iscte
INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

UPskill

<Welcome To>

python™

1.

Overall Program Content

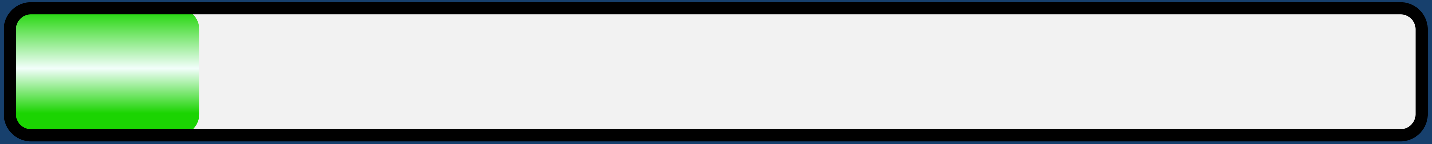
Web development with Python	Hours
Work skills development	50
Python Programming Introduction	150
Web Programming Introduction (html/css)	100
Databases Concepts and Structures	50
Web Servers Programming	150
Web services development	150
Total	650

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

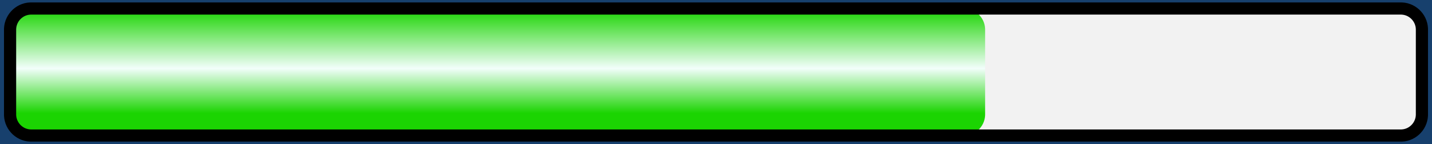
2. Data types/outputs/inputs
3. Operators
4. Functions and Modules

Python programming Introduction Content

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

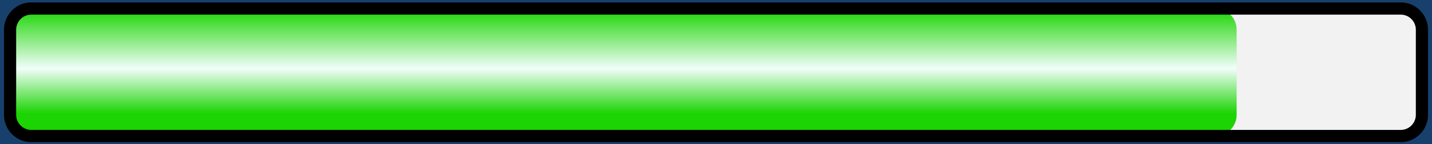
Python programming Introduction Content

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

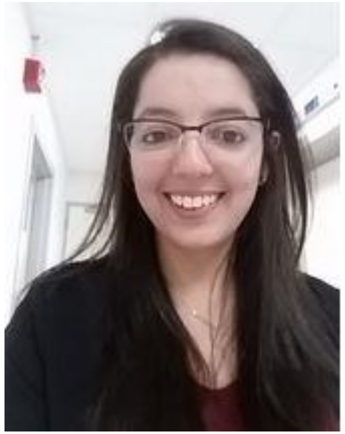


Python programming Introduction Content

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

100% Loaded

Our Professors:



Joseanne Viana (Josi)

