

ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES I

**Outros comandos: break,
continue e pass**

BREAK

O comando `break`, quando inserido dentro de um loop (`for` ou `while`), faz com que aquele laço seja finalizado, passando o controle para o próximo bloco de instruções não indentado.

```
for var in sequence:
    # codes inside for loop
    if condition:
        break
    # codes inside for loop
# codes outside for loop
```

```
while test expression:
    # codes inside while loop
    if condition:
        break
    # codes inside while loop
# codes outside while loop
```

BREAK

Exemplo:

Implemente uma função que verifica se um número é primo ou não.

```
def primo(num):  
    i = 2  
    while i < num:  
        if num % i == 0:  
            break  
        i += 1  
    return i == num
```

CONTINUE

O comando `continue`, quando inserido dentro de um loop (`for` ou `while`), faz com que a nova iteração do laço seja forçada, a partir daquele ponto onde o comando foi inserido.

```
for var in sequence:
    # codes inside for loop
    if condition:
        continue
    # codes inside for loop

# codes outside for loop
```

```
while test expression:
    # codes inside while loop
    if condition:
        continue
    # codes inside while loop

# codes outside while loop
```

CONTINUE

Exemplo:

Dada a função primo() implementada no slide anterior, faça um programa que imprima todos os números primos de 2 a 100.

```
for n in range(2, 100):  
    if not primo(n):  
        continue  
    print(n)
```

PASS

Todos os comandos def, if, else, for ou while precisam ter necessariamente um bloco indentado não vazio.

Apesar de poder ser evitado na maioria das situações, a linguagem Python oferece o comando pass, que é uma instrução válida que não faz nada.

PASS

Exemplo:

```
if n % 2 == 0:
```

```
    pass
```

```
else:
```

```
    print(n)
```

ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES I

**Outros comandos: break,
continue e pass**