

PyBR - Guide pour Débutants

Maxwell Anderson Ielpo do Amaral

Guide Complet pour Débutants

Apprendre la programmation Python en français

Janeiro/2026

PyBR - Python International
<https://github.com/maxwellamaral/pybr>

Table des matières

Apprendre à Programmer avec PyBR - Guide pour Débutants	2
Bienvenue dans le Monde de la Programmation ! ☐	2
Table des Matières	2
☐ Utilisation du Terminal	3
☐ Installation de Python	3
Exécuter PyBR	3
Votre Premier Programme	4
Variables - La Mémoire de l'Ordinateur	4
Calculs et Opérations Mathématiques	5
Entrée et Sortie	5
Prendre des Décisions - Conditionnelles	5
Répéter des Actions - Boucles	6
Organiser le Code - Fonctions	6
Fonctions Avancées	7
Créer des Objets - Classes	7
Projets Pratiques	7
Conseils Finaux	8
Félicitations ! ☐	8

Apprendre à Programmer avec PyBR - Guide pour Débutants

Auteur : Maxwell Anderson Ielpo do Amaral Traduction : AI Assistant

Publié en Janvier 2026

Bienvenue dans le Monde de la Programmation ! ☐

Ce guide a été créé spécialement pour vous qui n'avez jamais programmé auparavant et souhaitez apprendre facilement et en français ! Avec **PyBR**, vous apprendrez à programmer en utilisant des mots en français au lieu de l'anglais traditionnel de Python.

Table des Matières

1. Utilisation du Terminal
2. Installation de Python
3. Exécuter PyBR
4. Qu'est-ce que la Programmation ?
5. Votre Premier Programme
6. Variables - La Mémoire de l'Ordinateur
7. Calculs et Opérations Mathématiques
8. Entrée et Sortie
9. Prendre des Décisions - Conditionnelles
10. Répéter des Actions - Boucles
11. Organiser le Code - Fonctions

-
- 12. Fonctions Avancées
 - 13. Créer des Objets - Classes
 - 14. Projets Pratiques
-

□ Utilisation du Terminal

Si vous n'avez jamais utilisé le **Terminal**, ne vous inquiétez pas ! C'est plus simple qu'il n'y paraît.

Commandes de Base

Action	Windows	Mac/Linux
Où suis-je ?	cd	pwd
Lister les fichiers	dir	ls
Entrer dans un dossier	cd dossier	cd dossier
Revenir en arrière	cd ..	cd ..
Effacer l'écran	cls	clear

□ Installation de Python

Avant de commencer, vous devez avoir **Python** installé.

1. Ouvrez le terminal et tapez : `python --version`
 2. Si Python 3.x.x apparaît, vous êtes prêt !
 3. Sinon, téléchargez-le sur python.org.
 - **Windows** : Important ! Cochez “**Add Python to PATH**” lors de l’installation !
-

Exécuter PyBR

Prérequis

□ Python 3.6+ □ Fichiers PyBR (pybr.py)

Façons d’Exécuter

Option 1 : Mode Interactif (REPL) Parfait pour des tests rapides. Dans le terminal :

```
python pybr.py --lang fr
```

Vous verrez :

```
PyBR - Python International (Multilingue)
Tapez 'quitter()' pour terminer.
>>>
```

Option 2 : Exécuter des Fichiers Créez un fichier `mon_programme.pybr` et exécutez-le :

```
python pybr.py mon_programme.pybr --lang fr
```

Votre Premier Programme

Commençons par le classique “Bonjour le Monde” :

```
imprimer("Bonjour le Monde !")
```

Essayez vous-même :

```
imprimer("J'apprends à programmer avec PyBR !")
```

Variables - La Mémoire de l'Ordinateur

Les **variables** sont comme des boîtes où vous stockez des informations.

