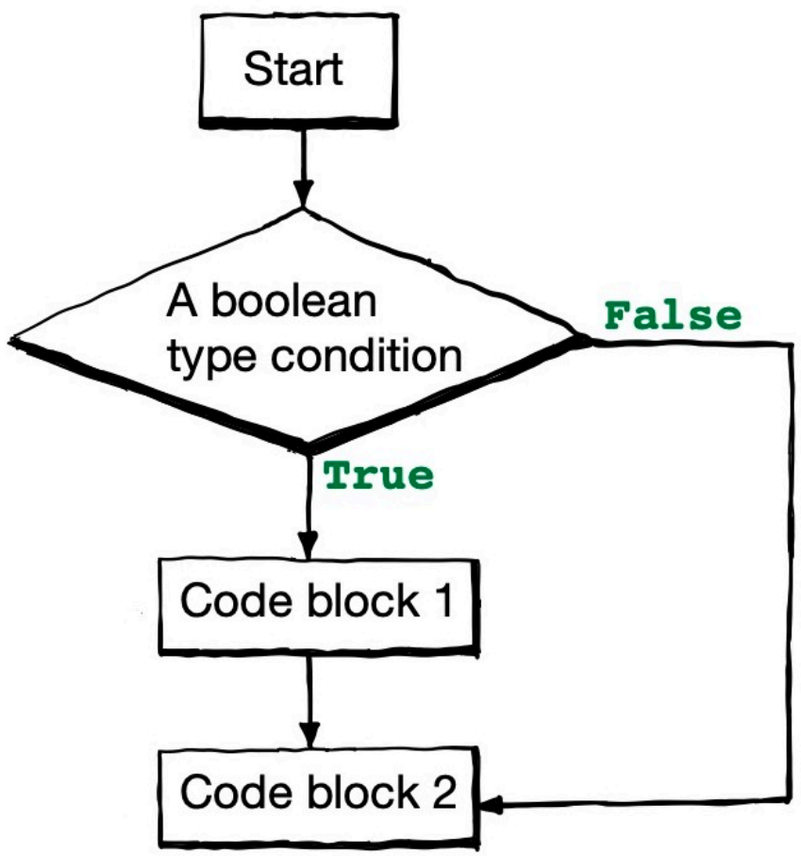
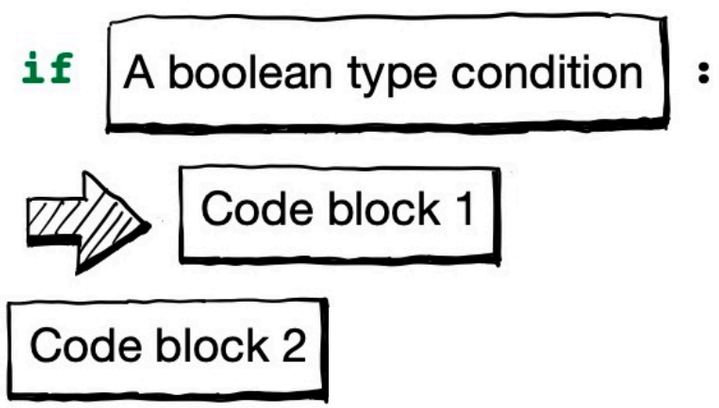
Control Flow of Python Programs

Contents

Conditional Statements

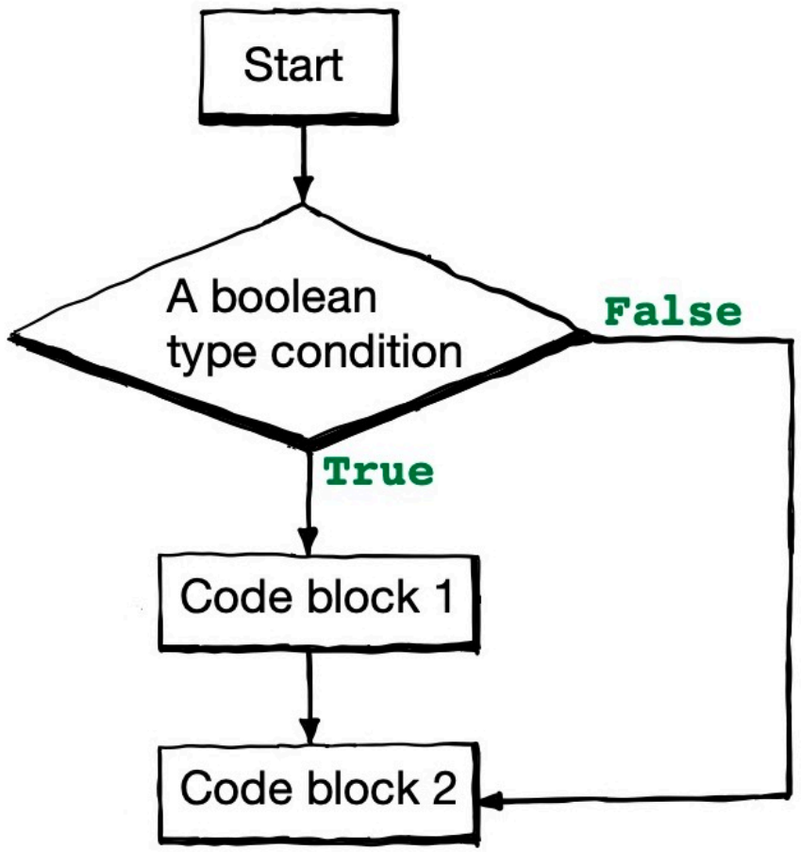
- Syntax rules of `if`-statements
  - The simplest case

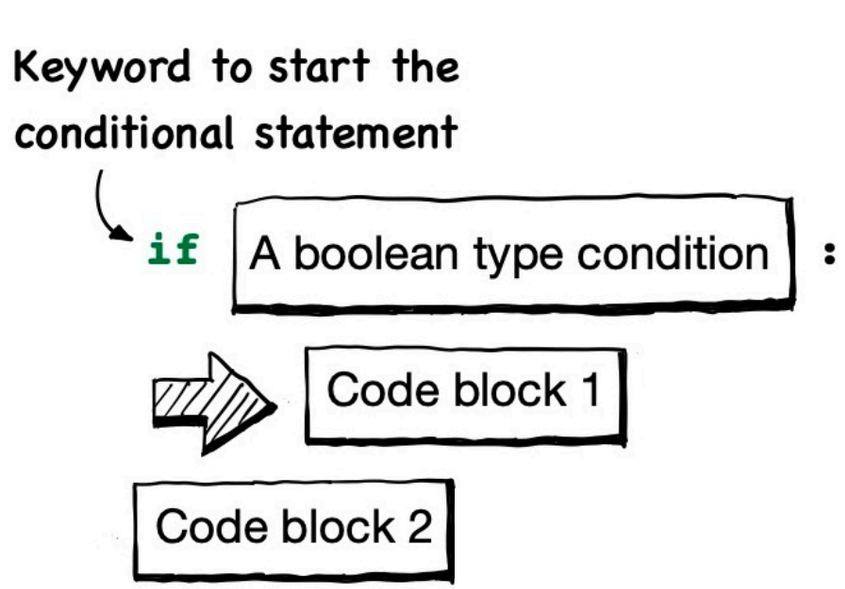




Conditional Statements

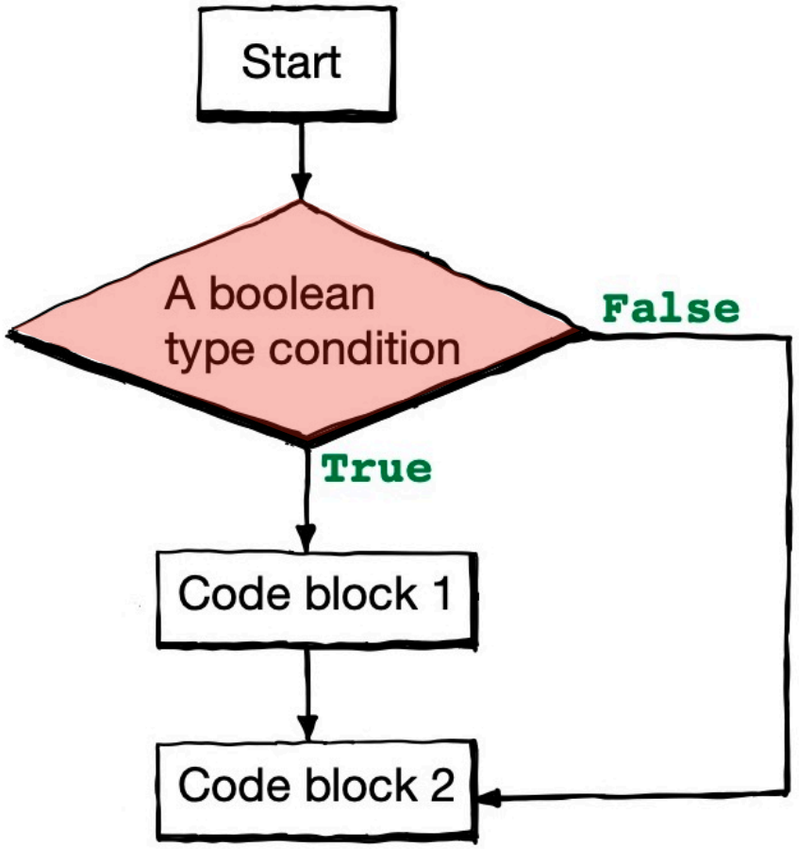
- Syntax rules of `if`-statements
  - The simplest case
    - ✓ The statement is started by the keyword `if`

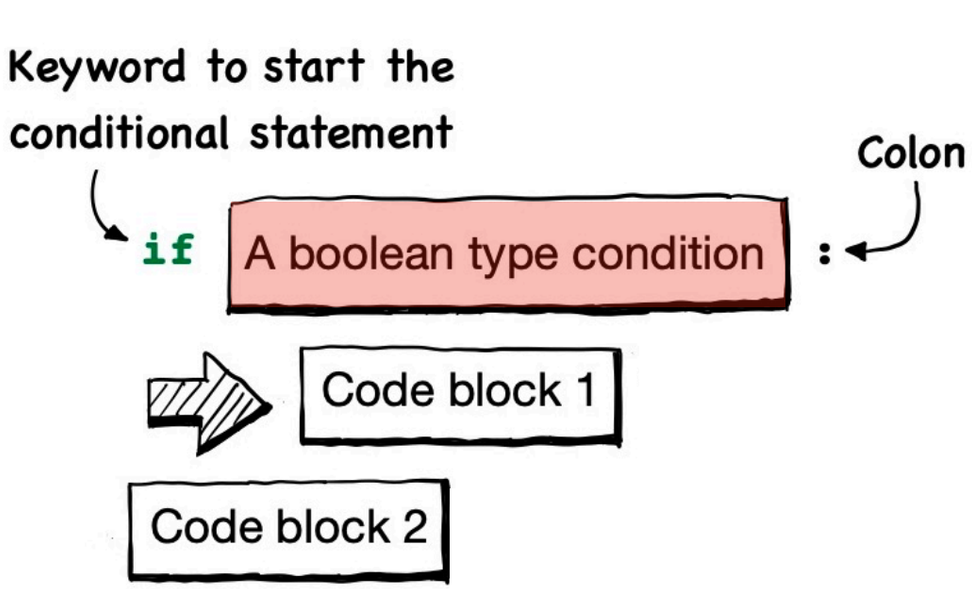




Conditional Statements

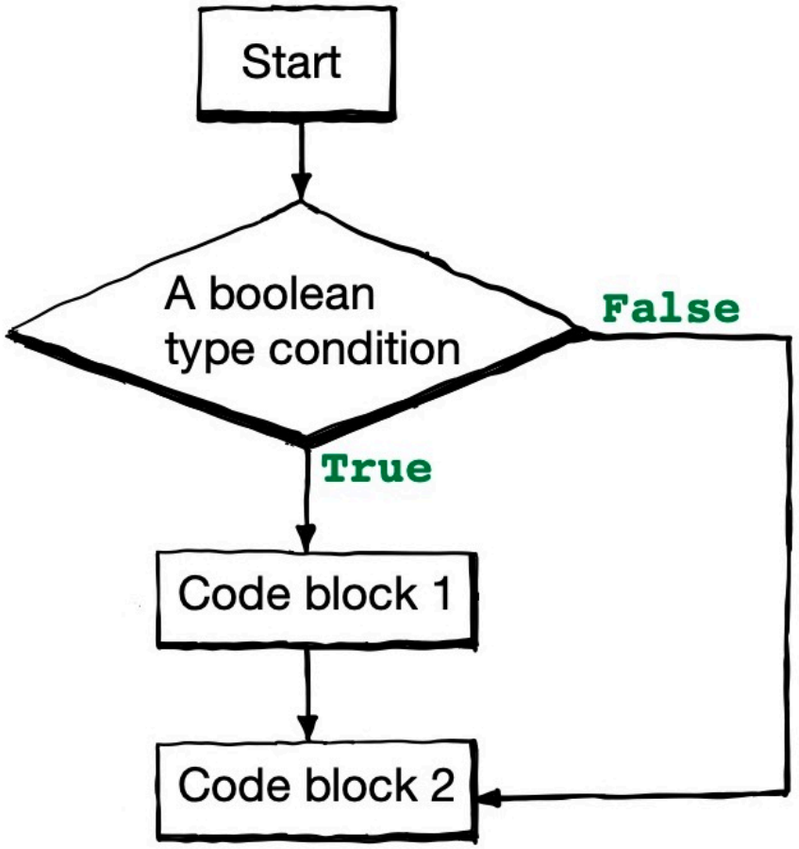
- Syntax rules of `if`-statements
  - The simplest case
    - ✓ The `if` keyword is followed by a boolean type condition and a colon

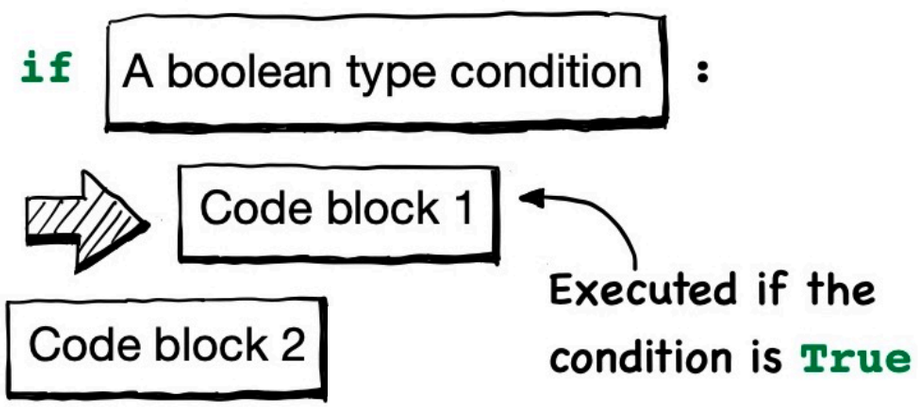




Conditional Statements

- Syntax rules of `if`-statements
  - The simplest case
    - ✓ Execute **Code block 1** if the boolean condition is `True`, and it is skipped if the condition is `False`

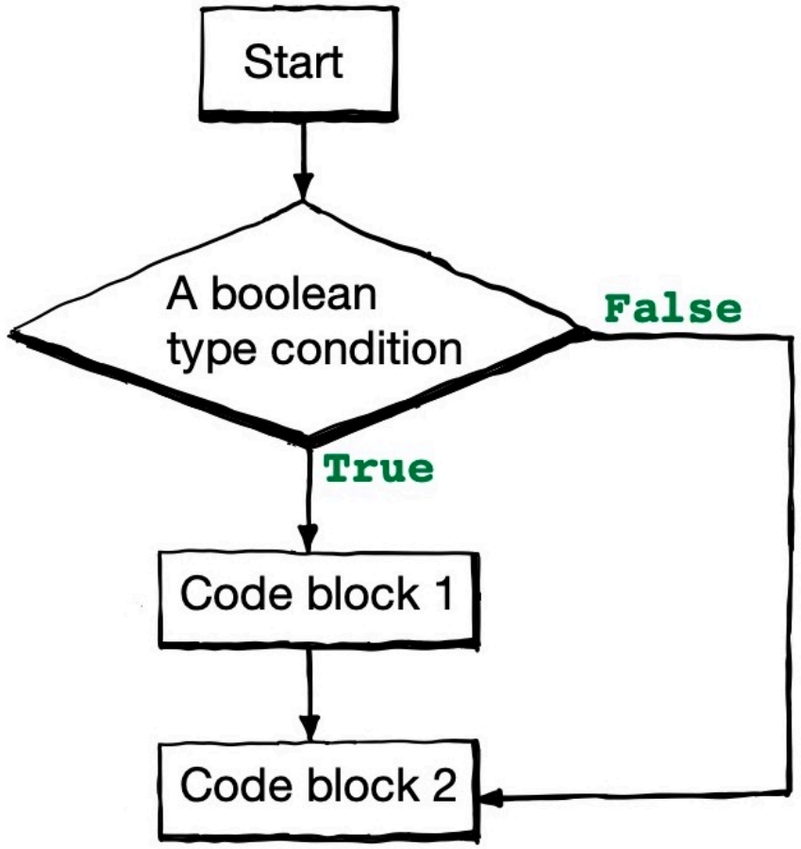


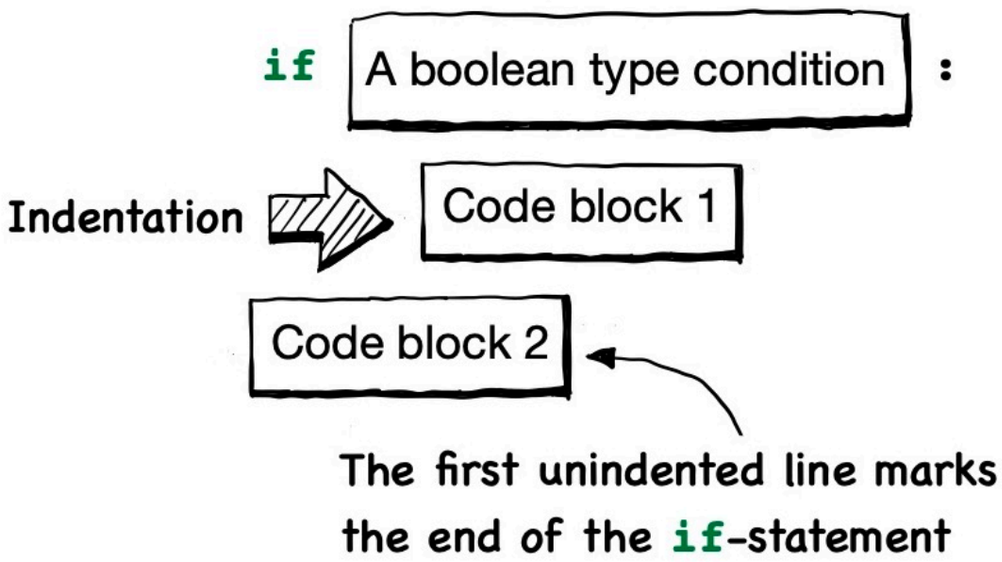


Conditional Statements

- Syntax rules of `if`-statements
  - The simplest case

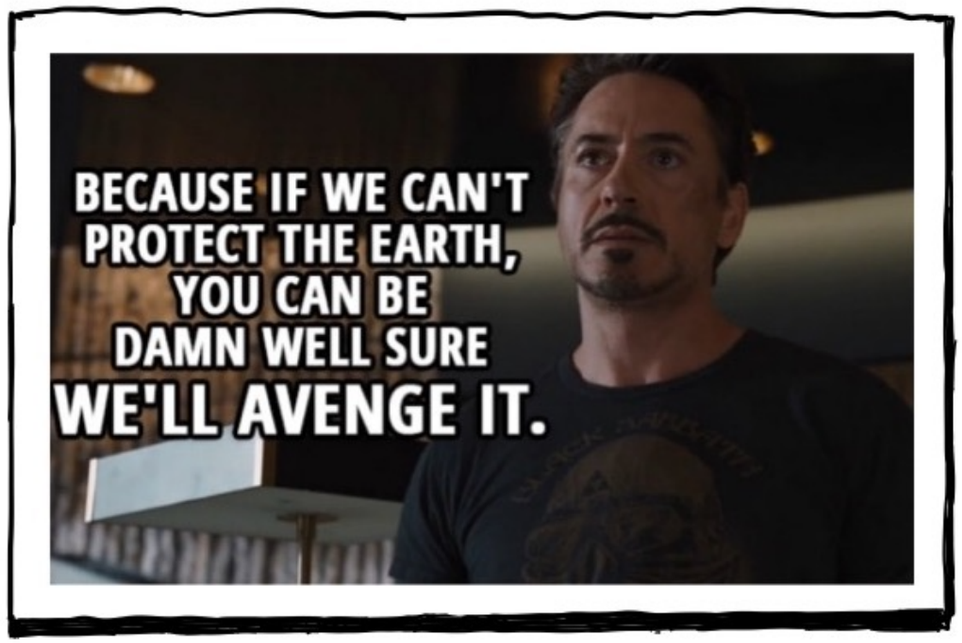
Notes: In Python, indentation (a tab or four spaces) is used to differentiate **Code block 1** from other parts of code. The first unindented line (**Code block 2**) marks the end of the conditional statement.

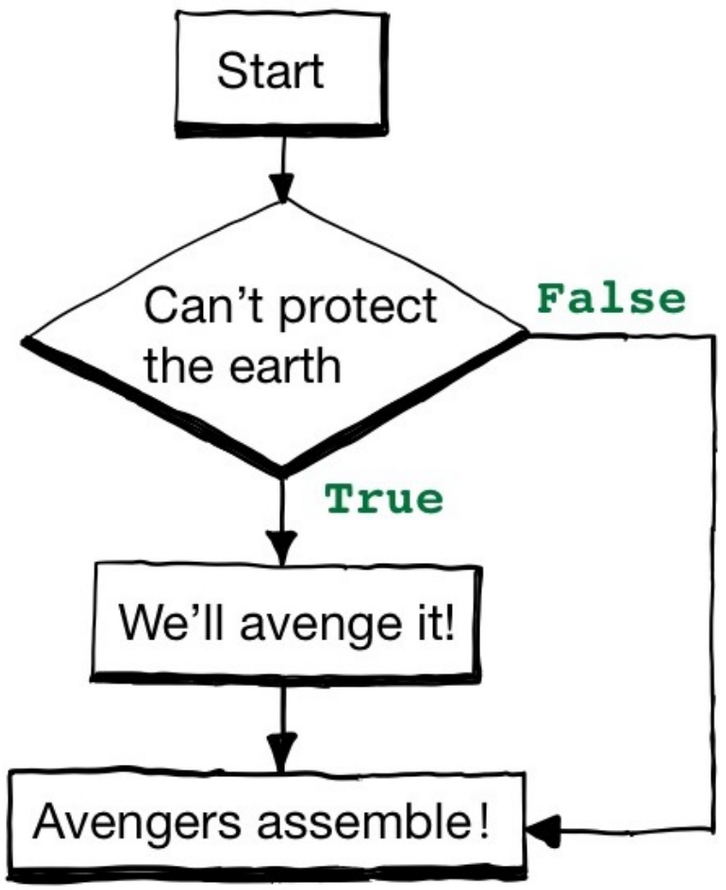




Conditional Statements

- Syntax rules of `if`-statements
  - The simplest case



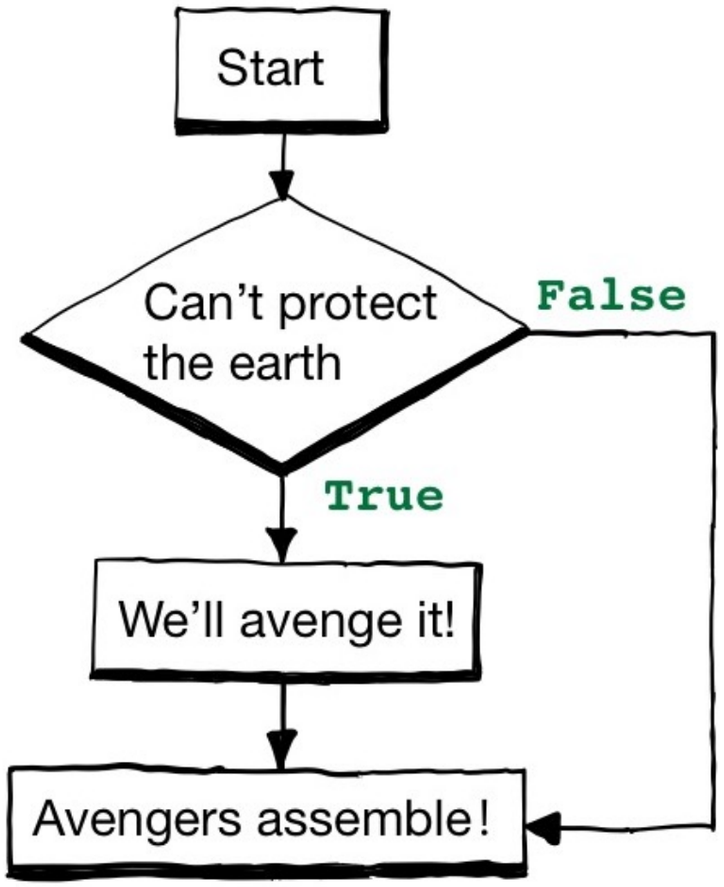


Conditional Statements

- Syntax rules of `if`-statements
  - The simplest case

```
cant_protect_earth = True

if cant_protect_earth:
    print("We'll avenge it!”)

print('Avengers assemble!')
```

```
We'll avenge it!
Avengers assemble!
```

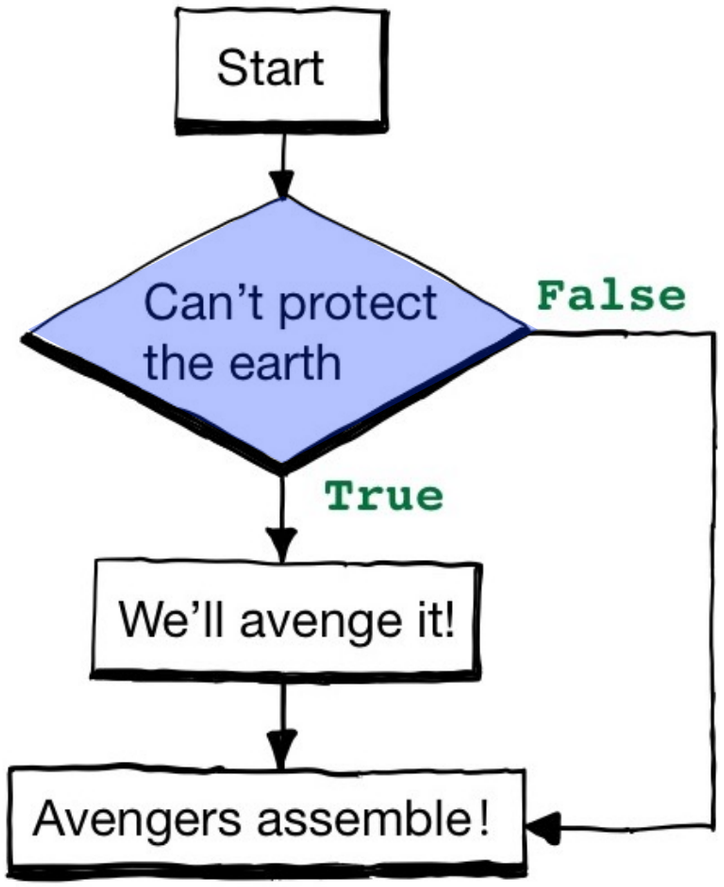


Conditional Statements

- Syntax rules of `if`-statements
  - The simplest case

```
cant_protect_earth = True

if cant_protect_earth:
    print("We'll avenge it!”)

print('Avengers assemble!')
```

```
We'll avenge it!
Avengers assemble!
```

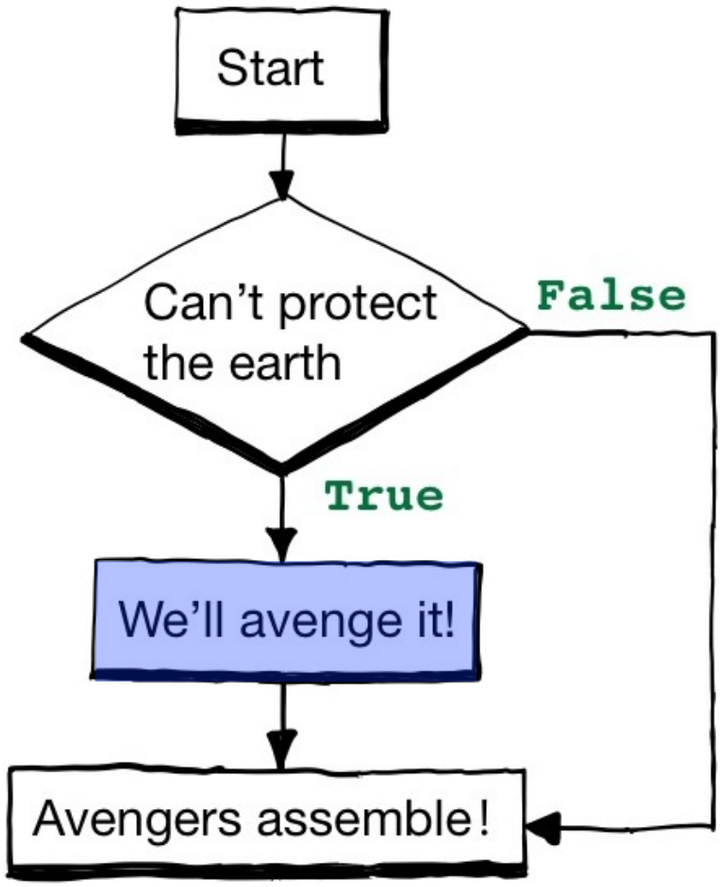


Conditional Statements

- Syntax rules of `if`-statements
  - The simplest case

```
cant_protect_earth = True

if cant_protect_earth:
    print("We'll avenge it!")

print('Avengers assemble!')
```

```
We'll avenge it!
Avengers assemble!
```

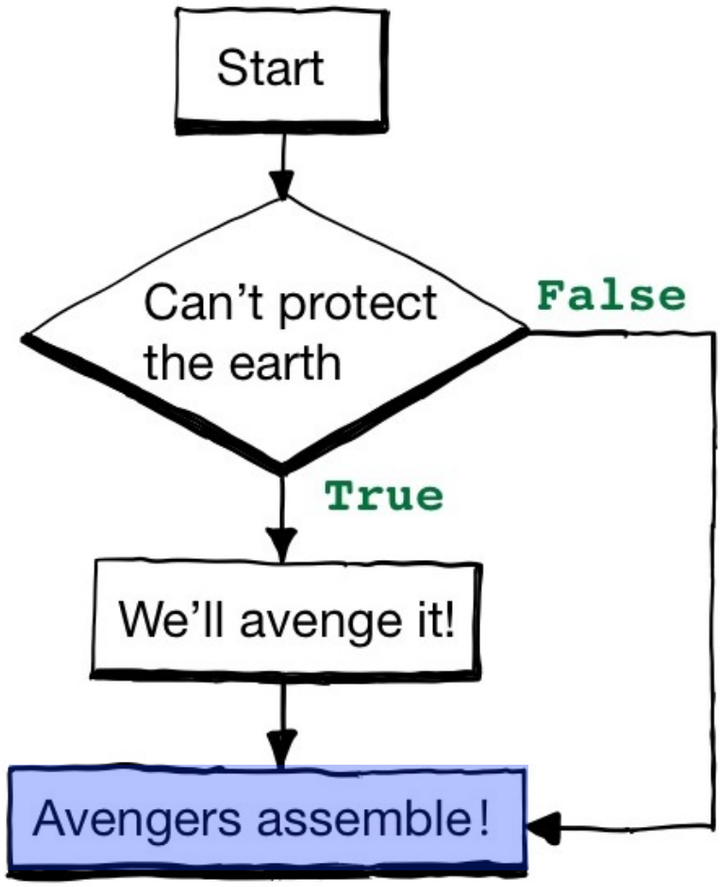


Conditional Statements

- Syntax rules of `if`-statements
  - The simplest case

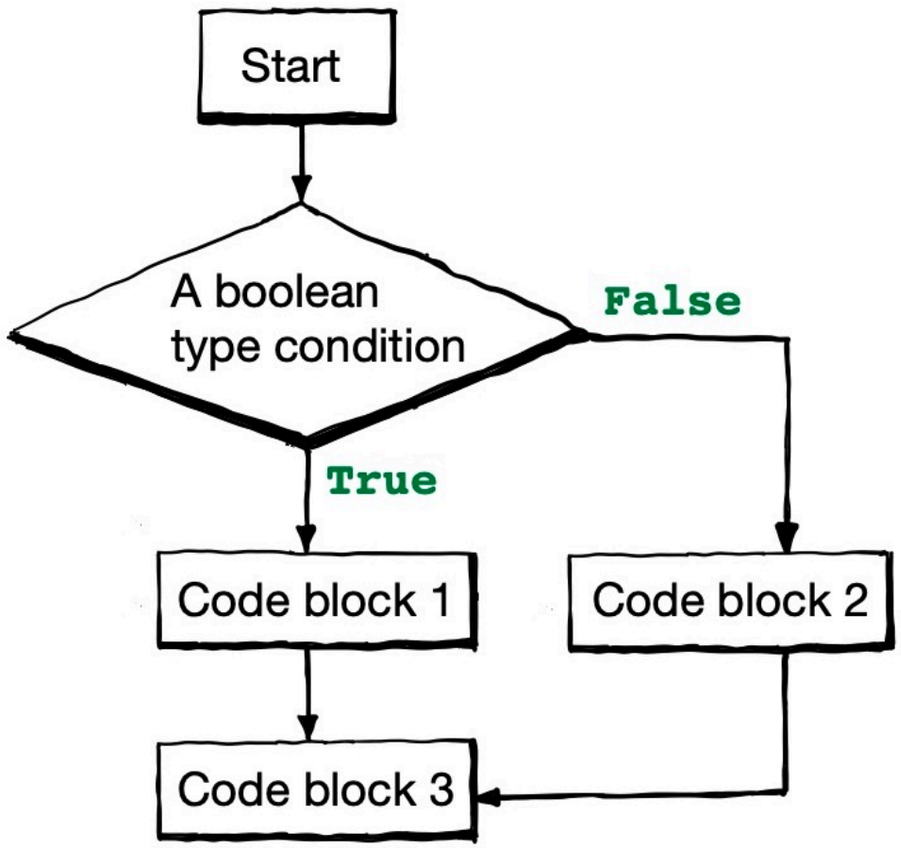
```
cant_protect_earth = True

if cant_protect_earth:
    print("We'll avenge it!")

print('Avengers assemble!')
```

