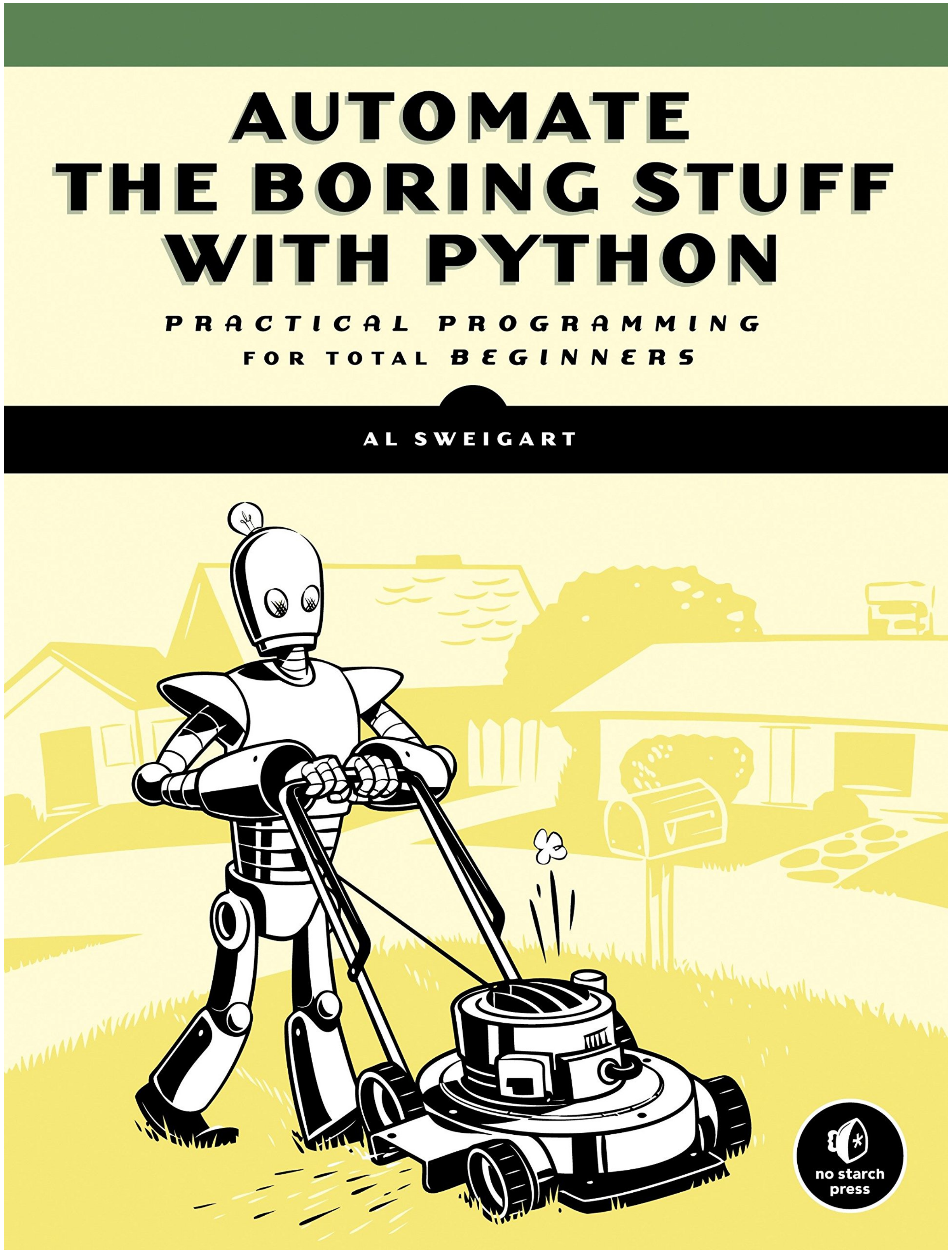
Flow Control



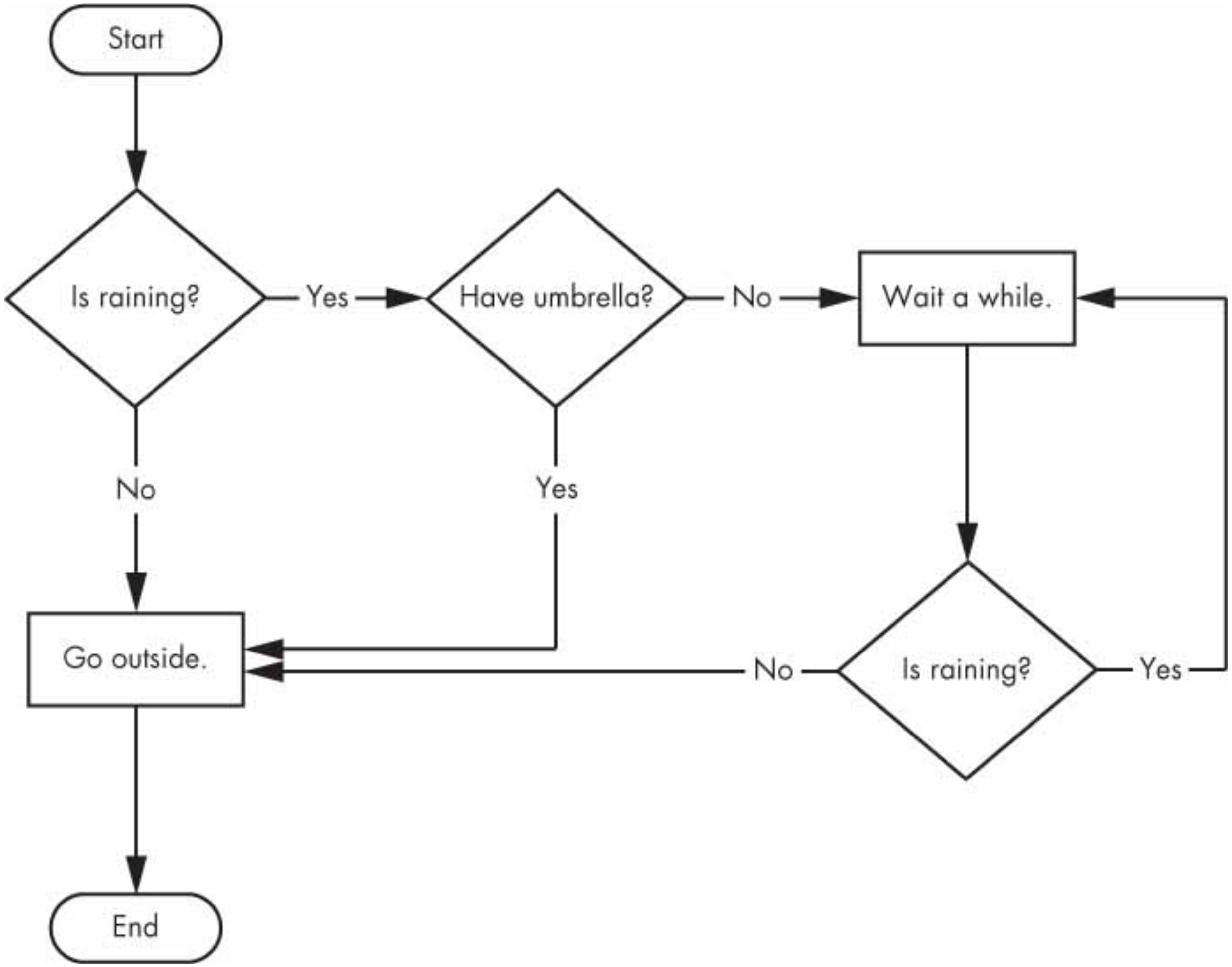
AUTOMATE THE BORING STUFF WITH PYTHON

PRACTICAL PROGRAMMING FOR TOTAL BEGINNERS

AL SWEIGART

no starch press

