



LISBON
SCHOOL OF
ECONOMICS &
MANAGEMENT
UNIVERSIDADE DE LISBOA

Carlos J. Costa

PYTHON PROGRAMMING FUNDAMENTALS: CONTROL STRUCTURES



Index

- Learning Objectives
- Conditions and Branching
- Loops



Learning Objectives

- Know key control structures
- Know how to do small programs



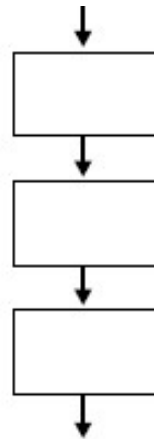
Control Structure

- Sequencing
- Decision
- Loops



Sequencing

- By default, in a structured language code is interpreted from right to left and top to bottom.



Decision

- The **IF-THEN** statement is a simple control that tests whether a condition is true or false.

```
# A simple decision
```

```
age=?
```

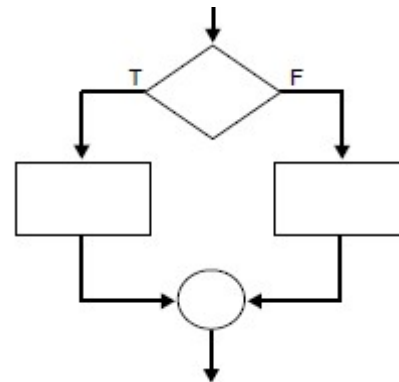
```
if age >= 18:
```

```
    print("You can vote!")
```



Decision

- IF-THEN statements test for only one action.
- If the condition is true, then an action occurs. If the condition is false, take an alternate action.



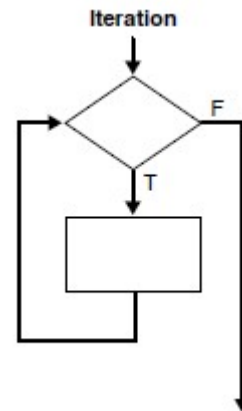
Decisão

```
#If-elif-else structure
if age < 3:
    ticketPrice= 0
elif age < 18:
    ticketPrice = 5
else: ticketPrice = 10
```



Loops

- A **WHILE** loop is a process in which a loop is initiated until a condition has been met.
- This structure is useful when performing iterative instructions to satisfy a certain parameter.



Loops

- The loop ends as a result of value entered by user

```
msg = ''  
while msg != 'exit':  
    msg = input("Write a message: ")  
    print(msg)
```

Loops

- A for loop is used for iterating over a sequence (e.g. list, tuple, dictionary, set, or string).
- Similar to an iterator method

```
insects = ["fly", "ant", "ladybirth"]  
for x in insects:  
    print(x)
```

Loops

- **break** statement - stop the loop before it has looped through all the items

```
insects = ["fly", "ant",  
"ladybirth"]
```

```
for x in insects:  
    if x == "ant":  
        break  
print(x)
```

- **continue** statement - stop the current iteration of the loop, and continue with the next

```
insects = ["fly", "ant",  
"ladybirth"]
```

```
for x in insects:  
    if x == "ant":  
        break  
print(x)
```



Loops

- To loop through a set of code a specified number of times, use the `range()` function

```
insects = ["fly", "ant", "ladybirth"]  
for x in range(3):  
    print(insects[x])
```

Loops

- **Nested Loop** is a loop inside a loop.

```
insects = ["fly", "ant", "ladybirth"]
adj = ["nice", "hugly", "disgusting"]
for x in adj:
    for y in insects:
        print(x, y)
```