Créer des variables :

```
# Sauvegarder un nom  
nom = "Marie"
```

```
# Sauvegarder un âge  
age = 25
```

```
# Utiliser les variables  
imprimer(nom)  
imprimer(age)
```

Types de Données :

```
# TEXTE (String)  
ville = "Paris"
```

```
# NOMBRES ENTIERS (Int)  
quantite = 10
```

```
# NOMBRES DÉCIMAUX (Float)  
prix = 19.99
```

```
# VRAI ou FAUX (Boolean)  
est_lundi = Vrai  
il_pleut = Faux
```

Calculs et Opérations Mathématiques

```
# ADDITION (+)
somme = 10 + 5
imprimer(somme) # Affiche : 15

# SOTTRAZIONE (-)
difference = 20 - 8
imprimer(difference) # Affiche : 12

# MULTIPLICATION (*)
produit = 6 * 7
imprimer(produit) # Affiche : 42

# DIVISION (/)
resultat = 15 / 3
imprimer(resultat) # Affiche : 5.0
```

Entrée et Sortie

Sortie (Afficher des informations) :

```
nom = "Pierre"
imprimer(f"Je m'appelle {nom}")
```

Entrée (Recevoir des informations) :

```
nom = saisir("Comment t'appelles-tu ? ")
imprimer(f"Bonjour, {nom} !")
```

```
# Pour les nombres, il faut convertir :
age = entier(saisir("Quel âge as-tu ? "))
imprimer(f"Tu as {age} ans")
```

Prendre des Décisions - Conditionnelles

Le programme prend des décisions avec `si`, `sinon_si` et `sinon`.

```
age = 18

si age >= 18:
    imprimer("Tu es majeur")
sinon:
    imprimer("Tu es mineur")
```

Exemple avec SI-SINON_SI-SINON :

```
note = 85

si note >= 90:
    imprimer("Excellent !")
sinon_si note >= 70:
    imprimer("Bien !")
sinon:
    imprimer("Doit s'améliorer")
```

Répéter des Actions - Boucles

Boucle POUR (for) :

```
# Compter de 0 à 4
pour i dans intervalle(5):
    imprimer(i)
```

Boucle TANT_QUE (while) :

```
compteur = 0

tant_que compteur < 5:
    imprimer(f"Compteur : {compteur}")
    compteur = compteur + 1
```

Organiser le Code - Fonctions

Les fonctions sont des blocs de code réutilisables.

```
definir saluer(nom):
    imprimer(f"Bonjour, {nom} !")

saluer("Sophie")
saluer("Marc")
```

Fonctions avec Retour :

```
definir additionner(a, b):
    retourner a + b

resultat = additionner(10, 20)
imprimer(resultat) # 30
```

Fonctions Avancées

Lambda :

Petites fonctions anonymes.

```
double = lambda x: x * 2
imprimer(double(5)) # 10
```

Appliquer (map) :

```
nombres = [1, 2, 3, 4]
carres = liste(appliquer(lambda x: x ** 2, nombres))
imprimer(carres) # [1, 4, 9, 16]
```

Filtrer (filter) :

```
nombres = [1, 2, 3, 4, 5, 6]
pairs = liste(filtrer(lambda x: x % 2 == 0, nombres))
imprimer(pairs) # [2, 4, 6]
```

Créer des Objets - Classes

Les classes sont des "moules" pour créer des objets.

classe Chien:

```
definir __init__(soi, nom, race):
    soi.nom = nom
    soi.race = race

definir aboyer(soi):
    imprimer(f"{soi.nom}: Wouf wouf !")

# Créer des objets
rex = Chien("Rex", "Berger Allemand")
rex.aboyer()
```

Projets Pratiques

Projet 1 : Liste de Tâches

```
classe GestionnaireTaches:
    definir __init__(soi):
        soi.taches = []

    definir ajouter(soi, tache):
        soi.taches.append(tache)
        imprimer(f"Tâche ajoutée : {tache}")
```

```
definir lister(soi):
    imprimer("--- Mes Tâches ---")
    pour i, tache dans enumerate(soi.taches):
        imprimer(f"{i + 1}. {tache}")

gestionnaire = GestionnaireTaches()
gestionnaire.ajouter("Apprendre PyBR")
gestionnaire.ajouter("Pratiquer Python")
gestionnaire.lister()
```

Projet 2 : Convertisseur de Température

```
definir celsius_en_fahrenheit(c):
    retourner (c * 9/5) + 32

temp_c = flottant(saisir("Température en Celsius : "))
temp_f = celsius_en_fahrenheit(temp_c)
imprimer(f"{temp_c}°C est égal à {temp_f}°F")
```

Conseils Finaux

1. **Pratiquez tous les jours** : La régularité est la clé.
2. **Lisez les erreurs** : Elles vous disent ce qui ne va pas.
3. **Commentez le code** : Utilisez # pour expliquer.

Félicitations ! ☺

Vous avez terminé le guide complet de PyBR en français. Vous êtes maintenant un programmeur !