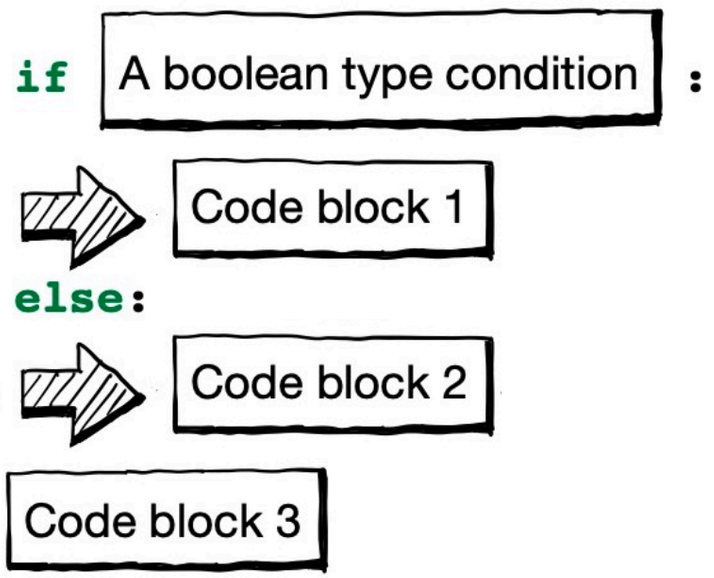
```
We'll avenge it!
Avengers assemble!
```



Conditional Statements

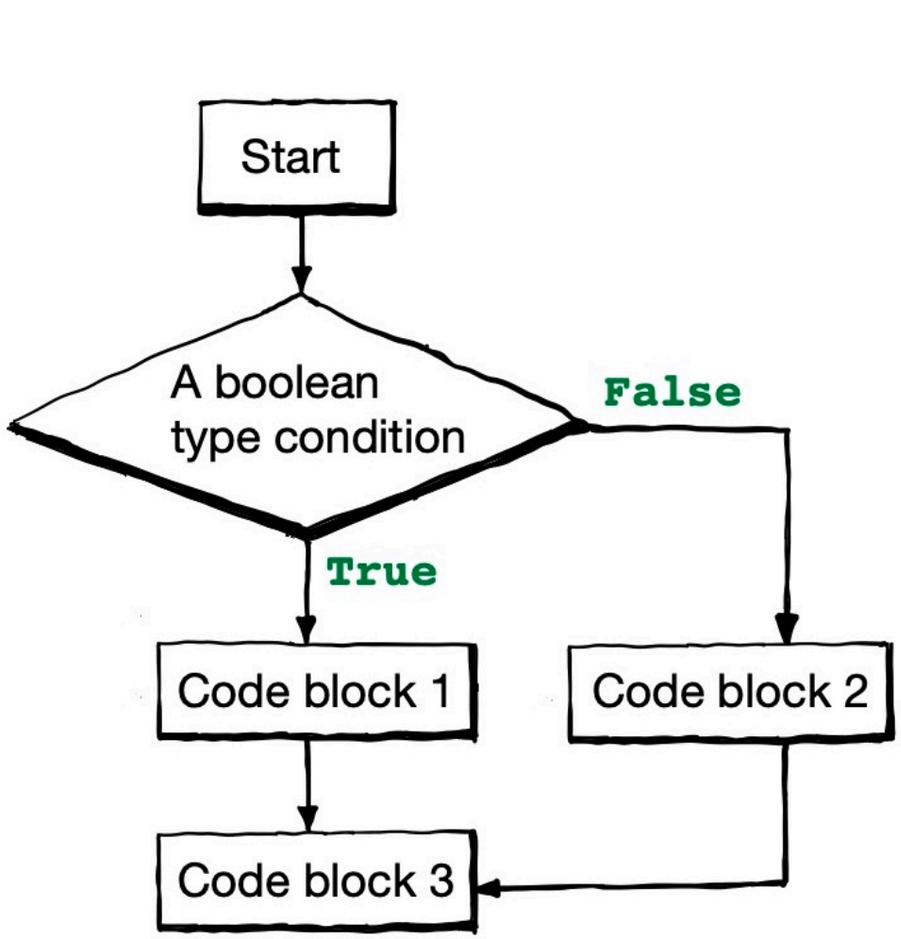
- Syntax rules of `if`-statements
  - The case with an `else` situation

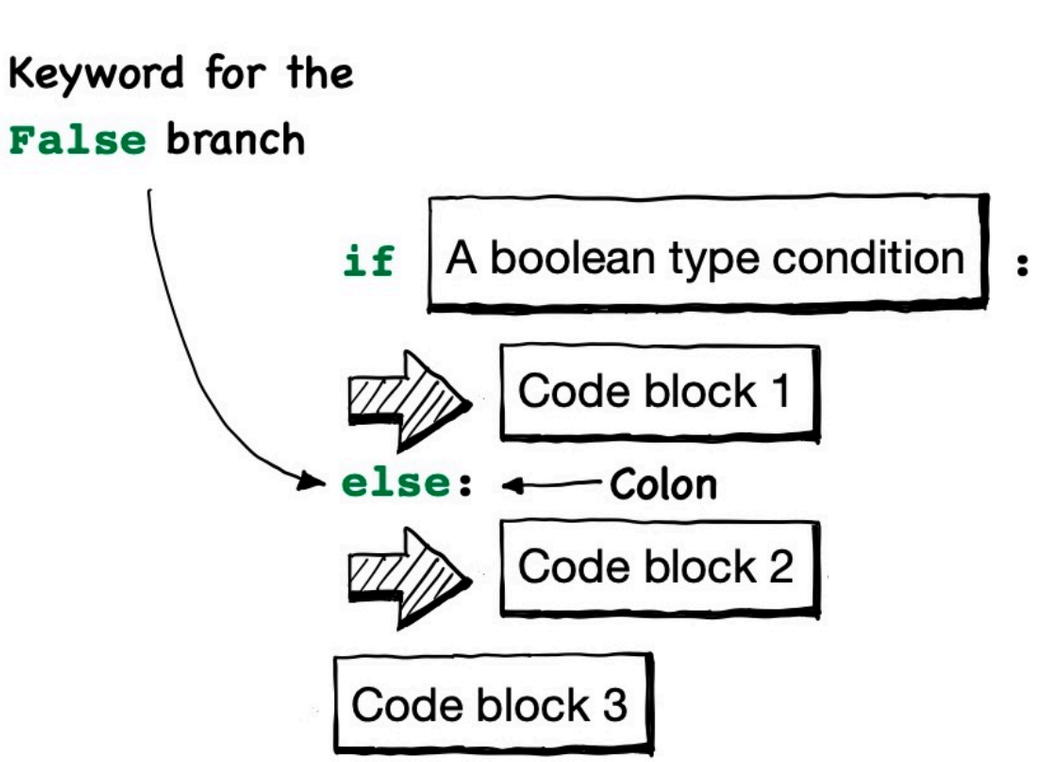




Conditional Statements

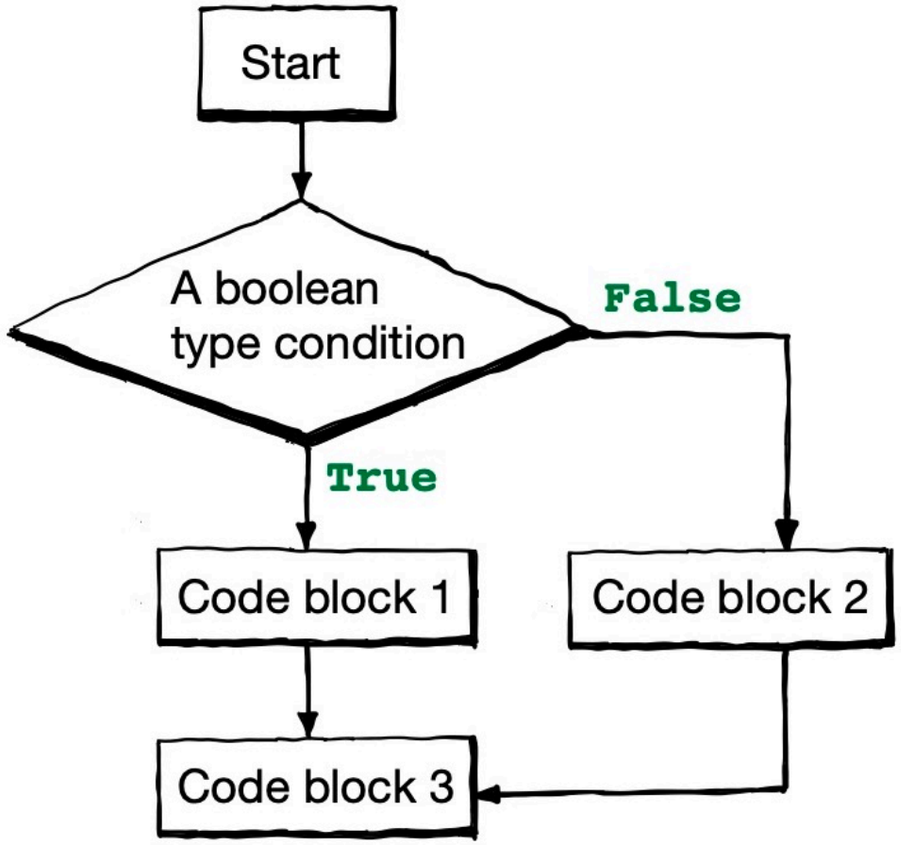
- Syntax rules of `if`-statements
  - The case with an `else` situation
    - ✓ The `False` branch is started by the keyword `else` and a colon

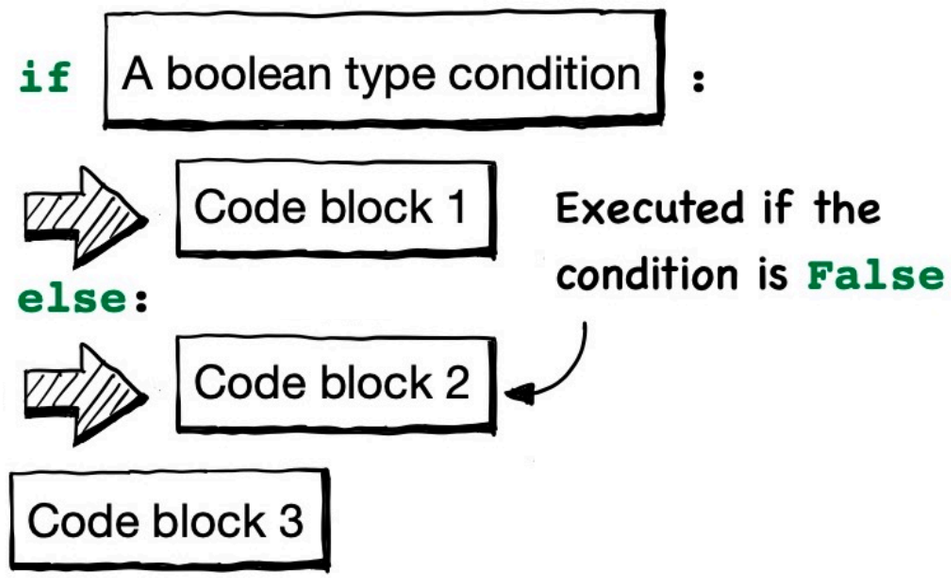




Conditional Statements

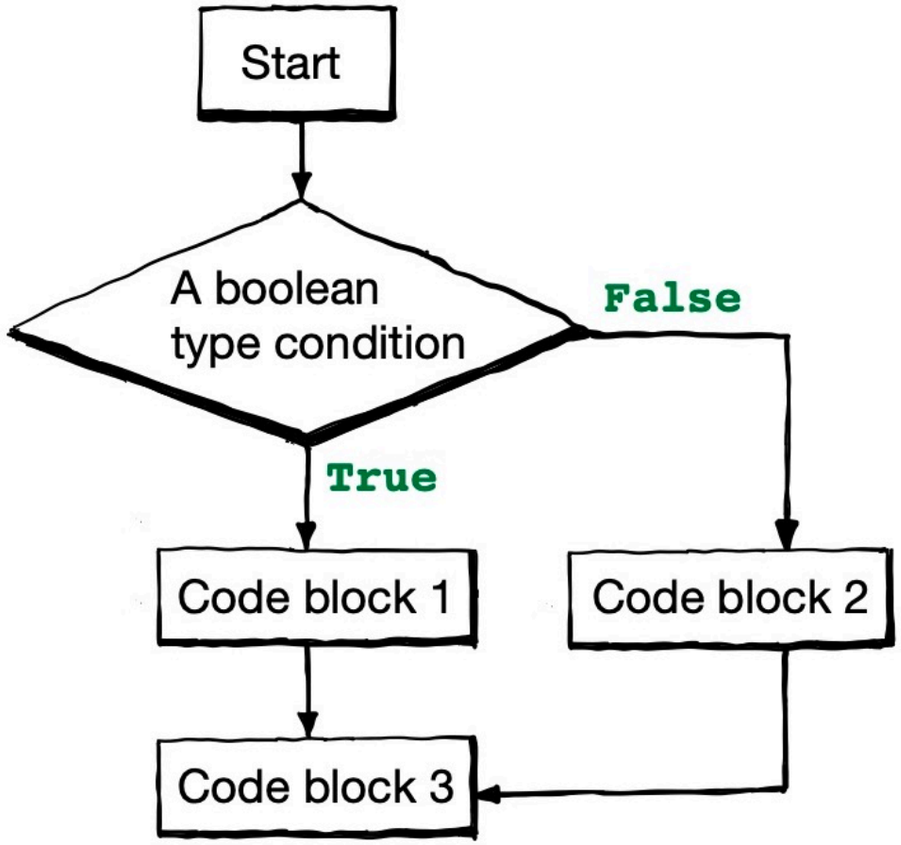
- Syntax rules of `if`-statements
  - The case with an `else` situation
    - ✓ Execute **Code block 2** if the boolean type condition is `False`

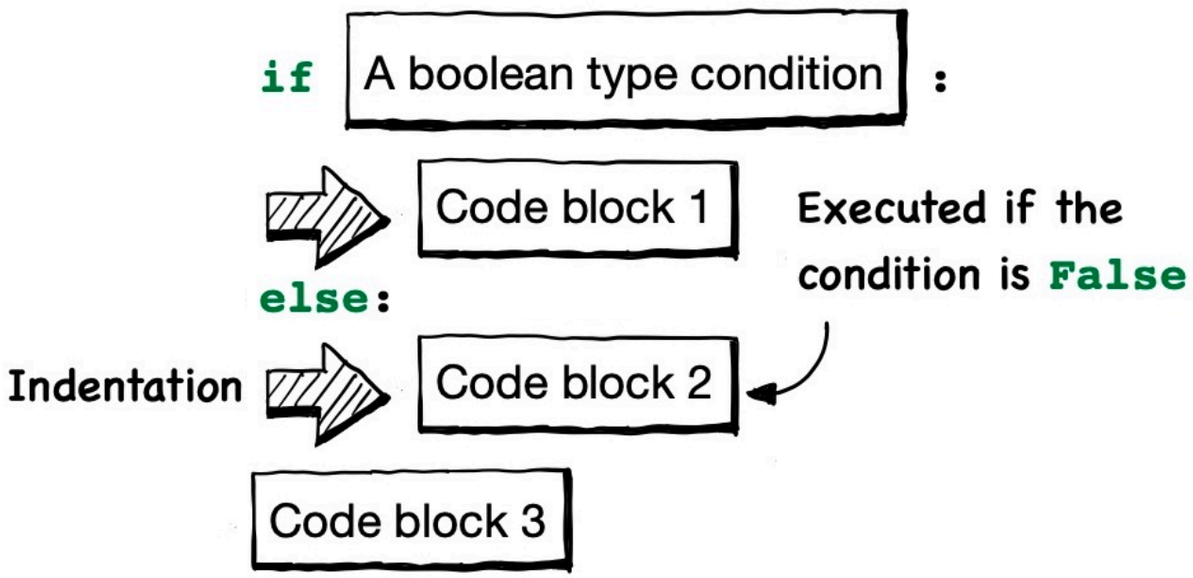




Conditional Statements

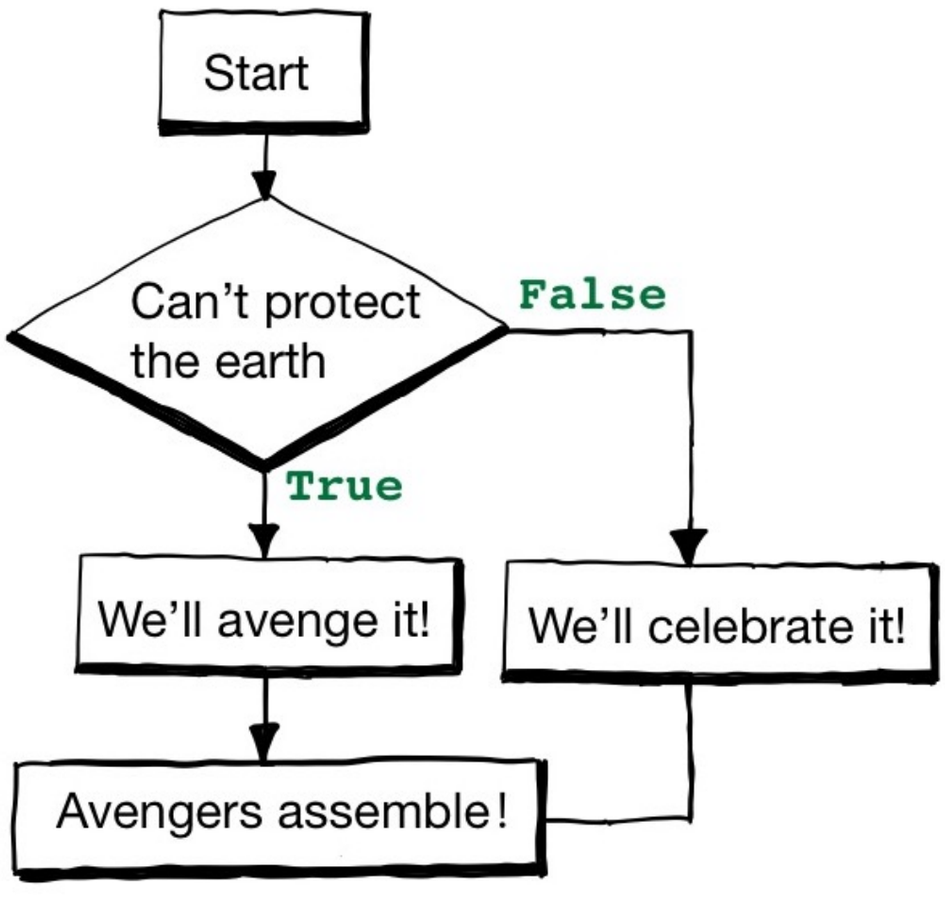
- Syntax rules of `if`-statements
  - The case with an `else` situation
    - ✓ **Code block 2** has the same level of indentation as **Code block 1**





Conditional Statements

- Syntax rules of `if`-statements
  - The case with an `else` situation

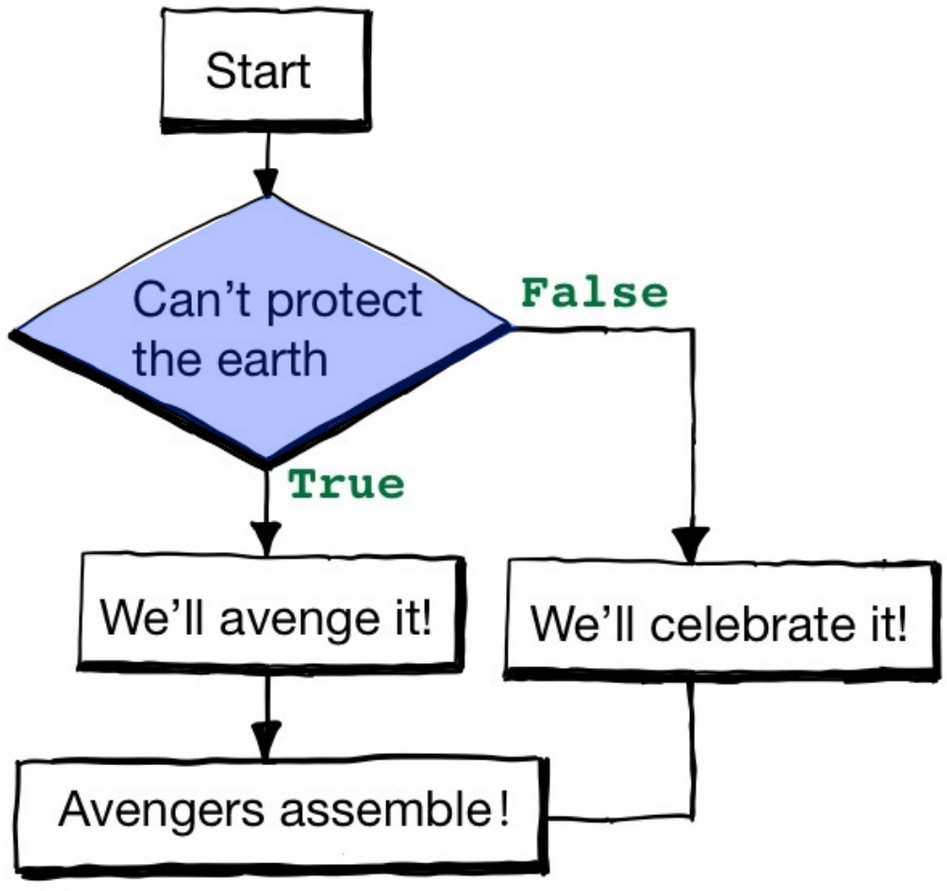


Conditional Statements

- Syntax rules of `if`-statements
  - The case with an `else` situation

```
cant_protect_earth = False

if cant_protect_earth:
    print("We'll avenge it!")
else:
    print("We'll celebrate it!")

print('Avengers assemble')
```

```
We'll celebrate it!
Avengers assemble!
```

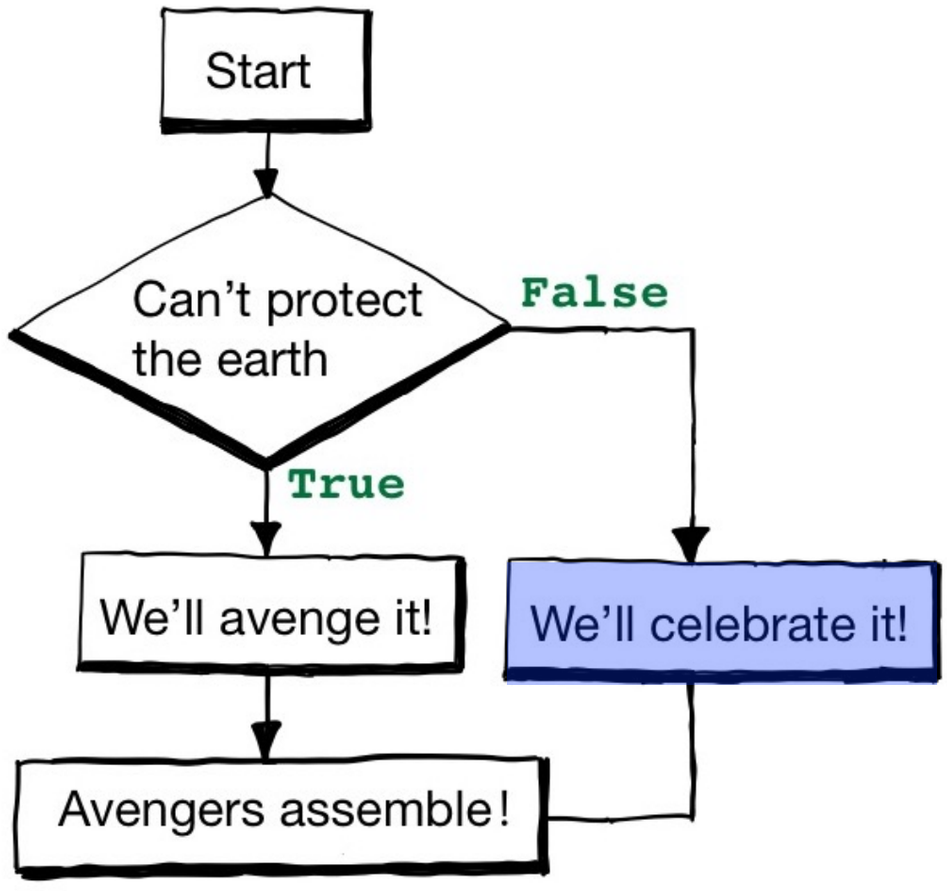


Conditional Statements

- Syntax rules of `if`-statements
  - The case with an `else` situation

```
cant_protect_earth = False

if cant_protect_earth:
    print("We'll avenge it!")
else:
    print("We'll celebrate it!")

print('Avengers assemble')
```

```
We'll celebrate it!
Avengers assemble!
```

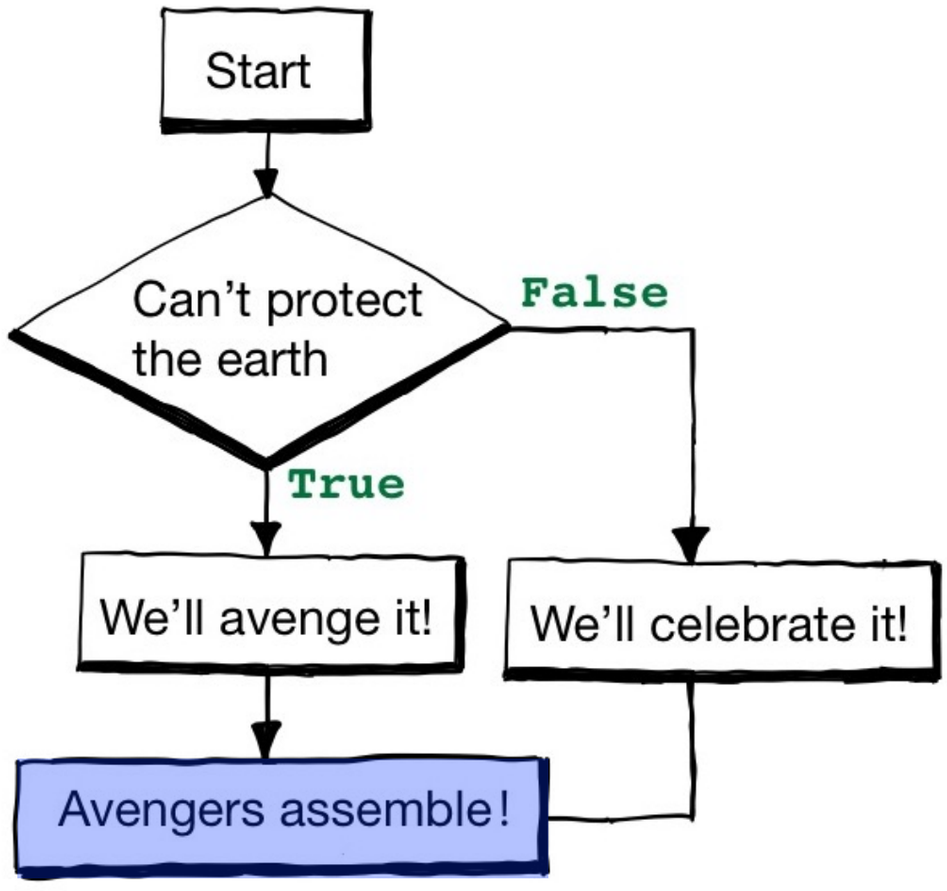


Conditional Statements

- Syntax rules of `if`-statements
  - The case with an `else` situation

```
cant_protect_earth = False

if cant_protect_earth:
    print("We'll avenge it!")
else:
    print("We'll celebrate it!")

print('Avengers assemble')
```

```
We'll celebrate it!
Avengers assemble!
```



Conditional Statements

- Syntax rules of `if`-statements
  - The case with an `else` situation

> **Question 1:** In the code segment above, the variable **`cant_protect_earth`** represents a boolean value which is **`True`** if we can not protect the earth and **`False`** otherwise. Please rewrite the program with the same logic using a boolean type variable **`can_protect_earth`**, which is **`True`** if we can protect the earth and **`False`** otherwise.

```
can_protect_earth = True     # True if we can protect the earth
```

Conditional Statements

- Syntax rules of `if`-statements
  - The case with an `else` situation

```
can_protect_earth = True          # True if we can protect the earth
```

```
if can_protect_earth:
    print("We'll celebrate it!")
else:
    print("We'll avenge it!")

print('Avengers assemble!')
```

Exchange two branches

```
We'll celebrate it!
Avengers assemble!
```

Conditional Statements

- Syntax rules of `if`-statements
  - The case with an `else` situation

```
can_protect_earth = True          # True if we can protect the earth
```

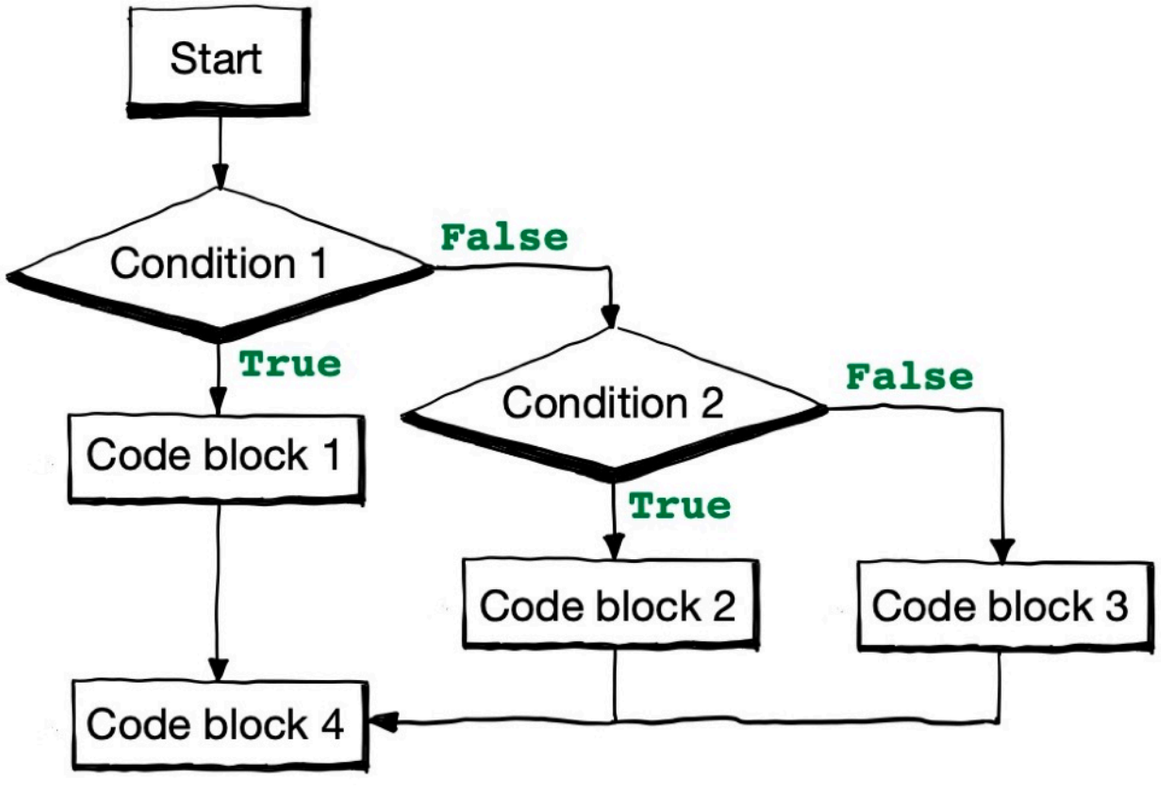
Invert the status of the boolean condition

