
iscte
INSTITUTO
UNIVERSITÁRIO
DE LISBOA
emprego
digital
UPskill


<Welcome To>
python™

1.

Overall Program Content

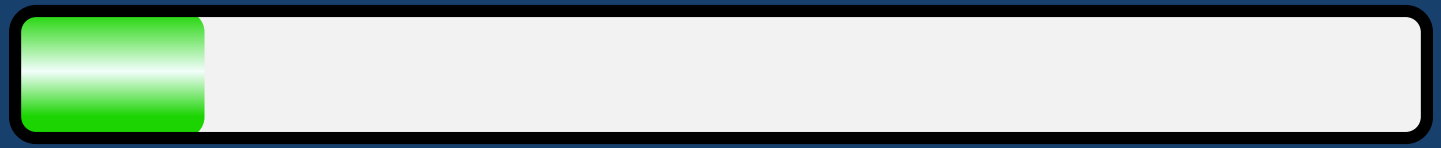
Web development with Python	Hours
Work skills development	50
Python Programming Introduction	150
Web Programming Introduction (html/css)	100
Databases Concepts and Structures	50
Web Servers Programming	150
Web services development	150
Total	650

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

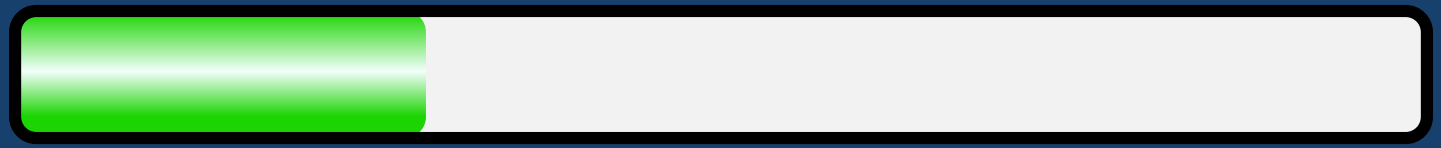
Python programming Introduction Content

1. Course Introduction
- Why Python?
- Python Applications
- Installation Tools
- Building your code catalog
- Useful websites