Email: jcova1@iscte-iul.pt

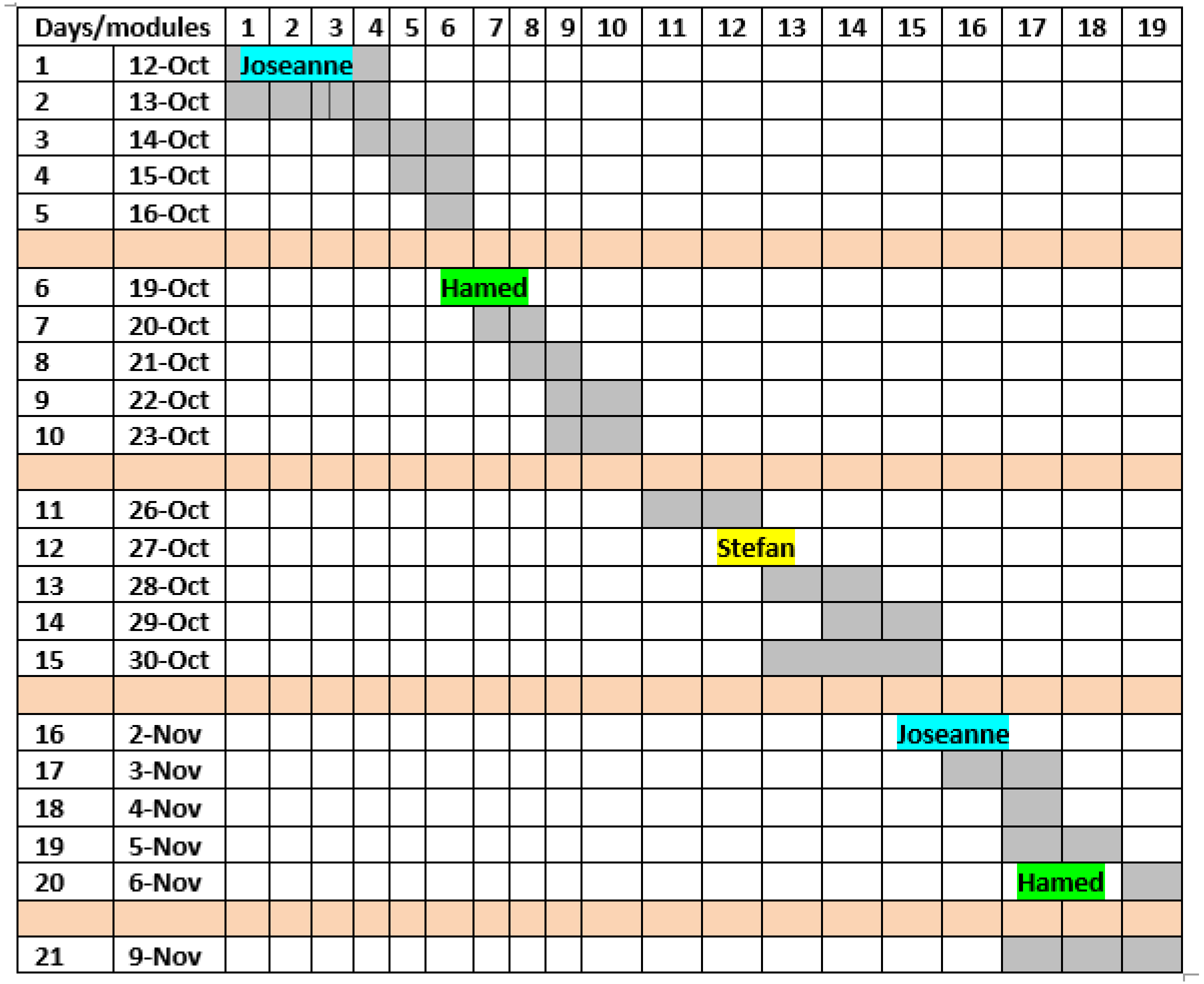
Stefan Postolache

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 >

S
U
M
M
A
R
Y

SUMMARY

iscte
INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

Contents

Data types
Outputs
Inputs

iscte
INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

Variables

Input:

Name	Age
Josi	31
Tom	34
Joe	22
Leo	45
Samuel	38

Output:

```
Josi is 31 years old
Tom  is 34 years old
Joe  is 22 years old
Leo  is 45 years old
Samuel is 38 years old
```

Put Name and age in variables and change the variable values every time.

Variable Examples

Input:

```
name ="Josi“
age = 31
```

Output:

```
Josi is 31 years old
```

```
print('%s is %d years old' % (name, age))
print('{} is {} years old'.format(name, age))
print(f'{name} is {age} years old’)
```

Simplified:

```
print(name, "is", age, "years old")

print(name + " is " + str(age) + " years old")  🡐 all should be string
```

Change the code for different input age types without raising errors : “31”, “31.5”, 31.5

output ➔ Josi is 31 years old

Answer → `print(name, "is", int(float(age)), "years old")`

Variables Name

All identifiers must start with a letter or underscore (_), you can't start with digits.

Identifiers can contain letters, digits and underscores (_).

Identifiers can't be a keyword.

They can be of any length.

```
print ( ' a2 ' . isidentifier() )      # True
print ( ' 2a ' . isidentifier() )      # False
```

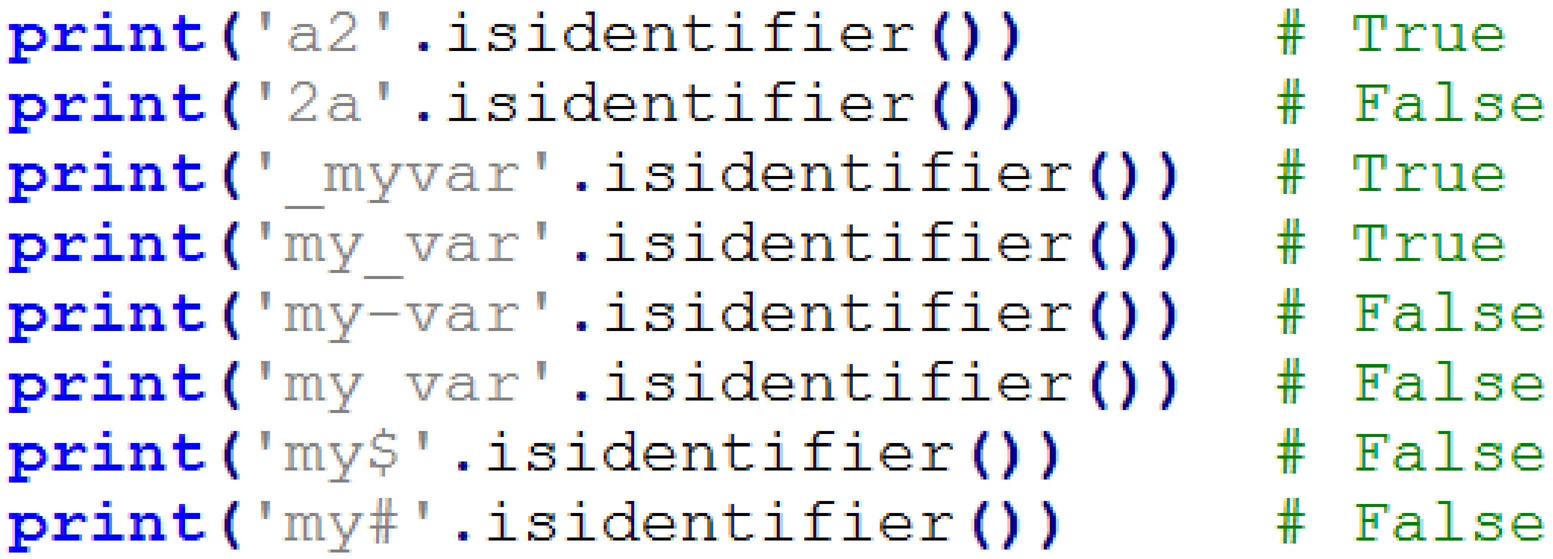
Variables Name

You cannot use reserved words as variable names (keywords)

False	class	return	is	finally	None	if
for	lambda	continue	True	def	from	while
nonlocal	and	del	global	not	with	as
elif	try	or	yield	assert	else	Import
pass	break	except	in	raise		

```
from        keyword   import      iskeyword
print ( iskeyword (‘ if ‘ ) )                        # True
```

Variables Name Examples



```
print('a2'.isidentifier())       # True
print('2a'.isidentifier())       # False
print('_myvar'.isidentifier())   # True
print('my_var'.isidentifier())   # True
print('my-var'.isidentifier())   # False
print('my var'.isidentifier())   # False
print('my$'.isidentifier())      # False
print('my#'.isidentifier())      # False
```

