

```
In [1]: a = [1, 3, 5]
```

```
In [2]: b = [1, 3, 5]
```

```
In [84]: students = [  
    'Ippokratis',  
    'Polina',  
    'Elsa',  
    'Giannis',  
    'Nikos',  
    'Danai',  
    'Maria',  
    'Aimilios',  
    'Athanasia',  
    'Klara',  
    'Artemis',  
]  
import random  
def pick():  
    return random.choice(students)
```

```
In [6]: pick()
```

```
Out[6]: 'Nikos'
```

```
In [5]: a==b
```

```
Out[5]: True
```

```
In [7]: a is b
```

```
Out[7]: False
```

```
In [8]: c = b
```

```
In [9]: c is b
```

```
Out[9]: True
```

```
In [10]: b
```

```
Out[10]: [1, 3, 5]
```

```
In [11]: b[0] = 7
```

```
In [12]: b
```

```
Out[12]: [7, 3, 5]
```

```
In [13]: pick()
```

```
Out[13]: 'Giannis'
```

```
In [14]: c
```

```
Out[14]: [7, 3, 5]
```

```
In [15]: a == b
```

```
Out[15]: False
```

```
In [16]: c == b
```

```
Out[16]: True
```

```
In [17]: a = 'Mitsos'
```

```
In [18]: b = a
```

```
In [20]: a = a+' P'
```

```
In [21]: a
```

```
Out[21]: 'Mitsos P'
```

```
In [22]: b
```

```
Out[22]: 'Mitsos'
```

```
In [ ]: a=b
```

```
In [24]: type(a) == str
```

```
Out[24]: True
```

```
In [25]: type(a) is str
```

```
Out[25]: True
```

```
In [27]: c
```

```
Out[27]: [7, 3, 5]
```

```
In [28]: type(c) is list
```

```
Out[28]: True
```

```
In [29]: c
```

```
Out[29]: [7, 3, 5]
```

```
In [30]: d = [c,c,c,c,c,c]
```

```
In [31]: d
```

```
Out[31]: [[7, 3, 5], [7, 3, 5], [7, 3, 5], [7, 3, 5], [7, 3, 5], [7, 3, 5], [7, 3, 5]]
```

```
In [33]: d[1][1]=8
```

```
In [34]: d
```

```
Out[34]: [[7, 8, 5], [7, 8, 5], [7, 8, 5], [7, 8, 5], [7, 8, 5], [7, 8, 5], [7, 8, 5]]
```

```
In [36]: nick()
```

```
Out[36]: 'Polina'
```

```
In [37]: d[1] = 8
```

```
In [38]: d
Out[38]: [[7, 8, 5], 8, [7, 8, 5], [7, 8, 5], [7, 8, 5], [7, 8, 5], [7, 8, 5]]

In [40]: 'hello'.replace('l', 'i')
Out[40]: 'hello'

In [41]: a = 'mitsos'

In [42]: a.replace('os', 'i')
Out[42]: 'mitsi'

In [44]: print(a)
mitsos

In [43]: nick()
Out[43]: 'Maria'

In [45]: a = a.replace('os', 'i')

In [46]: print(a)
mitsi

In [47]: 'mitsososososos'.replace('os', 'i')
Out[47]: 'mitsiiiiii'

In [49]: 'a+b+c'.split('+')
Out[49]: ['a', 'b', 'c']

In [50]: 'lkriqh wkleriqh lekghdklfahidlskfjgh skldfjgh sdklfjg '.split(' ')
Out[50]: ['lkriqh', 'wklerjgh', 'lekghdklfghjdlskfjgh', 'skldfjgh', 'sdklfjg', '']

In [52]: ' _ sdklfjgh sldkfjgh _ sdklfik ghs f fa hik hkl sdhiklh ikl _ sdfhkjgh lsdkf gh'
Out[52]: ['sdklfjgh',
'sldkfjgh',
'sdklfjk',
'ghs',
'f',
'fg',
'hjk',
'hjkl',
'sdhjklh',
'jkl',
'sdfhkjgh',
'lsdkf',
'ghskl']

In [56]: ' _ _ _ _ _ '.join(['a', 'b', 'c'])
Out[56]: 'a _ _ _ _ b _ _ _ _ c'

In [57]: ' _ sdklfjgh sldkfjghsdkljfhgsdklfjg _ '.strip()
Out[57]: 'sdklfjgh sldkfjghsdkljfhgsdklfjg'
```

```
In [58]: a = '''
jgsfklgsdklfjghskldfghsldkfghsldkfjghskldjfg
'''
```

```
In [60]: a.strip()
```

```
Out[60]: 'jgsfklgsdklfjghskldfghsldkfghsldkfjghskldjfg'
```

```
In [65]: a=['a', 'b', 'c']
r = ''
for x in a:
    r += '+' + x
r
```

```
Out[65]: '+a+b+c'
```

```
In [ ]:
```

```
In [66]: a = [3, 6, 8, 9]
```

```
In [67]: for index in range(0, len(a)):
        print (a[index])
3
6
8
9
```

```
In [68]: for x in a:
        print (x)
3
6
8
9
```

```
In [70]: list(enumerate(a))
```

```
Out[70]: [(0, 3), (1, 6), (2, 8), (3, 9)]
```

```
In [72]: for index, element in enumerate(a):
        print (element)
3
6
8
9
```

```
In [73]: a
```

```
Out[73]: [3, 6, 8, 9]
```

```
In [74]: list(enumerate(a))
```

```
Out[74]: [(0, 3), (1, 6), (2, 8), (3, 9)]
```

```
In [75]: b = [8, 7, 6, 5]
```

```
In [108]: a
```

```
Out[108]: [[1, 2], [5, 6], [8, 9], [11, 12]]
```

```
In [77]: b
```

```
Out[77]: [8, 7, 6, 5]
```

```
In [78]: list(zip(a,b))
```

```
Out[78]: [(3, 8), (6, 7), (8, 6), (9, 5)]
```