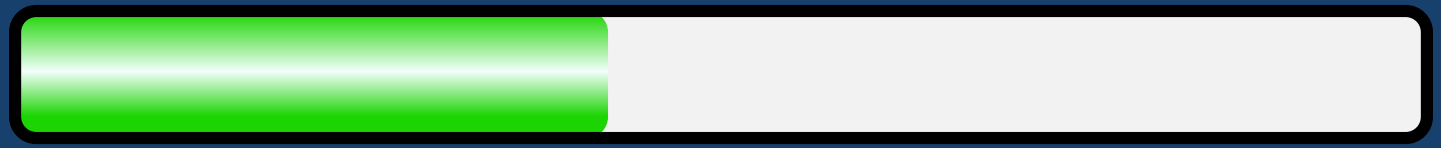
Python programming Introduction Content

2. Data types/outputs/inputs
3. Operators
4. Functions and Modules



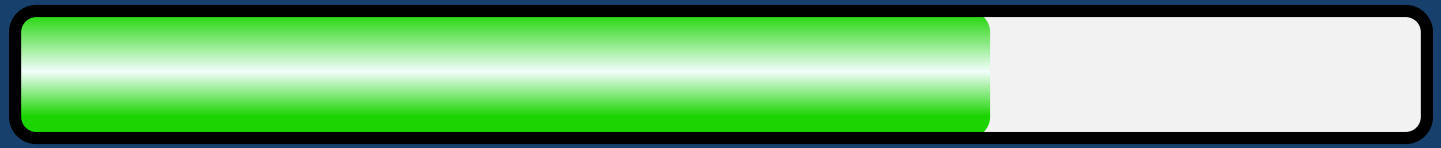
Python programming Introduction Content

5. Conditional statements and expression
6. Loops
7. Work with standard Library and Modules



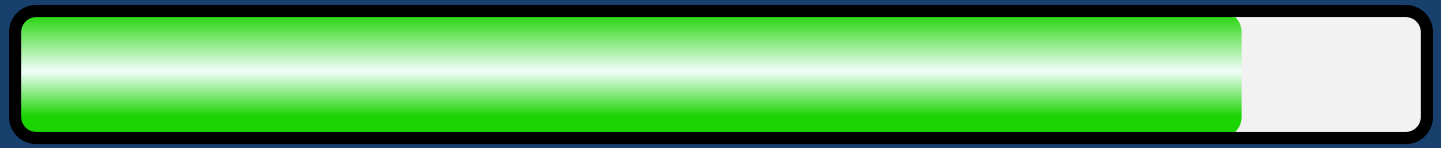
Python programming Introduction Content

8. Data structure in python
9. List,
10. Tuple,
11. Dictionaries,
12. Set



Python programming Introduction Content

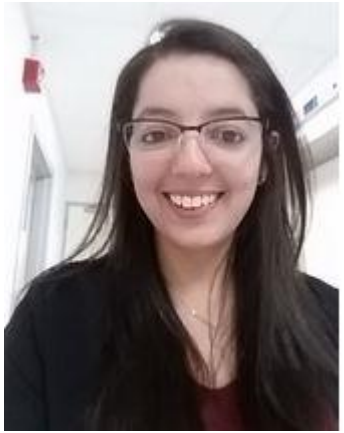
13. Files
14. Functions and Modules
15. Classes
16. Introduction to Numpy
17. Introduction to Pandas



Python programming Introduction Content

18. Introduction to matplotlib for data visualization
19. Data Preprocessing

Our Teachers:



Joseanne Viana (Josi)

Email: jcova1@iscte-iul.pt

Stefan Postolache

Email:stefanpostolache@edu.ulisboa.pt

Hamed Farkhari

Email:Hamed_Farkhari@iscte-iul.pt

Schedule

Days/modules		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	12-Oct	Joseanne																		
2	13-Oct																			
3	14-Oct																			
4	15-Oct																			
5	16-Oct																			
6	19-Oct						Hamed													
7	20-Oct																			
8	21-Oct																			
9	22-Oct																			
10	23-Oct																			
11	26-Oct																			
12	27-Oct												Stefan							
13	28-Oct																			
14	29-Oct																			
15	30-Oct																			
16	2-Nov															Joseanne				
17	3-Nov																			
18	4-Nov																			
19	5-Nov																			
20	6-Nov																	Hamed		
21	9-Nov																			


<Let’s get started >

Contents

Dictionary

Define Dictionary

use [ ] to define a List
use (,) to define a Tuple
use { : } to define a Dictionary,
Dictionary {key : value}

```
d = {
      'brand' : 'cherry' ,
      'model' : 'arizo5' ,
      'color' : 'white'
      }

print(type(d))      # <class dict>

print(len(d))       # 3
```

Question

Dictionaries are Ordered or Not?

Add new
<key : value>

Or

Change value

add new <key : value> to the 'd' dictionary

```
d = {
      'brand' : 'cherry' ,
      'model' : 'arizo5' ,
      'color' : 'white'
      }

d['year'] = '2010'
```

d[<key>] ==> <value>

```
print( d['model'])  # arizo5

d['color'] = 'Black' # change values
```

Get( )

access the members in tuple/list we use [<index>]

```
print(d)       # {'brand': 'cherry', 'model': 'arizo5',
                 'color': 'Black' , 'year' : '2010' }

x = d.get('model')          # dic_name.get(<key>) ==> <value>
print(x)                    # arizo5

x = d.get('cylinder')
print(x)                    # None, 'cylinder' key not found

x = d.get('cylinder',-1)    # if <key> not found, return -1
print(x)                    # -1
```

Keys( )
Values( )
Items( )

Access keys, values, Or both of them

```
print(d)      # {'brand': 'cherry', 'model': 'arizo5',
                'color': 'Black' , 'year' : '2010' }

print(list(d.keys()))    # ['brand', 'model', 'color', 'year']

print(list(d.values()))  # ['cherry', 'arizo5', 'Black', '2010']

print(list(d.items()))   # [ ( <key1> , <value1> ) , ... ]
#[('brand', 'cherry'), ('model', 'arizo5'), ('color', 'Black'), ('year', '2010')]


for k,v in d.items():
    print(k,':',v)
```

unpacking

pop(<key>)

```
d.pop()     # Error
Error: Pop() for dict expected at least 1 argument
```

```
d = {
      'brand' : 'cherry' ,
      'model' : 'arizo5' ,
      'color' : 'white'
      }

d.pop('model')     # dict_name.pop(<key>)

print(d)      # {'brand': 'cherry', 'color': 'white'}
```

popitem( )

remove the last item in dictionary
return removed item in output

```
d = {
    'brand' : 'cherry' ,
    'color' : 'white'
    }

f = d.popitem()
print(d)          # {'brand': 'cherry'}
print(f)          # ('color', 'white')
print(type(f))    # <class 'tuple'>
```

clear( )
del

```
d.clear()

print(d)      # {}

del d
```

Example

make a dict of {<key : value>}
value of each key is the number of key occurrences
Input ['x', 'y', 'x', 'z', 'y', 'x']
Output {'x': 3, 'y': 2, 'z': 1}