Multi assignment

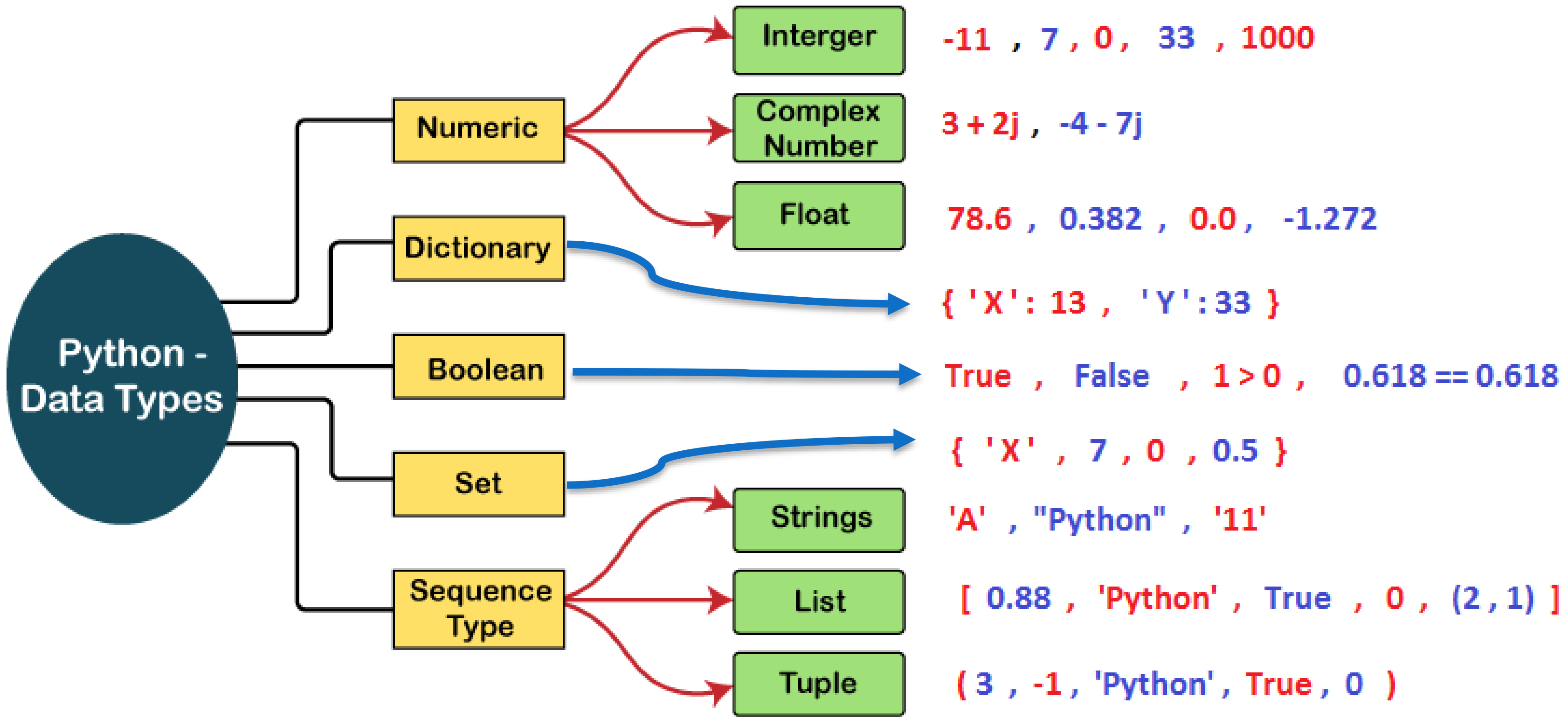
```
a = 5
b = 1
print('Five plus one is {a + b}')     # Five plus one is {a + b}
print(f'Five plus one is {a + b}')    # Five plus one is 6
```

```
a = b = c = 5          # this statement assign 5 to c, b and a.
print(a, b, c)         # 5 5 5 , a, b and c are independent
```

```
x = 1
y = 2
y, x = x, y            # assign y value to x and x value to y
print(x)               # 2
print(y)               # 1
```

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital



Check data types of each one with ' type() ' command

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA
emprego digital

Data Types Examples

Python Data Types: *int , float , complex , str , bool , list , tuple , set , dict , bytes , …*

```
s = "Python Course"
print(type(s))          # str

i = 2
print(type(i))          # int

f = 2.5
print(type(f))          # float
```

```
c = 2 + 3j          # 2 is the real part and 3 is imaginary
print(type(c))      # complex
```

INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

Data Types
Examples

```
print(bool(5))          # True
print(bool(-2))         # True
print(bool('Hamed'))    # True
```

```
print(bool(0))          # False
print(bool(''))         # False
```

```
print(bool([]))       # False (empty list)
print(bool({}))       # False (empty dictionary)
print(bool(()))       # False (empty tuple)
```

```
b = True
c = 5<2
print(type(b))      # bool
print(type(c))      # bool
```

Data Types Examples

```python
l = ["apples", "grapes", "oranges"]
print(type(l))        # list
```

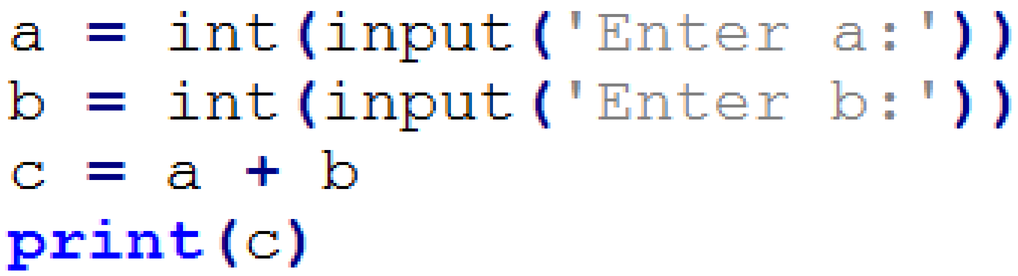
```python
t = ("apple", "banana", "cherry")
print(type(t))        # tuple
```

```python
d = {'id': '123', 'name': 'farshid'}
print(type(d))      # dict
```

```python
s = {'apple', 'banana', 'cherry'}
print(type(s))      # set
```