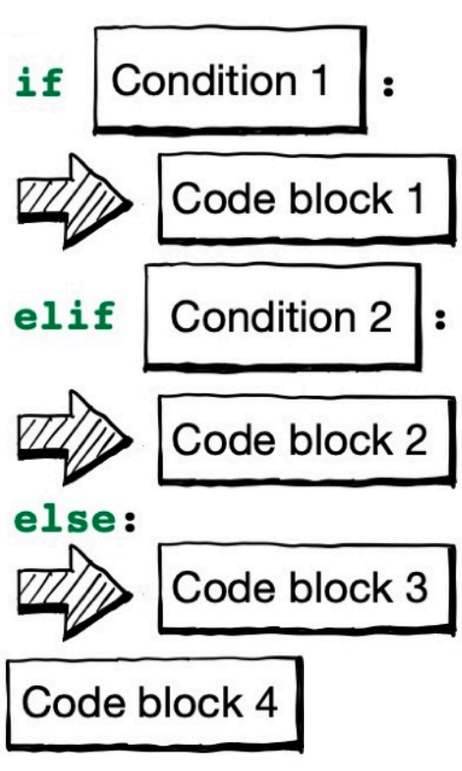
```
if not can_protect_earth:
    print("We'll avenge it!")
else:
    print("We'll celebrate it!")

print('Avengers assemble!’)
```

```
We'll celebrate it!
Avengers assemble!
```

Conditional Statements

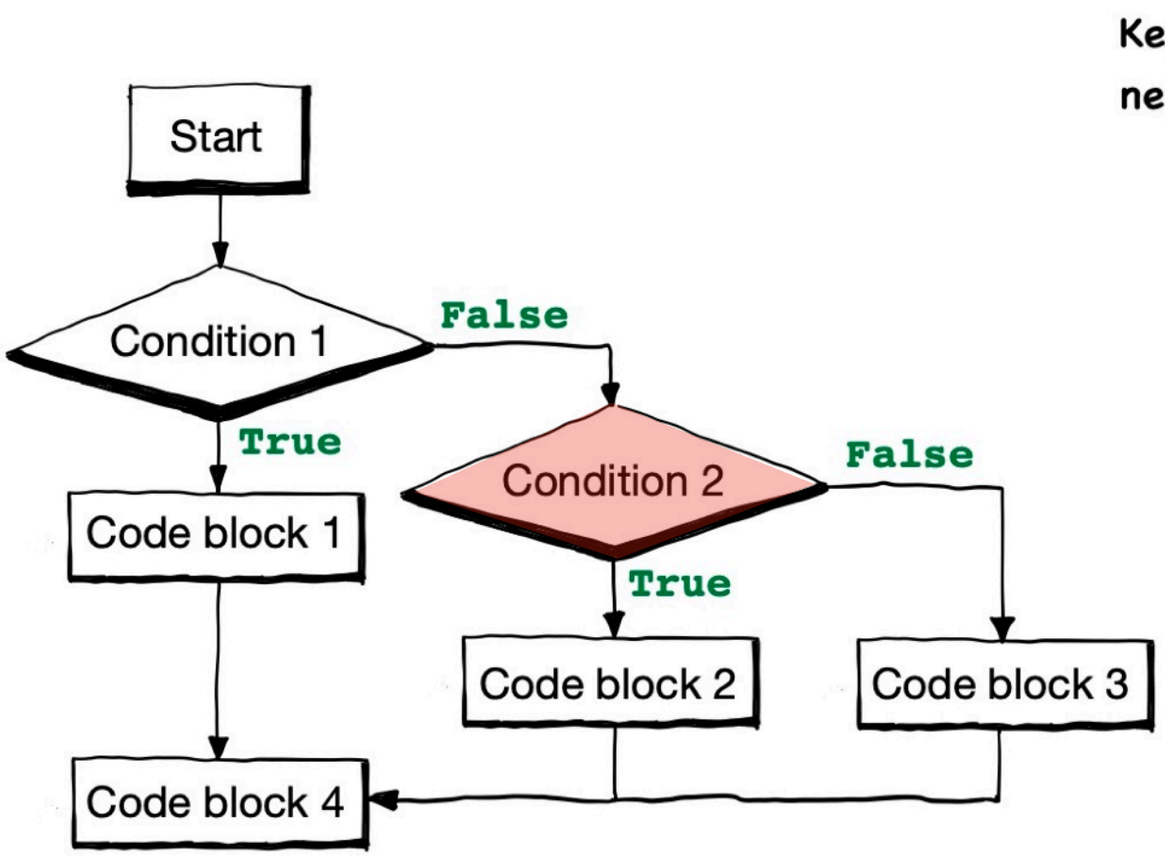
- Syntax rules of `if`-statements
  - The case with `elif` situation(s)

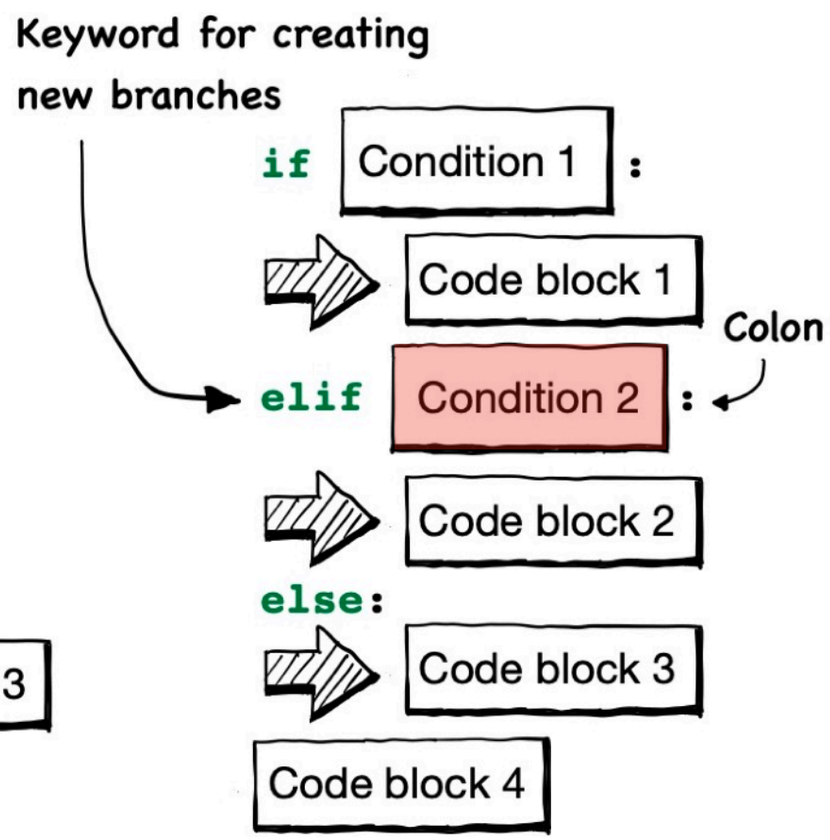




Conditional Statements

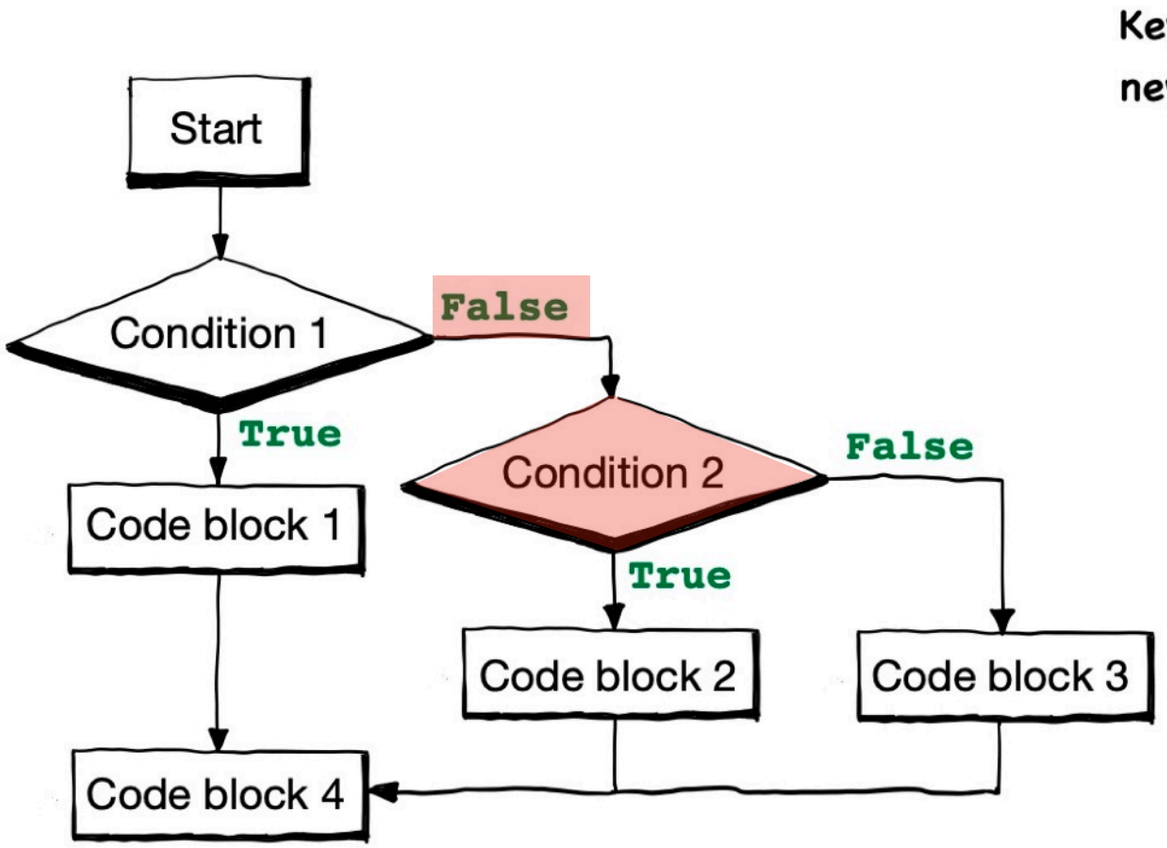
- Syntax rules of `if`-statements
  - The case with `elif` situation(s)
    - ✓ The `elif` keyword starts a new condition, followed by a colon

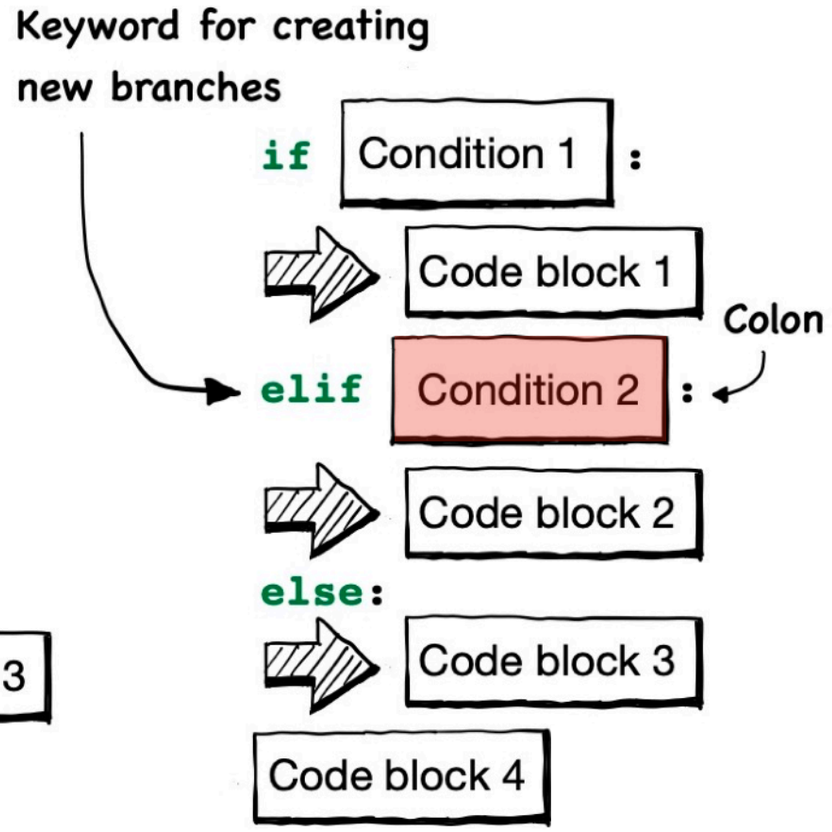




Conditional Statements

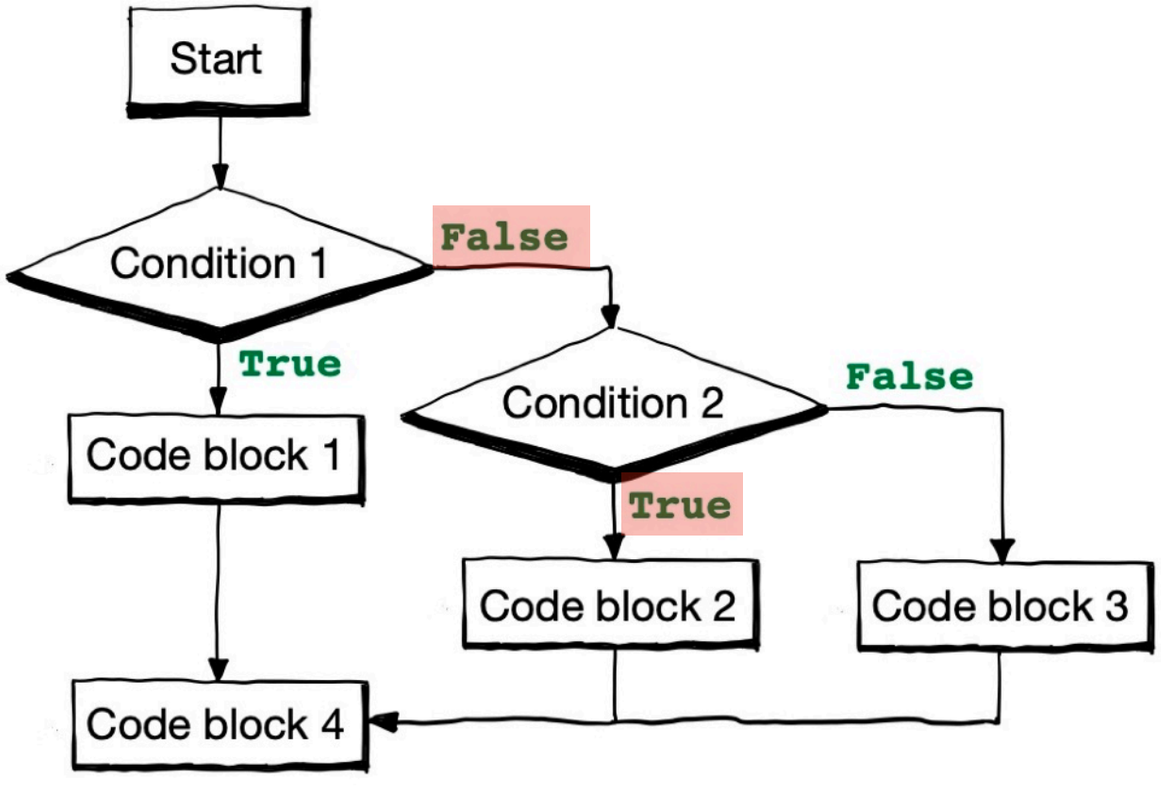
- Syntax rules of `if`-statements
  - The case with `elif` situation(s)
    - ✓ The `elif` condition is examined if all previous conditions are `False`

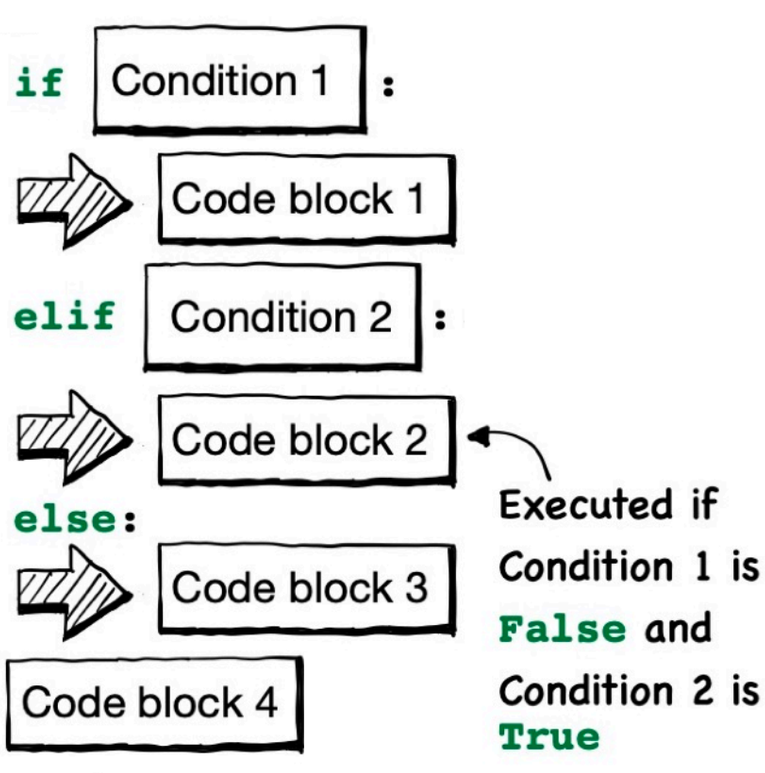




Conditional Statements

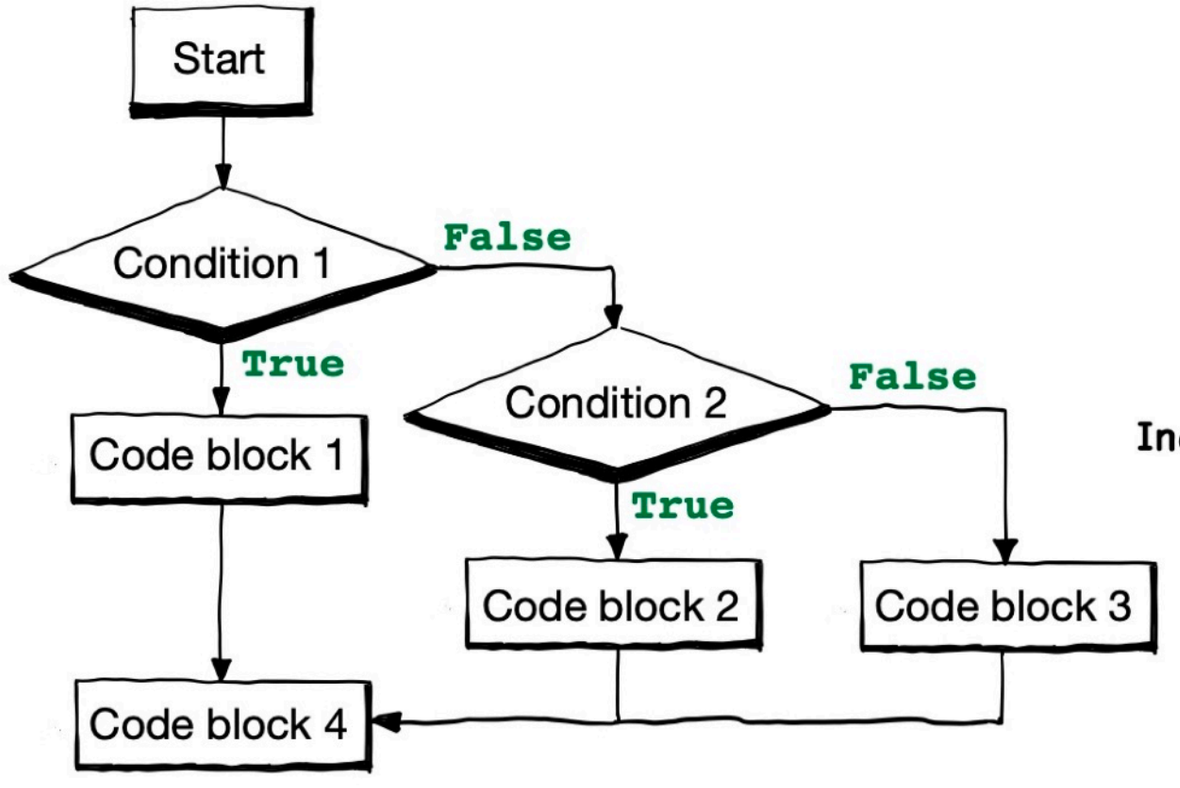
- Syntax rules of `if`-statements
  - The case with `elif` situation(s)
    - ✓ Execute **Code block 2** if **Condition 2** is `True` and all previous conditions are `False`

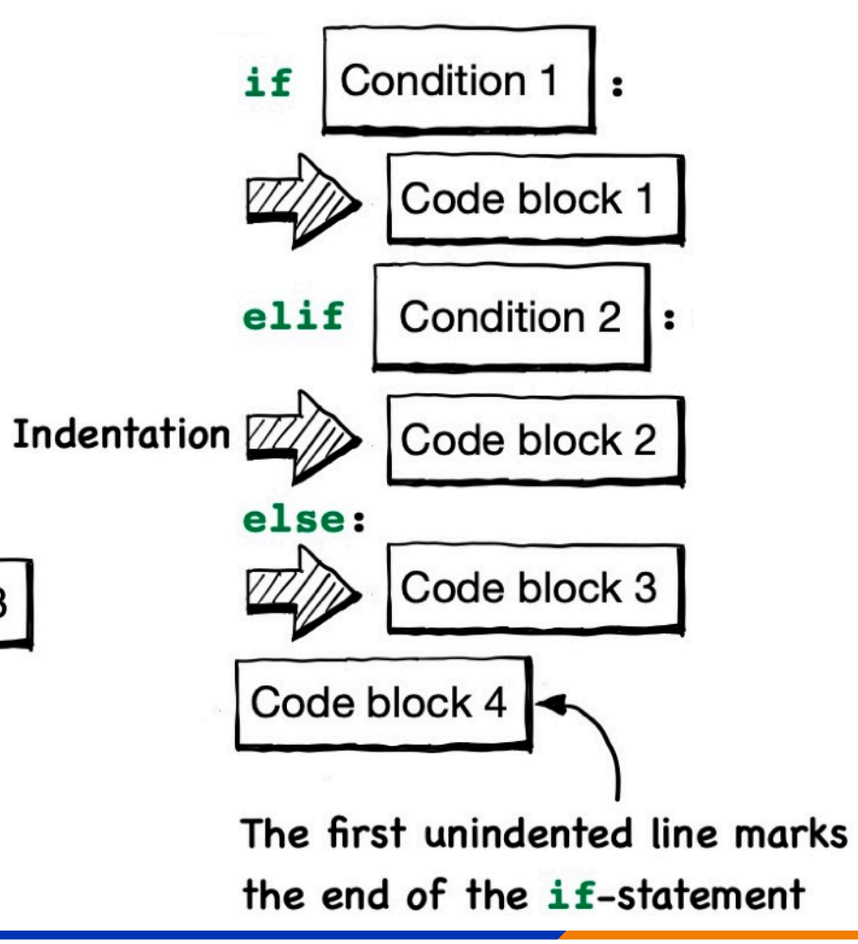




Conditional Statements

- Syntax rules of `if`-statements
  - The case with `elif` situation(s)
    - ✓ **Code block 2** has the same level of indentation as **Code block 1** and **Code block 3**





Conditional Statements

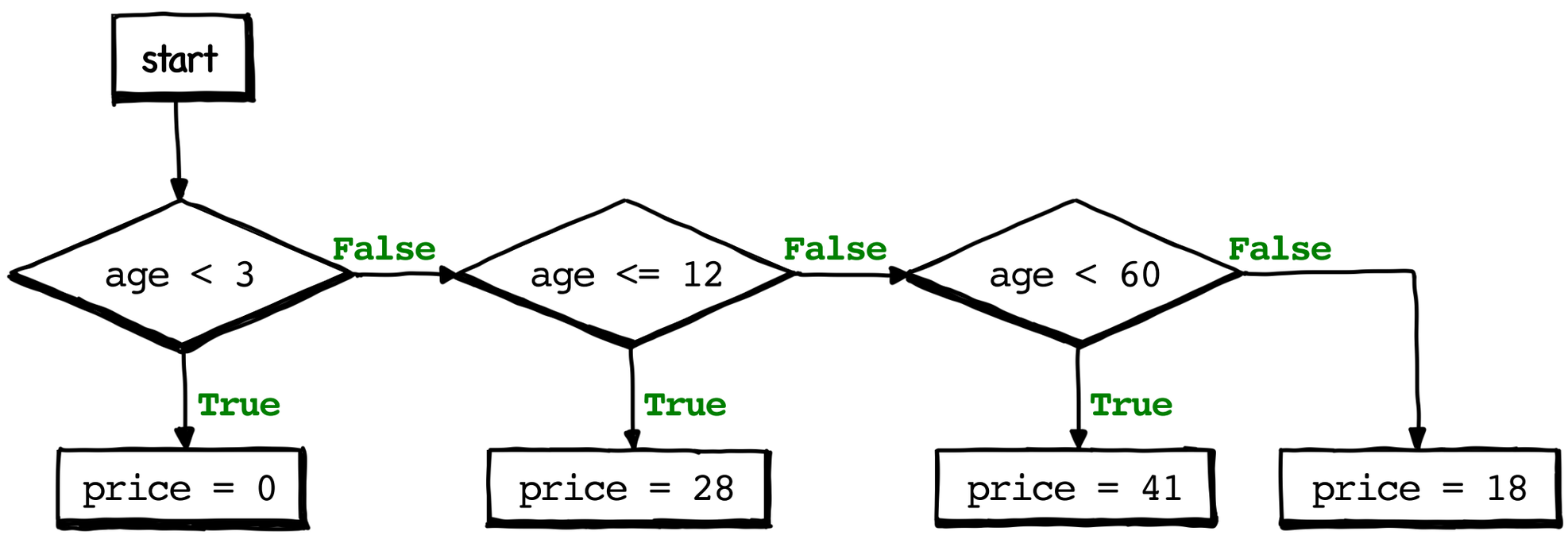
- Syntax rules of `if`-statements
  - The case with `elif` situation(s)

> **Example 1:** The price of ticket for admission to Singapore Zoo is $41 for adults, and $18 for senior citizens, who are 60 years old or above. The ticket price for children who aged 3 to 12 years old, is $28, and children under 3 years old can enjoy a free admission to the zoo. Given a visitor's age, write a program to print the ticket price.

Conditional Statements

- Syntax rules of `if`-statements
  - The case with `elif` situation(s)

> **Example 1:** The price of ticket for admission to Singapore Zoo is $41 for adults, and $18 for senior citizens, who are 60 years old or above. The ticket price for children who aged 3 to 12 years old, is $28, and children under 3 years old can enjoy a free admission to the zoo. Given a visitor's age, write a program to print the ticket price.



Conditional Statements

- Syntax rules of `if`-statements
  - The case with `elif` situation(s)

```
age = 20

if age < 3:
    ticket_price = 0
elif age <= 12:
    ticket_price = 28
elif age < 60:
    ticket_price = 41
else:
    ticket_price = 18

print('The ticket price is: $' + str(ticket_price))
```

```
The ticket price is: $41
```

Conditional Statements

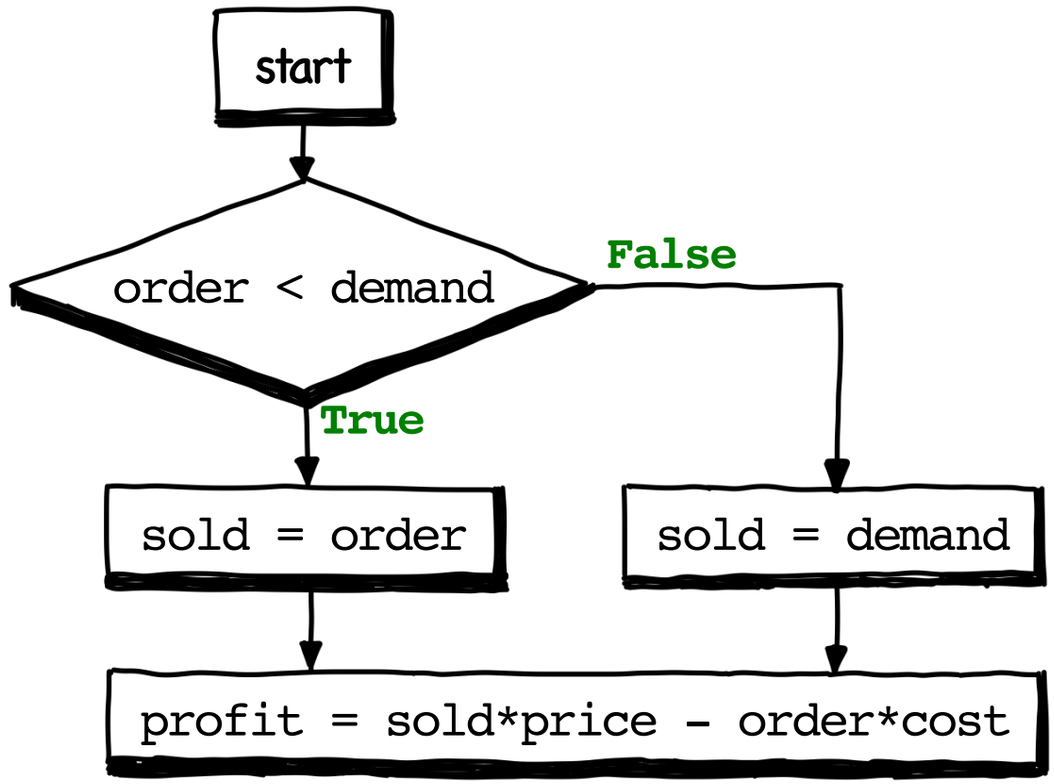
- Special cases of conditional statements
  - The `if-else` ternary expression

Example 2: A newsboy ordered 550 newspapers to sell today. Each piece of newspaper costs 10 cents and can be sold at a price of 60 cents. Write a program to calculate the total profit, given the demand quantity to be 300.

Conditional Statements

- Special cases of conditional statements
  - The `if-else` ternary expression

Example 2: A newsboy ordered 550 newspapers to sell today. Each piece of newspaper costs 10 cents and can be sold at a price of 60 cents. Write a program to calculate the total profit, given the demand quantity to be 300.

