, b, c, d, f, g, h, i ·]</code>

Nom des variables

Ne jamais utiliser les caractères suivants pour donner des noms de variables

- 🚫 "l" (L minuscule)
- 🚫 "O" (de Oh)
- 🚫 "I" (i majuscule)

Les variables qui représentent des constantes sont notées en CAPITALES (souvent au début) :

`ITERATION_NUMBER = 4`

Chez python, "Soyez Zen" (cf. PEP 20)

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!



Accès aux pages identiques de PEP8