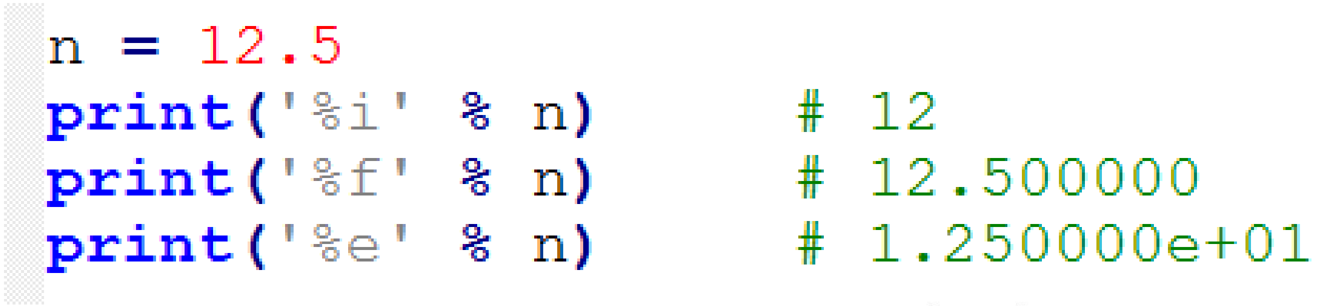
Input

Receiving input from Console
The output of ' input() ' command is string



```
a = int(input('Enter a:'))
b = int(input('Enter b:'))
c = a + b
print(c)
```

Output



```
n = 12.5
print('%i' % n)          # 12
print('%f' % n)          # 12.500000
print('%e' % n)          # 1.250000e+01
```

Operators

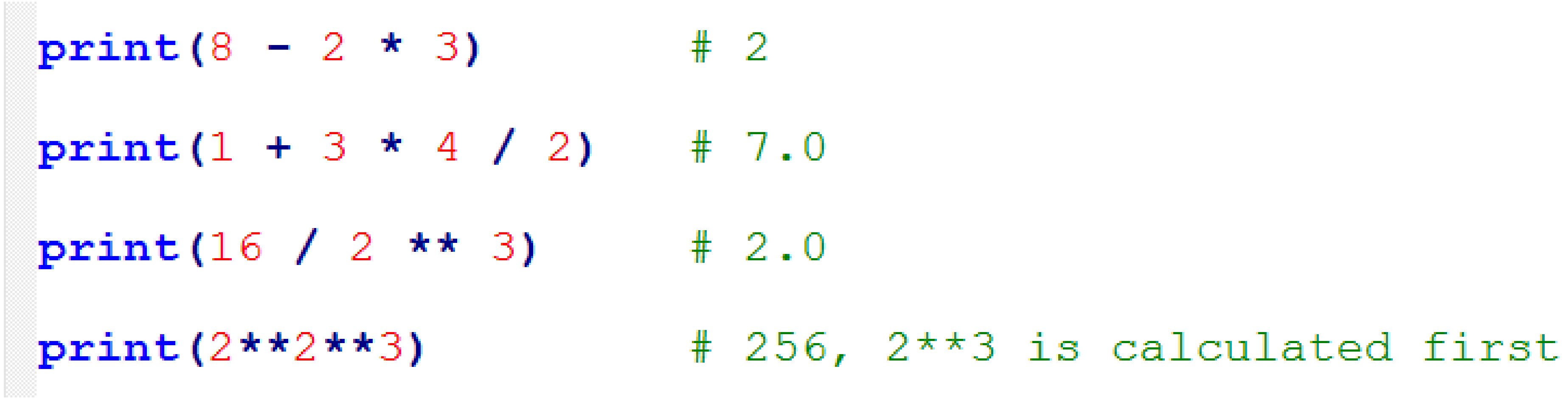
Operators

Arithmetic	+ , - , * , / , % , ** , //
Assignment	= , += , -= , *= , /= , %= , //= , **=
Comparison	== , != , > , < , >= , <=
Logical	and , or , not
Membership	in , not in
Bitwise	&, | , ^ , ~ , << , >>

Arithmetic Operators Examples

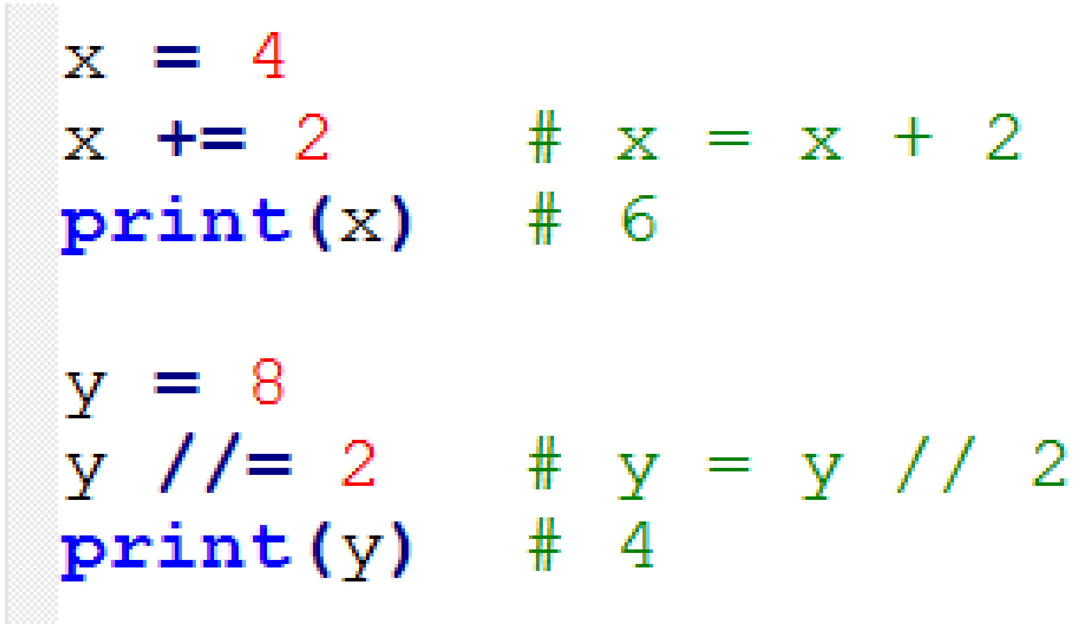
```
#Addition
print(1 + 3)          # 4

#Subtraction
print(5 - 3)          # 2

#Multiplication
print(2 * 3)          # 6

#Float Division
print(3 / 2)          # 1.5

#Integer Division
print(3 // 2)         # 1

#Remainder
print(17 % 5)         # 2

# Exponentiation
print(2 ** 3)         # 8
print(0 ** 0)         # 1 #########
print(6 ** 0)         # 1
```

Operators Precedence Examples



```
print(8 - 2 * 3)           # 2

print(1 + 3 * 4 / 2)       # 7.0

print(16 / 2 ** 3)         # 2.0

print(2**2**3)             # 256, 2**3 is calculated first
```

Augmented Assignment Operator Examples



```
x = 4
x += 2       # x = x + 2
print(x)     # 6

y = 8
y //= 2      # y = y // 2
print(y)     # 4
```