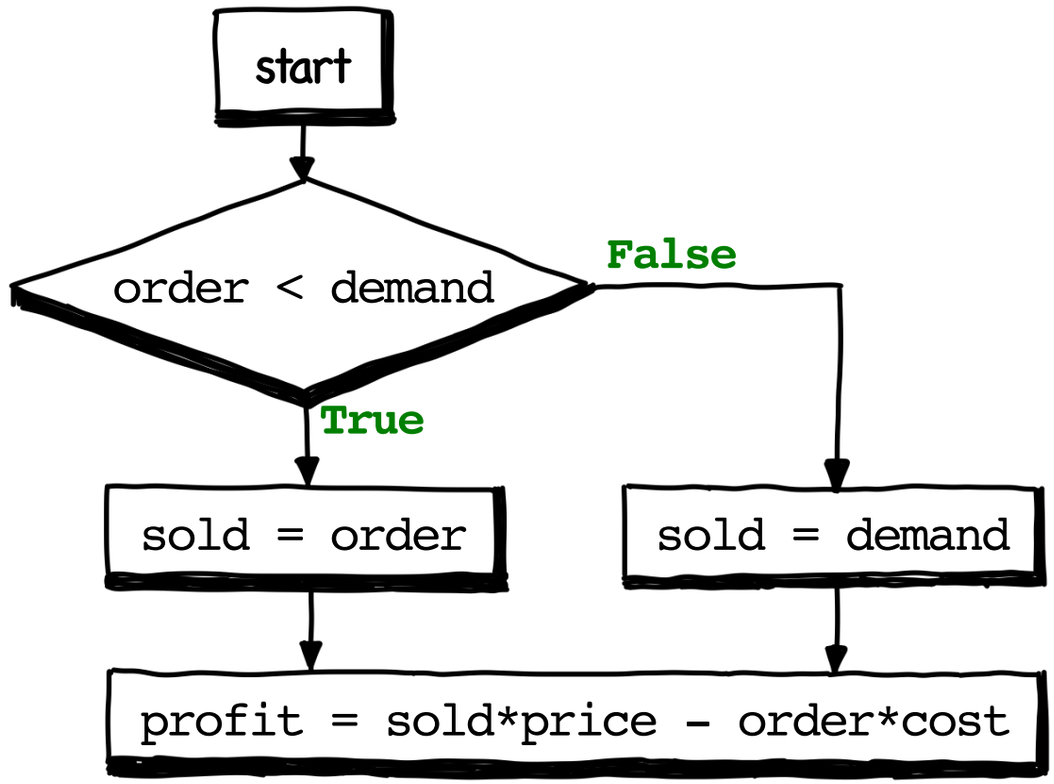


Conditional Statements

- Special cases of conditional statements
  - The `if-else` ternary expression

Example 2: A newsboy ordered 550 newspapers to sell today. Each piece of newspaper costs 10 cents and can be sold at a price of 60 cents. Write a program to calculate the total profit, given the demand quantity to be 300.

```
if order < demand:
    sold = order
else:
    sold = demand

profit = price*sold - cost*order
```



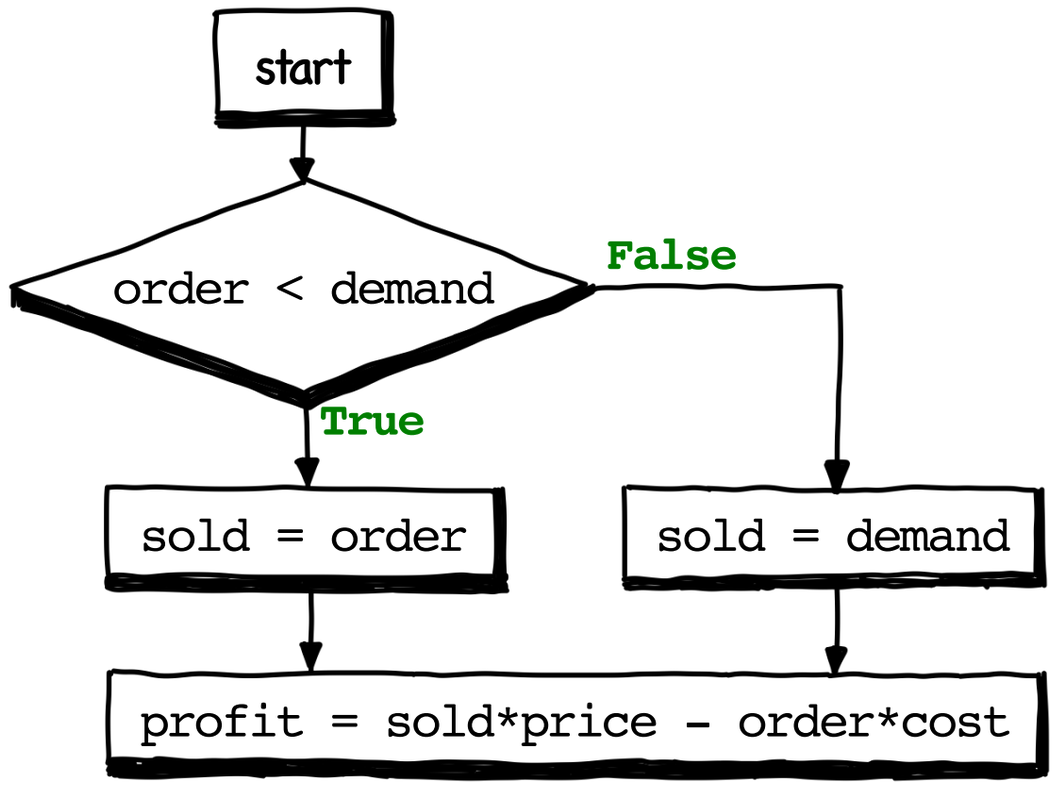
Conditional Statements

- Special cases of conditional statements
  - The `if-else` ternary expression

```
sold = demand if demand < order else order
sold
```

```
300
```

```
if order < demand:
    sold = order
else:
    sold = demand

profit = price*sold - cost*order
```

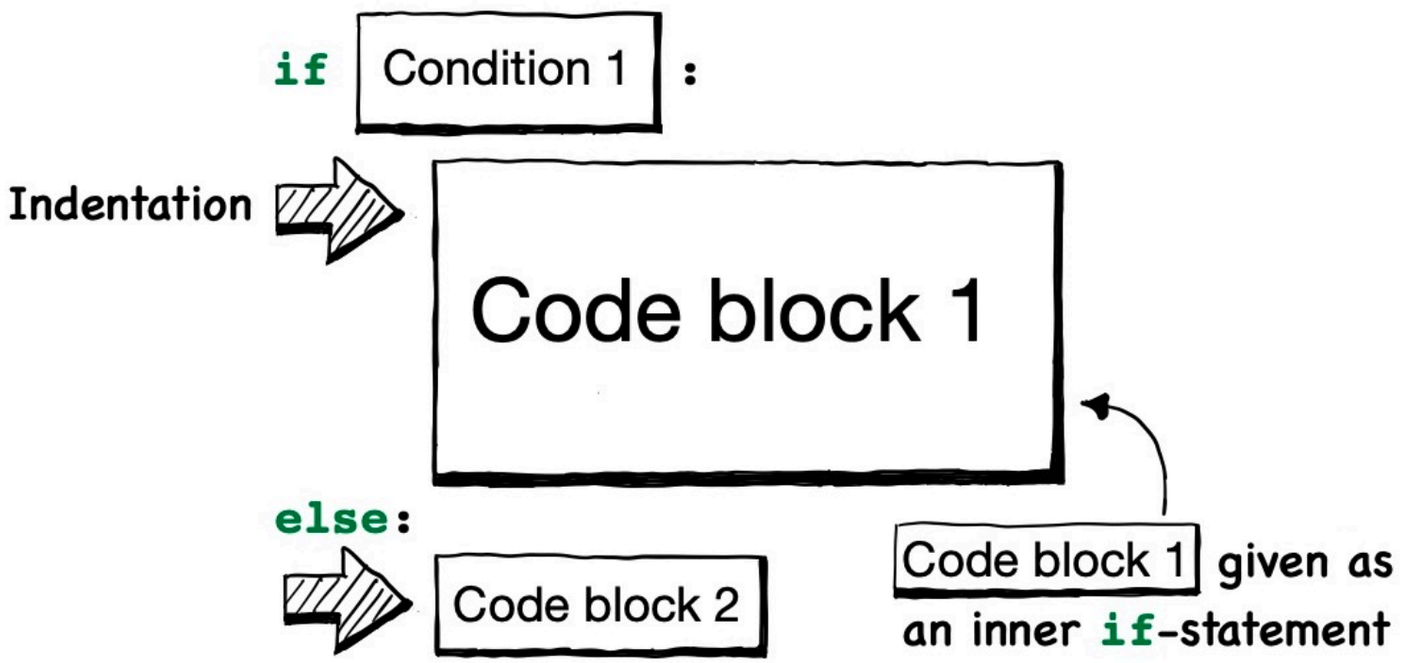


Conditional Statements

- Special cases of conditional statements
  - The `if-else` ternary expression
    - ✓ Simple action to take for each branch
    - ✓ Limited number of branches

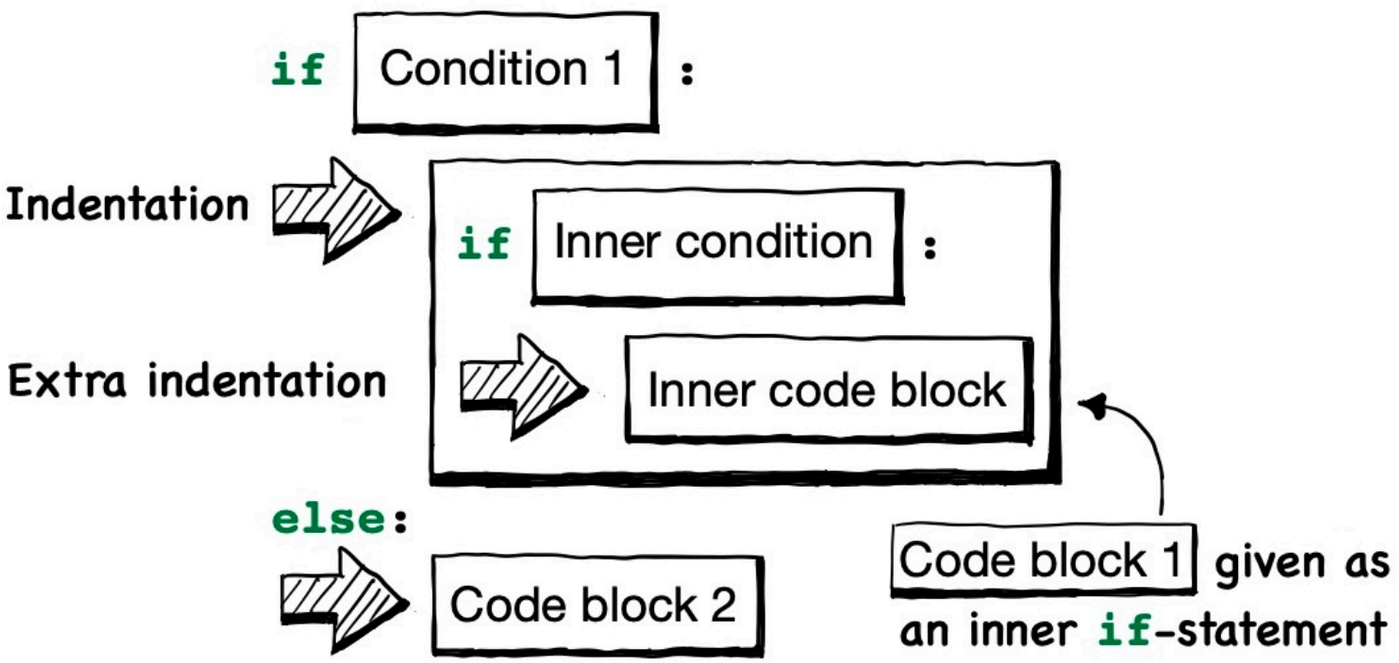
Conditional Statements

- Special cases of conditional statements
  - Nested `if`-statements



Conditional Statements

- Special cases of conditional statements
  - Nested `if`-statements



Conditional Statements

- Special cases of conditional statements
  - Nested `if`-statements

Example 3: A programming course is taken by both full-time and part-time students. Their grades for this course are calculated separately, as shown in the table below. Write a program that asks students two questions:

- Are you a full-time student? Yes/No
- What is your score?

and then prints out the corresponding grades of the student.

Grades	Scores of full-time students	Scores of part-time students
A	Higher or equal to 90	Higher or equal to 85
B	Between 80 and 89	Between 75 and 84
C	Between 70 and 79	Between 65 and 74
D	Lower than 70	Lower than 65

Conditional Statements

- Special cases of conditional statements
  - Nested `if`-statements

```
if is_full_time == 'Yes':
    if score >= 90:
        grade = 'A'
    elif score >= 80:
        grade = 'B'
    elif score >= 70:
        grade = 'C'
    else:
        grade = 'D'
else:
    if score >= 85:
        grade = 'A'
    elif score >= 75:
        grade = 'B'
    elif score >= 65:
        grade = 'C'
    else:
        grade = 'D'
```

Conditional Statements

- Special cases of conditional statements
  - Nested `if`-statements

```
score = score - 5 if is_full_time == 'Yes' else score

if score >= 85:
    grade = 'A'
elif score >= 75:
    grade = 'B'
elif score >= 65:
    grade = 'C'
else:
    grade = 'D'
```

5 marks are deducted from full-time students' scores

Part-time students' scores remain the same

Grading according to part-time students' criteria

Conditional Statements

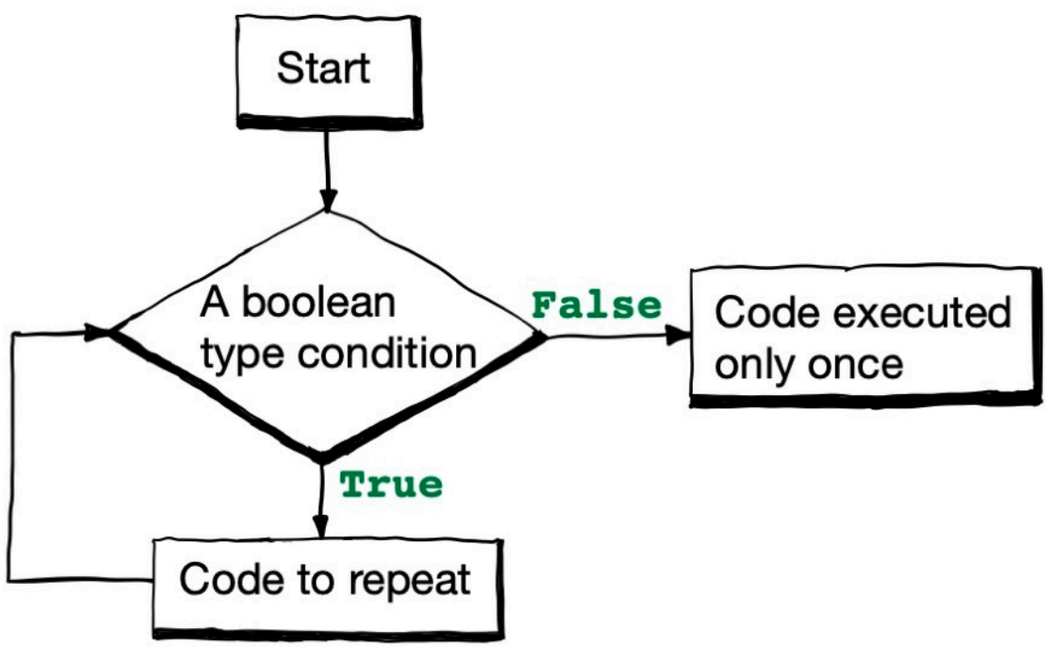
- Special cases of conditional statements
  - Nested `if`-statements

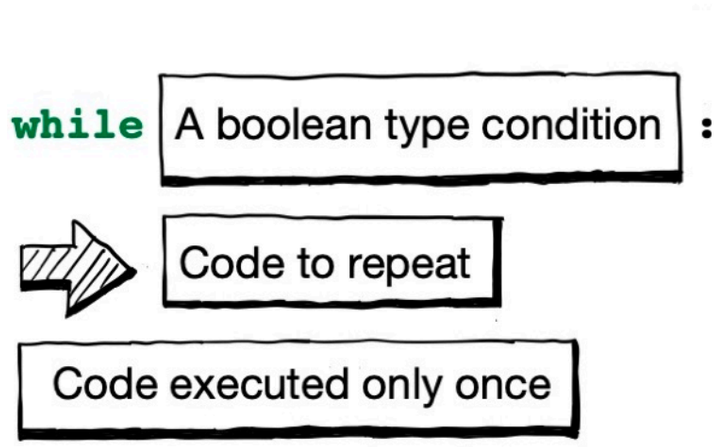
```
score = score - 5 if is_full_time == 'Yes' else score

if score >= 85:
    grade = 'A'
elif score >= 75:
    grade = 'B'
elif score >= 65:
    grade = 'C'
else:
    grade = 'D'
```

Notes: Python uses indentation to determine the logical connection between blocks of code. If a line of code is not correctly indented, you may encounter an **indentation error** message, or have incorrect results as the code is executed mistakenly.

Loops and Iterations

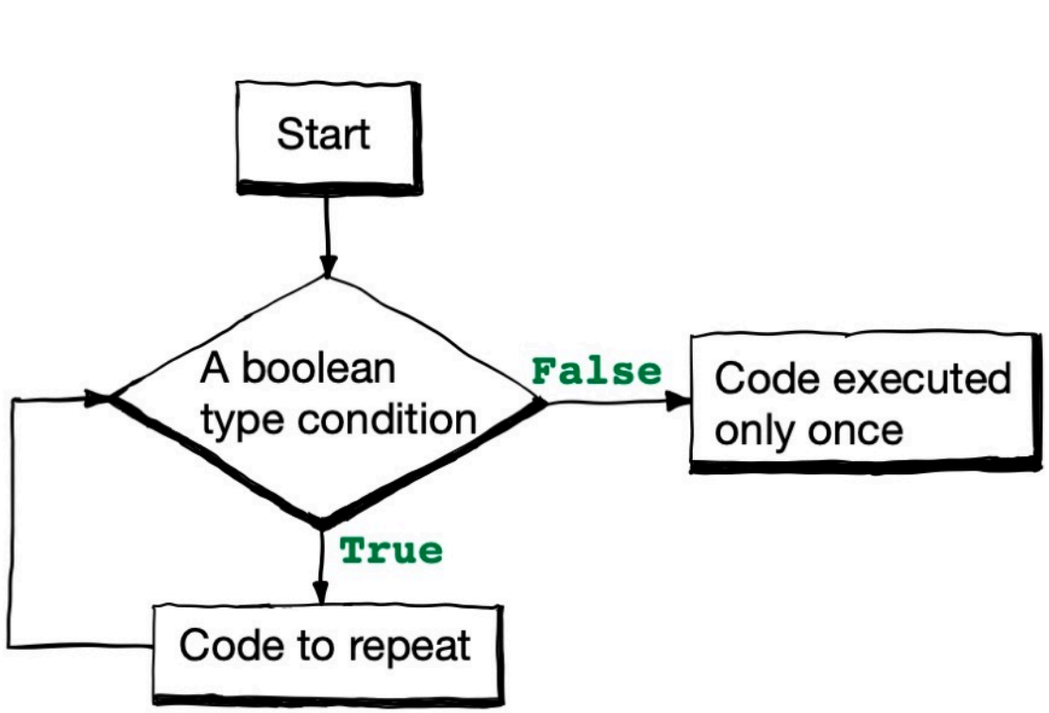
- Syntax rules of `while` loops

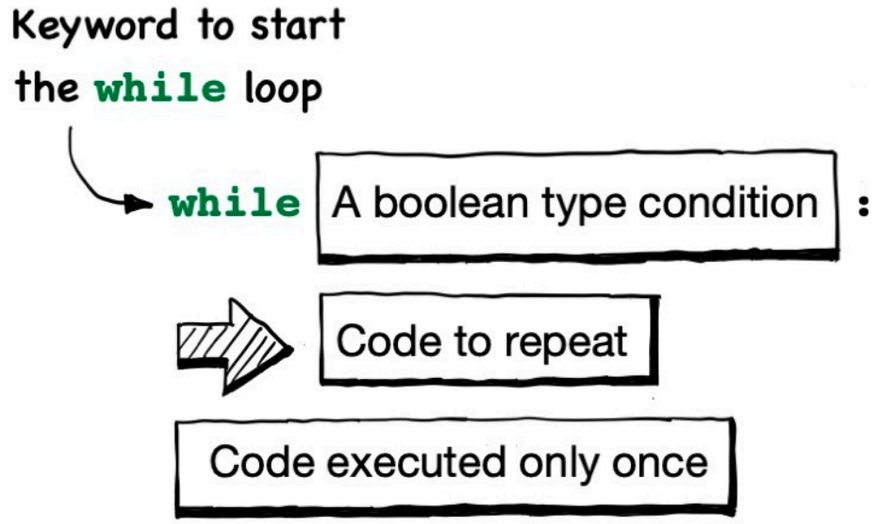




Loops and Iterations

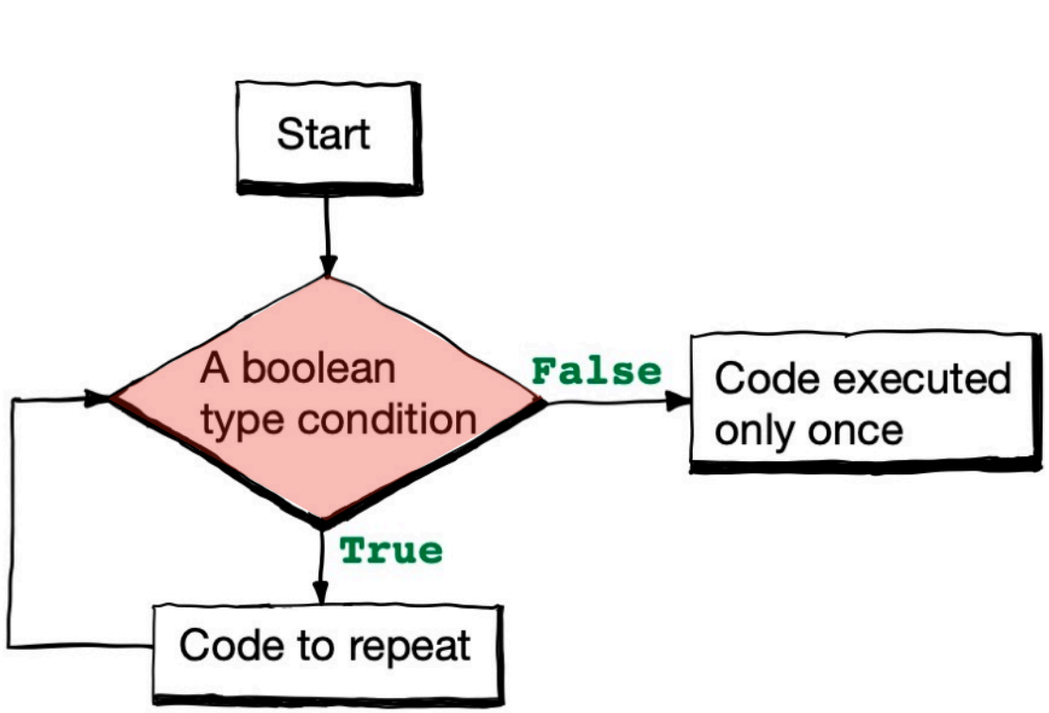
- Syntax rules of `while` loops
  - The loop is started with the keyword `while`

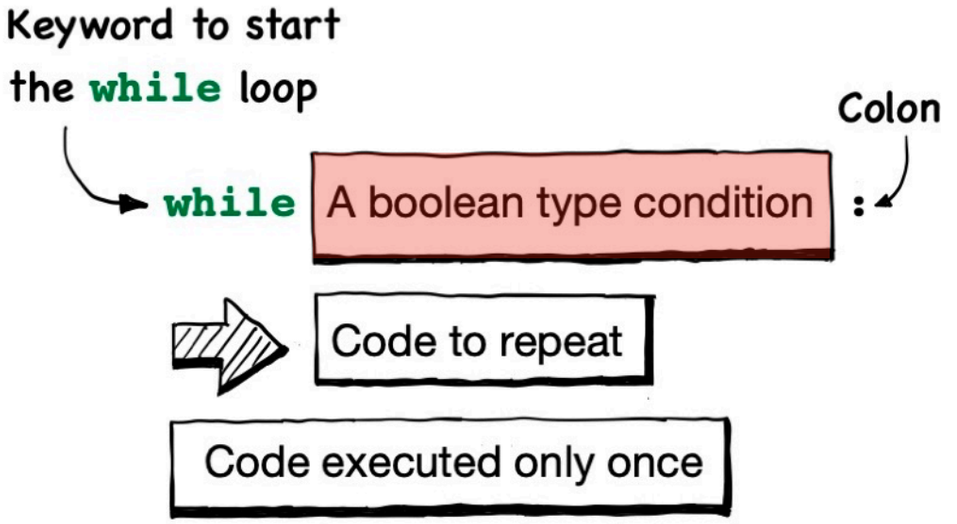




Loops and Iterations

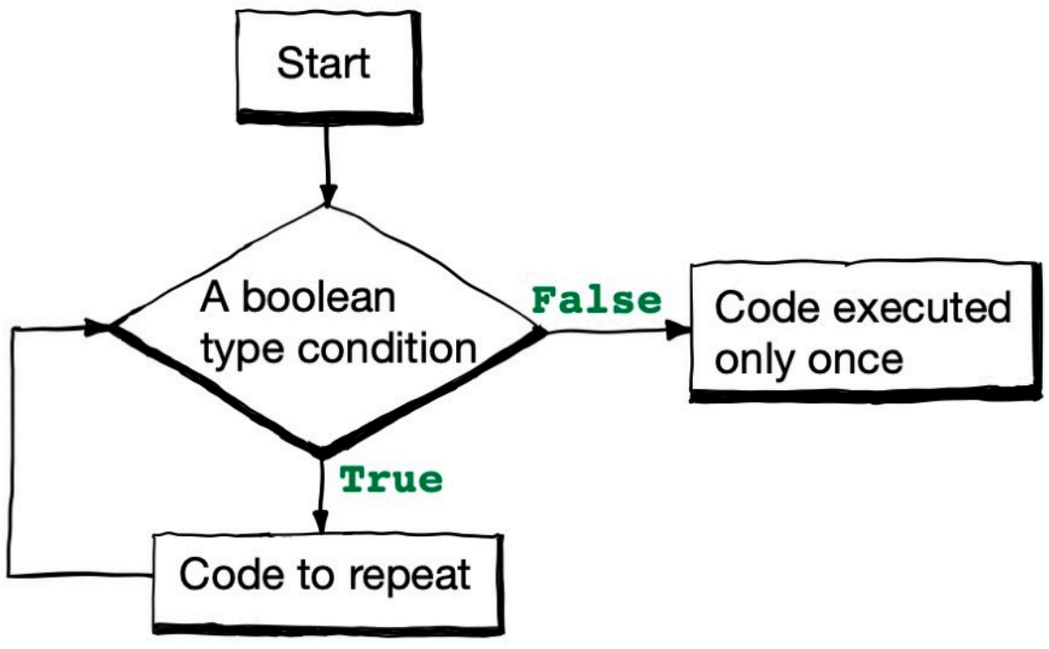
- Syntax rules of `while` loops
  - The keyword `while` is followed by a boolean type condition and a colon

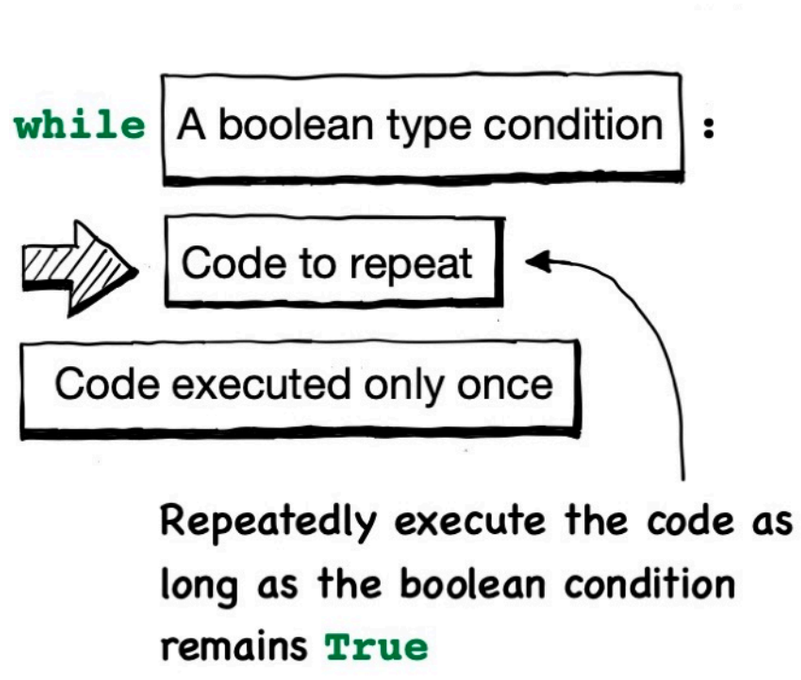




Loops and Iterations

- Syntax rules of `while` loops
  - The boolean type condition is examined in each iteration, and **Code to repeat** is repeatedly executed as long as the condition is `True`

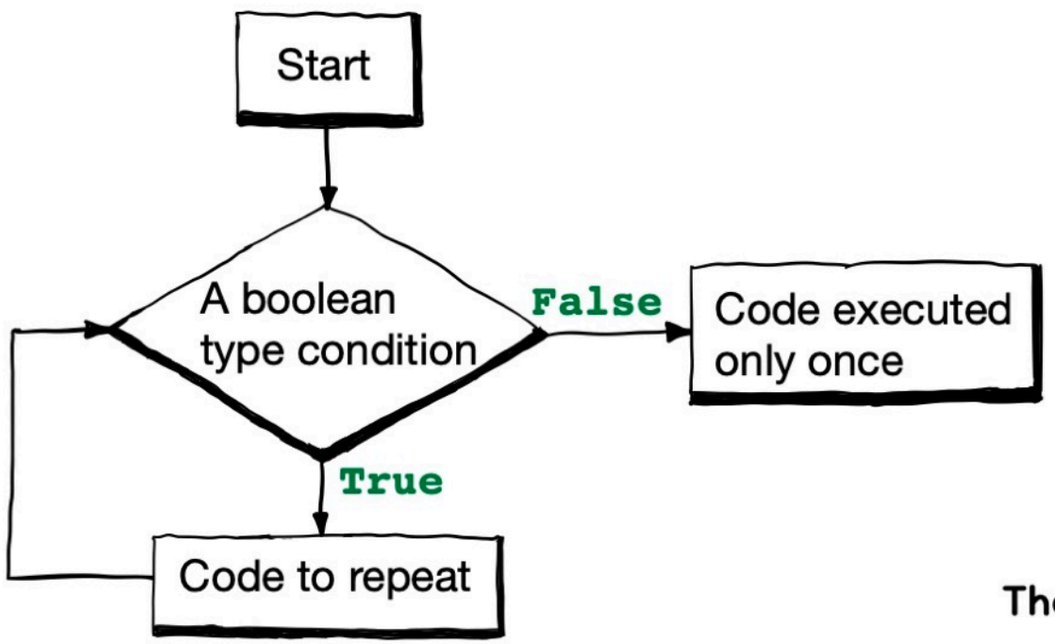


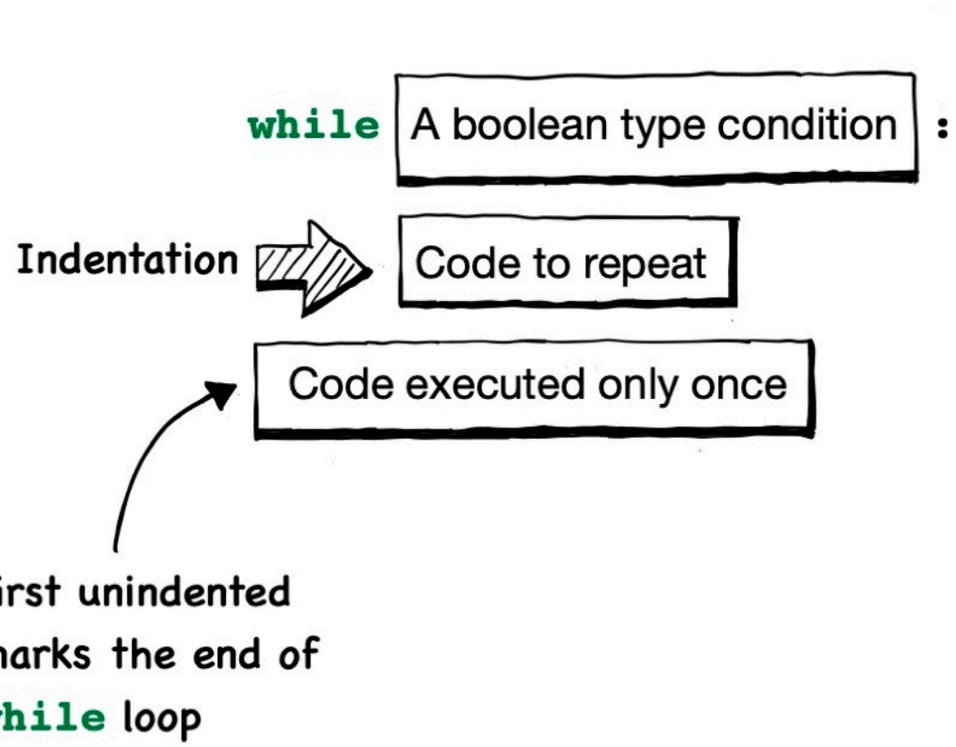


Loops and Iterations

- Syntax rules of `while` loops

Notes: Indentation is used to differentiate **Code to be repeated** and **Code executed only once**.