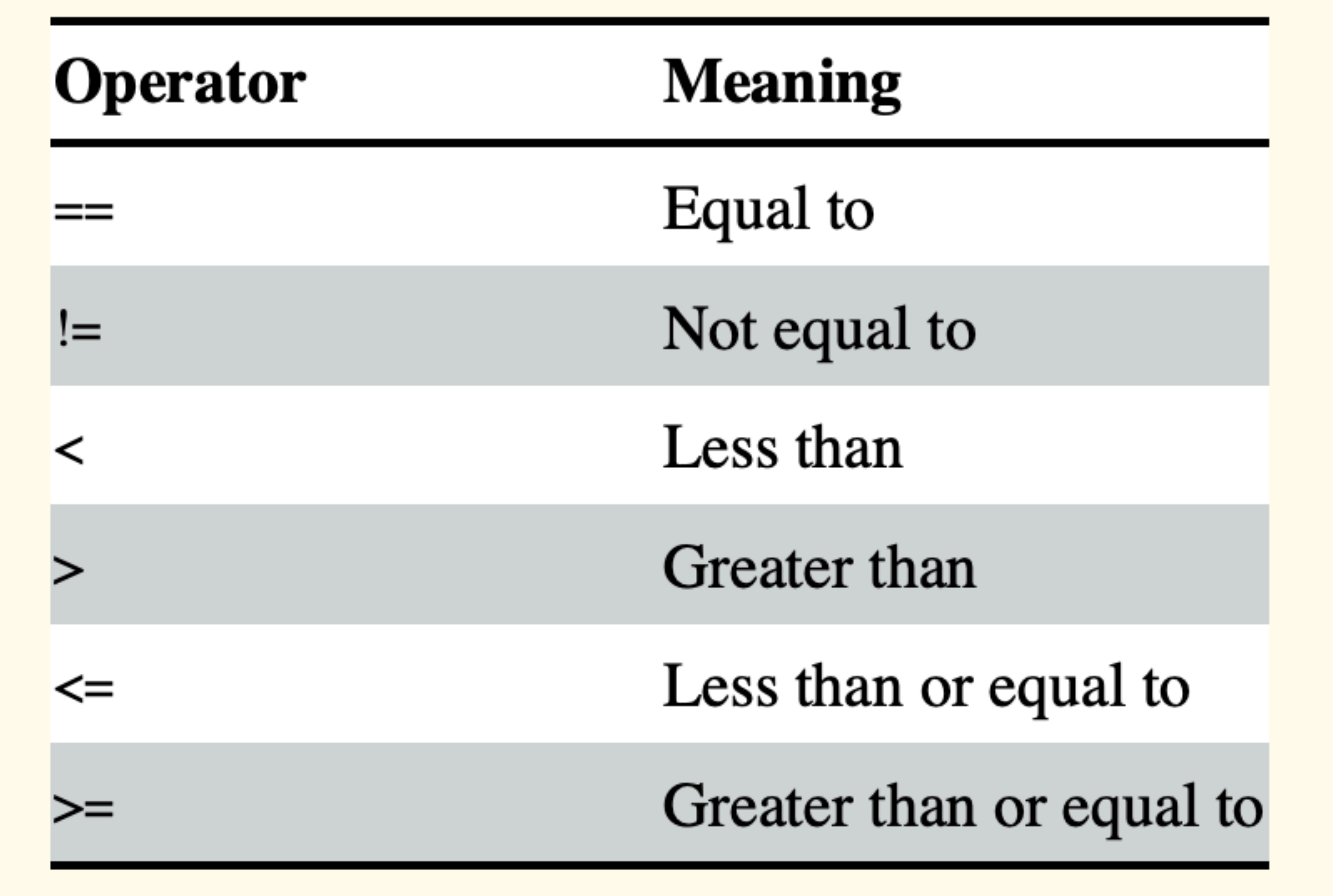
https://automatetheboringstuff.com


Start
Is raining?
Yes
Have umbrella?
No
Wait a while.
No
Yes
Go outside.
No
Is raining?
Yes
End

- Boolean values
- Comparison operators