Comparison Operators Examples

```
print(2 == 3)          # False
print(2 != 3)          # True
print(2 < 3)           # True
```

Logical Operators Examples

```
print(1<3  or 4>5)     # True
print(1<3 and 4>5)     # False
print(not 1<3)         # False
```

Short-circuit Examples

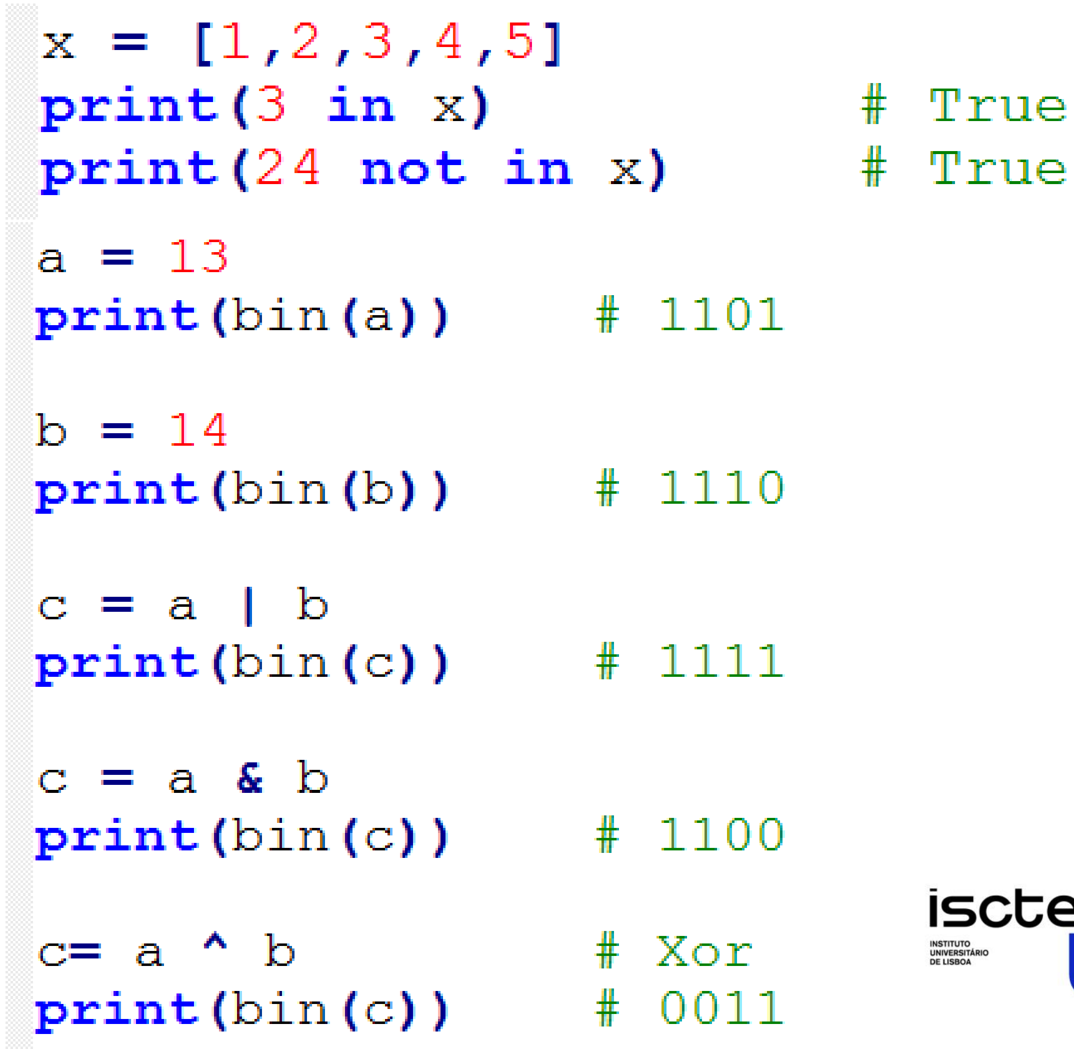
```
print(1 >= 2 and (5/0) > 2)    # False

print(3 >= 2 and (5/0) > 2)    # Error
```

In ‘and’ , if The first operation is false, the second operation will be skipped!

Error: division by zero

Membership Operators Examples

Bitwise Operators Examples



```
x = [1,2,3,4,5]
print(3 in x)                # True
print(24 not in x)           # True

a = 13
print(bin(a))       # 1101

b = 14
print(bin(b))       # 1110

c = a | b
print(bin(c))       # 1111

c = a & b
print(bin(c))       # 1100

c= a ^ b            # Xor
print(bin(c))       # 0011
```

Bitwise Operators Examples

```
a = 13                # Shift to left
print(a << 1)         # 26

a = 20                # Shift to right
print(a >> 1)         # 10

a = 18                # Shift 2 bits to right
print(a >> 2)         # 4
```

Operations on String Examples

```
s1 = 'Python'
s2 = ' Course'
s3 = s1 + s2
print(s3)         # Python Course

s = 'sara'
print(2*s)        # sarasara
```

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

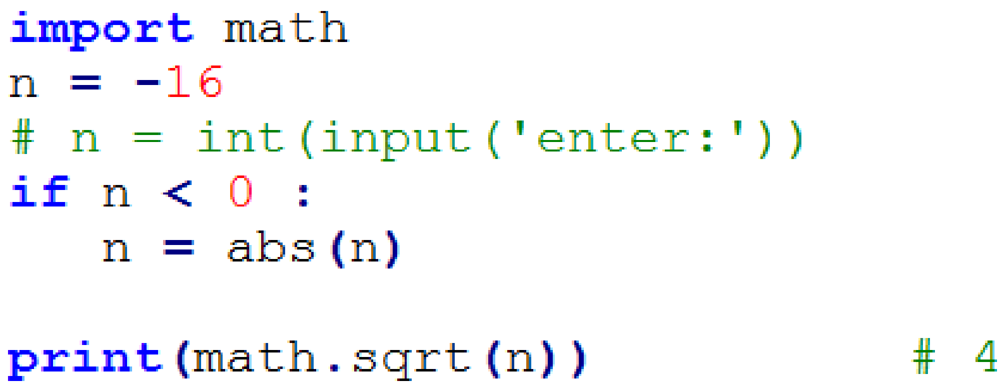
emprego digital

Conditional statements & expression

Control Statements

if
if else → elif
else

we do not have 'switch case' in python :(



```
import math
n = -16
# n = int(input('enter:'))
if n < 0 :
    n = abs(n)

print(math.sqrt(n))                    # 4
```

