

PyBR - Guida per Principianti

Maxwell Anderson Ielpo do Amaral

Guida Completa per Principianti

Impara a programmare in Python usando l'italiano

Janeiro/2026

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

Indice

Impara a Programmare con PyBR - Guida per Principianti	2
Benvenuti nel Mondo della Programmazione! □	2
Indice	2
□ Uso del Terminale	3
□ Installare Python	3
Eseguire PyBR	3
Il tuo Primo Programma	4
Variabili - La Memoria del Computer	4
Calcoli e Operazioni Matematiche	5
Input e Output	5
Prendere Decisioni - Condizionali	5
Ripetere Azioni - Cicli	6
Organizzare il Codice - Funzioni	6
Funzioni Avanzate	7
Creare Oggetti - Classi	7
Progetti Pratici	7
Consigli Finali	8
Congratulazioni! □	8

Impara a Programmare con PyBR - Guida per Principianti

Autore: Maxwell Anderson Ielpo do Amaral Traduzione: AI Assistant

Pubblicato nel Gennaio 2026

Benvenuti nel Mondo della Programmazione! □

Questa guida è stata creata appositamente per te che non hai mai programmato prima e vuoi imparare in modo semplice e in italiano! Con **PyBR**, imparerai a programmare usando parole in italiano invece del tradizionale inglese di Python.

Indice

1. Uso del Terminale
2. Installare Python
3. Eseguire PyBR
4. Cos'è la Programmazione?
5. Il tuo Primo Programma
6. Variabili - La Memoria del Computer
7. Calcoli e Operazioni Matematiche
8. Input e Output
9. Prendere Decisioni - Condizionali
10. Ripetere Azioni - Cicli
11. Organizzare il Codice - Funzioni
12. Funzioni Avanzate

- 13. Creare Oggetti - Classi
- 14. Progetti Pratici

□ Uso del Terminale

Se non hai mai usato il **Terminale** (o **Prompt dei Comandi**), non preoccuparti! È più semplice di quanto sembri.

Comandi Base

Azione	Windows	Mac/Linux
Dove sono?	cd	pwd
Elencare file	dir	ls
Entrare in cartella	cd cartella	cd cartella
Tornare indietro	cd ..	cd ..
Pulire schermo	cls	clear

□ Installare Python

Prima di iniziare, devi avere **Python** installato.

1. Apri il terminale e digita: `python --version`
 2. Se appare Python 3.x.x, sei pronto!
 3. Se no, scaricalo su python.org.
 - **Windows:** Importante! Spunta **“Add Python to PATH”** durante l’installazione!
-

Eseguire PyBR

Requisiti

□ **Python 3.6+** □ **File PyBR** (`pybr.py`)

Modi per Eseguire

Opzione 1: Modalità Interattiva (REPL) Perfetto per test rapidi. Nel terminale:

```
python pybr.py --lang it
```

Vedrai:

PyBR - Python Internazionale (Multilingue)

Digita 'esci()' per terminare.

```
>>>
```

Opzione 2: Eseguire File Crea un file `mio_programma.pybr` ed esegilo:

```
python pybr.py mio_programma.pybr --lang it
```

Il tuo Primo Programma

Iniziamo con il classico “Ciao Mondo”:

```
stampa("Ciao Mondo!")
```

Prova tu stesso:

```
stampa("Sto imparando a programmare con PyBR!")
```

Variabili - La Memoria del Computer

Le **variabili** sono come scatole dove conservi le informazioni.

Creare variabili:

```
# Salvare un nome
```

```
nome = "Maria"
```

```
# Salvare un'età
```

```
eta = 25
```

```
# Usare le variabili
```

```
stampa(nome)
```

```
stampa(eta)
```

Tipi di Dati:

```
# TESTO (String)
```

```
citta = "Roma"
```

```
# NUMERI INTERI (Int)
```

```
quantita = 10
```

```
# NUMERI DECIMALI (Float)
```

```
prezzo = 19.99
```

```
# VERO o FALSO (Boolean)
```

```
e_lunedì = Vero
```

```
piove = Falso
```

