PROGRAMMING 1 - WEEK 5

Tuples

Lists

UC Leuven Limburg
MOVING MINDS

Recap

- Arithmetic
- Booleans
- Conditionals
- None
- Strings

- Loops

Permanent Evaluation

- Be quiet - no talking
- Don't cheat, look at your own screen
- ONLY allowed to be on ANS
  - No GenAI – No VSCode – No ppt – No notes – ...
- Browser full screen, display light 100%
- Scrap paper: strict rules (ask lecturer before test starts)
- Duration: 10 minutes
- Guess correction: - 1 / (# options – 1)
- Close your laptop when finished

PROGRAMMING 1 - WEEK 5

Tuples

Lists

Collections

- Basic building blocks = int, float, bool, string

- Glueing them together = collection
  - **Tuples** ~~/ Named Tuples~~
  - Lists
  - Sets
  - Dictionaries
  - …

Tuples

A tuple is an object that can hold an arbitrary number of other objects

```
x = ( a , b , c , d , ...)
```

name of the tuple

a, b, c, d … are its content

( )

- int
- float
- boolean
- string
- Tuple (or other collections)

or any combination of the above

Tuples - Examples

```
point2D = (1,2)
```

```
vowels = ('a','e','i','o','u','y')
```

```
card = (8, 'hearts')
```

```
hand_of_cards = ((8, 'hearts'),(3,'clubs'),(12,'spades'))
```

```
tuple_one_element = (3,)
```

Tuples - Indexing

```
color_rgb = (250, 200, 50)
```

Each item in a tuple can be referenced with an index:

```
color_red = color_rgb[0]
color_green = color_rgb[1]
color_blue = color_rgb[2]
```

```
color_blue = color_rgb[-1]
color_gb = color_rgb[1:3]
length = len(color_rgb)
```

Destructuring:

```
color_red, color_green, color_blue = color_rgb
```

Tuples - Membership

```
3 in (1, 2, 3, 4, 5)    # True

6 in (1, 2, 3, 4, 5)    # False
```

```
example = (1,(2,3))
len(example)    #2
3 in example    # False
```

```
example = (1,2,(3,))
len(example)    # 3
3 in example    # False
(3,) in example # True
```

Tuples – Iteration

```
for item in (1, 2, 3, 4, 5):
    print(item)
```

```
colors = ((10,16,200),
          (56,104,30),
          (230,154,32))

for color in colors:
    r = color[0]
    g = color[1]
    b = color[2]
    ...
```

```
colors = ((10,16,200),
          (56,104,30),
          (230,154,32))

for r,g,b in colors:
    ...
```

Tuples - DEMO

An entrance exam consists of a combination of tests, each one being graded on 20. The rules for passing are

- It is allowed to skip at most one test.
- The average of the tests that were taken must be at least 12.

A skipped test will be represented by the value None.

Write a function **entrance_exam(grades)** with grades being a tuple. The function returns a tuple:

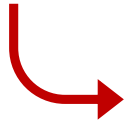
(pass , average)

Pass is True if the given grades are good enough to pass, False otherwise.

Tuples - Functions

UC Leuven Limburg
MOVING MINDS

min(t) , max(t) , sum(t)



when you pass a single argument, these functions expect that argument to be iterable

```
t = (5,2,3,9)

min(t)      # 2
max(t)      # 9
sum(t)      # 19
```

sorted(t)

returns a sorted list of the elements of t
ONLY if the values are comparable!

```
t = (5,2,3,9)
sorted(t)  # [2,3,5,9]
```

PROGRAMMING 1 - WEEK 5

Tuples

Lists

Collections

- Basic building blocks = int, float, bool, string

- Glueing them together = collection
  - Tuples
  - **Lists**
  - Sets
  - Dictionaries
  - …

Lists

Lists are very much like tuples, except they can be modified

```
x = [ a , b , c , d , ...]
```

[ ]

name of the list

a, b, c, d … are its content

- int
- float
- boolean
- string
- Tuple or List (or other collections)

or any combination of the above

Lists - Examples

```
point2D = [1,2]
```

```
vowels = ['a','e','i','o','u','y']
```

```
card = [8, 'hearts']
```

```
hand_of_cards = [[8, 'hearts'],[3,'clubs'],[12,'spades']]
```

```
list_one_element = [3]
```