Control Statements Examples

```
a = 5
if True:
    a = 6
print(a)            # 6
```

if - else example:

```
a = 20
if a % 2 == 0:
    print('even')       # even
else:
    print('odd')
```

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

Control Statements Examples

```
x = 3
y = 2
if x == 1 or y == 1:    # if 1 in(x,y)
    print('ok')
else:
    print('no')              # no


names = ['sara','taha','farshid']
if 'ali' in names:
    print('found')
else:
    print('not found')       # not found
```

Conditional Expression

Find minimum between a, b

```
a = 2
b = 5

if a < b:
    m = a
else:
    m = b
```

Conditional expression

```
m = a if a < b else b
```

Conditional Expression Examples

```
my_list  = ['a','e','o','i','u']

if 'o' in my_list:
     s = 'yes'
else:
     s = 'no'
```

Conditional expression

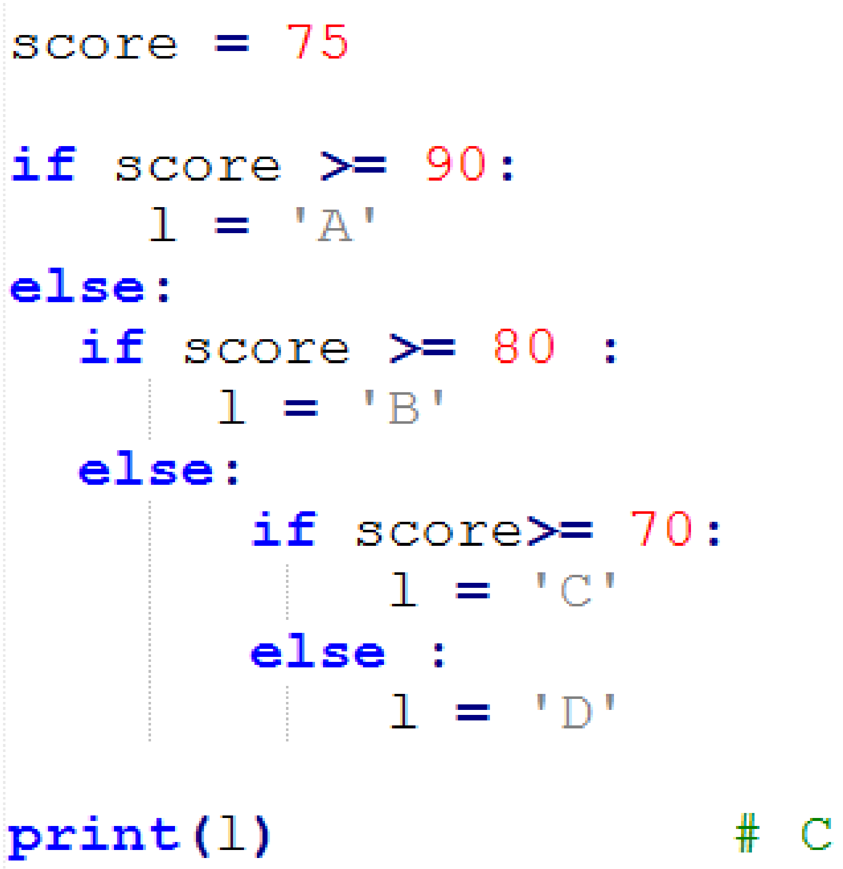
```
s = 'yes' if ('o' in my_list) else 'no'
print(s)                                # yes
```

Conditional Expression Examples

```
x = 2
y = 6

z = 1 + ( x if x > y else y+2)   # z = 1 + (y+2)

print(z)                         # 9
```

```
grade = 12
s = 'fail' if grade < 10 else 'pass'
print(s)                         # pass
```

Nested
if statements



```
score = 75

if score >= 90:
    l = 'A'
else:
  if score >= 80 :
      l = 'B'
  else:
       if score>= 70:
           l = 'C'
       else :
           l = 'D'

print(l)             # C
```

if - elif - else

```python
score = 75

if score >= 90:
    l = 'A'
elif score >= 80 :
    l = 'B'
elif score>= 70:
    l = 'C'
else :
    l = 'D'

print(l)          # C
```

Loops

for
range
Examples

range(start, end, step)
from start till end-1 with step=+2

```
for j in range(5,10,2):
    print(j , end = ' ' )    # 5 7 9
```

```
s = 'Python'
for ch in s:
    print(ch)
```

```
for _ in range(3):
    print('hello')
```

for range Examples

range(end)
start from 0 till end-1 with step=+1

```
for i in range(4):
    print(i , end = ' ')   # 0 1 2 3
```

range(start, end)
from start till end-1 with step=+1

```
for i in range(3, 8):
    print(i , end= ' ')    # 3 4 5 6 7
```

for
range
Examples

Count the number of characters in word

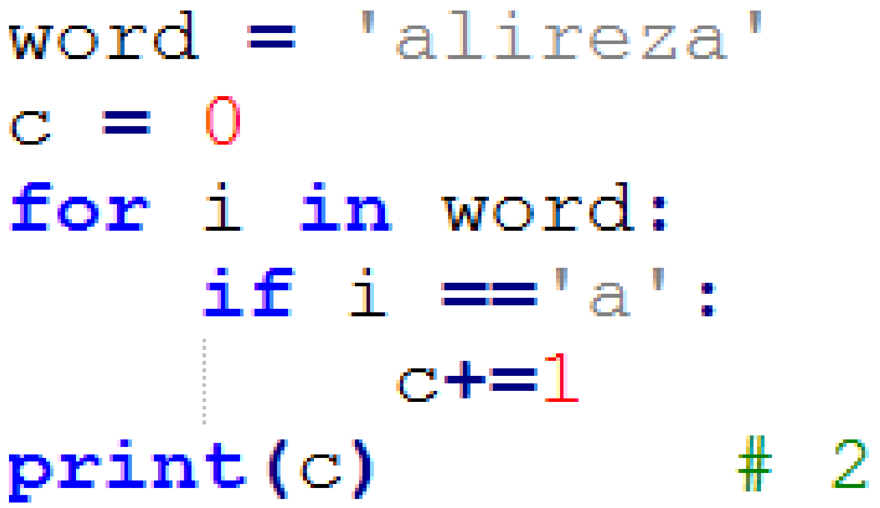
```
word = 'python'
c = 0
for i in word:
    c+=1
print(c)          # 6
```