Example Solution 1

make a dict of {<key : value>}
value of each key is the number of key occurrences
Input ['x', 'y', 'x', 'z', 'y', 'x']
Output {'x': 3, 'y': 2, 'z': 1}

```
a = ['x', 'y', 'x', 'z', 'y', 'x']
d = {}

for i in a :
    if i not in d:
        d[i] = 1
    else:
        d[i] += 1          # d[i] = d[i] + 1
print(d)                   # {'x': 3, 'y': 2, 'z': 1}
```

Example Solution 2 get( )

make a dict of {<key : value>}
value of each key is the number of key occurrences
Input ['x', 'y', 'x', 'z', 'y', 'x']
Output {'x': 3, 'y': 2, 'z': 1}

```
a = ['x', 'y', 'x', 'z', 'y', 'x']
d = {}

for i in a:
    d[i] = d.get(i,0) +1

print(d)
```

Example Solution 3 Setdefault(,)

make a dict of {<key : value>}
value of each key is the number of key occurrences
Input ['x', 'y', 'x', 'z', 'y', 'x']
Output {'x': 3, 'y': 2, 'z': 1}

```
a = ['x', 'y', 'x', 'z', 'y', 'x']
d = {}


for i in a:
    d[i] = d.setdefault(i, 0) + 1

print(d)
```

Setdefault(,)

Good for initialize dictionary inside a loop

```
"""
 d3 = { 1 : '1' ,
        2 : '2' ,
        3 : '3' ,
        ...     ,
        100 : '100' }
"""
d3 = {}
for i in range(1, 101):
    d3.setdefault(i,str(i) )
```

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA emprego digital

Copy( )

make a copy with copy()

```
a = {}          # 'a' is an empty dict

b = a           # 'a' and 'b' are dependent

c = a.copy()    # 'a' and 'c' are independent
```

Dependencies are like as Lists


<Exercise 1>

Exercise

make a dict of {<key : value>}

value of each key is the number of key occurrences

Input 'abfabdcaa' ← string

Output {'a': 4, 'b': 2, 'f': 1, 'd': 1, 'c': 1}

Exercise Solution

make a dict of {<key : value>}

value of each key is the number of key occurrences

Input 'abfabdcaa' ← string

Output {'a': 4, 'b': 2, 'f': 1, 'd': 1, 'c': 1}

```
s = 'abfabdcaa'

d = {}

for i in s:
    d[i] = d.get(i,0) + 1

print(d)   #{'a': 4, 'b': 2, 'f': 1, 'd': 1, 'c': 1}
```


<Exercise 2>

Exercise

make a dict of {<key : value>}

value of each key is the number of key occurrences

Input

line ='a dictionary is a datastructure·'

Output

{'a': 2, 'dictionary': 1, 'is': 1, 'datastructure·': 1}

find repeats in a text line

Exercise Solution

Input line ='a dictionary is a datastructure.'

Output

{'a': 2, 'dictionary': 1, 'is': 1, 'datastructure.': 1}

```
line ='a dictionary is a datastructure.'
d = {}
s = line.split() # split string by spaces
print(s)  # ['a', 'dictionary', 'is', 'a', 'datastructure.']

for i in s:
    d[i] = d.get(i,0) + 1

print(d)
# {'a': 2, 'dictionary': 1, 'is': 1, 'datastructure.': 1}
```


<Exercise 3>

Exercise development

make a dict of {<key : value>}

value of each key is the number of key occurrences

Input

```
lines ='a dictionary is a datastructure \n
        a set is also a datastructure.'
```

Output

```
{'a': 4, 'dictionary': 1, 'is': 2, 'datastructure': 2,
  'set' : 1, 'also' : 1}
```

find repeats in a text lines

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

Exercise Solution

```
"""
input
lines ='a dictionary is a datastructure
        a set is also a datastructure.'
Output
{'a': 4, 'dictionary': 1, 'is': 2, 'datastructure': 2,
  'set' : 1, 'also' : 1}
**************************************************

line1 = lines.split("\n")[0]
line1 = line1.split(".")[0]

line1 = lines.split("\n")[0].split(".")[0]

**************************************************

line2 = lines.split("\n")[1]
line2 = line2.split(".")[0]

line2 = lines.split("\n")[1].split(".")[0]

****** replace [...] with 'i'  ******
i = 0, 1
line = lines.split("\n")[i].split(".")[0]
"""
```

Exercise Solution

We developed the codes of previous exercise!

