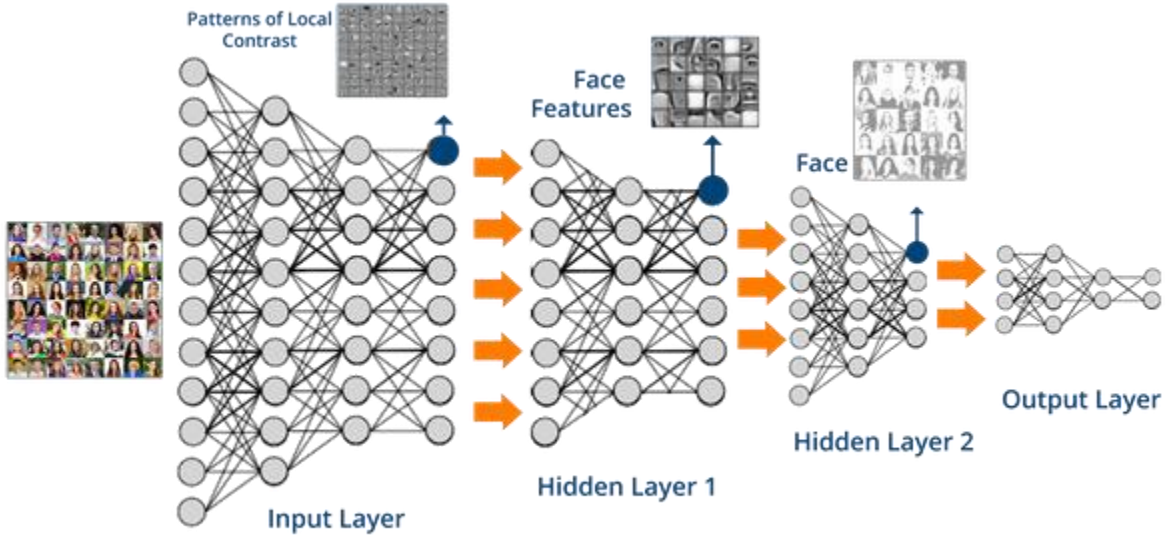
Control Structures

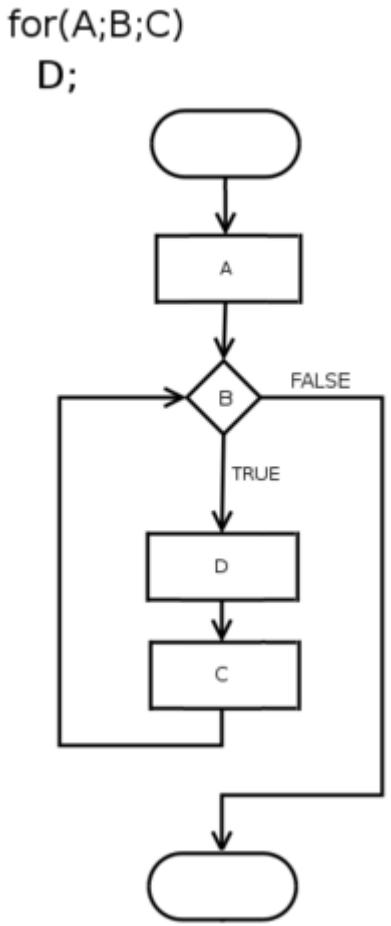
Iterative Control

Can computers Think?



Computers are as powerful as the algorithm they run!!

What is control flow?