Lists - functionality

All functionality available on tuples also works on lists:

len(), indexing, slicing, sum(), min(), max(),

iteration, destructuring …

(Actually, tuples have a subset of the functionality of lists)

Extra functionality:

updating, adding, removing …

Lists – update & add

```python
lst = [1,2,3,4]

lst[0] = 5
#lst = [5,2,3,4]

lst[-1] *= 2
# lst = [5,2,3,8]
```

```python
xs = [1,2,3]

xs.append(4)
# xs = [1,2,3,4]

xs.insert(0,9)
# xs = [9,1,2,3,4]
```

Lists – remove

```
xs = [1,2,3,4,5]

xs.pop(1)
# xs = [1,3,4,5]

xs.pop()
# xs = [1,3,4]
```

```
xs = [4,2,3,4,9,4]

xs.remove(2)
# xs = [4,3,4,9,4]

xs.remove(4)
xs = [3,4,9,4]
```

```
xs = [1,2,3,4,5]

del xs[-1]
# xs = [1,2,3,4]

del xs[:2]
xs = [3,4]
```

Lists - Example

```
xs = [5,3,8]

xs.append(2)
xs.insert(1,2)
xs.pop()
xs[2] = 4
xs.append(2)
xs.remove(2)
del xs[:1]
```

Lists - DEMO

An entrance exam consists of a combination of tests, each one being graded on 20. The rules for passing are

- It is allowed to skip at most one test.
- The average of the tests that were taken must be at least 12.

A skipped test will be represented by the value None.

Write a function **entrance_exam(grades)** with grades being a list. The function returns a tuple:

(pass , average)

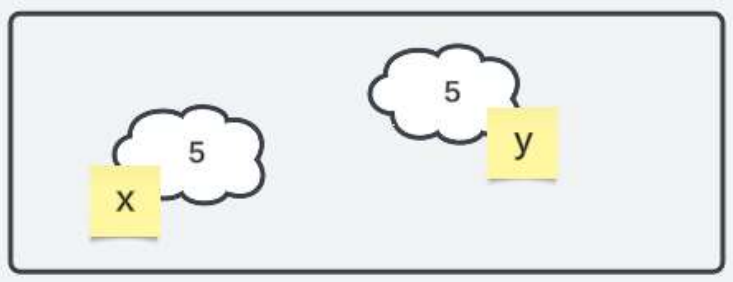
Pass is True if the given grades are good enough to pass, False otherwise.

Lists - DEMO

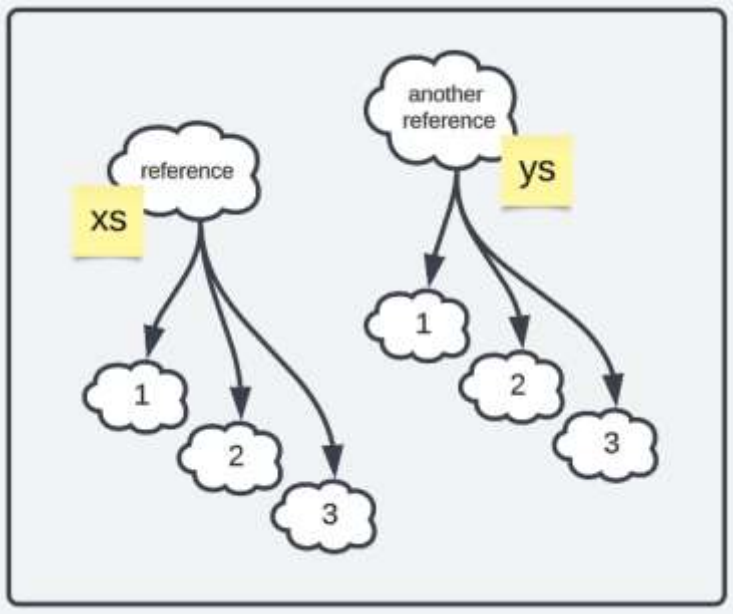
Make sure the function **entrance_exam(grades)** does not only return a tuple with the desired information. It should also clean up the given list: all values that are None should be removed from the list

