

# Blocos de Código (Indentação)

Prof. Alberto Costa Neto  
Programação em Python

# Indentação

- Aumentar a indentação depois de comandos `if/for/while` (a seguir)
- Mantenha a indentação para indicar o `escopo` do bloco (quais linhas são afetadas pelo `if/for/while`)
- Reduza a indentação *de volta ao* nível do comando `if/for/while` para indicar o fim do bloco de comandos
- Linhas em branco são ignoradas – não afetam em nada a indentação
- Comentários em uma linha por sim só são ignorados no que diz respeito a indentação

# Raciocinando sobre o início e fim dos blocos

```
x = 5
if x > 2 :
    print('Maior que 2')
    print('Ainda maior')
print('Fim do if x > 2')
```

```
for i in range(5) :
    print(i)
    if i > 2 :
        print('Maior que 2')
    print('Fim de i', i)
print('Fim')
```

Aumentar depois de if ou for para incluir os comandos dentro deles

Diminua para indicar o fim do bloco

```
→ x = 5
→ if x > 2 :
→     print('Maior que 2')
→     print('Ainda maior')
← print('Fim do if x > 2')

→ for i in range(5) :
→     print(i)
→     if i > 2 :
→         print('Maior que 2')
←     print('Fim de i', i)
← print('Fim')
```

# Python dá \*muita\* importância à indentação

- Um caractere a mais ou a menos muda o escopo
- Se você misturar **tabulação** e **espaços**, provavelmente terá “**indentation errors**”
- Mesmo que aparentemente o código esteja correto!

# O lado bom de Python!

- Não precisa de delimitadores como em outras linguagens de programação, como `{ }` em C ou `begin end` em Pascal
- O código fonte em Python é mais enxuto e indentado
- Indentar não é obrigatório em outras linguagens, mas é uma boa prática

**QUEM NÃO  
INDENTA O  
CÓDIGO  
PODE  
MORRER.**

Programa sempre pensando que o próximo programador pode ser um assassino psicopata.

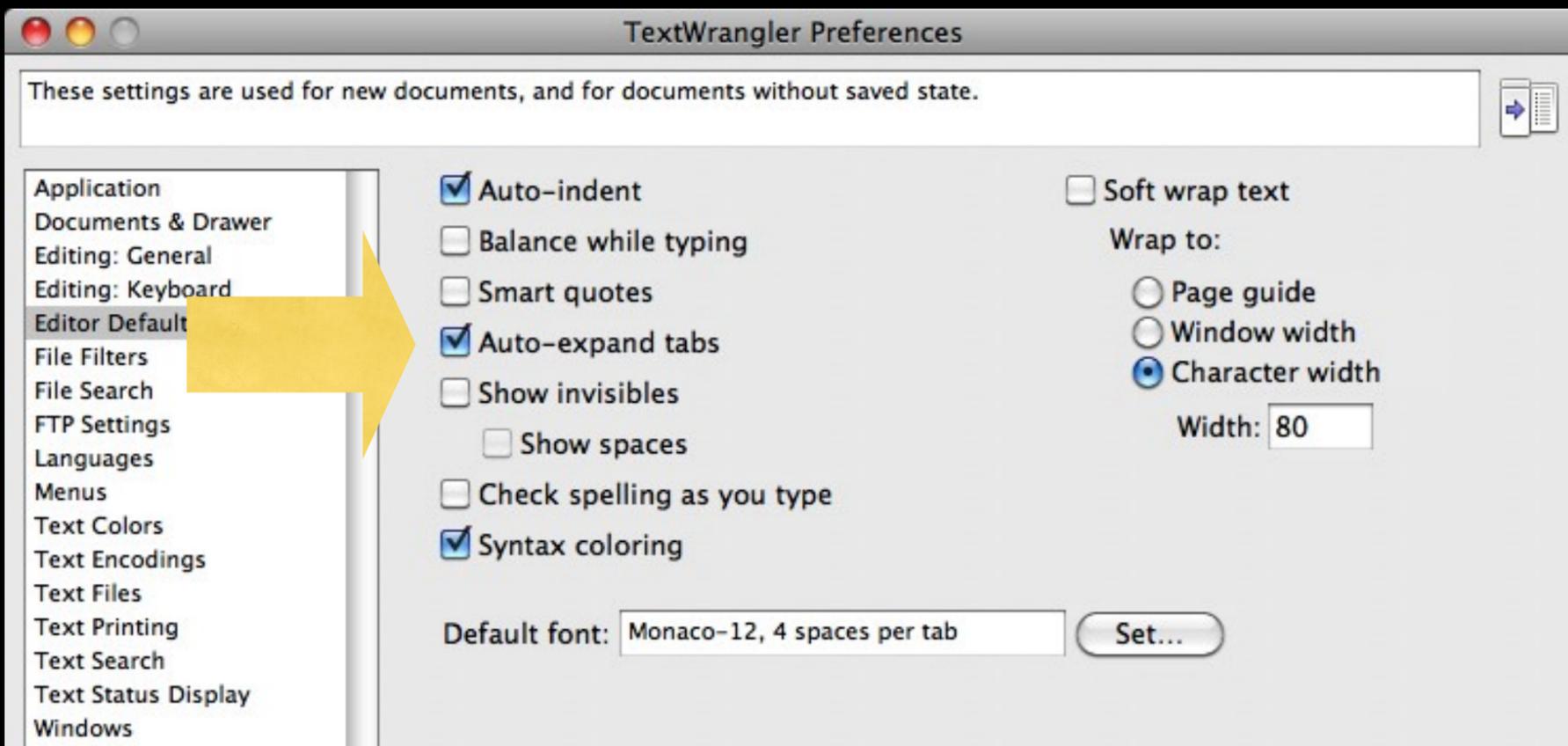
**NA DÚVIDA, USE PYTHON.**

VIDA DE  
PROGRAMADOR  
COMER



# Alerta: Não use Tabulação

- A maioria dos editores de texto pode transformar **tabulação** em **espaços** - tenha certeza que habilitou esta funcionalidade no editor
  - > **NotePad++**: Configurações -> Preferências -> Language -> Tab Settings
  - > **TextWrangler**: TextWrangler -> Preferences -> Editor Defaults
  - > **SublimeText**: View → Indentation → Indent using spaces
  - > **IDLE**: Converte automaticamente. Escolha quantos espaços correspondem a um Tab em: **Options → Configure IDLE → Indentation with**



Essa simples configuração irá lhe poupar horas de raiva!