DICTIONARIES

```
In [79]: a = [4,5,6,7,8,9]
```

```
In [81]: 7 in a
```

```
Out[81]: True
```

```
In [116]: b = {  
    '1': 5,  
    'mitsos': 'kostas',  
    9: 7.8,  
}
```

```
In [85]: b['1']
```

```
Out[85]: 5
```

```
In [86]: b['mitsos']
```

```
Out[86]: 'kostas'
```

```
In [87]: b[9]
```

```
Out[87]: 7.8
```

```
In [89]: b['1'] = 77
```

```
In [90]: b
```

```
Out[90]: {'1': 77, 'mitsos': 'kostas', 9: 7.8}
```

```
In [93]: len(b)
```

```
Out[93]: 3
```

```
In [96]: len({})
```

```
Out[96]: 0
```

```
In [97]: b
```

```
Out[97]: {'1': 77, 'mitsos': 'kostas', 9: 7.8}
```

```
In [99]: b.keys()
```

```
Out[99]: dict_keys(['1', 'mitsos', 9])
```

```
In [100]: b.values()
```

```
Out[100]: dict_values([77, 'kostas', 7.8])
```

```
In [101]: b[5,6] = 'hello'
```

```
In [103]: b
Out[103]: {'1': 77, 'mitsos': 'kostas', 9: 7.8, (5, 6): 'hello'}
```

```
In [104]: b[[5,6]] = '34534534'

-----
TypeError                                Traceback (most recent call last)
<ipython-input-104-9ecd564edbba> in <module>
----> 1 b[[5,6]] = '34534534'

TypeError: unhashable type: 'list'
```

```
In [105]: a=[[1,2], [5,6], [8,9], [11,12]]
```

```
In [106]: dict(a)
Out[106]: {1: 2, 5: 6, 8: 9, 11: 12}
```

```
In [107]: dict([[1,2], [5,6], [8,9], [11,12,13]])

-----
ValueError                                Traceback (most recent call last)
<ipython-input-107-2aac9227b7cb> in <module>
----> 1 dict([[1,2], [5,6], [8,9], [11,12,13]])

ValueError: dictionary update sequence element #3 has length 3; 2 is required
```