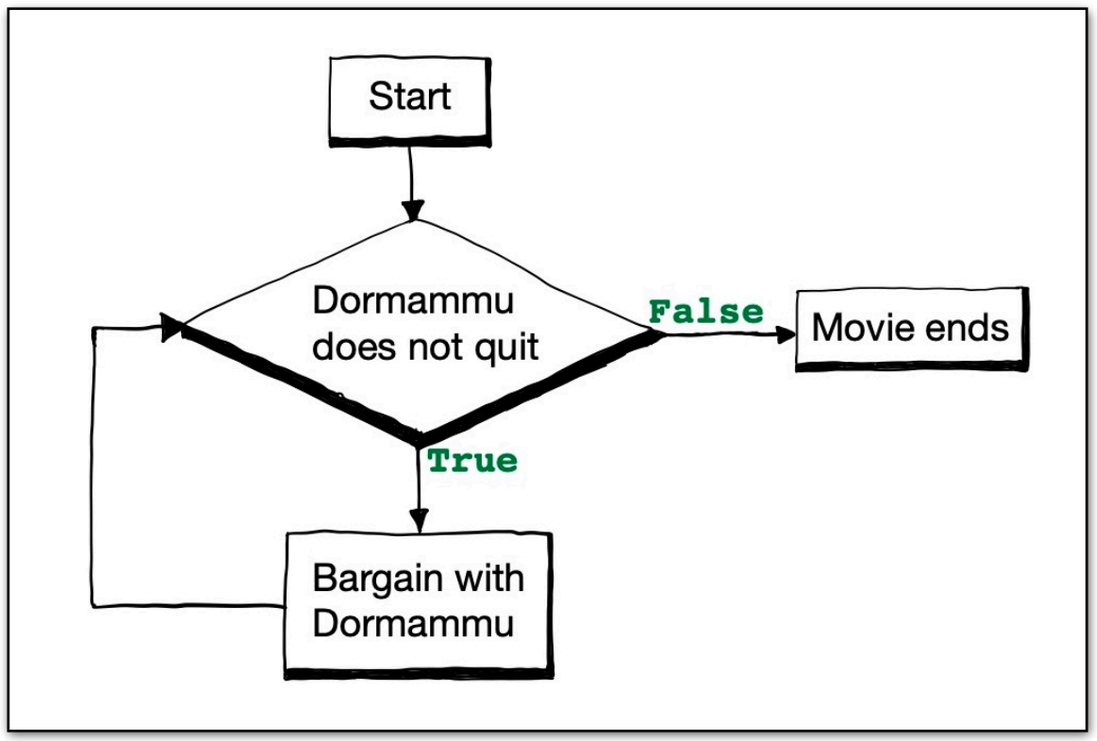




Loops and Iterations

- Syntax rules of `while` loops

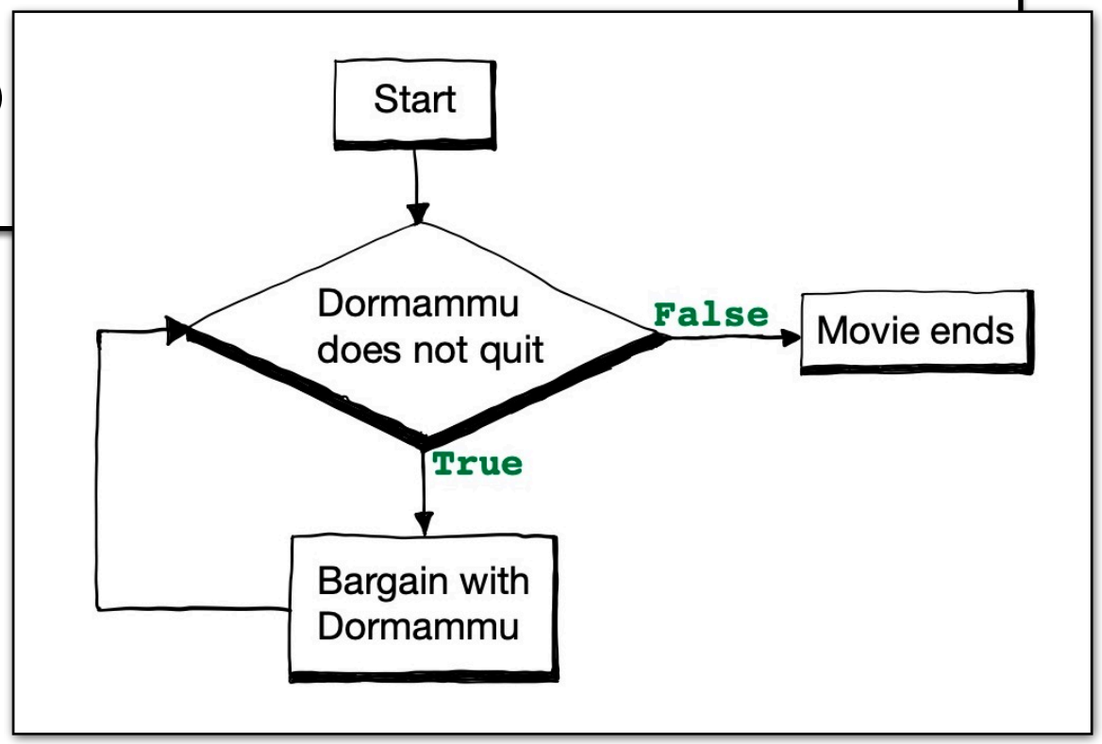
> **Example 4:** In the movie Dr. Strange (2016), Steven Strange beat Dormammu by keeping annoying him until he was extremely bored and frustrated. Write a program to act like Dr. Strange. The "time" loop can only be broken when you type in "I quit" or "You win".



Loops and Iterations

- Syntax rules of `while` loops

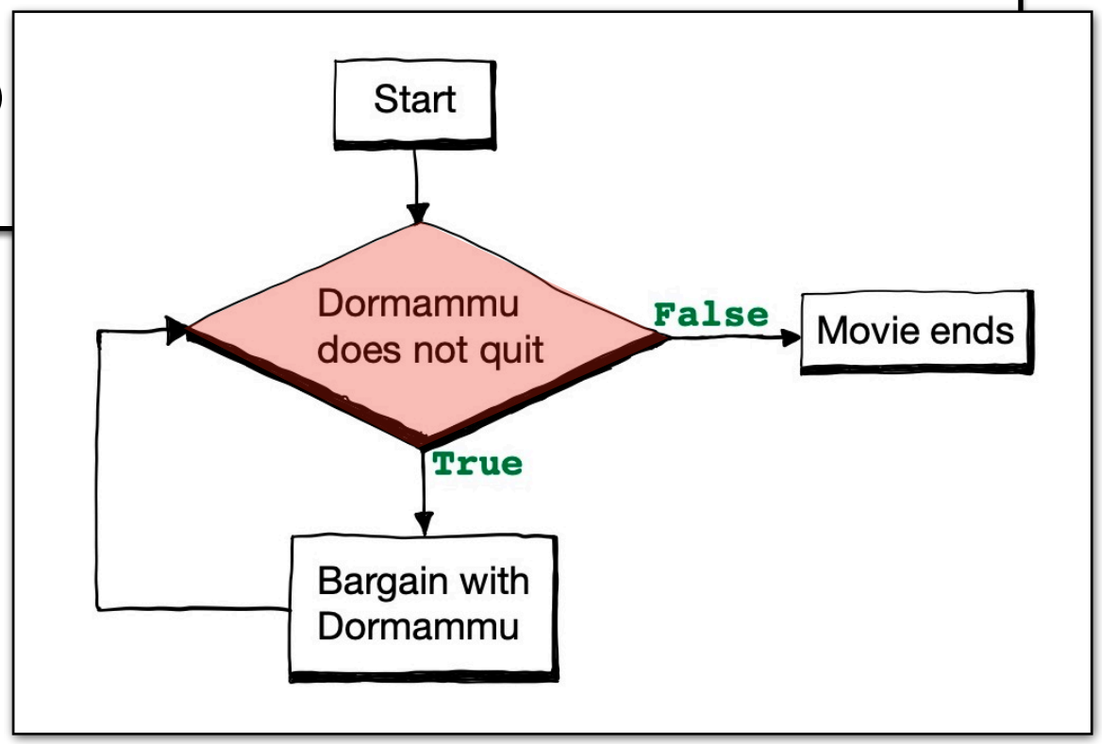
```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```



Loops and Iterations

- Syntax rules of `while` loops

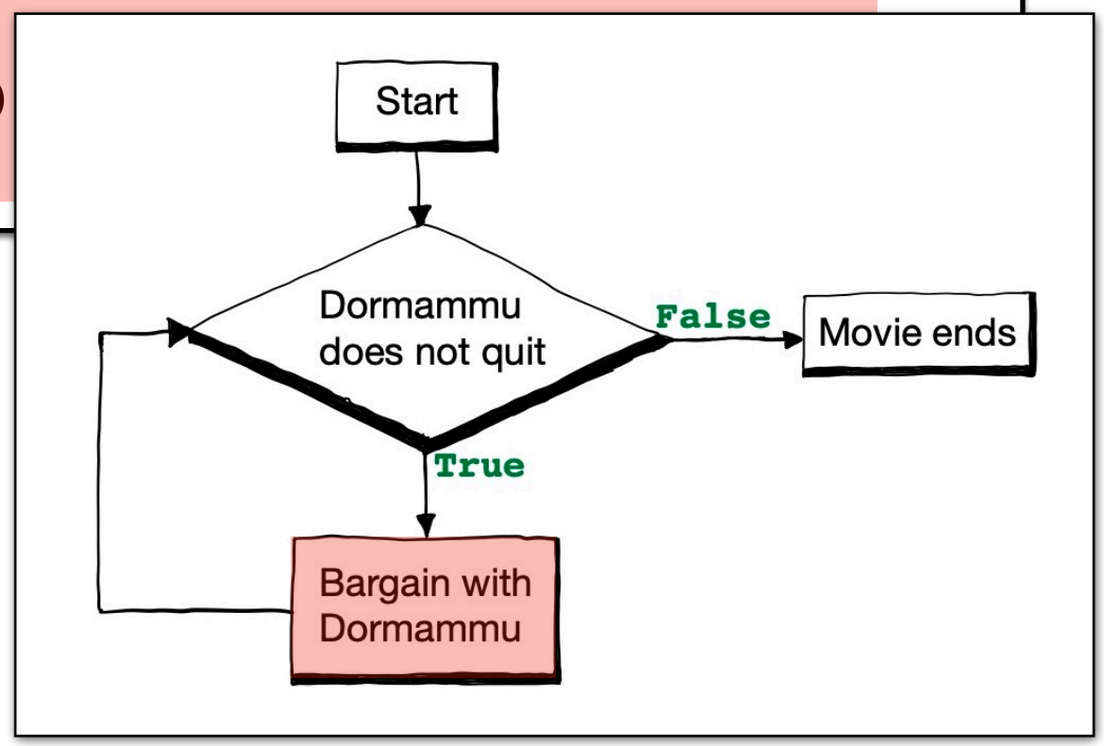
```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```



Loops and Iterations

- Syntax rules of `while` loops

```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```



Loops and Iterations

- Syntax rules of `while` loops


```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

Loops and Iterations

- Syntax rules of `while` loops


```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

True

Loops and Iterations

- Syntax rules of `while` loops


```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
```

Loops and Iterations

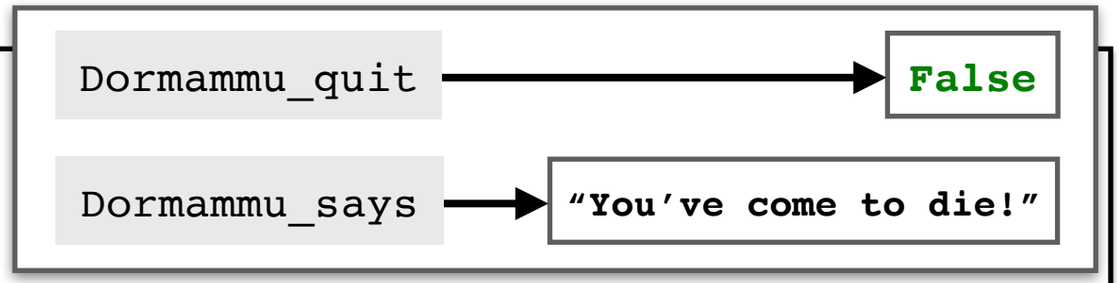
- Syntax rules of `while` loops


```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
```

Loops and Iterations

- Syntax rules of `while` loops

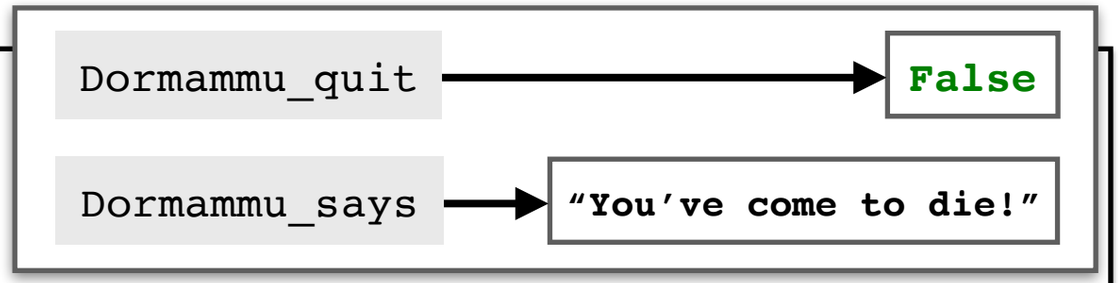



```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
```

Loops and Iterations

- Syntax rules of `while` loops

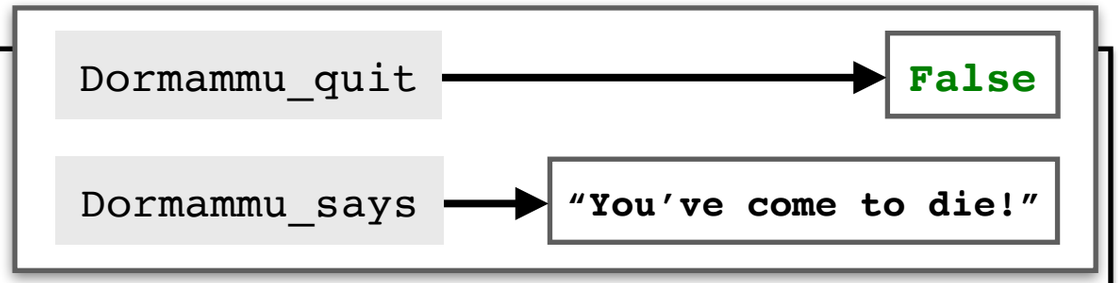



```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~
```

Loops and Iterations

- Syntax rules of `while` loops



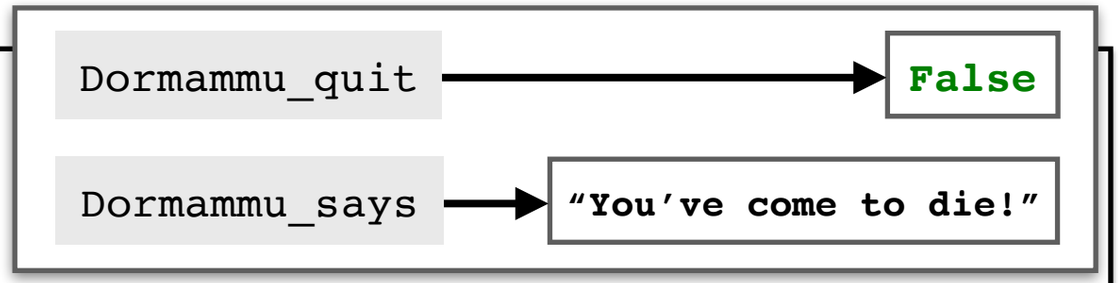

```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

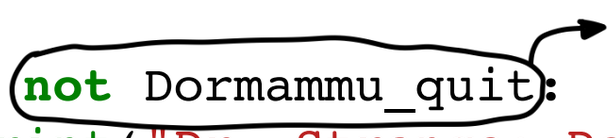
```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~
```

Loops and Iterations

- Syntax rules of `while` loops

```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

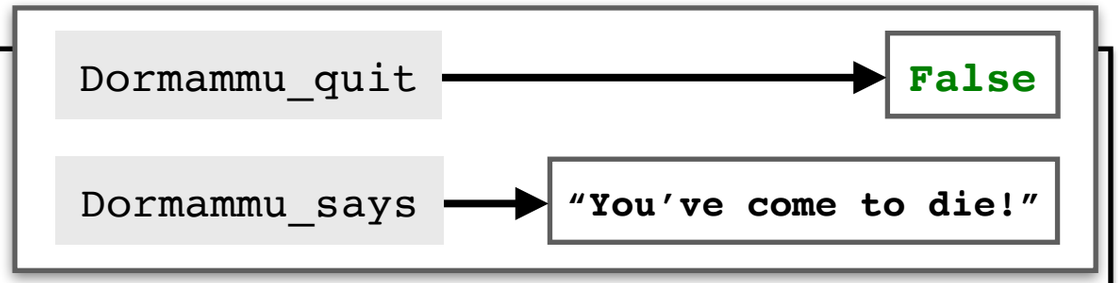





```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~
```

Loops and Iterations

- Syntax rules of `while` loops

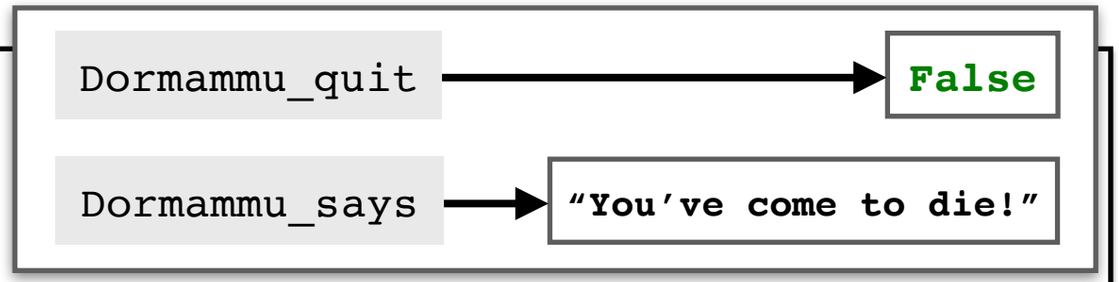



```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~


Dr. Strange: Dormammu, I've come to bargain!
```

Loops and Iterations

- Syntax rules of `while` loops



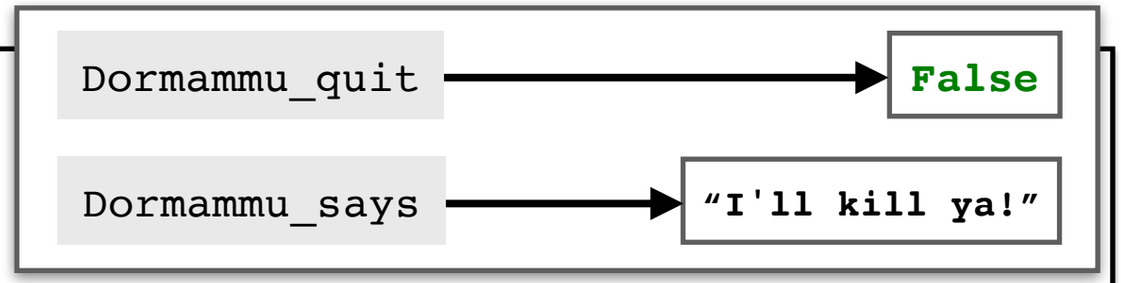

```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~


Dr. Strange: Dormammu, I've come to bargain!
Dormammu: I'll kill ya!
```

Loops and Iterations

- Syntax rules of `while` loops

```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

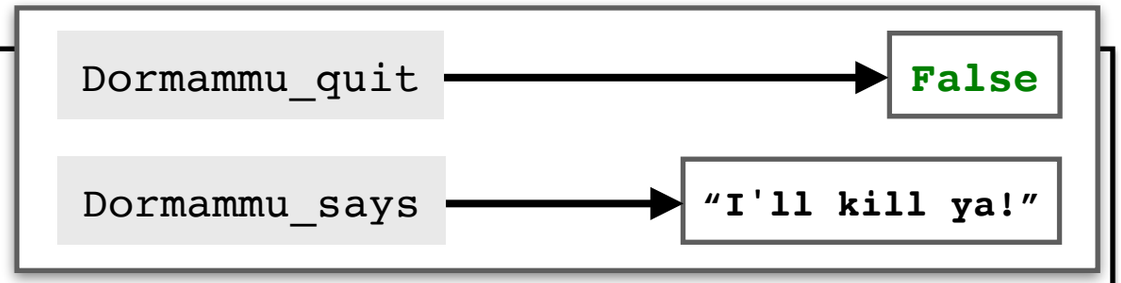



```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~


Dr. Strange: Dormammu, I've come to bargain!
Dormammu: I'll kill ya!
```

Loops and Iterations

- Syntax rules of `while` loops

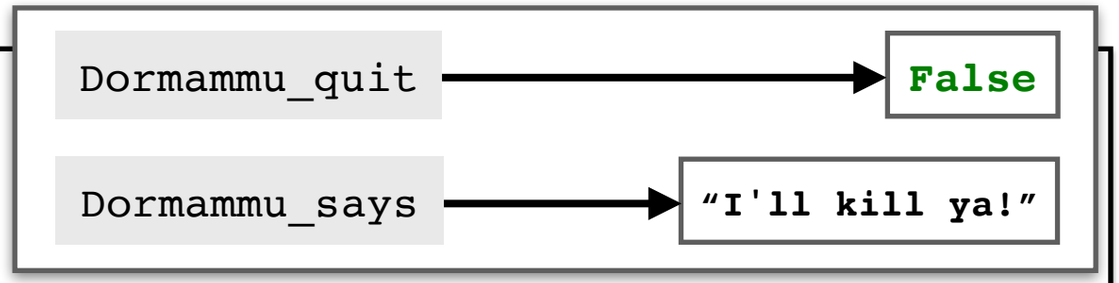
```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```




```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~


Dr. Strange: Dormammu, I've come to bargain!
Dormammu: I'll kill ya!
Dr. Strange: Ah~~~
```

Loops and Iterations

- Syntax rules of `while` loops

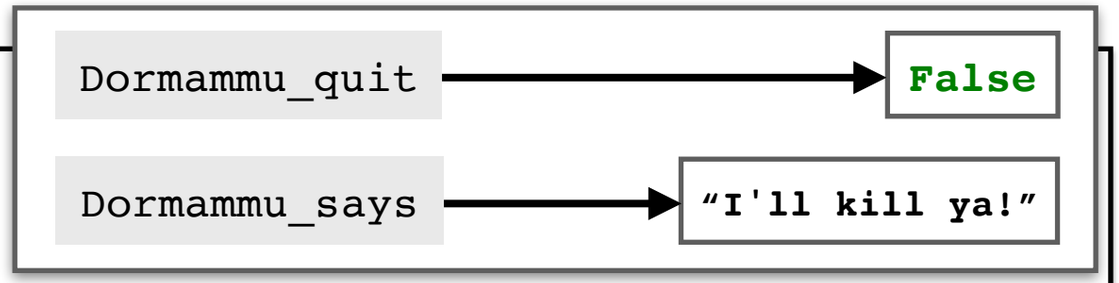



```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~


Dr. Strange: Dormammu, I've come to bargain!
Dormammu: I'll kill ya!
Dr. Strange: Ah~~~
```

Loops and Iterations

- Syntax rules of `while` loops



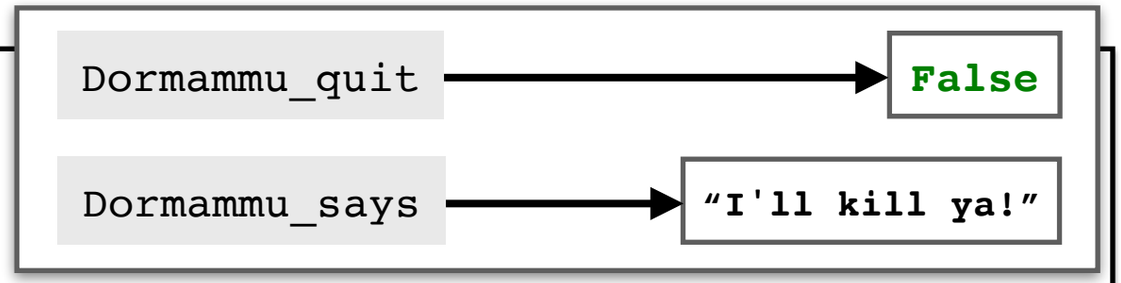

```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
...
...
...


Dr. Strange: Dormammu, I've come to bargain!
```

Loops and Iterations

- Syntax rules of `while` loops

```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

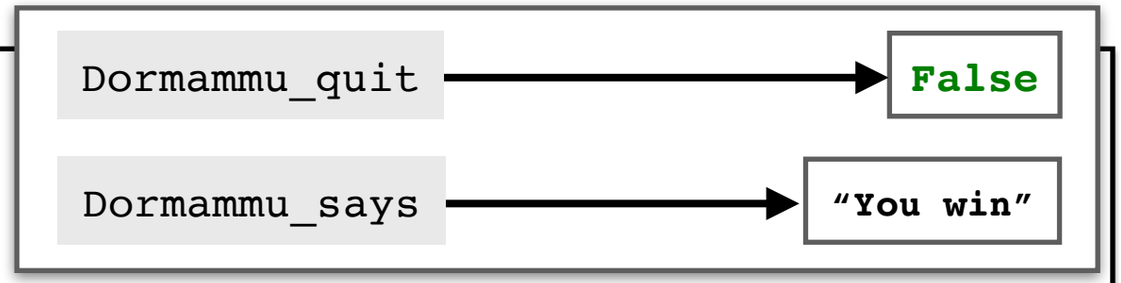



```
...
...
...


Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You win
```

Loops and Iterations

- Syntax rules of `while` loops

```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```



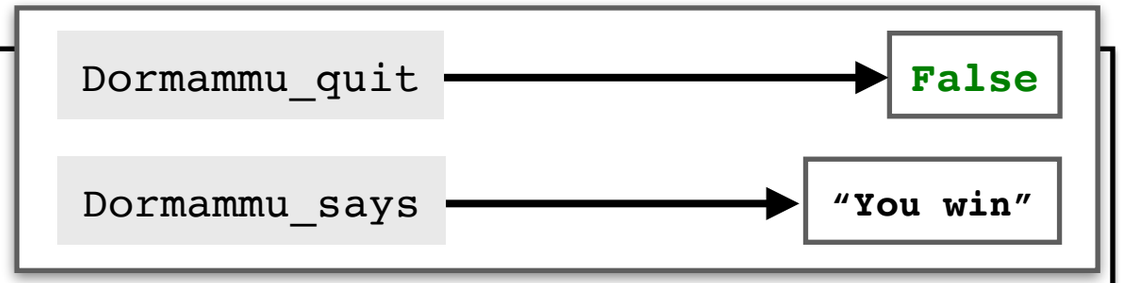

```
...
...
...
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You win
```

Loops and Iterations

- Syntax rules of `while` loops

```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```



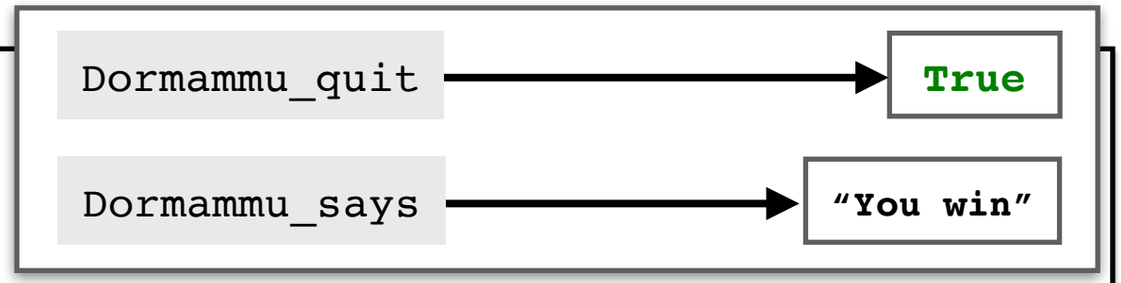

```
...
...
...
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You win
```

Loops and Iterations

- Syntax rules of `while` loops

```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```



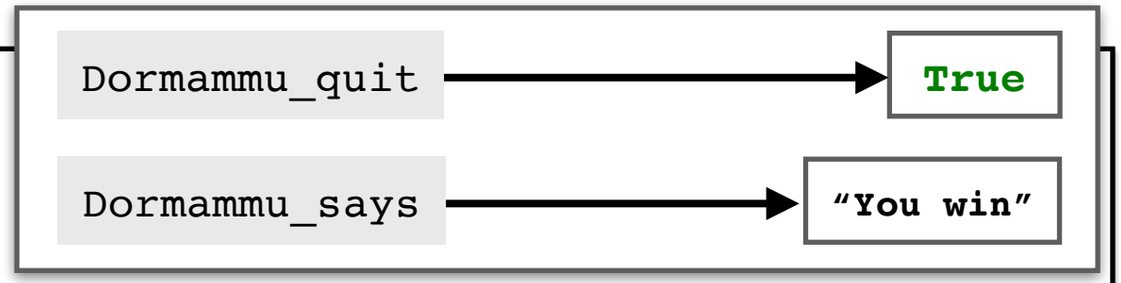

```
...
...
...
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You win
```

Loops and Iterations

- Syntax rules of `while` loops

```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

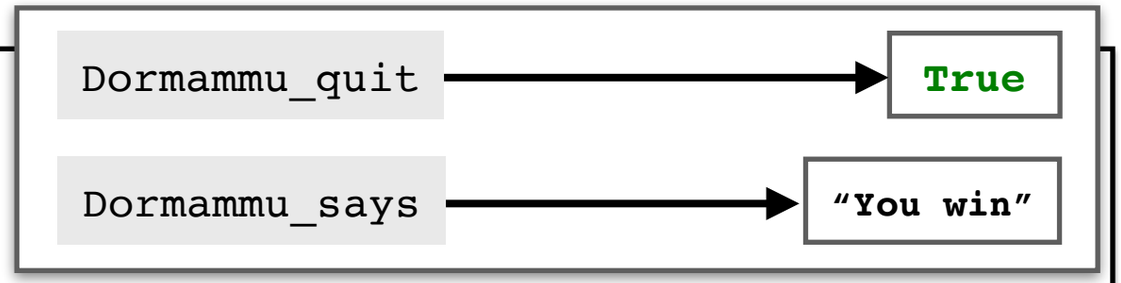



```
...
...
...


Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You win
Dr. Strange: Wise choice bro!
```

Loops and Iterations

- Syntax rules of `while` loops

```
Dormammu_quit = False

while not Dormammu_quit:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        Dormammu_quit = True
        print('Dr. Strange: Wise choice bro!')
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```



False

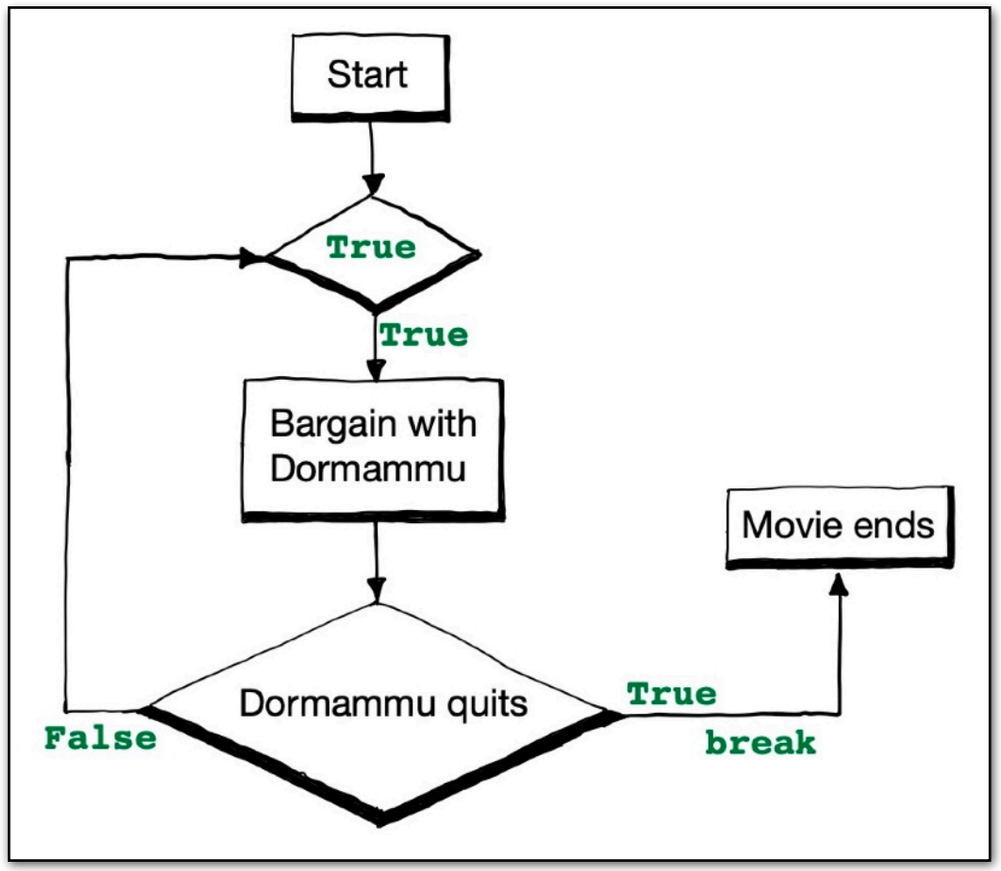
...
...
...

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You win
Dr. Strange: Wise choice bro!
```

Loops and Iterations

- Syntax rules of `while` loops

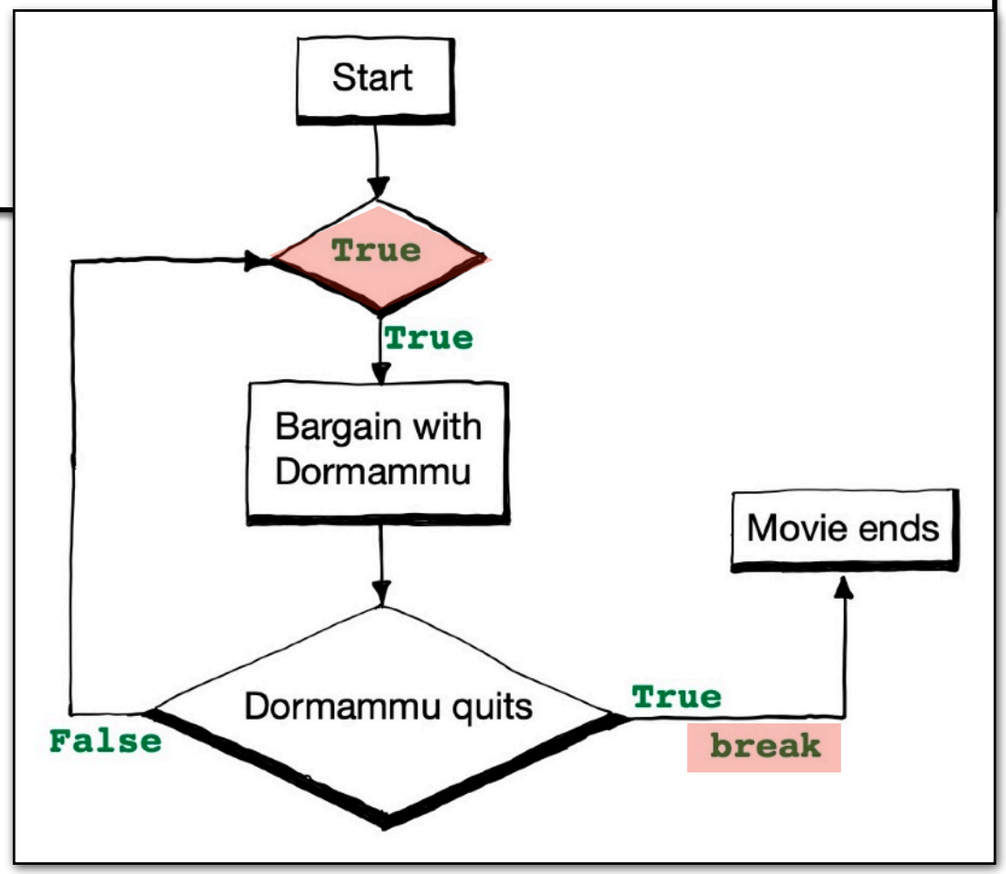
> **Example 4:** In the movie Dr. Strange (2016), Steven Strange beat Dormammu by keeping annoying him until he was extremely bored and frustrated. Write a program to act like Dr. Strange. The "time" loop can only be broken when you type in "I quit" or "You win".



