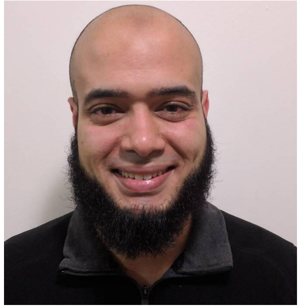
# Python Programming

# Practice

**Mostafa S. Ibrahim**

*Teaching, Training and Coaching since more than a decade!*

*Artificial Intelligence & Computer Vision Researcher*
*PhD* from Simon Fraser University - Canada
*Bachelor / Msc* from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

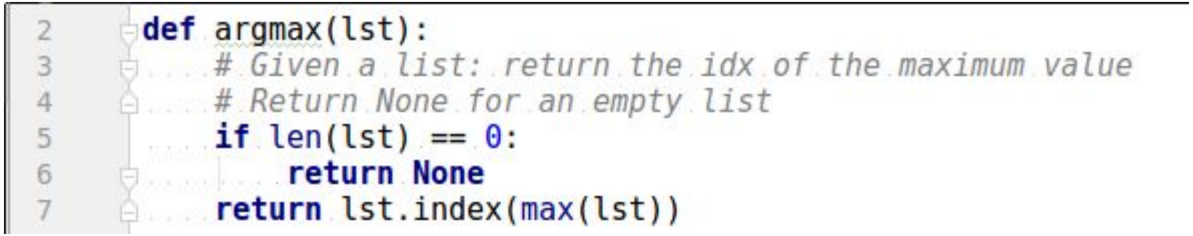


# Practice: Find the index of the top 2 maximum values

- Read a line of N integers (all on same line)
- Find the index of the the maximum and 2nd maximum values
  - If there are more than an answer: find the first match
- Input:
  - 10 **20** 3 **30** 7
    - idx1 3 value 30
    - idx2 1 value 20
  - 10 20 30 25 30 17
    - idx1 2 value 30
    - idx2 4 value 30
- Stop the video and code it

# Find the index of maximum value in a list

- max() function returns the max but not its index.
    - The index is called: argmax
- There are some efficient ways to do that.
- For simplicity we go the this approach
    - Find the maximum. Find its position. This is iterating on whole list twice. Slow, but simple

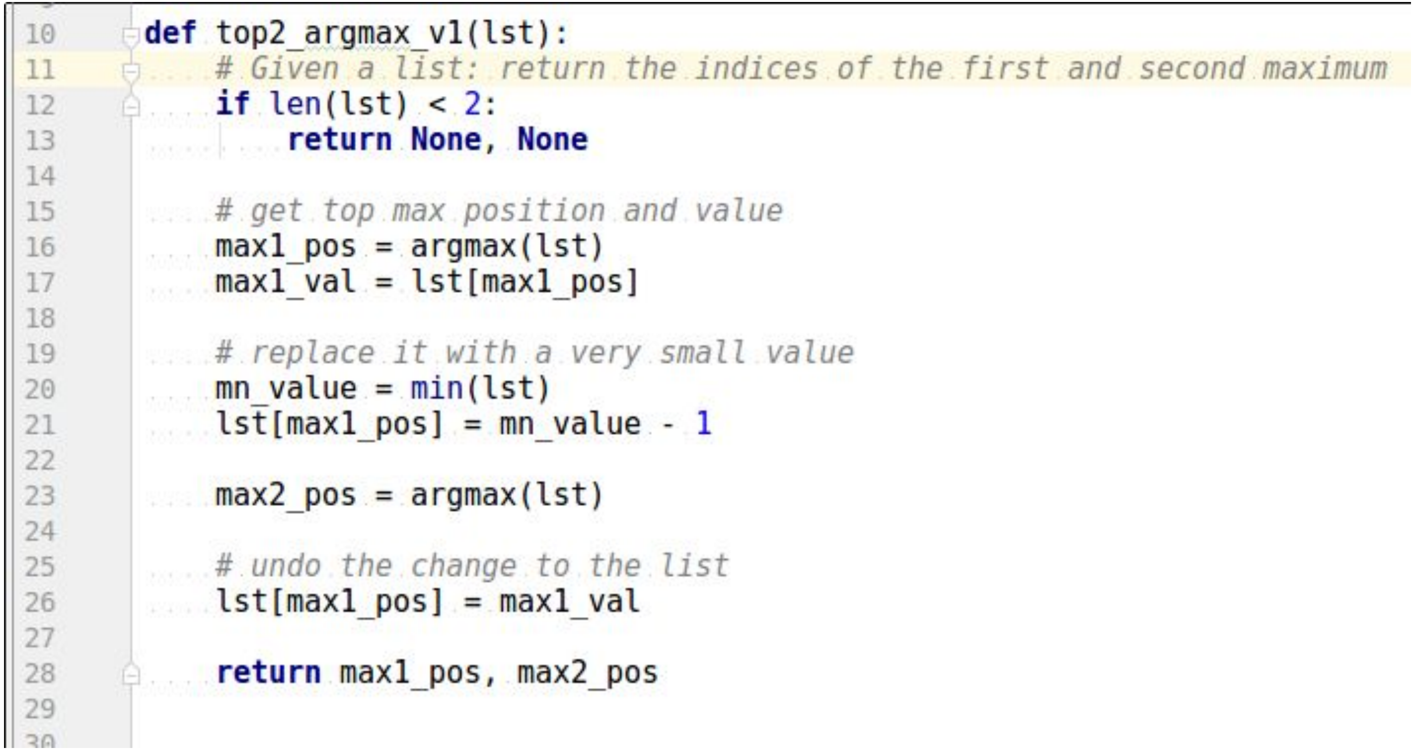


```python
def argmax(lst):
    # Given a list: return the idx of the maximum value
    # Return None for an empty list
    if len(lst) == 0:
        return None
    return lst.index(max(lst))
```

# Practice: Find the index of the top 2 maximum values

- One simple idea is as following:
- Find the argmax
- Reset its position with a very small value NOT in the list
- Find the argmax again. This now corresponds to the 2nd argmax
- Undo the list change
- Try to code it
- Drawback: many iterations on the list!