Step = -3

```
for i in range(9,2,-3)     :
    print(i , end='  ' )     # 9 6 3
```

for
range
Examples

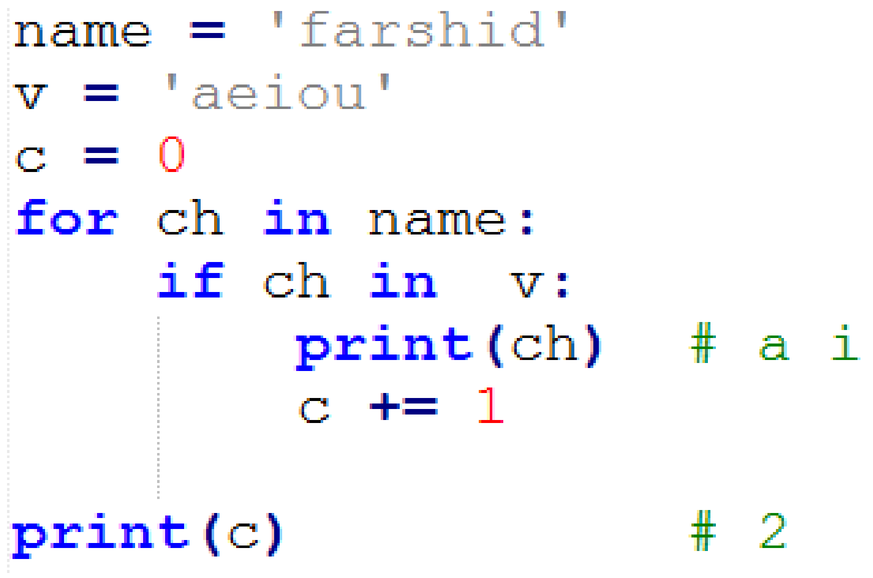
Count how many 'a' are in word



```
word = 'alireza'
c = 0
for i in word:
    if i =='a':
        c+=1
print(c)            # 2
```

for
range
Examples

Vowels in name



```
name = 'farshid'
v = 'aeiou'
c = 0
for ch in name:
    if ch in  v:
        print(ch)   # a i
        c += 1

print(c)            # 2
```

Nested for range Examples

```python
name = 'farshid'
v = 'aeiou'
a = [ch for ch in name if ch in v]
print(a)                    # ['a', 'i']
```

Nested For

```python
for i in range(1,4):
    for j in range(2,4):
        print(j,end=' ')
    print()
'''
i = 1  : j=2 , j=3
i = 2  : j=2 , j=3
i = 3  : j=2 , j=3
'''
```

break
continue
Examples

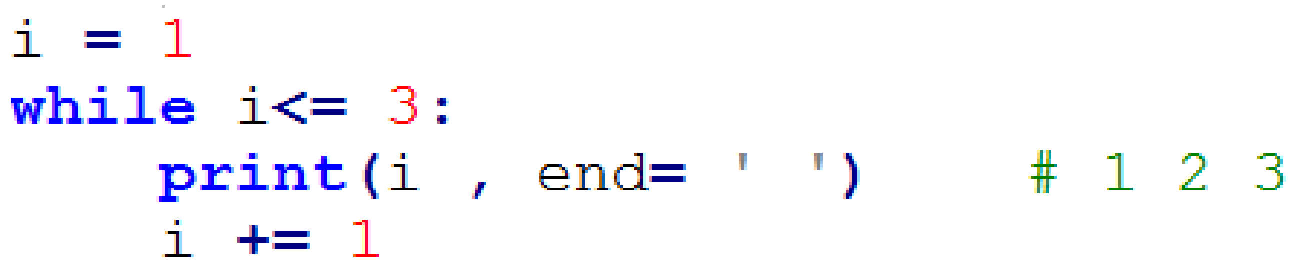
break

```
for i in range(5):
    if i == 3 :
        break
    else:
        print(i,end=' ') # 0 1 2
```

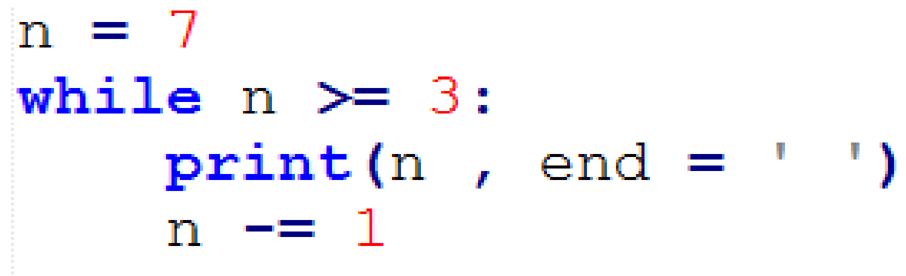
continue

```
for i in range(5):
    if i == 3 :
        continue
    else:
        print(i,end=' ')   # 0 1 2 4
```

while
Examples

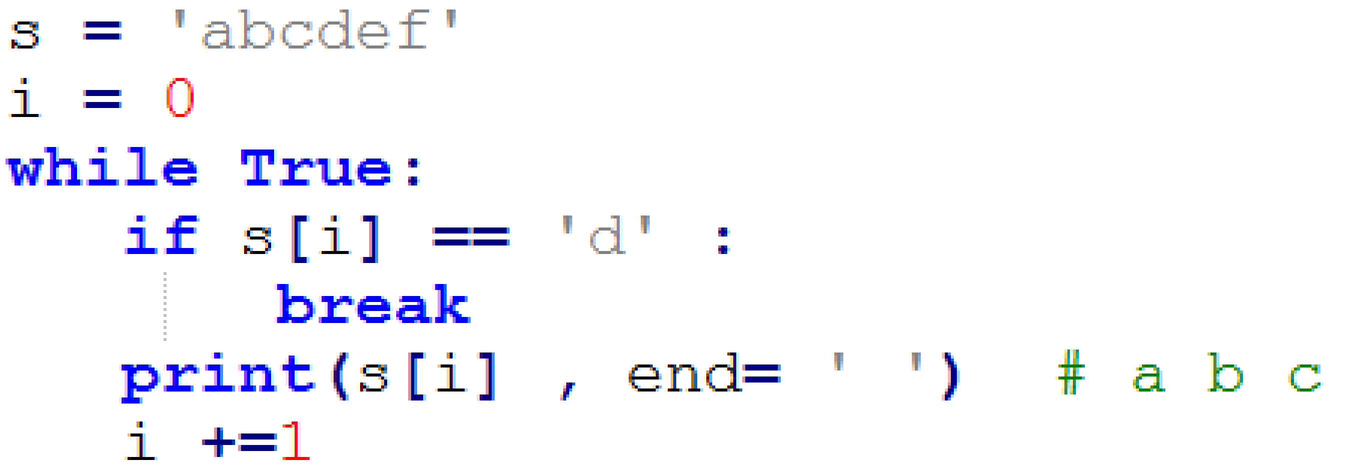


```
i = 1
while i<= 3:
    print(i , end= ' ')     # 1 2 3
    i += 1
```



```
n = 7
while n >= 3:
    print(n , end = ' ')
    n -= 1
```