Loops and Iterations

- Syntax rules of `while` loops

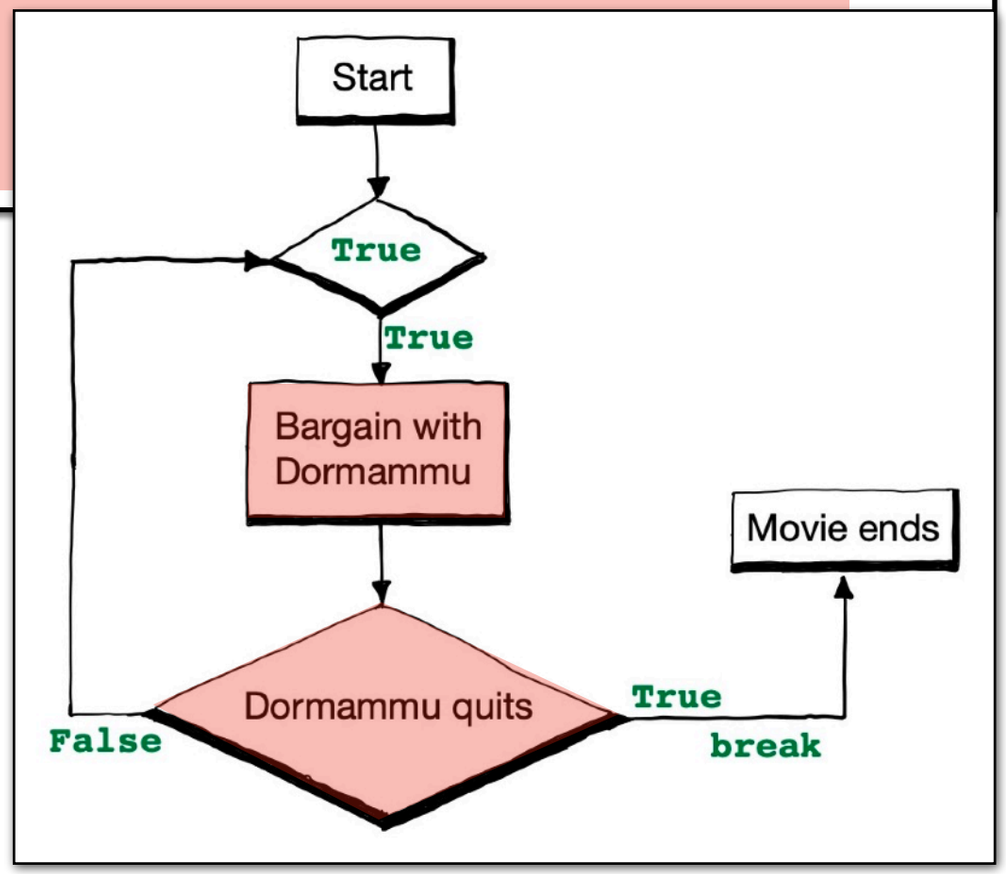
```
while True:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        print('Dr. Strange: Wise choice bro!')
        break
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```



Loops and Iterations

- Syntax rules of `while` loops

```
while True:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        print('Dr. Strange: Wise choice bro!')
        break
    else:
        print("Dr. Strange: Ah~~~")
print('\n')
```



Loops and Iterations

- Syntax rules of `while` loops

```
while True:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        print('Dr. Strange: Wise choice bro!')
        break
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

Loops and Iterations

- Syntax rules of `while` loops

```
while True:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        print('Dr. Strange: Wise choice bro!')
        break
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

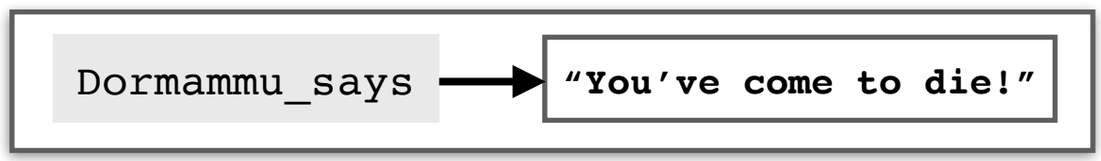
```
Dr. Strange: Dormammu, I've come to bargain!
```

Loops and Iterations

- Syntax rules of `while` loops

```
while True:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        print('Dr. Strange: Wise choice bro!')
        break
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
```

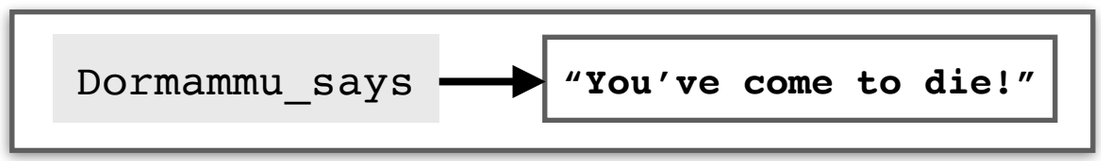


Loops and Iterations

- Syntax rules of `while` loops

```
while True:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        print('Dr. Strange: Wise choice bro!')
        break
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
```

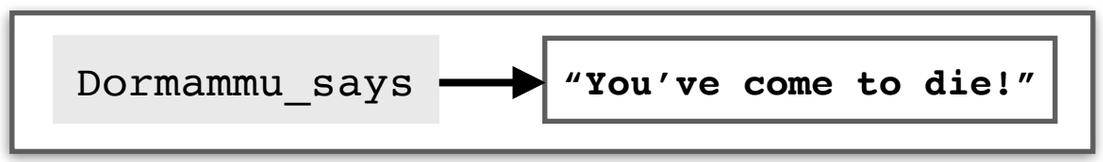


Loops and Iterations

- Syntax rules of `while` loops

```
while True:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        print('Dr. Strange: Wise choice bro!')
        break
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~
```

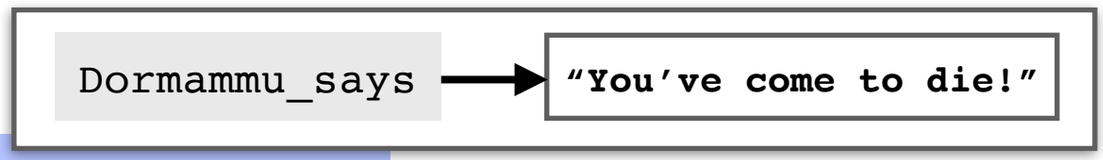


Loops and Iterations

- Syntax rules of `while` loops

```
while True:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        print('Dr. Strange: Wise choice bro!')
        break
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~
```

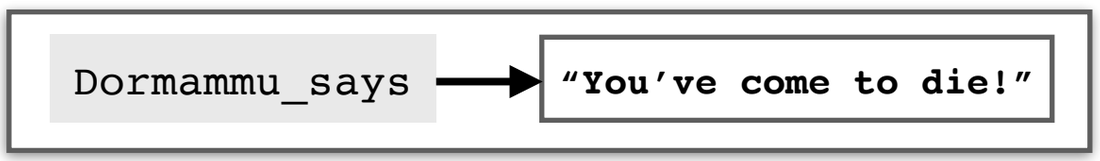


Loops and Iterations

- Syntax rules of `while` loops

```
while True:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        print('Dr. Strange: Wise choice bro!')
        break
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~
```

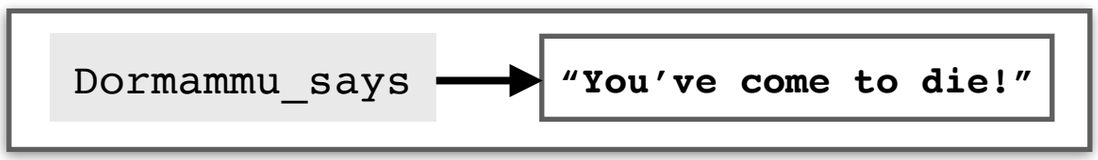


Loops and Iterations

- Syntax rules of `while` loops

```
while True:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        print('Dr. Strange: Wise choice bro!')
        break
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~
```



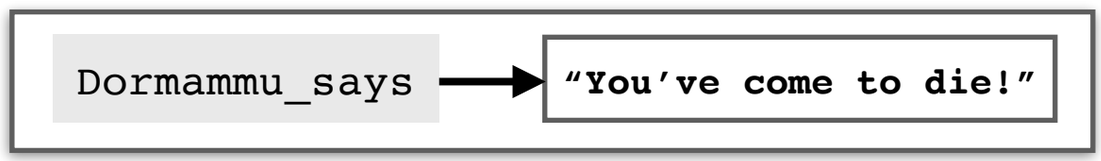

```
Dr. Strange: Dormammu, I've come to bargain!
```

Loops and Iterations

- Syntax rules of `while` loops

```
while True:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        print('Dr. Strange: Wise choice bro!')
        break
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~
```




```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: I quit
```

Loops and Iterations

- Syntax rules of `while` loops

```
while True:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        print('Dr. Strange: Wise choice bro!')
        break
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~
```


```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: I quit
```

Loops and Iterations

- Syntax rules of `while` loops

```
while True:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        print('Dr. Strange: Wise choice bro!')
        break
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~
```


```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: I quit
```

Loops and Iterations

- Syntax rules of `while` loops

```
while True:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        print('Dr. Strange: Wise choice bro!')
        break
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~
```


```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: I quit
Dr. Strange: Wise choice bro!
```

Loops and Iterations

- Syntax rules of `while` loops

```
while True:
    print("Dr. Strange: Dormammu, I've come to bargain!")
    Dormammu_says = input("Dormammu: ")
    if Dormammu_says == 'I quit' or Dormammu_says == 'You win':
        print('Dr. Strange: Wise choice bro!')
        break
    else:
        print("Dr. Strange: Ah~~~")
    print('\n')
```

Break out the loop

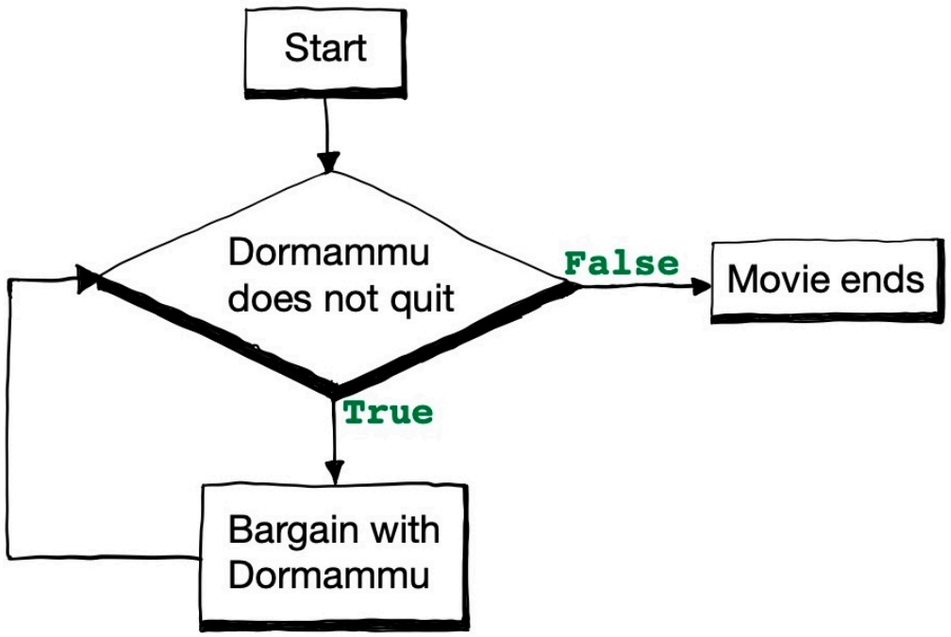
```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: You've come to die!
Dr. Strange: Ah~~~
```

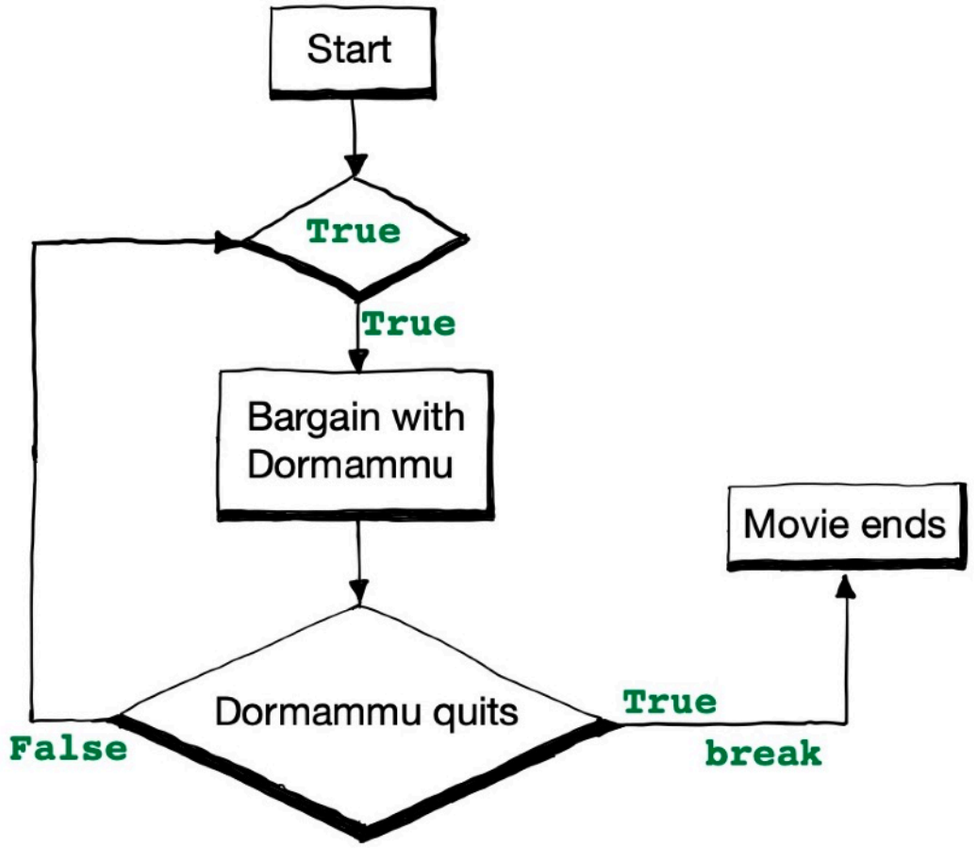

```
Dr. Strange: Dormammu, I've come to bargain!
Dormammu: I quit
Dr. Strange: Wise choice bro!
```

Loops and Iterations

- Syntax rules of `while` loops

> **Notes:** A loop can be broken by using the keyword **`break`**, or by enforcing the boolean type condition to be **`False`**. If neither method stops the loop, the program is trapped in an infinite loop.





Loops and Iterations

- Syntax rules of `while` loops

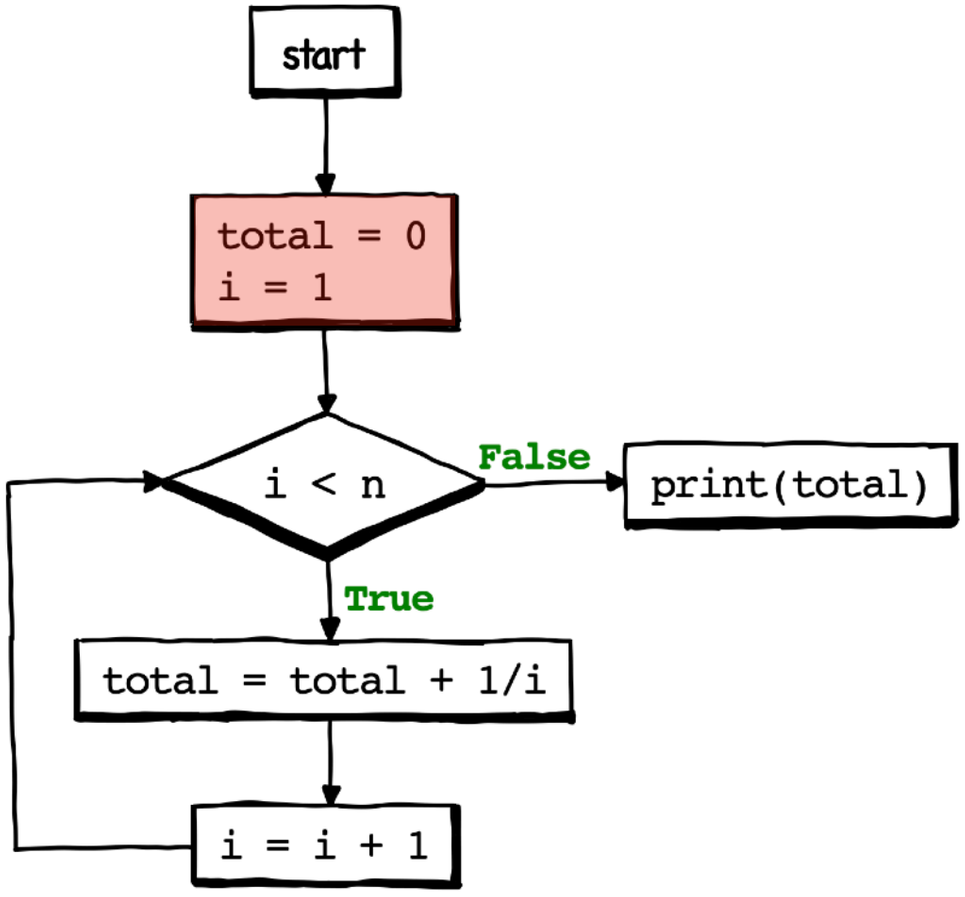
> **Example 5:** PUBG is an online game where n players are killing each others and the final survivor is the winner. If each player is equally good, we can prove that the expected number of kills made by the winner is $1 + 1/2 + 1/3 + \ldots + 1/(n-1)$. Given the number of players, write a program to calculate this expected number of kills.

Loops and Iterations

- Syntax rules of `while` loops

> **Example 5:** PUBG is an online game where n players are killing each others and the final survivor is the winner. If each player is equally good, we can prove that the expected number of kills made by the winner is $1 + 1/2 + 1/3 + \ldots + 1/(n-1)$. Given the number of players, write a program to calculate this expected number of kills.

```
total = 0
i = 1

while i < n:
    total += 1/i
    i += 1

print(total)
```

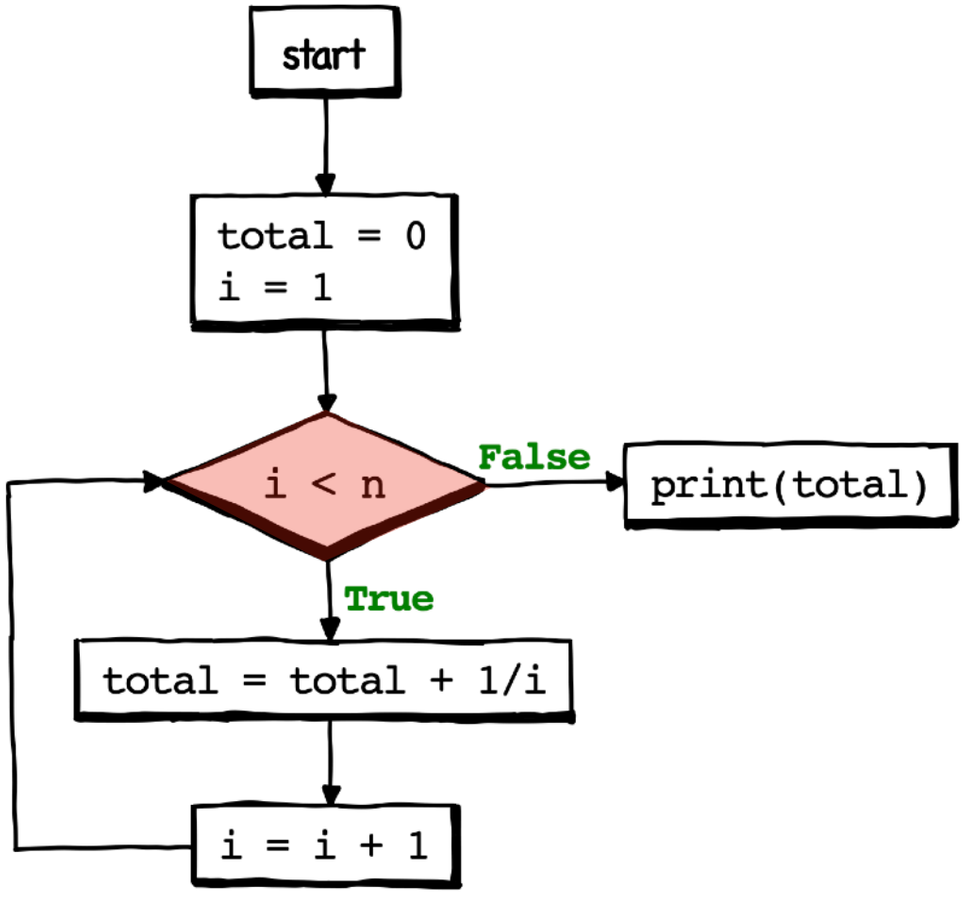


Loops and Iterations

- Syntax rules of `while` loops

> **Example 5:** PUBG is an online game where n players are killing each others and the final survivor is the winner. If each player is equally good, we can prove that the expected number of kills made by the winner is $1 + 1/2 + 1/3 + \ldots + 1/(n-1)$. Given the number of players, write a program to calculate this expected number of kills.

```
total = 0
i = 1

while i < n:
    total += 1/i
    i += 1

print(total)
```

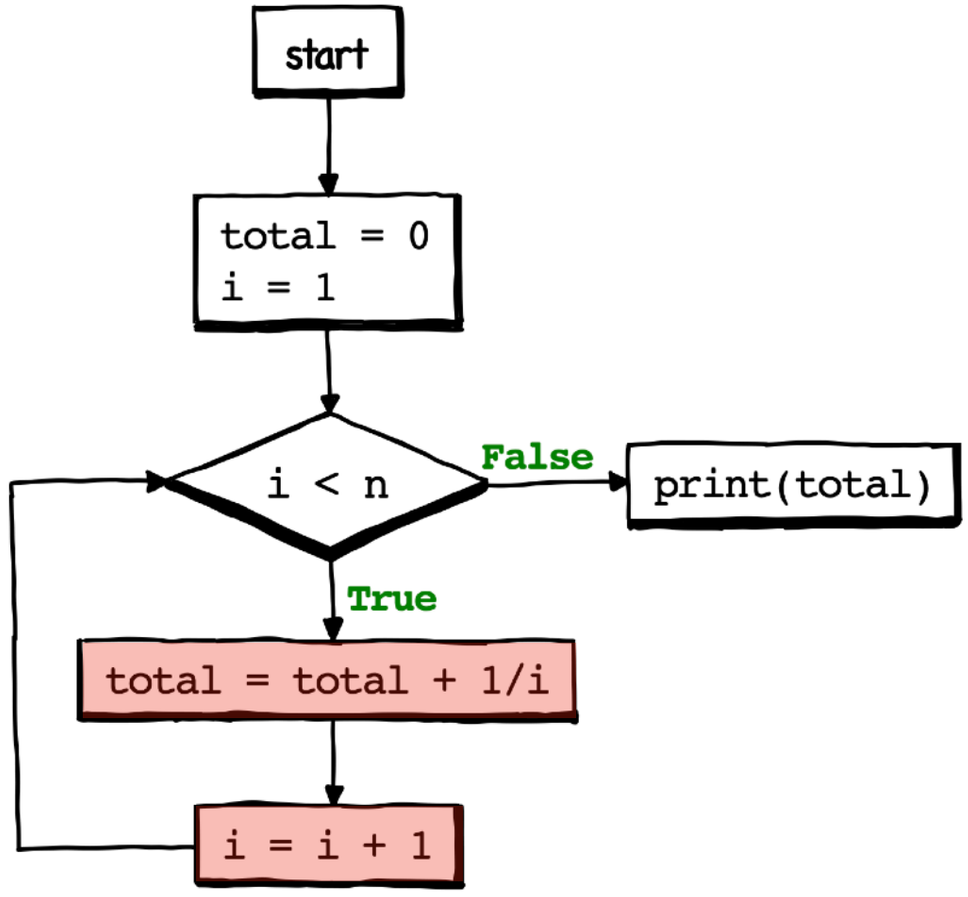


Loops and Iterations

- Syntax rules of `while` loops

> **Example 5:** PUBG is an online game where n players are killing each others and the final survivor is the winner. If each player is equally good, we can prove that the expected number of kills made by the winner is $1 + 1/2 + 1/3 + \ldots + 1/(n-1)$. Given the number of players, write a program to calculate this expected number of kills.

```
total = 0
i = 1

while i < n:
    total += 1/i
    i += 1

print(total)
```

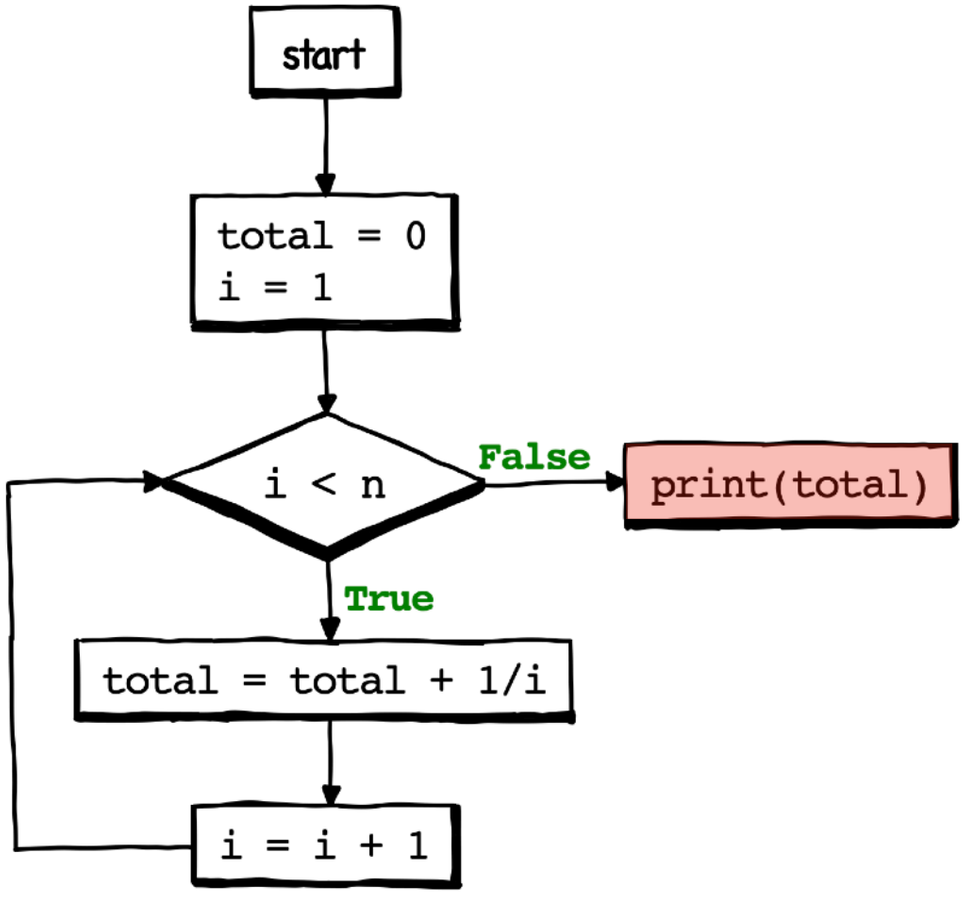


Loops and Iterations

- Syntax rules of `while` loops

> **Example 5:** PUBG is an online game where n players are killing each others and the final survivor is the winner. If each player is equally good, we can prove that the expected number of kills made by the winner is $1 + 1/2 + 1/3 + \ldots + 1/(n-1)$. Given the number of players, write a program to calculate this expected number of kills.

```
total = 0
i = 1

while i < n:
    total += 1/i
    i += 1

print(total)
```

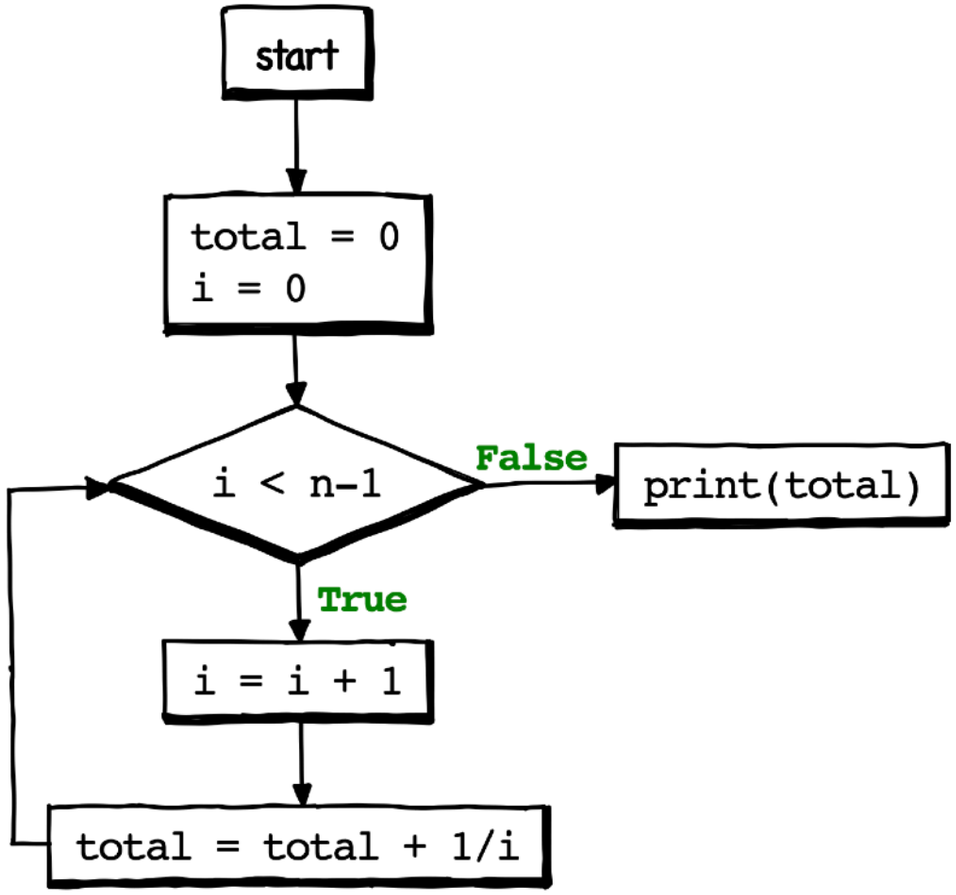


Loops and Iterations

- Syntax rules of `while` loops

Example 5: PUBG is an online game where n players are killing each others and the final survivor is the winner. If each player is equally good, we can prove that the expected number of kills made by the winner is $1 + 1/2 + 1/3 + \ldots + 1/(n-1)$. Given the number of players, write a program to calculate this expected number of kills.