- Boolean operators

- YES: True, False
- NO: TRUE,false,’true’

Comparison operators

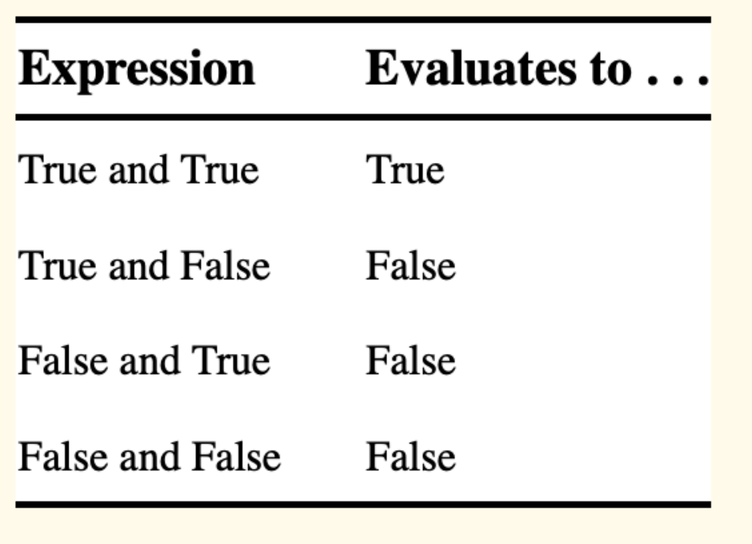


Operator	Meaning
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Boolean Operators