while
break
Examples



```
s = 'abcdef'
i = 0
while True:
    if s[i] == 'd' :
        break
    print(s[i] , end= ' ')  # a b c
    i +=1
```

while
break
continue
Examples

```
n = 8                          # while - break
while n > 2:
    n -= 1
    if n == 5:
        break
    print(n , end = ' ')       #7 6
```

```
n = 8                          # while - continue
while n > 2:
    n -= 1
    if n == 5:
        continue
    print(n , end = ' ')       #7 6 4 3 2
```

PEP 8

It is not standard of PEP8

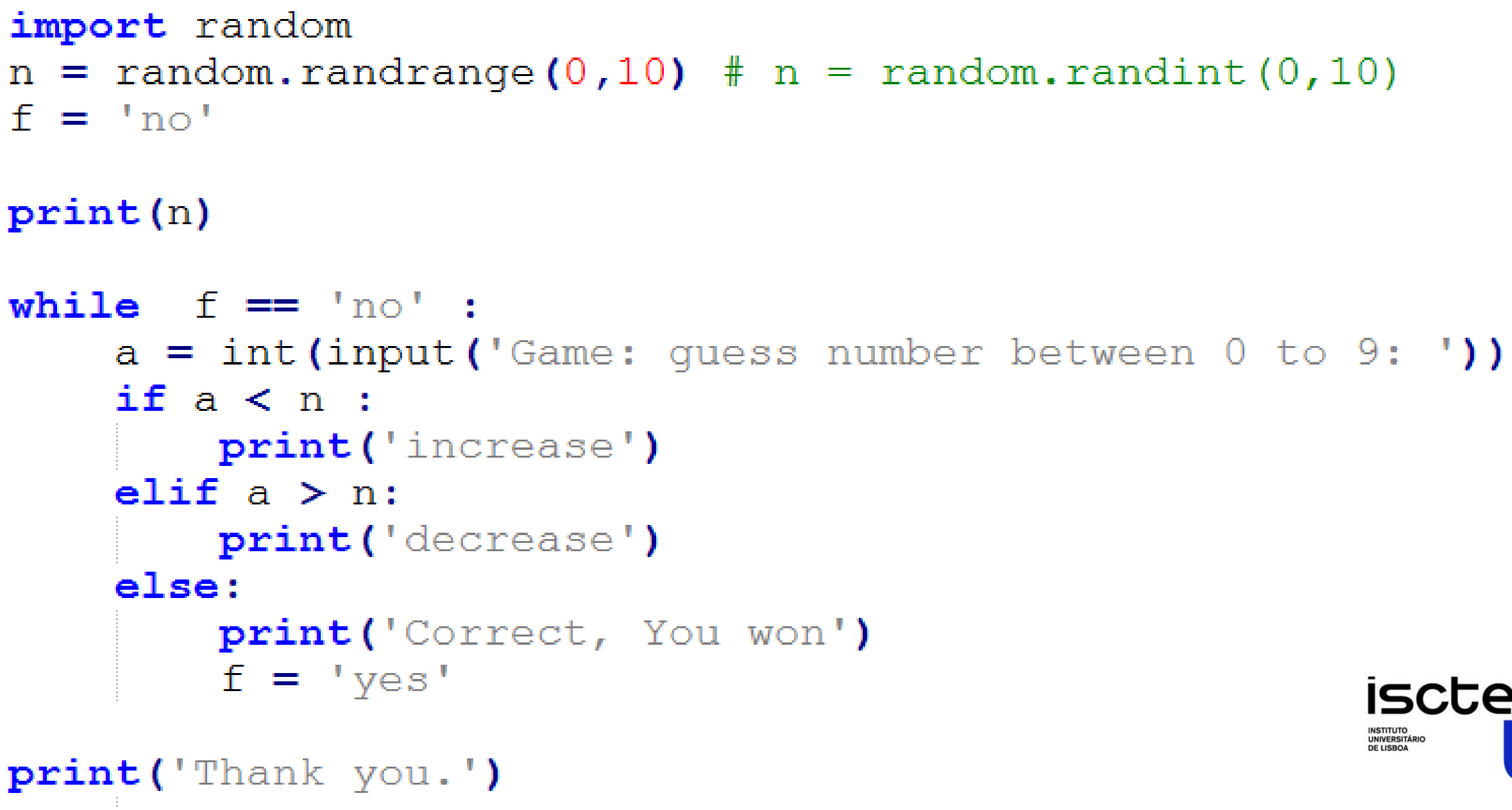
```
n = 1
while n <= 3 :  print(n) ; n+=1
```

It is standard (PEP8)

```
n = 1
while n <= 3 :
    print(n)
    n+=1
```

<Let’s Write a Game>

Game Guess the number between “0” to “9”



```python
import random
n = random.randrange(0,10) # n = random.randint(0,10)
f = 'no'

print(n)

while  f == 'no' :
    a = int(input('Game: guess number between 0 to 9: '))
    if a < n :
        print('increase')
    elif a > n:
        print('decrease')
    else:
        print('Correct, You won')
        f = 'yes'

print('Thank you.')
```

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA emprego digital

8.

Usefull links

- https://www.anaconda.com/products/individual/get-started
- https://www.python.org/
- https://github.com/
- https://git-scm.com/
- https://about.gitlab.com/
- https://bitbucket.org/dashboard/overview
- https://stackoverflow.com/

“

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

O futuro profissional começa aqui

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

UPskill