Lists – References

```
x = 5
y = 5
```

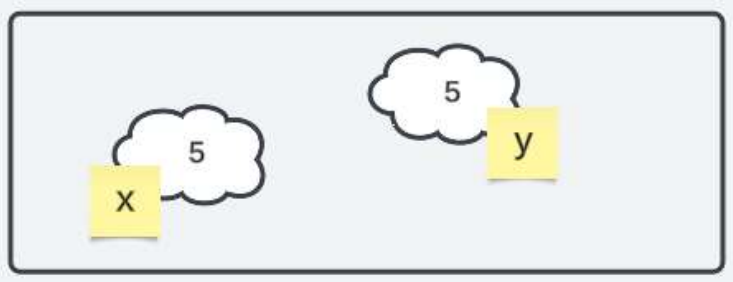



```
xs = [1, 2, 3]
ys = [1, 2, 3]
```



Lists – References

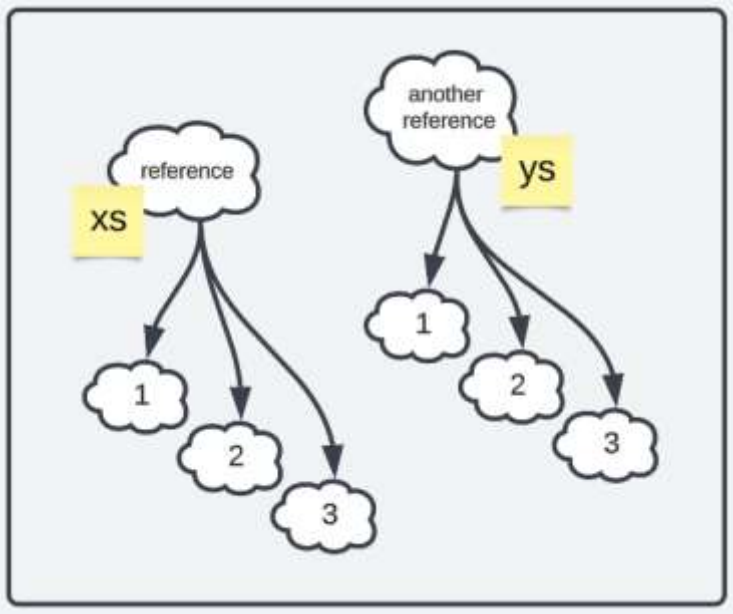
```
x = 5
y = 5
```




```
x == y      # True
```

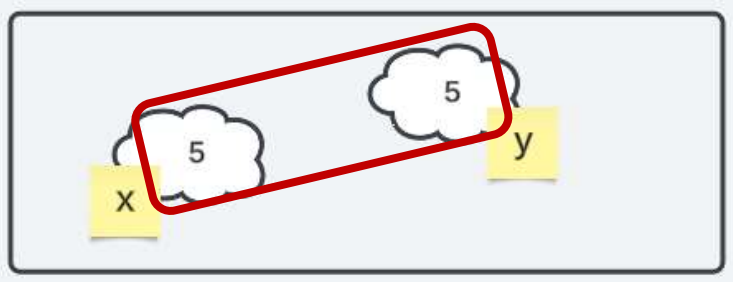
```
xs == ys    # True
```

```
xs = [1, 2, 3]
ys = [1, 2, 3]
```



Lists – References

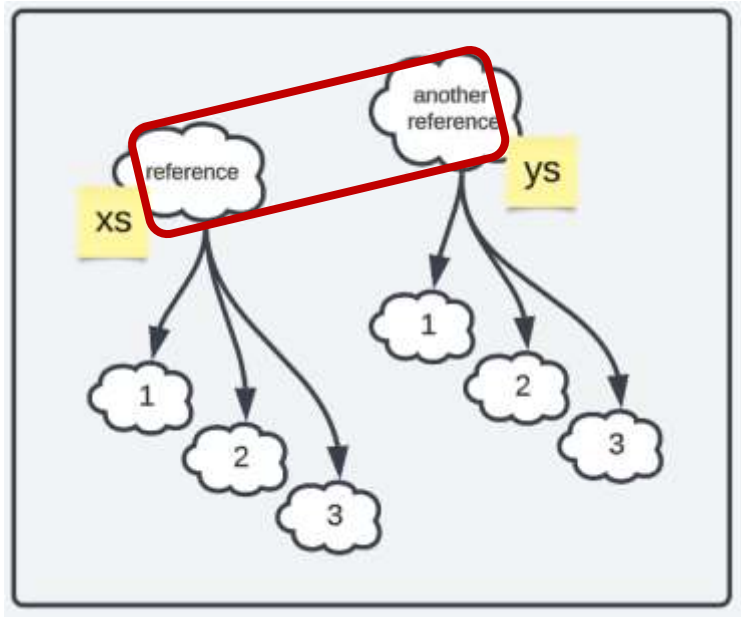
```
x = 5
y = 5
```




```
x is y      # True
```

```
xs is ys    # False
```

```
xs = [1, 2, 3]
ys = [1, 2, 3]
```