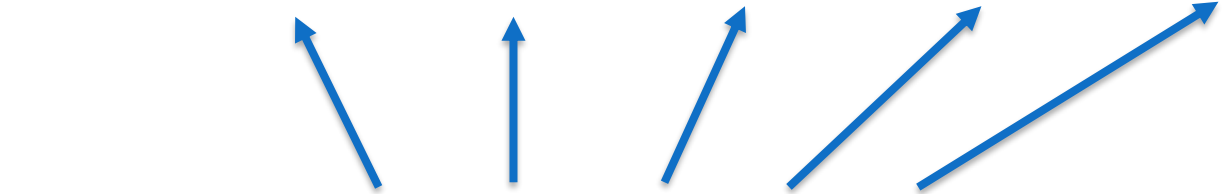
```
lines = 'a dictionary is a datastructure\na set is also a datastructure.'

d = {}
number_of_lines = len(lines.split("\n"))

for i in range(number_of_lines):
    line = lines.split("\n")[i].split(".")[0]

    s = line.split()
    # print(s)

    for i in s:
        d[i] = d.get(i, 0) + 1

print(d)
```


<Exercise 4>

Exercise

calculate sum of values in dict

```
d = {'a': 4, 'b': 2, 'f': 1, 'd': 1, 'c': 1}
```



Output *4 + 2 + 1 + 1 + 1 = 9*

Exercise Answer

calculate sum of values in dict

```
d = {'a': 4, 'b': 2, 'f': 1, 'd': 1, 'c': 1}

s = 0
for i in d:
    s += d[i]
print(s)        # 9
```

Or you can use dict_name.values() and sum()

```
print(sum(d.values()))  # 9
```

Sort
Operator module
Itemgetter( )

sort in dict
by keys / by values

```
d = {'a': 4, 'b': 2, 'f': 1, 'd': 1, 'c': 1}

import operator
k= operator.itemgetter(1)      # sort by values
print(sorted(d.items(),key = k))
# [('f', 1), ('d', 1), ('c', 1), ('b', 2), ('a', 4)]

k= operator.itemgetter(0)      # sort by keys
print(sorted(d.items(),key = k))
# [('a', 4), ('b', 2), ('c', 1), ('d', 1), ('f', 1)]
```

Sorted( )

Sort values

```
num ={
      'ali' : [12,13,8],
      'sara': [15,7,14],
      'taha': [5,18,13]
      }

d = {k : sorted(v)  for k,v in num.items()}

print(d)

# {'ali': [8, 12, 13], 'sara': [7, 14, 15], 'taha': [5, 13, 18]}
```

update( )

merge 2 dictionaries

```
d1 = {'x' : 3 , 'y': 2 , 'z':1}

d2 = {'w' : 8 , 't': 7 , 'z':5}


d1.update(d2)

print(d1)
# {'x': 3, 'y': 2, 'z': 5, 'w': 8, 't': 7}
```

new value of 'z' is replaced

update( )
{d1 , **d2}**

merge 2 dictionaries with for

```
d1 = {'x' : 3 , 'y': 2 , 'z':1}

d2 = {'w' : 8 , 't': 7 , 'z':5}

d = {}
for i in (d1,d2):
    d.update(i)
print(d)
```

```
# d = d1.copy()
# d.update(d2)
```