A. Escoger medio de transporte

B. ¿Esta bloqueado?

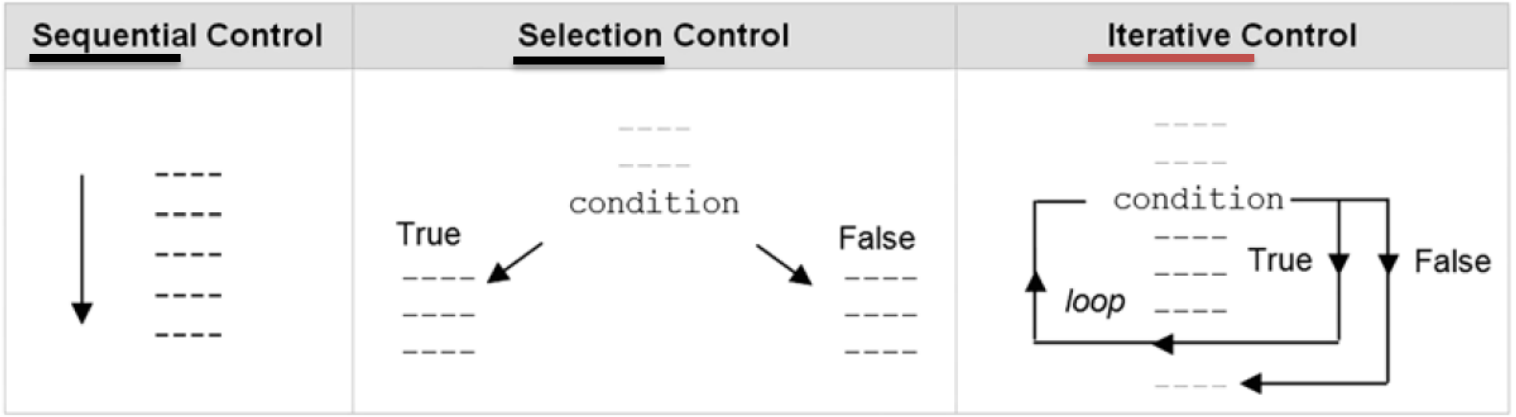
C. Escoger un medio de trasporte

D. Ir al medio de transporte

What is control flow?

- A **control statement** is a statement that determines the control flow of a set of instructions.

STATEMENT TO SPECIFY HOW CONTROL FLOW SHOULD CHANGE



Iterative Control

THE MATHEMATICAL ANALYSIS

OF LOGIC,

BEING AN ESSAY TOWARDS A CALCULUS
OF DEDUCTIVE REASONING.

BY GEORGE BOOLE.

Ἐπικοινωνοῦσι δὲ πᾶσαι αἱ ἐπιστῆμαι ἀλλήλαις κατὰ τὰ κοινά. Κοινὰ δὲ λέγω, οἷς χρῶνται ὡς ἐκ τούτων ἀποδεικνύντες· ἀλλ' οὐ περὶ ὧν δεικνύουσιν, οὐδὲ ὃ δεικνύουσι.

ARISTOTLE, *Anal. Post.*, lib. I. cap. XI.

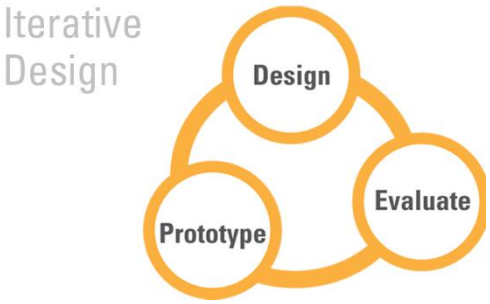
CAMBRIDGE:
MACMILLAN, BARCLAY, & MACMILLAN;
LONDON: GEORGE BELL.

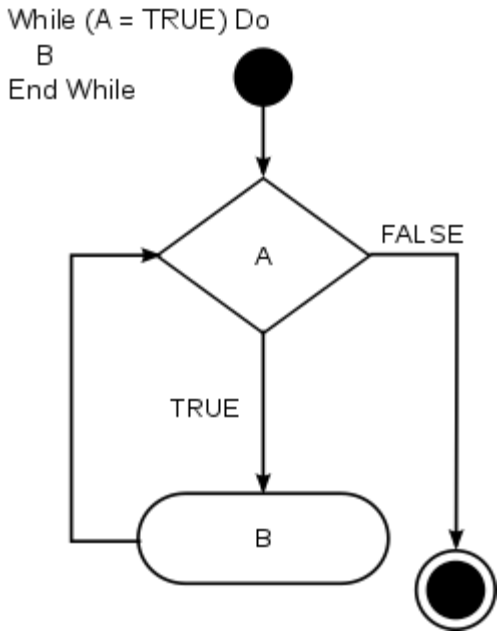
1847

"Todas las ciencias se asocian con otras respecto a elementos comunes. (Y yo llamo común a todo aquello que utilizan en sus demostraciones, no a aquello que puede ser o no ser probado)"

Iterative Control

- An **iterative control statement** is a control statement that allows for the repeated execution of a set of statements.