hash tables

```
In [109]: a = [1,2,3,4]
          b = [4,5,6,7]
In [112]: dict(zip(a,b))
Out[112]: {1: 4, 2: 5, 3: 6, 4: 7}
```

```
In [114]: list(zip(a,b))
Out[114]: [(1, 4), (2, 5), (3, 6), (4, 7)]
```

```
In [118]: b
Out[118]: {'1': 5, 'mitsos': 'kostas', 9: 7.8}
```

```
In [126]: for x in b:
          print (x)
          print (b[x])
```

```
1
5
mitsos
kostas
9
7.8
```

In [129]: `b['zfg']`

```
-----
KeyError                                Traceback (most recent call last)
<ipython-input-129-1dbc04684b5e> in <module>
----> 1 b['zfg']

KeyError: 'zfg'
```

In [130]: `c = [1,2,3,4]`
`d = [4,5,6,7]`

In [132]: `dict(zip(c,d))`

Out[132]: {1: 4, 2: 5, 3: 6, 4: 7}

In [134]: `b`

Out[134]: {'1': 5, 'mitsos': 'kostas', 9: 7.8}

In [136]: `list(b.items())`

Out[136]: [('1', 5), ('mitsos', 'kostas'), (9, 7.8)]

In [138]: `for k,v in b.items():`
 `print(k,'-',v)`
 1 = 5
 mitsos = costas
 9 = 7.8

In [151]: `for k,l,m in [[1,2,3], [6,7,8], [8,9,4], [10,11,12]]:`
 `print(l)`
 2
 7
 9
 11

In [141]: `for k in [[1,2,3], [6,7,8], [8,9,4], [10,11,12]]:`
 `print(k)`
 [1, 2, 3]
 [6, 7, 8]
 [8, 9, 4]
 [10, 11, 12]

In [142]: `k,l,m = [1,2,3]`

In [152]: `for k in [[1,2], [6,7,8], [8,9,4,3,4,5], [10,11,12]]:`
 `print(k)`
 [1, 2]
 [6, 7, 8]
 [8, 9, 4, 3, 4, 5]
 [10, 11, 12]

In [155]: `''.join(list('sfgsdfgsdf312312394523904850293485dfg'))`

Out[155]: 'sfgsdfgsdf312312394523904850293485dfg'

In [157]: `''.join(['a', 'b', 'c'])`

Out[157]: 'abc'

```
In [158]: int('343')
```

```
Out[158]: 343
```

```
In [159]: str(2342)
```

```
Out[159]: '2342'
```

```
In [160]: len('asasasas')
```

```
Out[160]: 8
```

```
In [161]: 'asdfasdfasdf'.len()
```

```
Out[161]: 12
```

```
In [162]: import this
```

The Zen of Python, by Tim Peters

Beautiful is better than ugly.
 Explicit is better than implicit.
 Simple is better than complex.
 Complex is better than complicated.
 Flat is better than nested.
 Sparse is better than dense.
 Readability counts.
 Special cases aren't special enough to break the rules.
 Although practicality beats purity.
 Errors should never pass silently.
 Unless explicitly silenced.
 In the face of ambiguity, refuse the temptation to guess.
 There should be one-- and preferably only one --obvious way to do it.
 Although that way may not be obvious at first unless you're Dutch.
 Now is better than never.
 Although never is often better than *right* now.
 If the implementation is hard to explain, it's a bad idea.
 If the implementation is easy to explain, it may be a good idea.
 Namespaces are one honking great idea -- let's do more of those!

```
In [166]: [x*2 for x in range(10)]
```

```
Out[166]: [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```