merge 2 dictionaries with {**d1 , **d2}

```
d = {**d1 , **d2}
print(d)
```

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA
emprego digital


<Exercise 5>

Exercise

merge 2 dictionaries together
For same keys, sum values

Input:
d1 = { 'x' : 3 , 'y': 2 , 'z':1 }
d2 = { 'w' : 8 , 't': 7 , 'z':5 }

Output:
{ 'x': 3, 'y': 2, 'z': 6, 'w': 8, 't': 7 }

Exercise Solution

Input:
d1 = { 'x' : 3 , 'y': 2 , 'z':1 }
d2 = { 'w' : 8 , 't': 7 , 'z':5 }

Output:
{ 'x': 3, 'y': 2, 'z': 6, 'w': 8, 't': 7 }

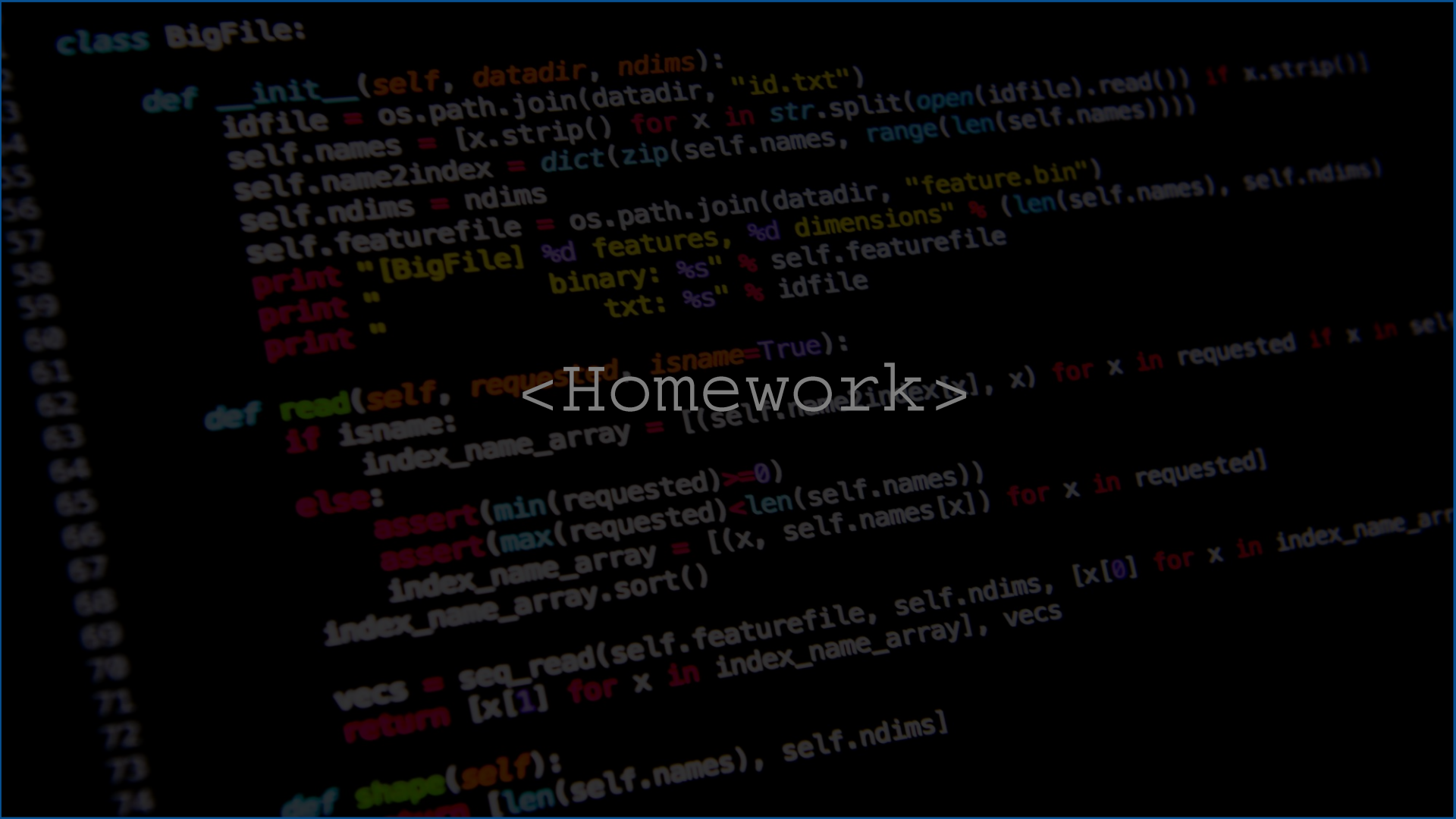
```
d1 = {'x' : 3 , 'y': 2 , 'z':1}
d2 = {'w' : 8 , 't': 7 , 'z':5}

for i,j in d2.items():
    if i in d1:
        d1[i] += d2[i]   # d1[i] = d1[i] + d2[i]
    else:
        d1.update({i : j})

print(d1)
```

We Continue on Monday...

It's Not Finished!!!

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital


<Homework>

Homework

Create a 'person' Dictionary with this details:

```
print(len(person))  # 4

print(person['phone']['home'])    # 01-4455

print(person['phone']['mobile'])  #918-123456

print(person['children'])     # ['Olivia', 'Sophia']
print(person['children'][0])  # Olivia

print(person.pop('age'))      # 48
```

“

- *Make it work*
- *Make it Right*
- *Make it Fast*


O futuro profissional
começa aqui
iscte
INSTITUTO
UNIVERSITÁRIO
DE LISBOA
emprego
digital
UPskill