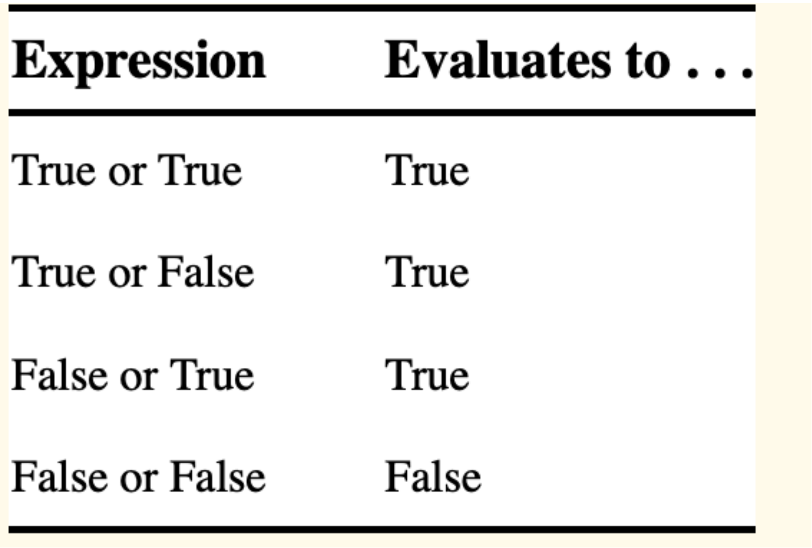
and, or, not

The and Operator’s Truth Table



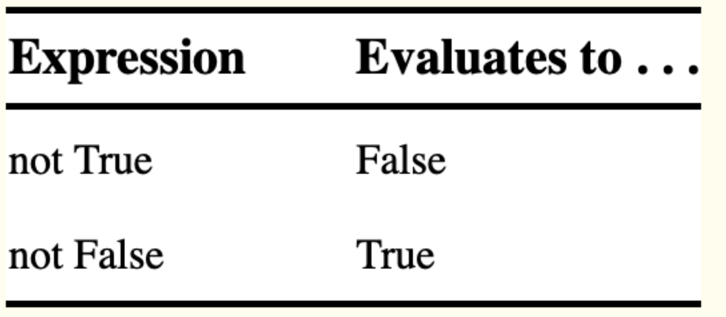
Expression	Evaluates to . . .
True and True	True
True and False	False
False and True	False
False and False	False

The or Operator’s Truth Table



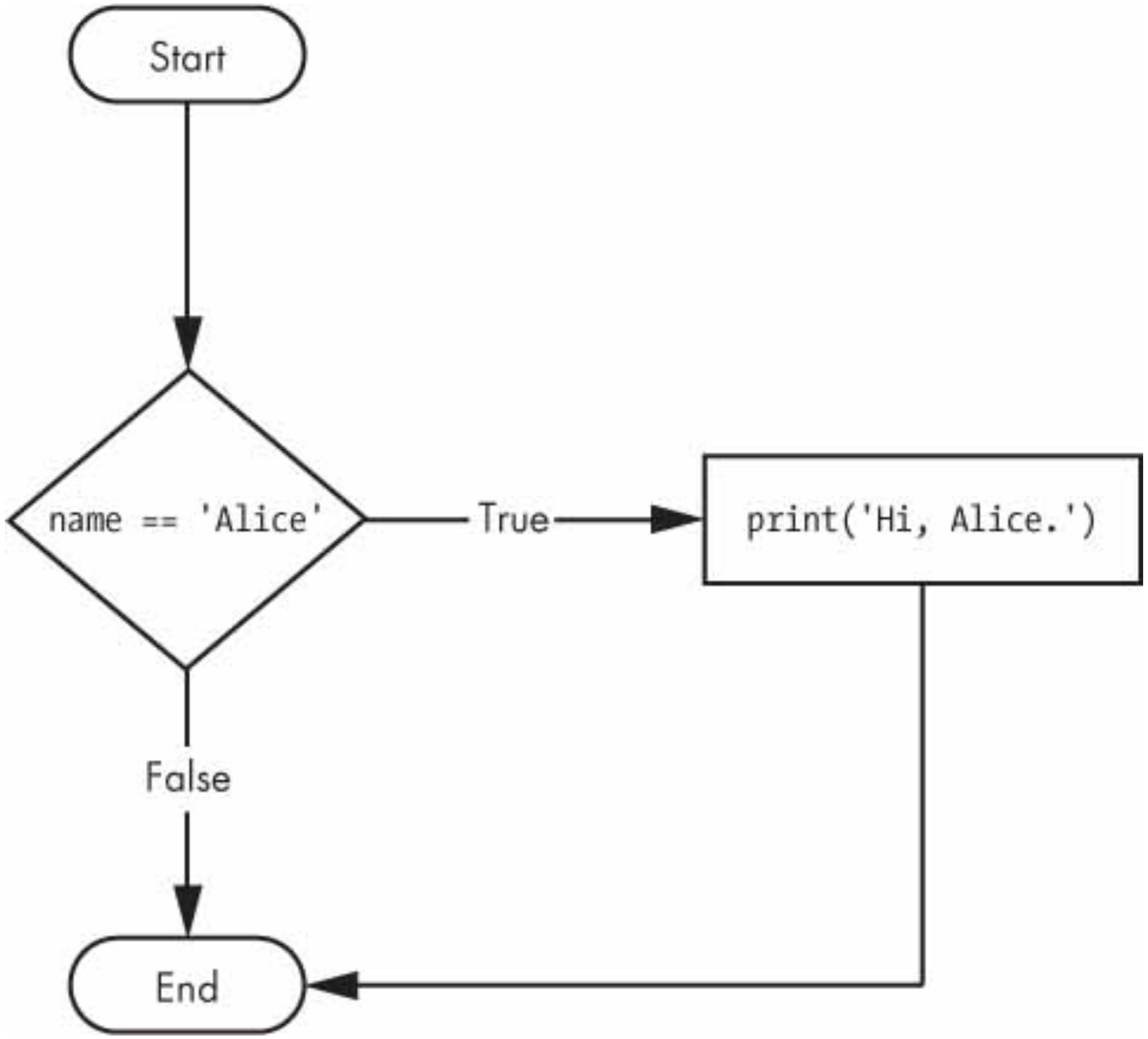
Expression	Evaluates to . . .
True or True	True
True or False	True
False or True	True
False or False	False