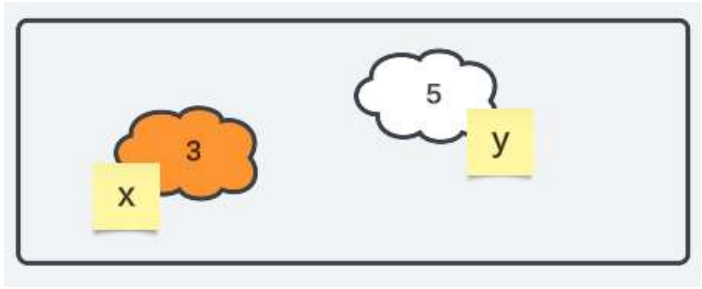


Lists – References

```
x = 5

y = x

x = 3
```




```
print(x)    # 3
print(y)    # 5
```

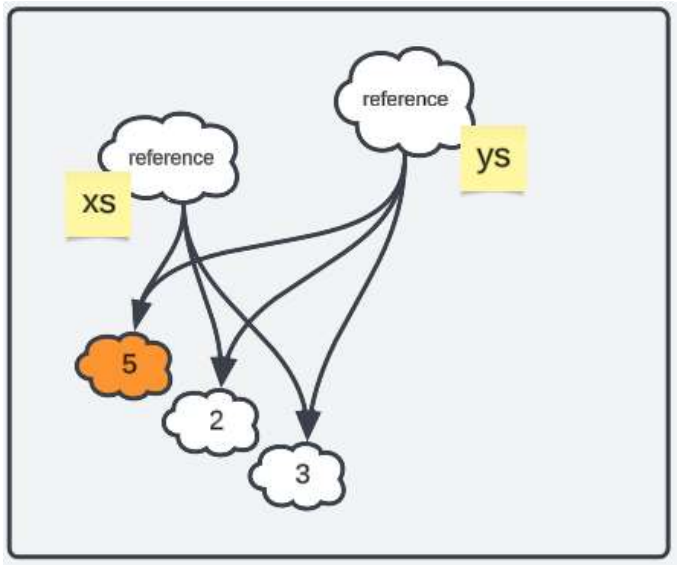
```
x is y      # False
```

Lists – References

```
xs = [1, 2, 3]

ys = xs

xs[0] = 5
```




```
print(xs)    # [5,2,3]
print(ys)    # [5,2,3]
```

```
xs is ys     # True
```

Lists – Usage

```
x = "An example"

a = list(x)          # a = ['A','n',' ','e','x','a','m','p','l','e']

b = x.split(" ")     # b = ['An','example']

c = x.split(",")     # c = ['An example']
```

Questions?

Toledo – ChatGPT

Info bij het OPO

- Planning
- Organisatie van de lessen
- Over de Evaluatie
- GenAI
  - Algemene info over GenAI
  - Hints en hulp via ChatGPT
  - Oefenen Permanente Evaluatie via ChatGPT
- ECTS-Fiche

Lists - DEMO

An entrance exam consists of a combination of tests, each one being graded on 20. The rules for passing are

- It is allowed to skip at most one test.
- The average of the tests that were taken must be at least 12.

A skipped test will be represented by the value None.

Write a function **entrance_exam(grades)** with grades being a list. The function returns a tuple: **(pass , average)**

Pass is True if the given grades are good enough to pass, False otherwise.

Make sure the function does not only return a tuple with the desired information. It should also clean up the given list: all values that are None should be removed from the list