```
total = 0
i = 0

while i < n - 1:
    i += 1
    total += 1/i

print(total)
```



Loops and Iterations

- Syntax rules of `while` loops

> **Question 2:** As the developer of the PUBG game, you want to make sure that the expected number of kills made by the winner of the game is at least four, assuming all players are equally good. Let n be the total number of players in one game, write a program to calculate the minimum value of n.

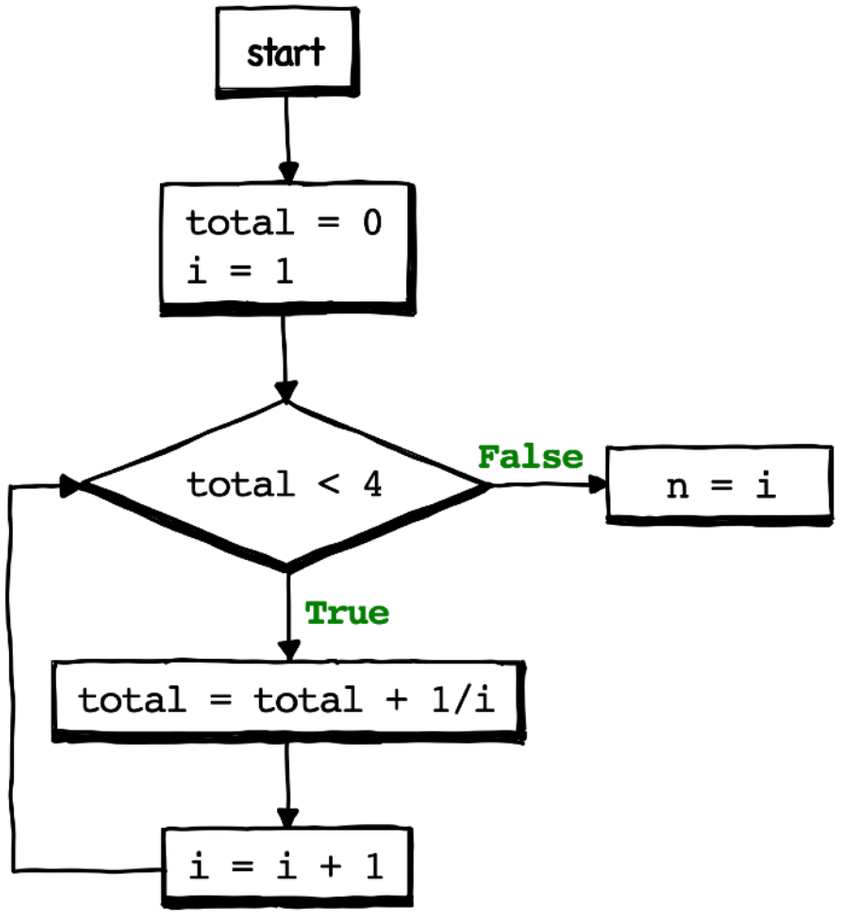
Loops and Iterations

- Syntax rules of `while` loops

> **Question 2:** As the developer of the PUBG game, you want to make sure that the expected number of kills made by the winner of the game is at least four, assuming all players are equally good. Let n be the total number of players in one game, write a program to calculate the minimum value of n.

```
total = 0
i = 1

while total < 4:
    total += 1/i
    i += 1

n = i
print(n)
```

```
32
```



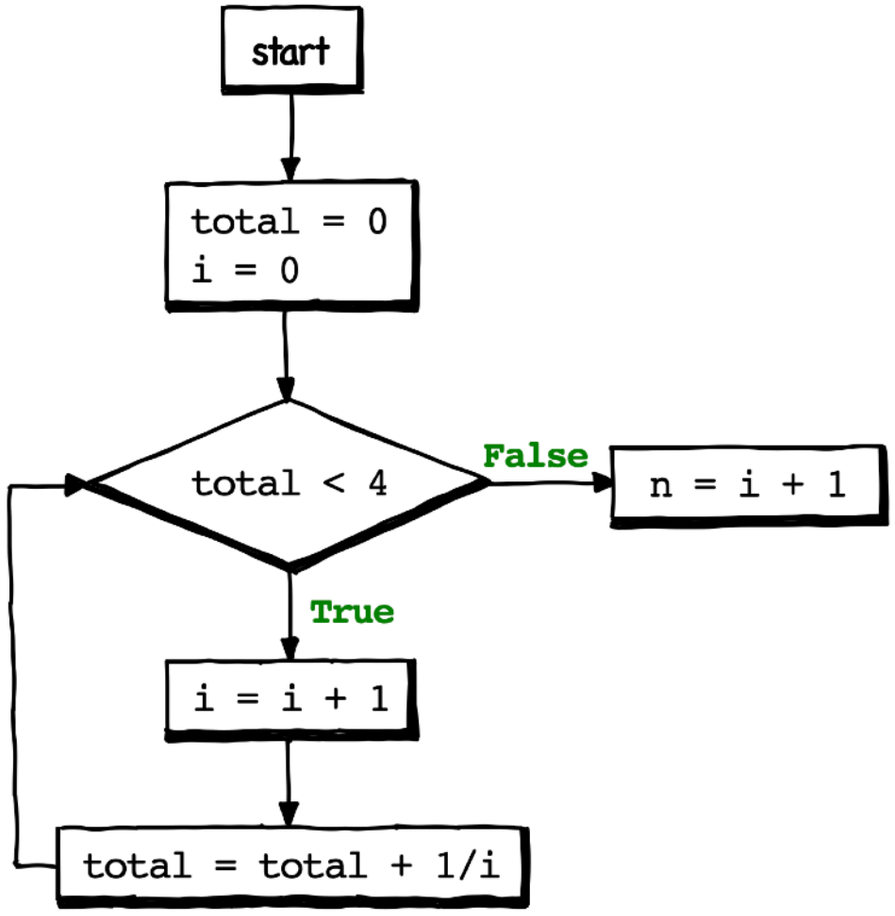
Loops and Iterations

- Syntax rules of `while` loops

> **Question 3:** As the developer of the PUBG game, you want to make sure that the expected number of kills made by the winner of the game is at least four, assuming all players are equally good. Let n be the total number of players in one game, write a program to calculate the minimum value of n.

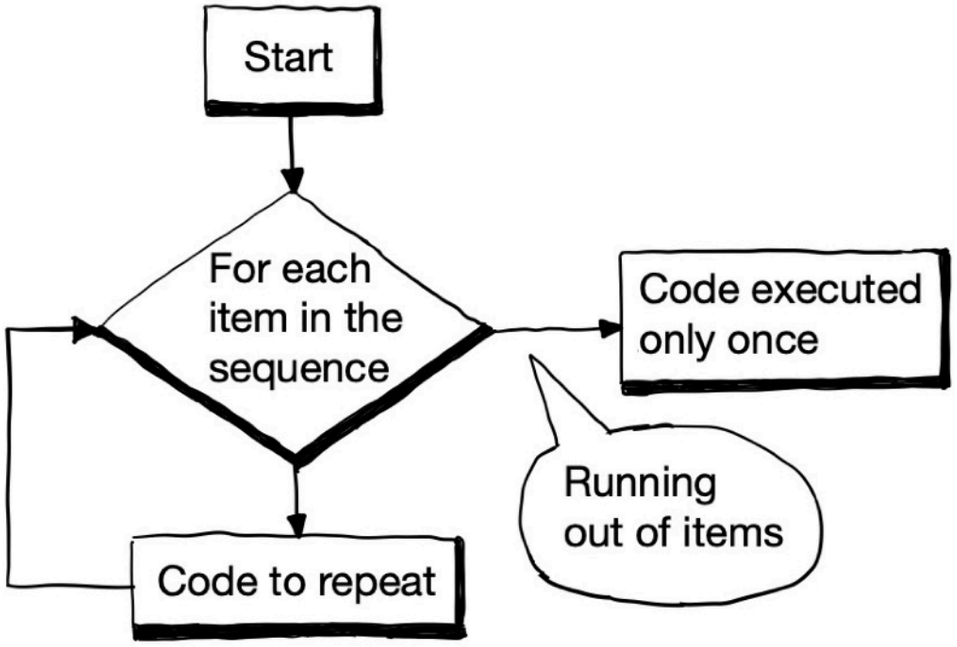
```
total = 0
i = 0

while total < 4:
    i += 1
    total += 1/i

n = i + 1
print(n)
```

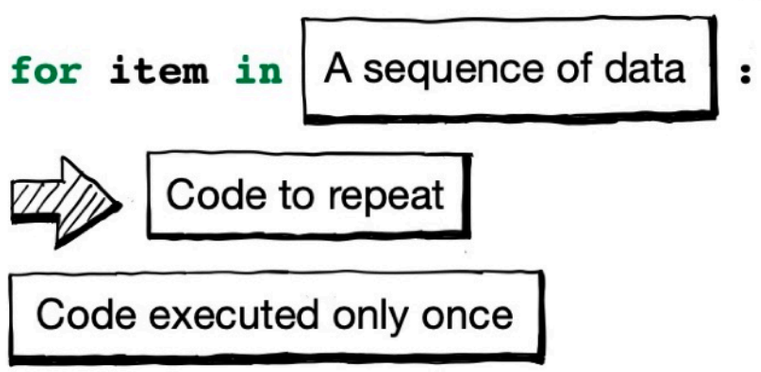
```
32
```



Loops and Iterations

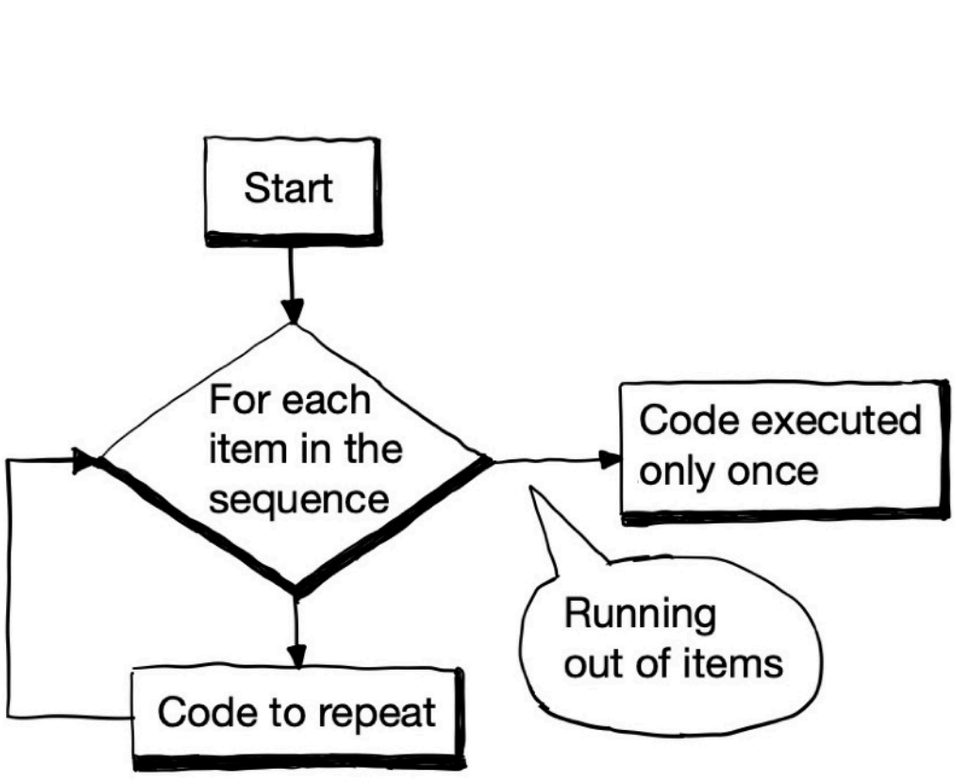
- Syntax rules of `for` loops

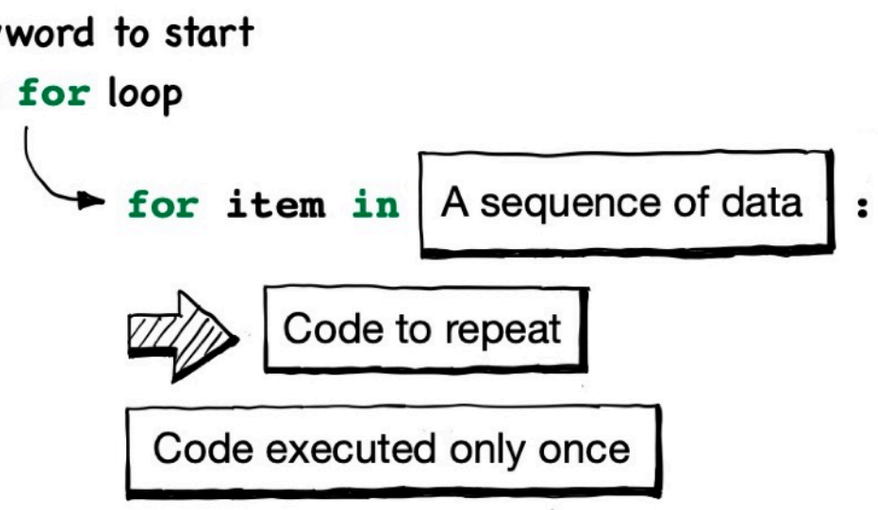




Loops and Iterations

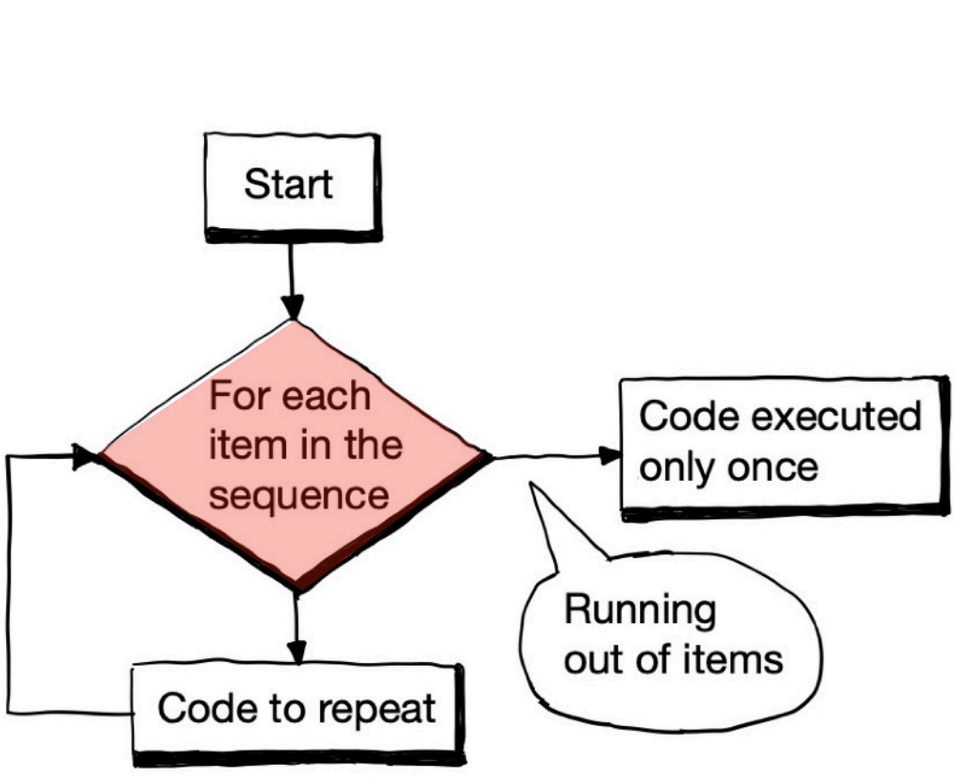
- Syntax rules of `for` loops
  - The loop is started by the keyword `for`

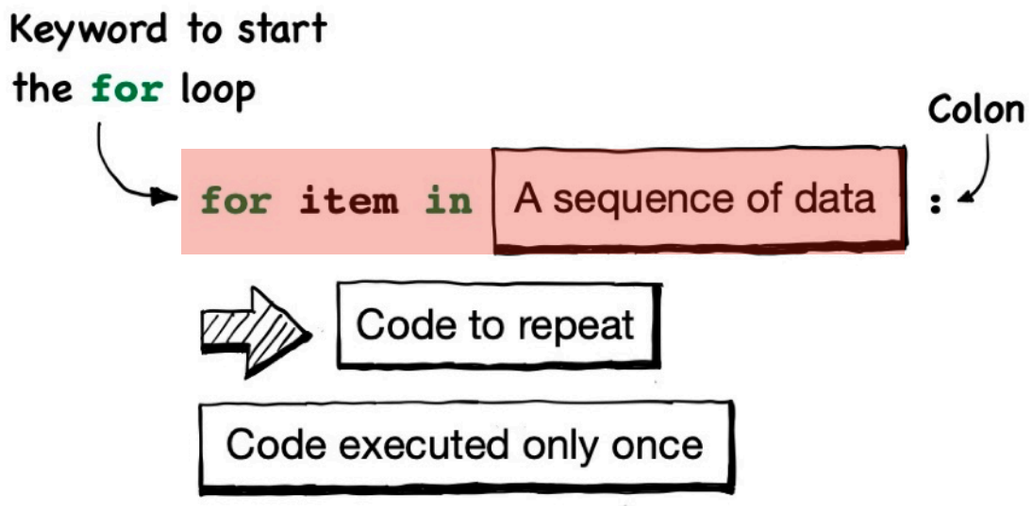




Loops and Iterations

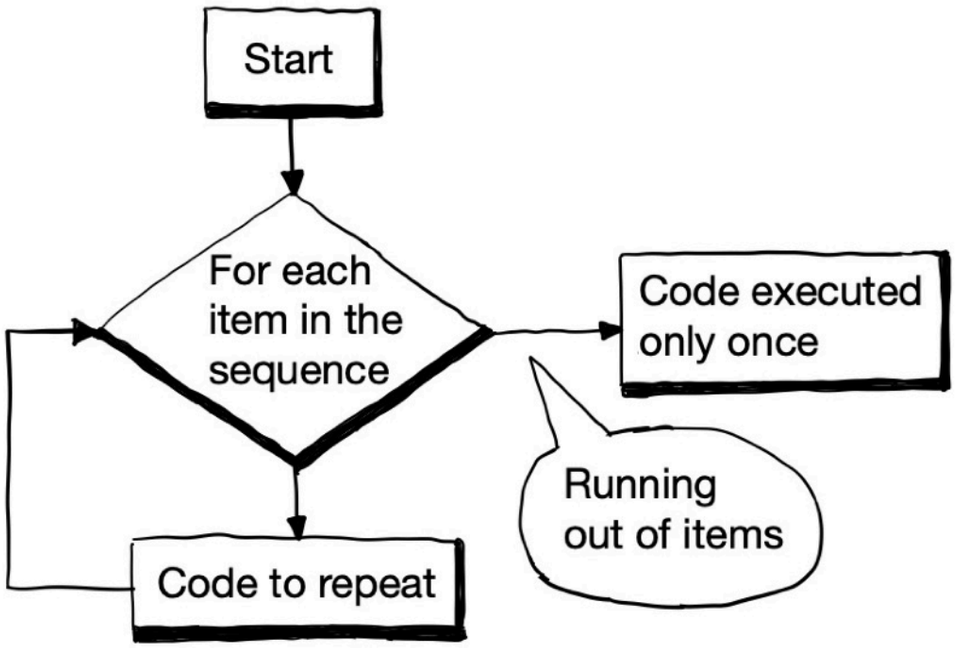
- Syntax rules of `for` loops
  - The keyword `for` is followed by `item in` a sequence of data, and a colon

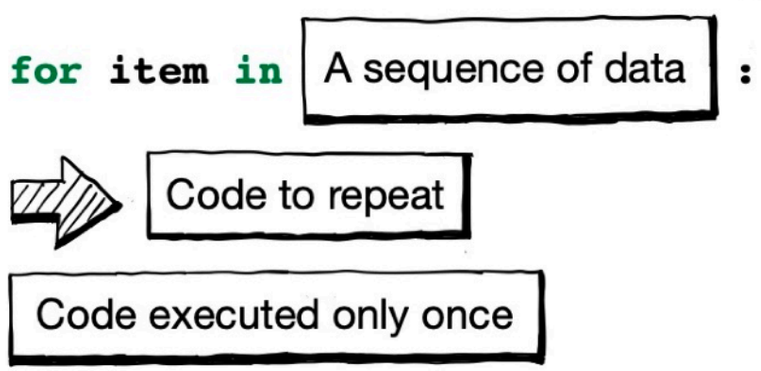




Loops and Iterations

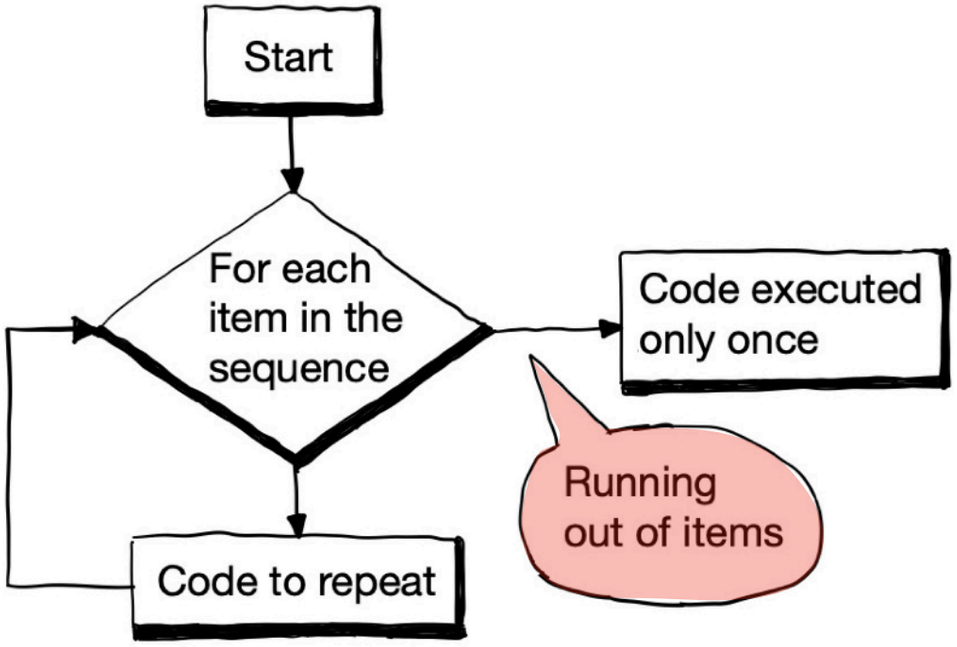
- Syntax rules of `for` loops
  - Here `item` can be any valid variable name, and it takes the value of each item in the data sequence in every iteration

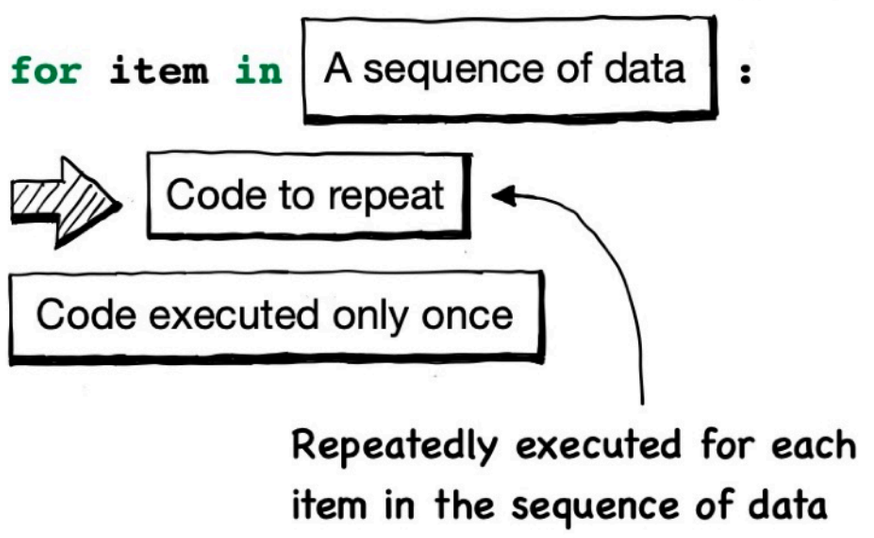




Loops and Iterations

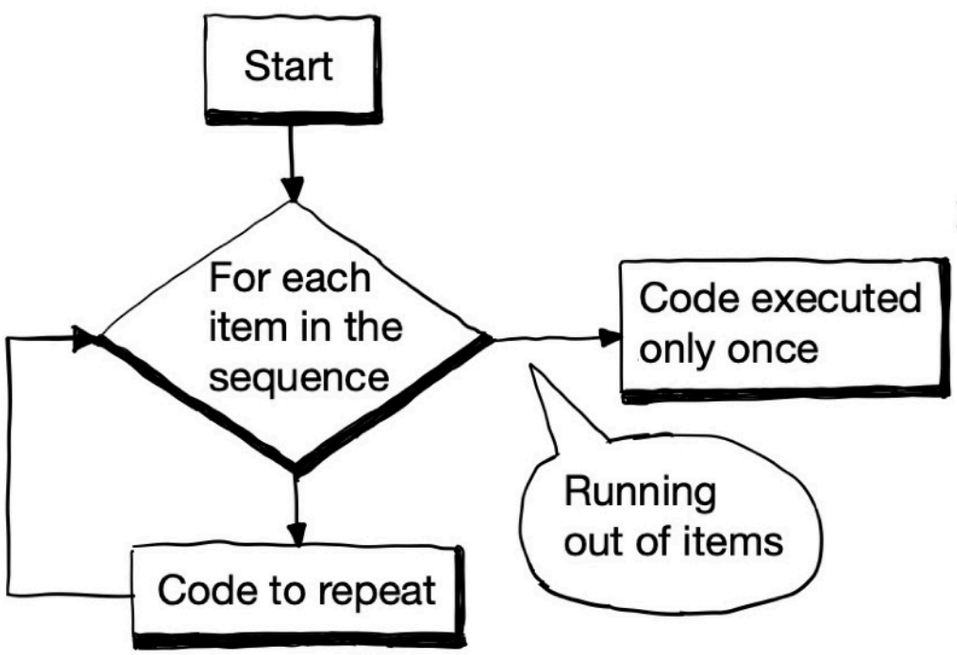
- Syntax rules of `for` loops
  - **Code to repeat** is repeatedly executed, until the program is running out of items in the sequence

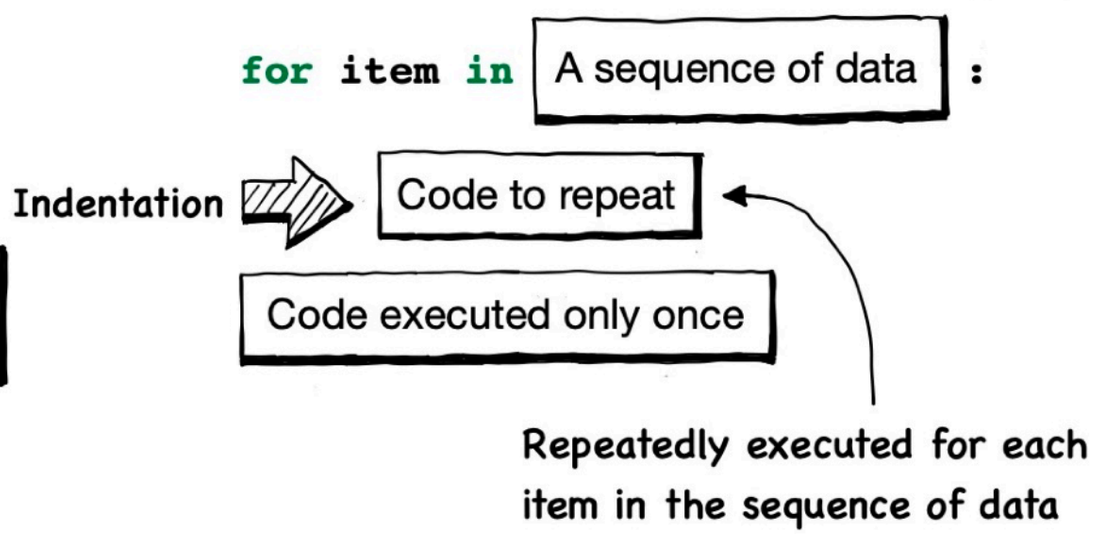




Loops and Iterations

- Syntax rules of `for` loops
  - Same indentation as the `while` loop





Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string

```
string = 'Jack'

for char in string:      # Iterate each character of string
    print(char)          # Print the character in each iteration
```

```
J
a
c
k
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string

```
string = 'Jack'

for char in string:      # Iterate each character of string
    print(char)          # Print the character in each iteration
```

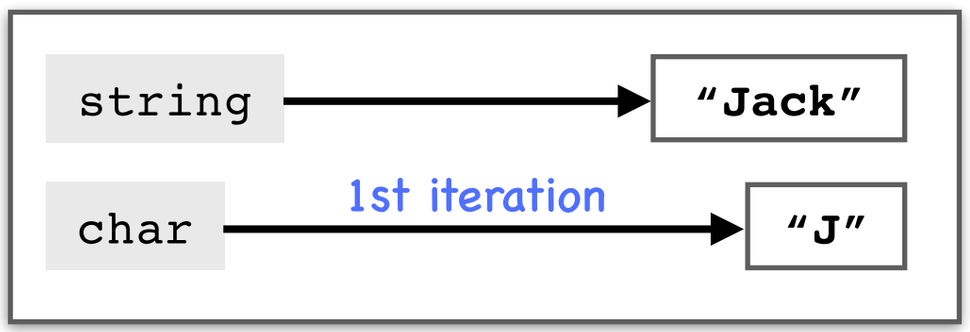
```
J
a
c
k
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string

```
string = 'Jack'

for char in string:      # Iterate each character of string
    print(char)          # Print the character in each iteration
```

```
J
a
c
k
```

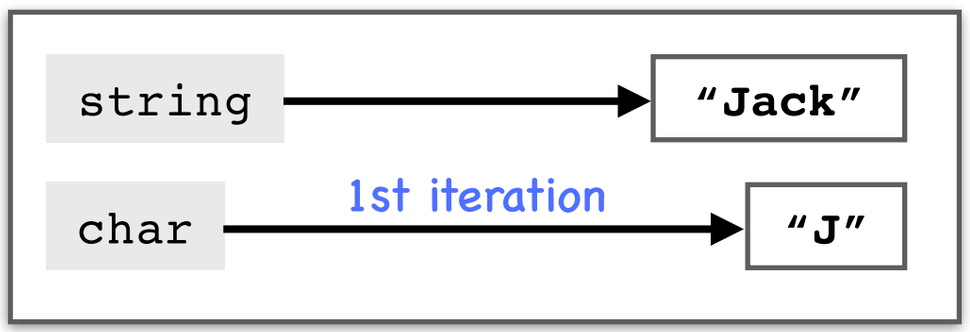


Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string

```
string = 'Jack'

for char in string:      # Iterate each character of string
    print(char)          # Print the character in each iteration
```

```
J
a
c
k
```

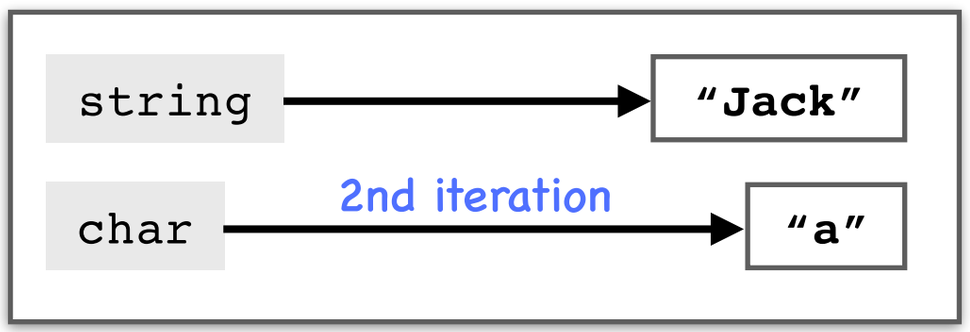


Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string

```
string = 'Jack'

for char in string:      # Iterate each character of string
    print(char)          # Print the character in each iteration
```

```
J
a
c
k
```

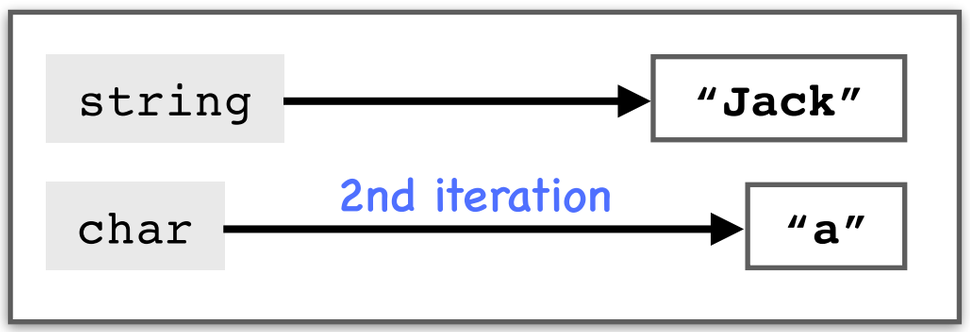


Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string