The not Operator’s Truth Table



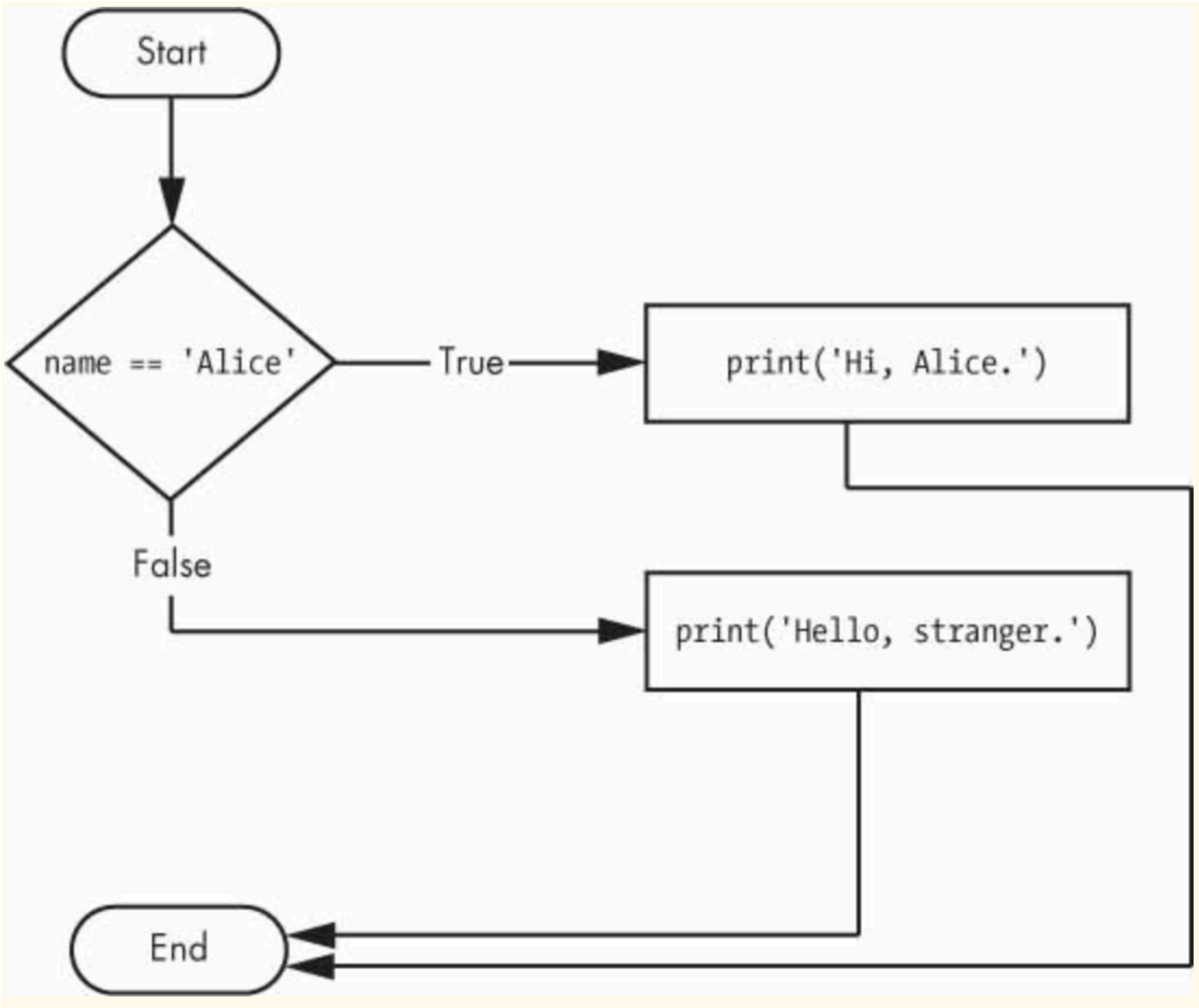
Expression	Evaluates to . . .
not True	False
not False	True