While Statement

- A **while statement** is an iterative control statement that repeatedly executes a set of statements based on a provided Boolean expression.

while statement	Example use

`while condition:`
`    suite`
	`sum = 0`
	`current = 1`
	
	`n = int(input('Enter value: '))`
	
	`while current <= n:`
	`    sum = sum + current`
	`    current = current + 1`

Iteration	sum	current	current <= 3	sum = sum + current	current = current + 1
1	0	1	True	sum = 0 + 1 (1)	current = 1 + 1 (2)
2	1	2	True	sum = 1 + 2 (3)	current = 2 + 1 (3)
3	3	3	True	sum = 3 + 3 (6)	current = 3 + 1 (4)
4	6	4	False	loop termination	

Error Checking

```
# Temperature Conversion Program (Celsius-Fahrenheit / Fahrenheit-Celsius)

# Display program welcome
print('This program will convert temperatures (Fahrenheit/Celsius)')
print('Enter (F) to convert Fahrenheit to Celsius')
print('Enter (C) to convert Celsius to Fahrenheit')

# Get temperature to convert
which = input('Enter selection: ')

while which != 'F' and which != 'C':
   which = input("Please enter 'F' or 'C': ")

temp = int(input('Enter temperature to convert: '))

# Determine temperature conversion needed and display results
if which == 'F':
    converted_temp = format((temp - 32) * 5/9, '.1f')
    print(temp, 'degrees Fahrenheit equals', converted_temp, 'degrees Celsius')
else:
    converted_temp = format((9/5 * temp) + 32, '.1f')
    print(temp, 'degrees Celsius equals', converted_temp, 'degrees Fahrenheit')
```

LET'S TRY IT

In IDLE, create and run a simple program containing the code below and observe the results. Make sure to indent the code exactly as shown.

```
n = 10
sum = 0
current = 1

while current <= n:
     sum = sum + current
     current = current + 1

print(sum)
```

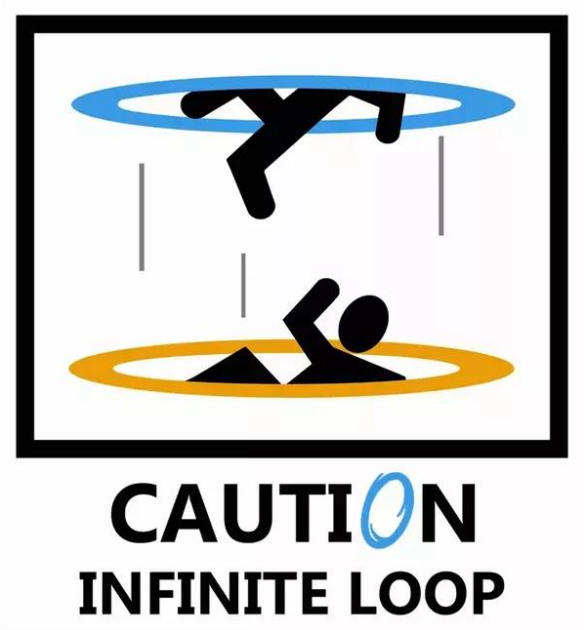
???

```
n = 10
sum = 0
current = 1

while current <= n:
     sum = sum + current
     current = current + 1

     print(sum)
```

???

Infinite Loops

- An **infinite loop** is an iterative control structure that never terminates (or eventually terminates with a system error).