```
string = 'Jack'

for char in string:      # Iterate each character of string
    print(char)          # Print the character in each iteration
```

```
J
a
c
k
```

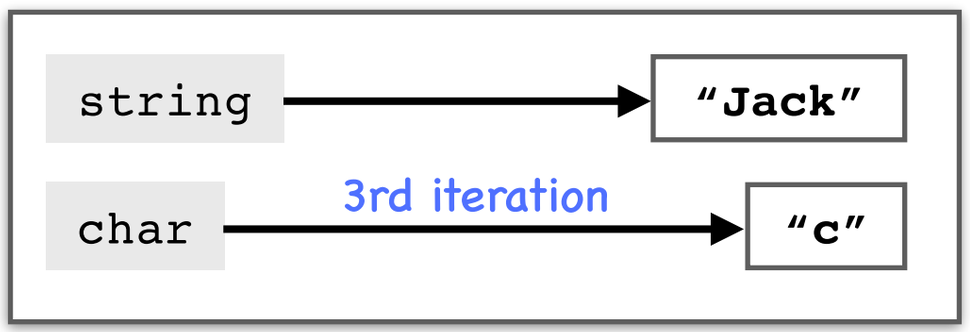


Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string

```
string = 'Jack'

for char in string:      # Iterate each character of string
    print(char)          # Print the character in each iteration
```

```
J
a
c
k
```

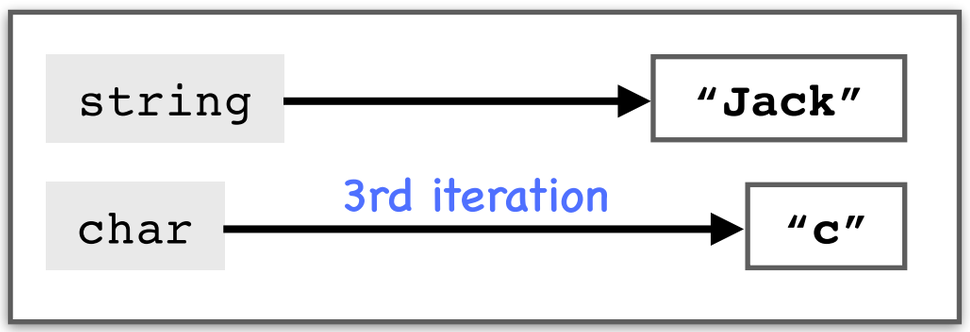


Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string

```
string = 'Jack'

for char in string:      # Iterate each character of string
    print(char)          # Print the character in each iteration
```

```
J
a
c
k
```

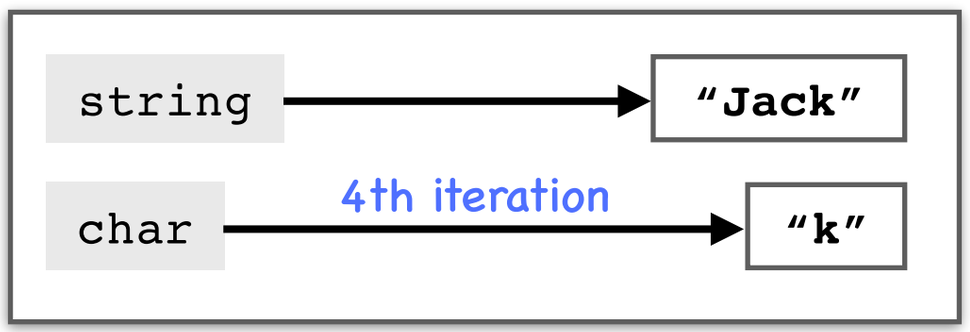


Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string

```
string = 'Jack'

for char in string:      # Iterate each character of string
    print(char)          # Print the character in each iteration
```

```
J
a
c
k
```

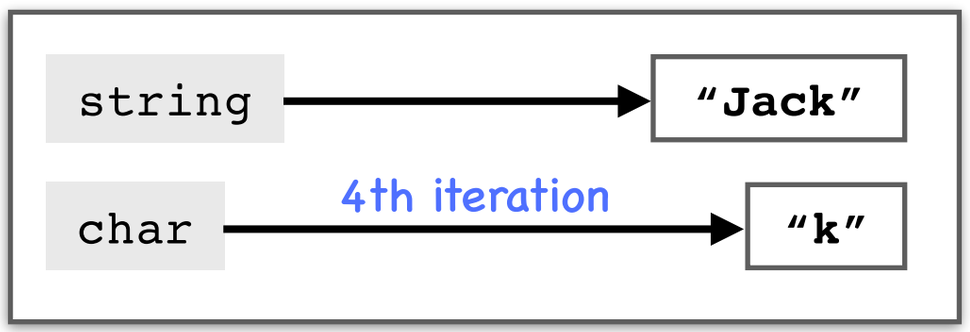


Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string

```
string = 'Jack'

for char in string:      # Iterate each character of string
    print(char)          # Print the character in each iteration
```

```
J
a
c
k
```



Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string

Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
Give me a a!
a!!!
Give me a c!
c!!!
Give me a k!
k!!!
What's that spell?
Jack!!!
Go! Go! Jack!!!
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string

Example 6: Cheerleader chant

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string

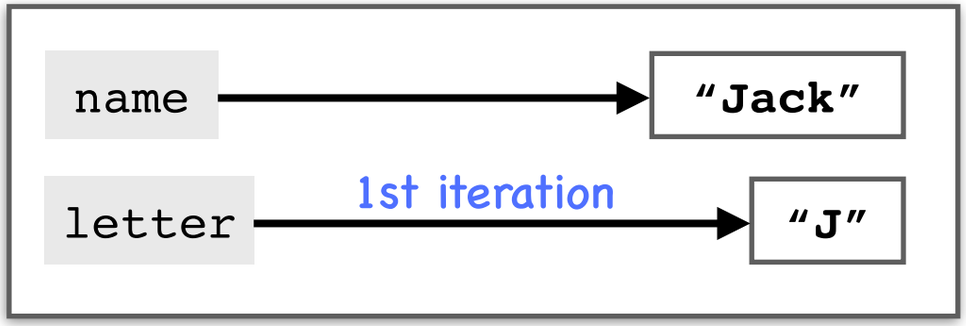
Example 6: Cheerleader chant

```
What is your name? Jack
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



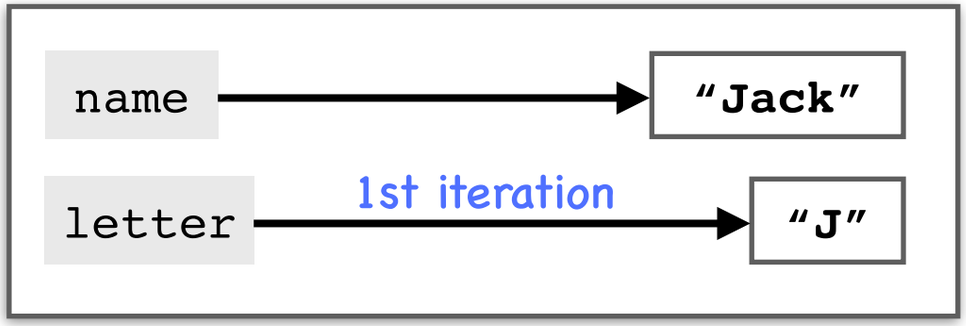
Example 6: Cheerleader chant

```
What is your name? Jack
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



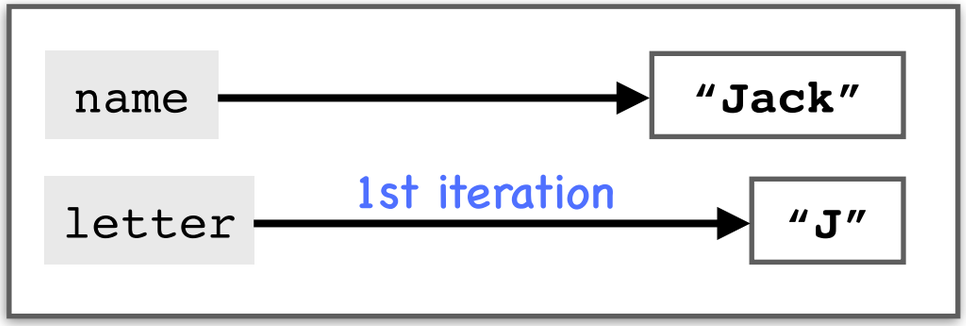
Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



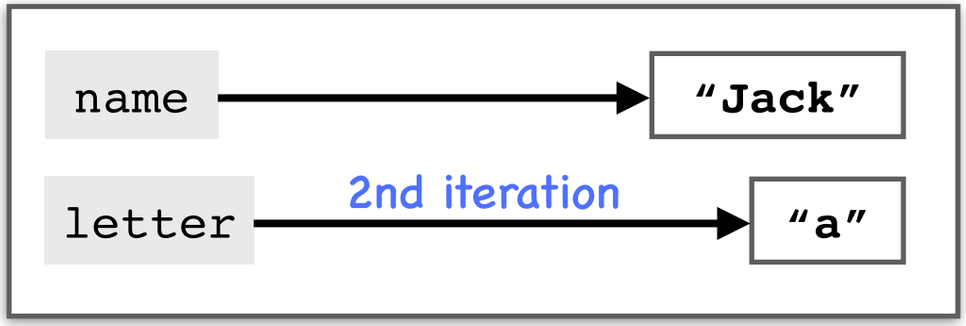
Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



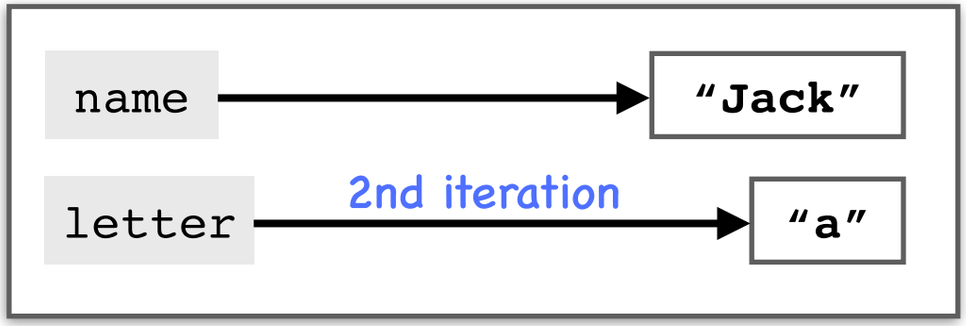
Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



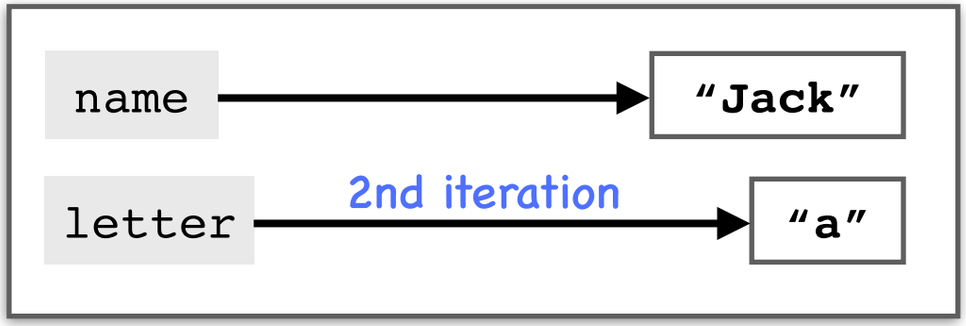
Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
Give me a a!
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



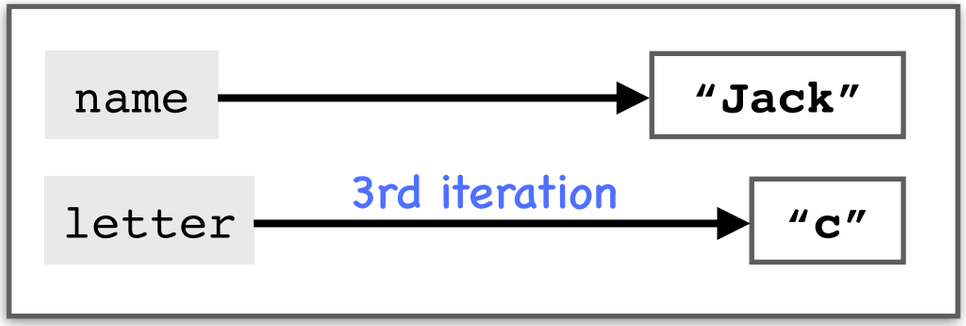
Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
Give me a a!
a!!!
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



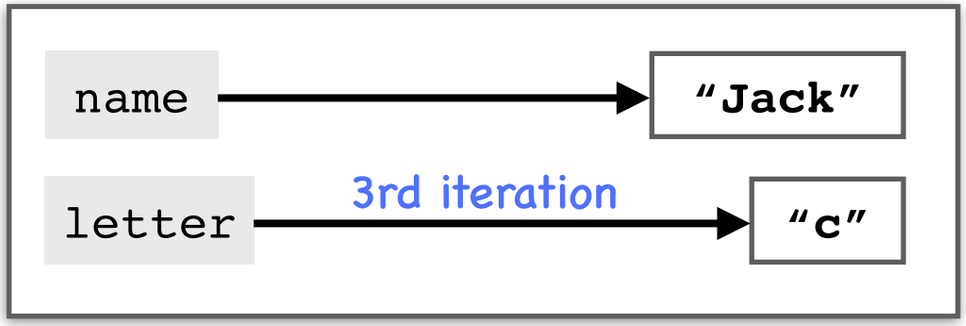
Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
Give me a a!
a!!!
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



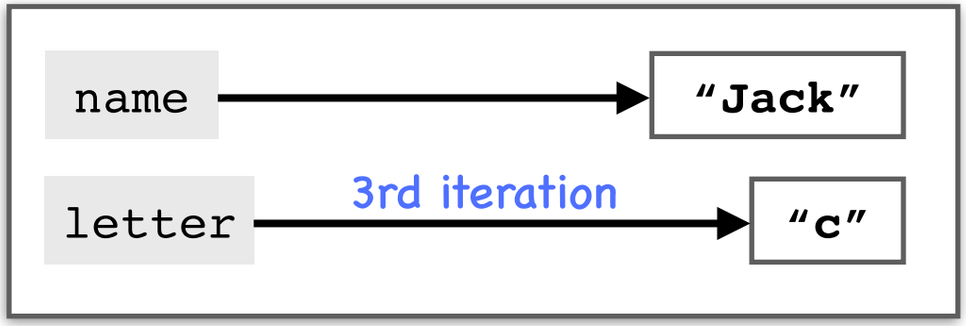
Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
Give me a a!
a!!!
Give me a c!
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



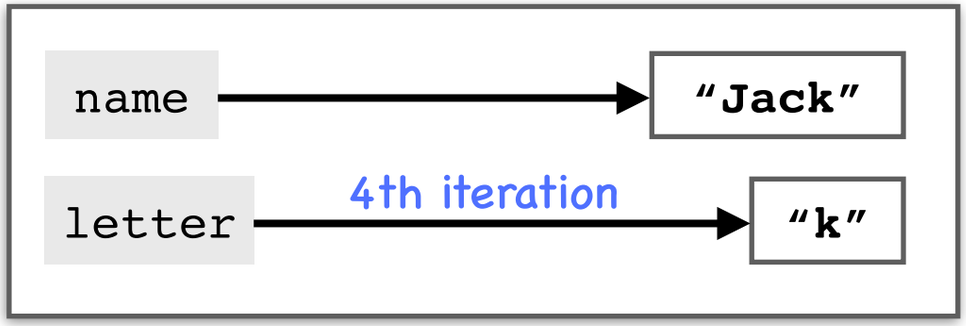
Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
Give me a a!
a!!!
Give me a c!
c!!!
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



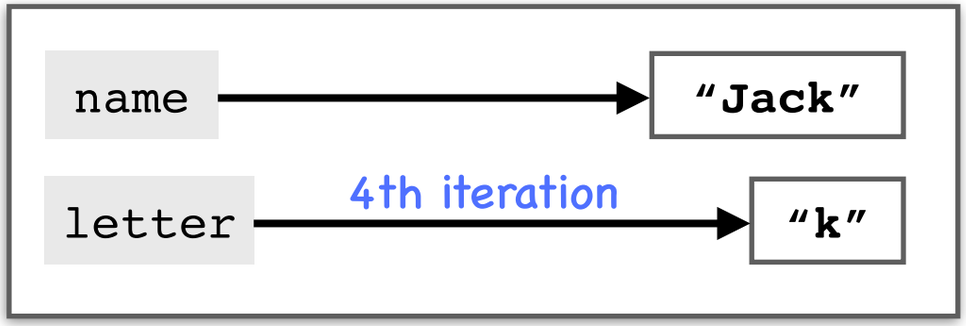
Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
Give me a a!
a!!!
Give me a c!
c!!!
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



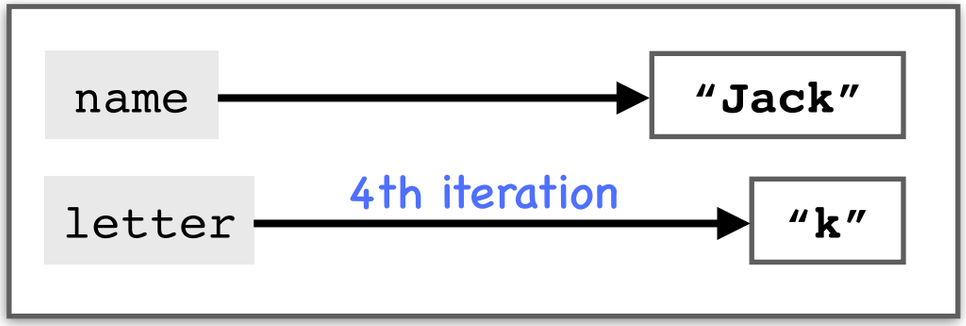
Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
Give me a a!
a!!!
Give me a c!
c!!!
Give me a k!
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



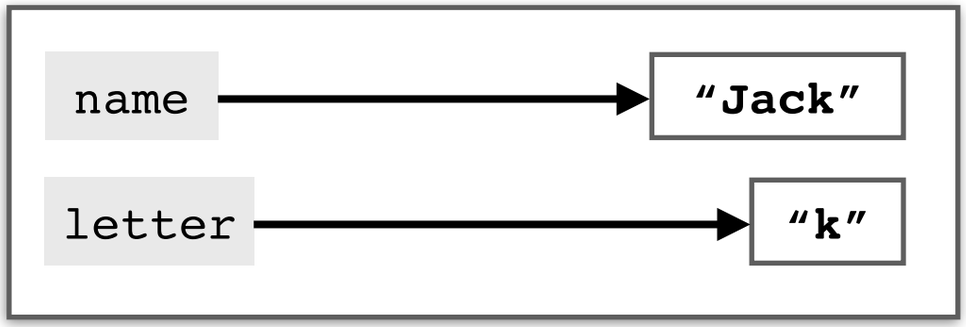
Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
Give me a a!
a!!!
Give me a c!
c!!!
Give me a k!
k!!!
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



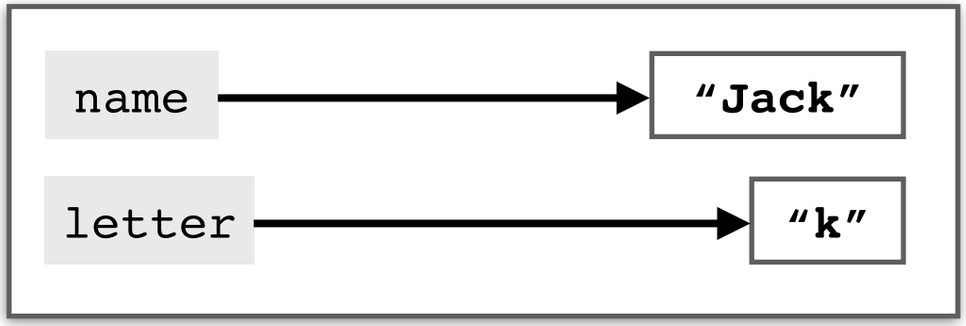
Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
Give me a a!
a!!!
Give me a c!
c!!!
Give me a k!
k!!!
What's that spell?
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



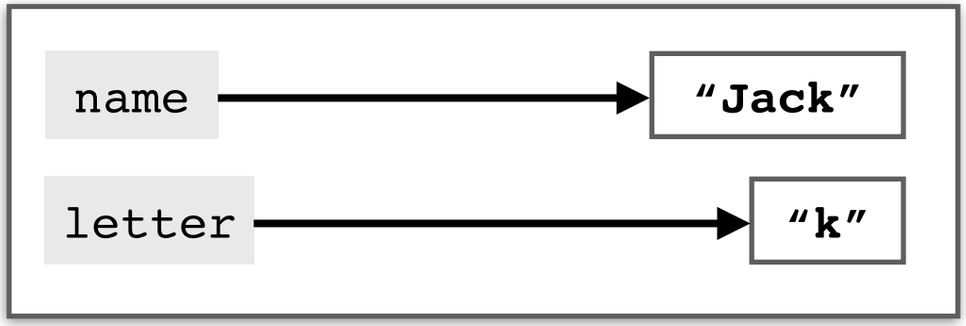
Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
Give me a a!
a!!!
Give me a c!
c!!!
Give me a k!
k!!!
What's that spell?
Jack!!!
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



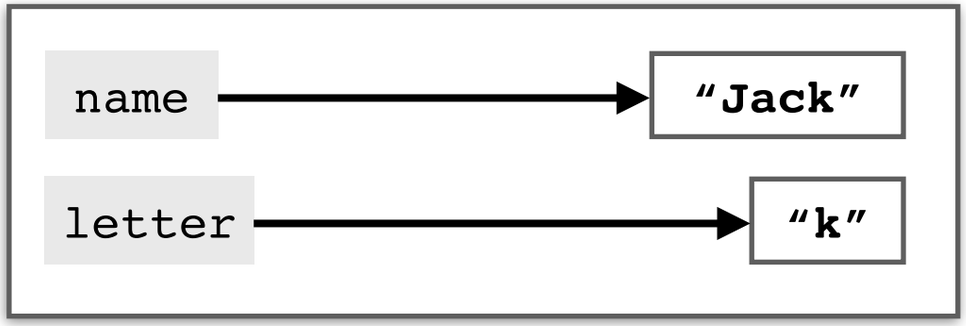
Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
Give me a a!
a!!!
Give me a c!
c!!!
Give me a k!
k!!!
What's that spell?
Jack!!!
Go! Go! Jack!!!
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



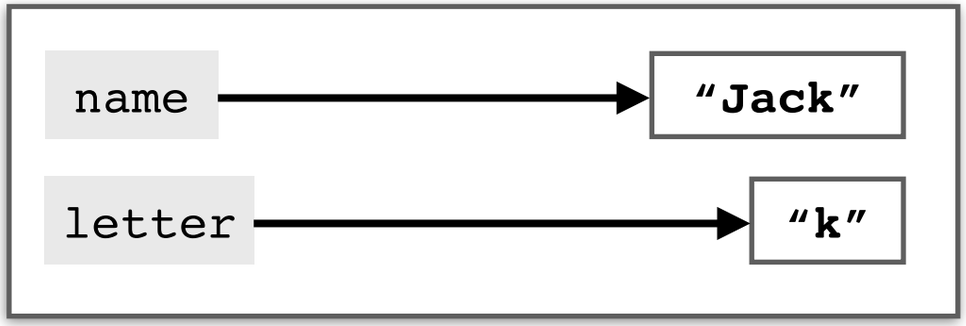
Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
Give me a a!
a!!!
Give me a c!
c!!!
Give me a k!
k!!!
What's that spell?
Jack!!!
Go! Go! Jack!!!
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string



Example 6: Cheerleader chant

```
What is your name? Jack
Give me a J!
J!!!
Give me a a!
a!!!
Give me a c!
c!!!
Give me a k!
k!!!
What's that spell?
Jack!!!
Go! Go! Jack!!!
```

```
name = input("What is your name? ")

for letter in name:
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Loops and Iterations

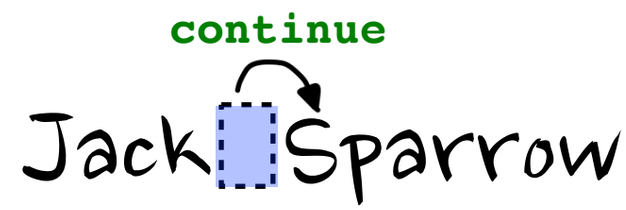
- Syntax rules of `for` loops
  - Iterating characters in a string
  - Go directly to the next iteration with `continue`

Jack Sparrow

```
...
Give me a  !
 !!!
...
```

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating characters in a string
  - Go directly to the next iteration with `continue`



```
name = input("What is your name? ")

for letter in name:
    if letter == ' ':
        continue
    print("Give me a " + letter + "!")
    print(letter + "!!!")

print("What's that spell?")
print(name + "!!!")
print("Go! Go! " + name + "!!!")
```

Go directly to the next iteration

Skip the subsequent code in the current iteration

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating integers with the `range()` function

```
range(start, stop, step)
```

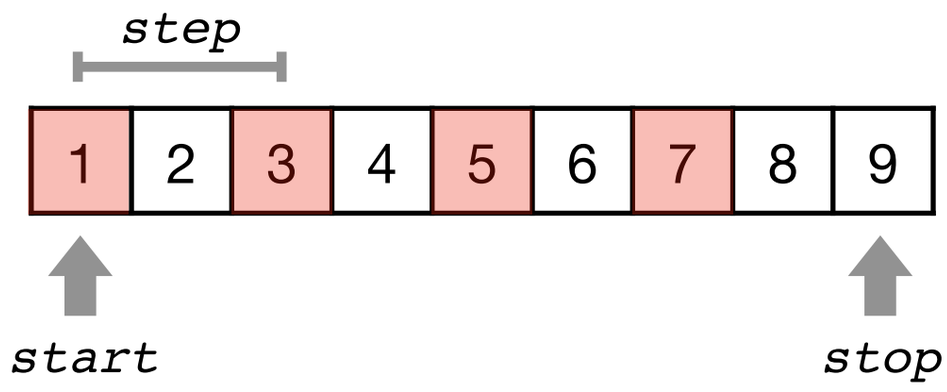
Arguments	Remarks	Default Values
start	The first integer of the number sequence	0
stop	The integer before which the sequence stops	-
step	The step length of the selected number sequence	1

Loops and Iterations

- Syntax rules of `for` loops
  - Iterating integers with the `range()` function
    - ✓ Sequence of integers created by `range(start, stop, step)`

```
for i in range(1, 9, 2):
    print(i)
```

```
1
3
5
7
```

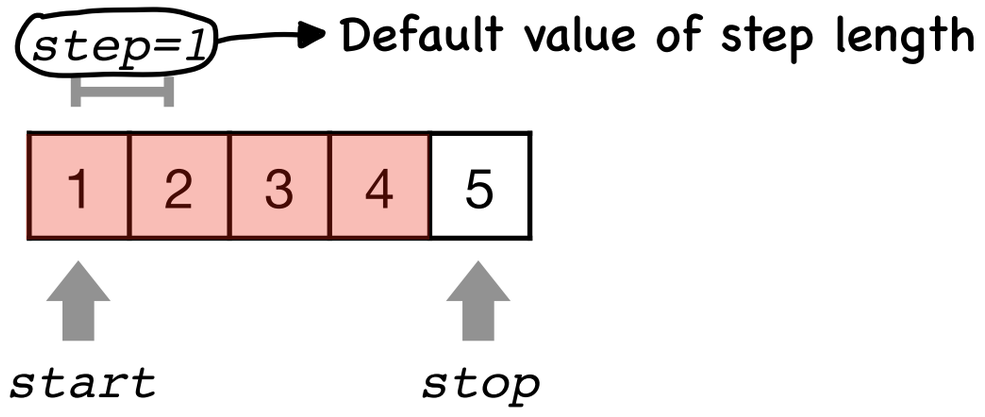


Loops and Iterations

- Syntax rules of `for` loops
  - Iterating integers with the `range()` function
    - ✓ Sequence of integers created by `range(start, stop)`

```
for i in range(1, 5):
    print(i)
```

```
1
2
3
4
```

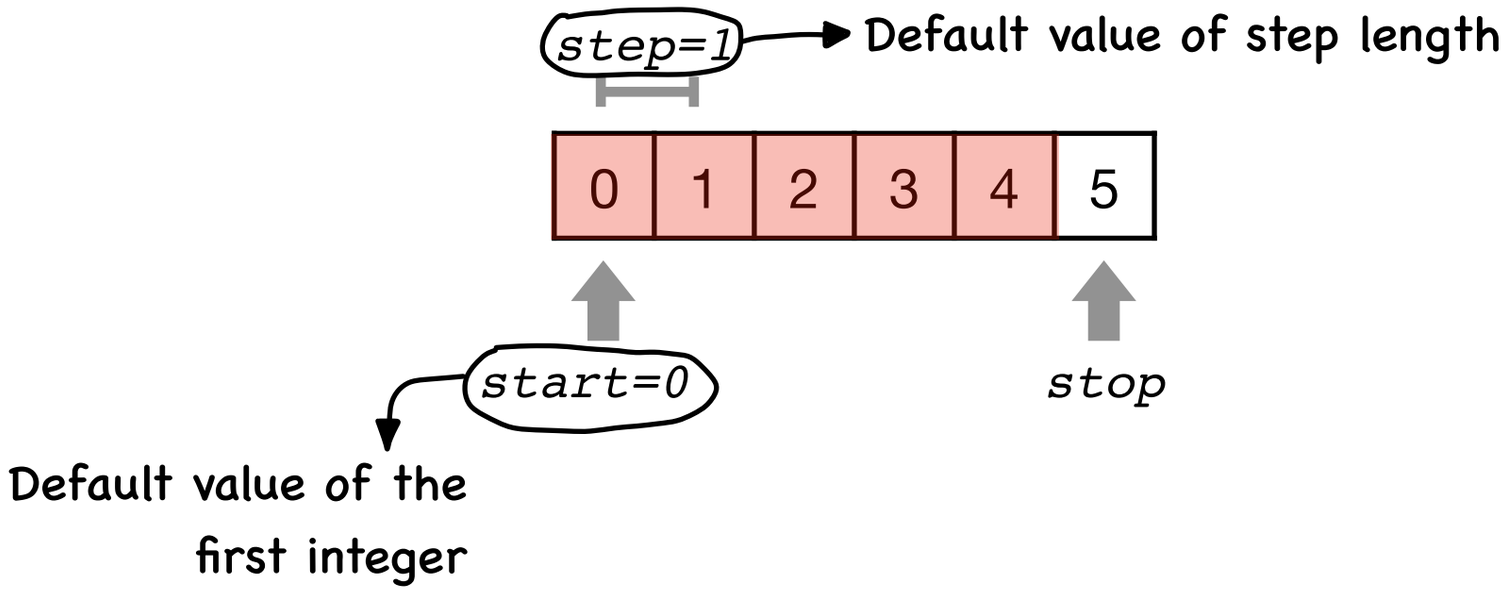


Loops and Iterations

- Syntax rules of `for` loops
  - Iterating integers with the `range()` function
    - ✓ Sequence of integers created by `range(stop)`

```
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

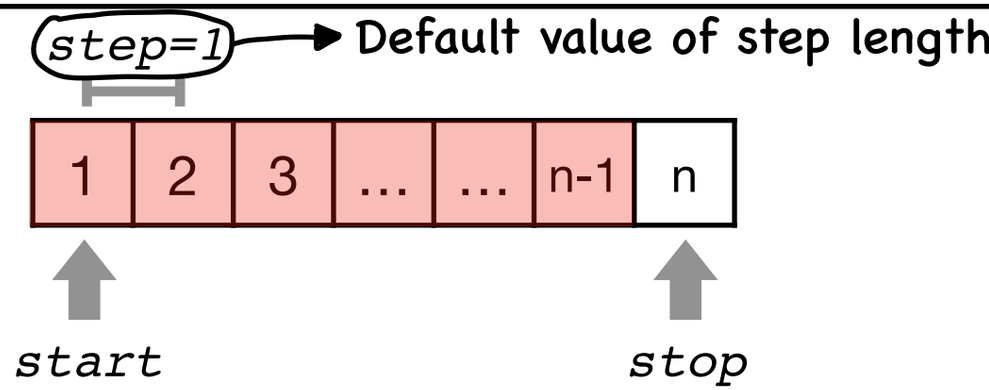


Loops and Iterations

- Syntax rules of `for` loops
  - Iterating integers with the `range()` function

Example 5: PUBG is an online game where n players are killing each others and the final survivor is the winner. If each player is equally good, we can prove that the expected number of kills made by the winner is $1 + 1/2 + 1/3 + \ldots + 1/(n-1)$.
Given the number of players, write a program to calculate this expected number of kills.

```
total = 0
for i in range(1, n):
    total += 1/i

print(total)
```



Loops and Iterations

- Comparison between `while` and `for` loops

`while` loop	`for` loop
Unknown number of iterations	Known number of iterations
The loop stops if the boolean type condition is `False`	The loop stops as the sequence is running out of items
Break the loop by `break`	Break the loop by `break`
Skip the subsequent code by `continue`	Skip the subsequent code by `continue`