if Statements



http://pythontutor.com

Else and Elif Statements



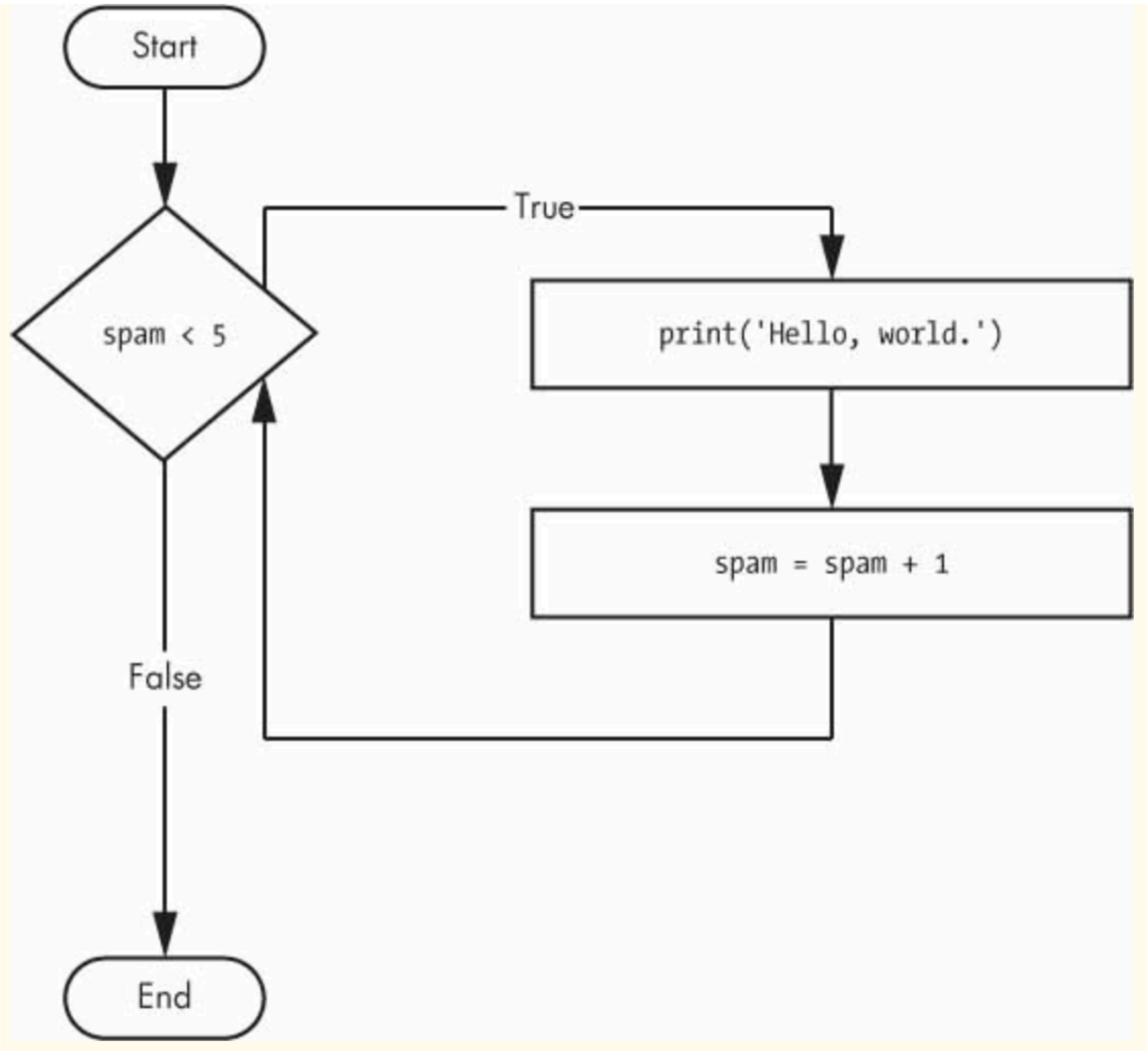
Recap

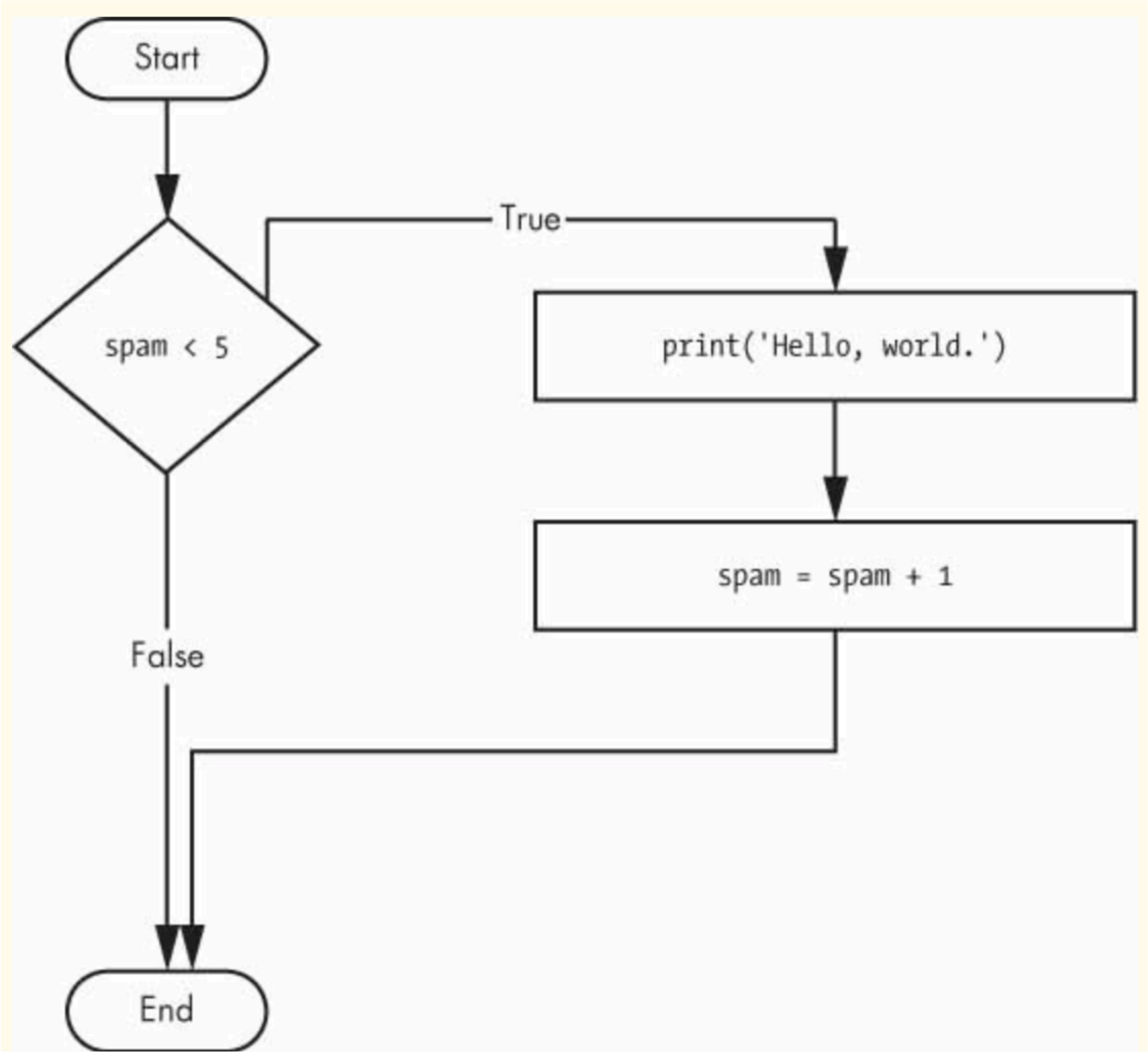
- An **if** statement can be used to conditionally execute code, depending on whether or not the if statement’s condition is **True** or **False**.
- An **elif** statement can follow an if statement. Its block executes if its condition is **True** and all of the previous conditions have been **False**.
- An **else** statement comes at the end. It’s block is executed if all of the previous conditions have been **False**.