Definite and Indefinite Loops

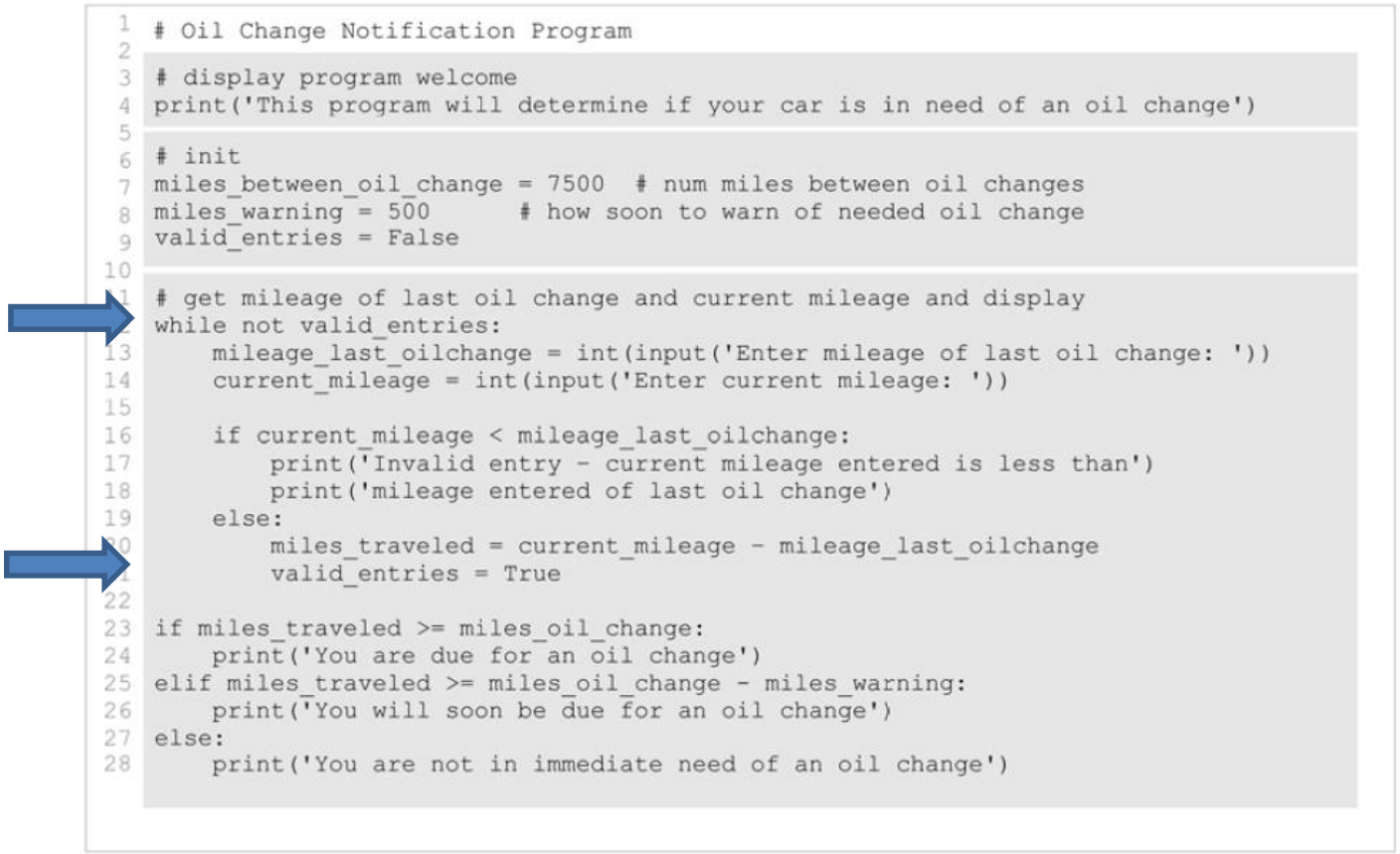
- A **definite loop** is a program loop in which the number of times the loop will iterate can be determined before the loop is executed. A **indefinite loop** is a program loop in which the number of times the loop will iterate is not known before the loop is executed.

```
sum = 0
current = 1
n = input('Enter value: ')
while current <= n:
    sum = sum + current
    current = current + 1
```

```
which = input("Enter selection: ")
while which != 'F' and which != 'C':
    which = input("Please enter 'F' or 'C': ")
```

Boolean Flag

- A single Boolean variable used as the condition of a given control statement is called a **Boolean flag**.



```
# Oil Change Notification Program

# display program welcome
print('This program will determine if your car is in need of an oil change')

# init
miles_between_oil_change = 7500  # num miles between oil changes
miles_warning = 500      # how soon to warn of needed oil change
valid_entries = False

# get mileage of last oil change and current mileage and display
while not valid_entries:
    mileage_last_oilchange = int(input('Enter mileage of last oil change: '))
    current_mileage = int(input('Enter current mileage: '))

    if current_mileage < mileage_last_oilchange:
        print('Invalid entry - current mileage entered is less than')
        print('mileage entered of last oil change')
    else:
        miles_traveled = current_mileage - mileage_last_oilchange
        valid_entries = True

if miles_traveled >= miles_oil_change:
    print('You are due for an oil change')
elif miles_traveled >= miles_oil_change - miles_warning:
    print('You will soon be due for an oil change')
else:
    print('You are not in immediate need of an oil change')
```