# Top max 2: Slow

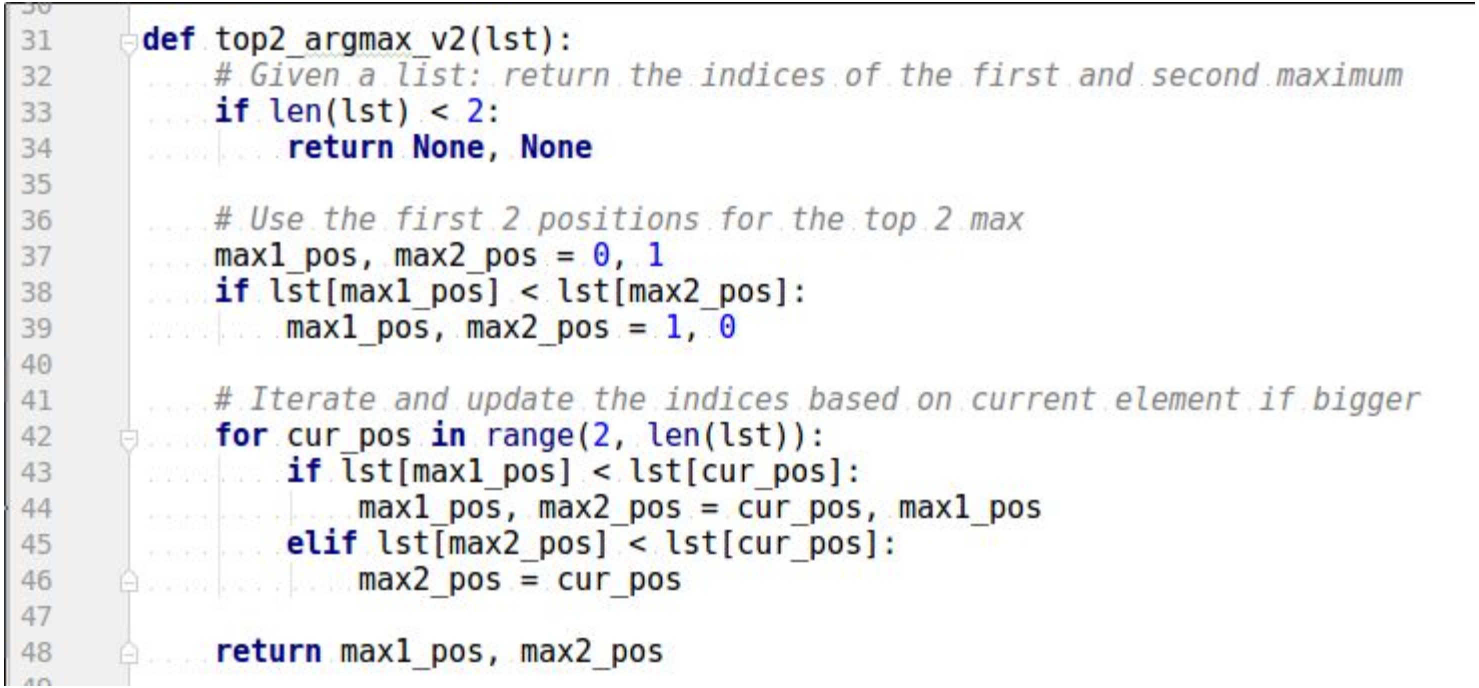


```python
def top2_argmax_v1(lst):
    # Given a list: return the indices of the first and second maximum
    if len(lst) < 2:
        return None, None

    # get top max position and value
    max1_pos = argmax(lst)
    max1_val = lst[max1_pos]

    # replace it with a very small value
    mn_value = min(lst)
    lst[max1_pos] = mn_value - 1

    max2_pos = argmax(lst)

    # undo the change to the list
    lst[max1_pos] = max1_val

    return max1_pos, max2_pos
```

# Faster solution

- Can you do an iterative code that finds the solution in a **single loop**?
- Maintain 2 variables for the 2 maximums
- Iterate on the list and update together
- Say we have so far 20 10
    - Current value is 30 ⇒ Now we should be 30 20
- Say we have so far 20 10
    - Current value is  15 ⇒ Now we should be 20 15
- Say we have so far 20 10
    - Current value is  20 ⇒ Now we should be 20 20

# Faster solution



```python
def top2_argmax_v2(lst):
    # Given a list: return the indices of the first and second maximum
    if len(lst) < 2:
        return None, None

    # Use the first 2 positions for the top 2 max
    max1_pos, max2_pos = 0, 1
    if lst[max1_pos] < lst[max2_pos]:
        max1_pos, max2_pos = 1, 0

    # Iterate and update the indices based on current element if bigger
    for cur_pos in range(2, len(lst)):
        if lst[max1_pos] < lst[cur_pos]:
            max1_pos, max2_pos = cur_pos, max1_pos
        elif lst[max2_pos] < lst[cur_pos]:
            max2_pos = cur_pos

    return max1_pos, max2_pos
```

# Tip

- Always challenge my solutions
- Are they pythonic enough?
  - The more experience in python you get, you will be amazed by what & how can be done
- Did we learn some syntax that can make it more simpler? Efficient?
- Message me for better solutions
  - Make sure to test well your code!

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”