While Statements

```
while spam <5 :
    print(“Hello, World”)
    spam=spam+1
```

```
if spam <5 :
    print(“Hello, World”)
    spam=spam+1
```

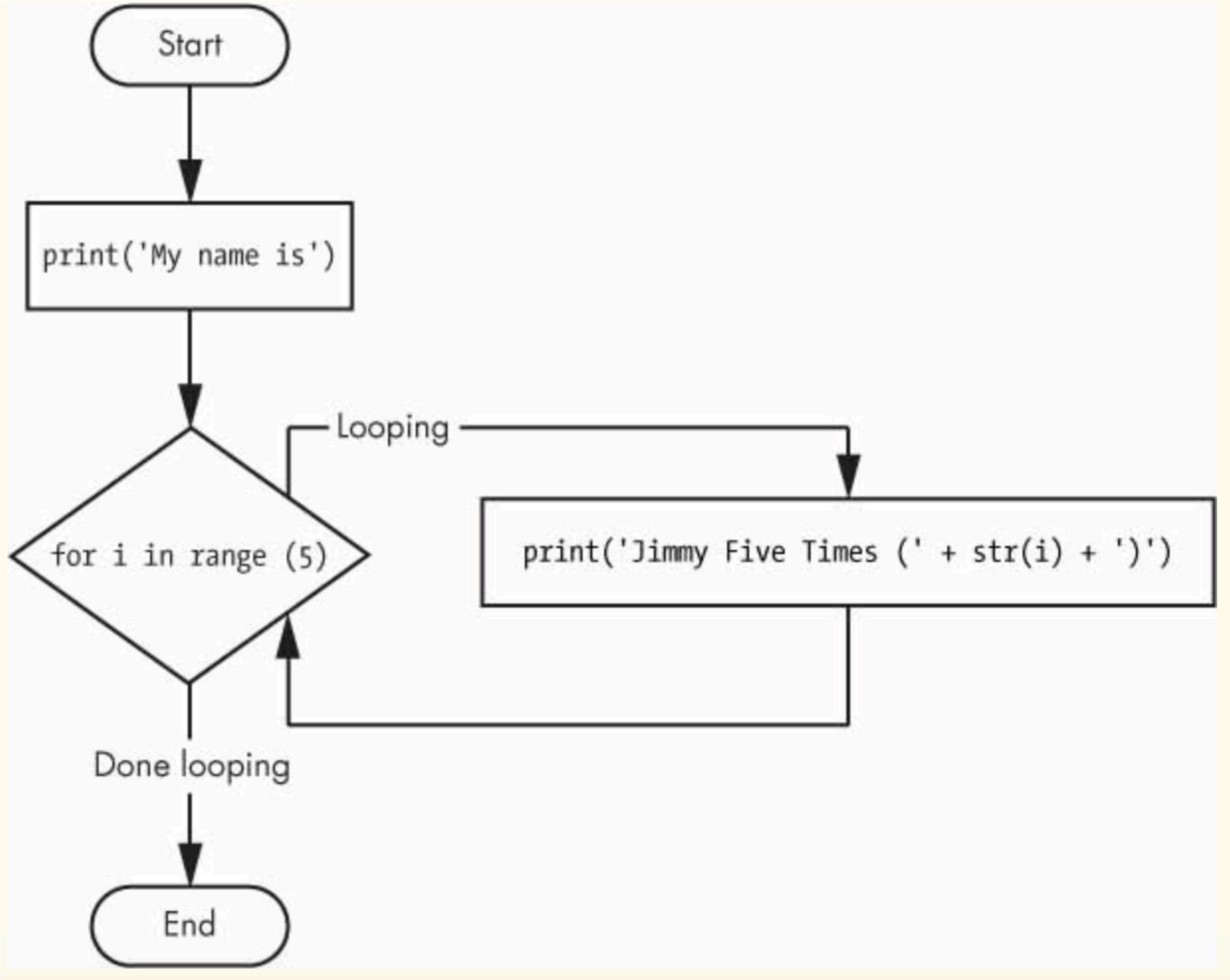




Just to recap

- When the Execution reaches the end of **While** statement’s block, it jumps back to the start to recheck
- A **break** statement causes the execution to immediately leave the loop, without re-checking the condition
- A **continue** statement causes the execution to immediately to jump back to the start of the loop and re-check the condition

For loop



Start

print('My name is')

for i in range (5)

Looping

print('Jimmy Five Times (' + str(i) + ')')

Done looping

End

Write a program to sum

sum=1+2+…..+100

Carl Friedrich Gauß (1777–1855)

- **for** loops will loop a specific number of times.
- The **range()** function called with one, two or three arguments.
- **Break** and **continue** statement can be used in for loops.

List

- A list is a value that contains multiple values
- Values in a list are called item
- Comma-delimited
- you can access items in a list with its indexed integers
- You can get multiple items from a list by slice