Python Programming

List of lists - Creation

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)

Why is that useful?

- Write a program that reads grades for students
  - 100 students
  - 20 subjects
- How can we code that?
  - Create 20 lists grade1[100], grade2[100], …..grade20[100]?
  - So impractical!
  - Just create a list of lists representing the grades of each subject!

Grades visualization: 7 students x 4 subjects

	Math	Science	History	Arts
Mostafa	50	33	40	30
Asmaa	35	50	44	17
Belal	30	35	50	37
Ziad	50	35	44	22
Safa	50	44	50	30
Ashraf	50	36	18	50
Mona	35	30	**47**	16

- This is called a matrix/table
  - The blue numbers
- 7 rows
  - Row 0, 1, 2, … 6
  - Row 0 for mostafa
  - Row 6 for mona
- 4 Columns
  - Column 0, 1, 2, 3
  - Column 0 for Math
- Value of table: row 6, col 2
  - 47 (Mona & History)
  - Notation: [6][2]

Creation

```
3   mostafa_grades = [50, 33, 40, 30]
4   asmaa_grades = [35, 50, 44, 17]
5   belal_grades = [30, 35, 50, 37]
6   ziad_grades = [50, 35, 44, 22]
7   safa_grades = [50, 44, 50, 30]
8   ashraf_grades = [50, 36, 18, 50]
9   mona_grades = [35, 30, 47, 16]
10
11  grades = [mostafa_grades, asmaa_grades, belal_grades,
12            ziad_grades, safa_grades, ashraf_grades, mona_grades]
13
14  print(grades[6])        # [35, 30, 47, 16]
15  print(grades[6][2])     # 47
```

Creation

```
grades = [  [50, 33, 40, 30],
            [35, 50, 44, 17],
            [30, 35, 50, 37],
            [50, 35, 44, 22],
            [50, 44, 50, 30],
            [50, 36, 18, 50],
            [35, 30, 47, 16]]

print(grades[6])        # [35, 30, 47, 16]
print(grades[6][2])     # 47
```

Shallow copy

```
3   grades = [  [50, 33, 40, 30],
4               [35, 50, 44, 17],
5               [30, 35, 50, 37],
6               [50, 35, 44, 22],
7               [50, 44, 50, 30],
8               [50, 36, 18, 50],
9               [35, 30, 47, 16]]
10
11  # similar to slicing: this creates a new list
12  # BUT items are just assigned
13  # we call this: shallow copy
14  lst2 = grades.copy()
15
16  print(id(grades[0]))
17  print(id(lst2[0]))
18
19  # later we learn how to make deep copy
20
```

Of other data structs

```
2   lst = ['mostafa', 'saad', 'ibrahim']
3   print(lst[2])       # ibrahim
4   print(lst[2][1])    # b
5
6   lst2 = [lst, (5, 7, 2)]
7   print(lst2[0][2][1])    # b
8   print(lst2[1][1])       # 7
9
10  lst.sort()
11  print(lst)
12  # ['ibrahim', 'mostafa', 'saad']
13
14
15  lst = [[[[1]]]]
16  print(lst)                  # [[[[1]]]]
17  print(lst[0])               # [[[1]]]
18  print(lst[0][0])            # [[1]]
19  print(lst[0][0][0])         # [1]
20  print(lst[0][0][0][0])      # 1
21
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."