Calcoli e Operazioni Matematiche

```
# ADDIZIONE (+)
somma = 10 + 5
stampa(somma) # Mostra: 15

# SOTTRAZIONE (-)
differenza = 20 - 8
stampa(differenza) # Mostra: 12

# MOLTIPLICAZIONE (*)
prodotto = 6 * 7
stampa(prodotto) # Mostra: 42

# DIVISIONE (/)
risultato = 15 / 3
stampa(risultato) # Mostra: 5.0
```

Input e Output

Output (Mostrare informazioni):

```
nome = "Marco"
stampa(f"Il mio nome è {nome}")
```

Input (Ricevere informazioni):

```
nome = input("Come ti chiami? ")
stampa(f"Ciao, {nome}!")

# Per i numeri, dobbiamo convertire:
eta = intero(input("Quanti anni hai? "))
stampa(f"Hai {eta} anni")
```

Prendere Decisioni - Condizionali

Il programma prende decisioni con `se`, `altrimenti_se` e `altrimenti`.

```
eta = 18

se eta >= 18:
    stampa("Sei maggiorenne")
altrimenti:
    stampa("Sei minorenn")
```

Esempio con SE-ALTRIMENTI_SE-ALTRIMENTI:

```
voto = 85

se voto >= 90:
    stampa("Eccellente!")
altrimenti_se voto >= 70:
    stampa("Buono!")
altrimenti:
    stampa("Devi migliorare")
```

Ripetere Azioni - Cicli

Ciclo PER (for):

```
# Contare da 0 a 4
per i in intervallo(5):
    stampa(i)
```

Ciclo MENTRE (while):

```
contatore = 0

mentre contatore < 5:
    stampa(f"Contatore: {contatore}")
    contatore = contatore + 1
```

Organizzare il Codice - Funzioni

Le funzioni sono blocchi di codice riutilizzabili.

```
definisci saluta(nome):
    stampa(f"Ciao, {nome}!")
```

```
saluta("Anna")
saluta("Pietro")
```

Funzioni con Ritorno:

```
definisci somma(a, b):
    ritorna a + b

risultato = somma(10, 20)
stampa(risultato) # 30
```

Funzioni Avanzate

Lambda:

Piccole funzioni anonime.

```
doppio = lambda x: x * 2
stampa(doppio(5)) # 10
```

Mappa (map):

```
numeri = [1, 2, 3, 4]
quadrati = lista(mappa(lambda x: x ** 2, numeri))
stampa(quadrati) # [1, 4, 9, 16]
```

Filtra (filter):

```
numeri = [1, 2, 3, 4, 5, 6]
pari = lista(filtra(lambda x: x % 2 == 0, numeri))
stampa(pari) # [2, 4, 6]
```

Creare Oggetti - Classi

Le classi sono “stampi” per creare oggetti.

```
classe Cane:
    definisci __init__(proprio, nome, razza):
        proprio.nome = nome
        proprio.razza = razza

    definisci abbaiare(proprio):
        stampa(f"{proprio.nome}: Bau bau!")

# Creare oggetti
rex = Cane("Rex", "Pastore Tedesco")
rex.abbaiare()
```

Progetti Pratici

Progetto 1: Lista di Cose da Fare

```
classe GestoreTask:
    definisci __init__(proprio):
        proprio.compiti = []

    definisci aggiungi(proprio, compito):
        proprio.compiti.append(compito)
        stampa(f"Compito aggiunto: {compito}")
```

```

definisci elenca(proprio):
    stampa("--- I Miei Compiti ---")
    per i, compito in enumera(proprio.compiti):
        stampa(f"{i + 1}. {compito}")

gestore = GestoreTask()
gestore.aggiungi("Imparare PyBR")
gestore.aggiungi("Fare esercizi")
gestore.elenca()

```

Progetto 2: Convertitore di Temperatura

```

definisci celsius_a_fahrenheit(c):
    ritorna (c * 9/5) + 32

temp_c = decimale(input("Temperatura in Celsius: "))
temp_f = celsius_a_fahrenheit(temp_c)
stampa(f"{temp_c}°C è uguale a {temp_f}°F")

```

Consigli Finali

1. **Pratica ogni giorno:** La costanza è fondamentale.
2. **Leggi gli errori:** Ti dicono cosa c'è che non va.
3. **Commenta il codice:** Usa # per spiegare.

Congratulazioni! 🎉

Hai completato la guida completa di PyBR in italiano. Ora sei un programmatore!