```
In [164]: [x for x in range(10)]
```

```
Out[164]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [165]: list(range(10))
```

```
Out[165]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [167]: {x:x+1 for x in range(10)}
```

```
Out[167]: {0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10}
```

```
In [168]: {x:x for x in range(10)}
```

```
Out[168]: {0: 0, 1: 1, 2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7, 8: 8, 9: 9}
```

```
In [170]: {x:x for x in range(10) if x > 4}
```

```
Out[170]: {5: 5, 6: 6, 7: 7, 8: 8, 9: 9}
```



```
In [173]: set([1,2,3,1])
```

```
Out[173]: {1, 2, 3}
```

```
In [174]: set([4,5,6,5,4,3,23,1,23,5,7,8,99,0])
```

```
Out[174]: {0, 1, 3, 4, 5, 6, 7, 8, 23, 99}
```

```
In [175]: a=set([1,2,3,4])
          b=set([4,5,6,7])
```

```
In [176]: a & b
```

```
Out[176]: {4}
```

```
In [178]: a | b
```

```
Out[178]: {1, 2, 3, 4, 5, 6, 7}
```

```
In [180]: a - b
```

```
Out[180]: {1, 2, 3}
```

```
In [183]: b - a
```

```
Out[183]: {5, 6, 7}
```

```
In [184]: a
```

```
Out[184]: {1, 2, 3, 4}
```

```
In [185]: a = a | set([5])
```

```
In [187]: a
```

```
Out[187]: {1, 2, 3, 4, 5}
```

```
In [188]: a.add(6)
```

```
In [190]: a
```

```
Out[190]: {1, 2, 3, 4, 5, 6}
```

```
In [192]: a.add(4)
```

```
In [193]: a
```

```
Out[193]: {1, 2, 3, 4, 5, 6}
```

```
In [194]: for x in a:
          print(x)
```

```
1
2
3
4
5
6
```

```
In [196]: len(a)
```

```
Out[196]: 6
```

```
In [197]: set()
```

```
Out[197]: set()
```

```
In [199]: list(a)
```

```
Out[199]: [1, 2, 3, 4, 5, 6]
```

```
In [200]: {x*2 for x in range(10)}
```

```
Out[200]: {0, 2, 4, 6, 8, 10, 12, 14, 16, 18}
```

```
In [202]: {x%3 for x in range(10)}
```

```
Out[202]: [0, 1, 2, 0, 1, 2, 0, 1, 2, 0]
```

```
In [204]: {x%3 for x in range(10)}
```

```
Out[204]: {0, 1, 2}
```

```
In [205]: a = [
    ('kosta', 45),
    ('maria', 36),
    ('sotiri', 40),
]
```

```
In [208]: min(dict(a).values())
```

```
Out[208]: 36
```

```
In [211]: minimum_value = 1000
for x in a:
    if x[1] < minimum_value:
        minimum_value = x[1]
        name = x[0]
name
```

```
Out[211]: 'maria'
```

```
In [212]: a
```

```
Out[212]: [('kosta', 45), ('maria', 36), ('sotiri', 40)]
```

```
In [220]: new_list = [(age, name) for name, age in a]
```

```
In [217]: min([(age, name) for name, age in a])
```

```
Out[217]: (36, 'maria')
```

```
In [219]: min([(age, name) for name, age in a])[1]
```

```
Out[219]: 'maria'
```

```
In [223]: min(new_list)[1]
```

```
Out[223]: 'maria'
```

```
In [233]: new_list
```

```
Out[233]: [(45, 'kosta'), (36, 'maria'), (40, 'sotiri')]
```

```
In [234]: min(new_list)
```

```
Out[234]: (36, 'maria')
```

```
In [235]: min(new_list)[1]
```

```
Out[235]: 'maria'
```

```
In [225]: a
```

```
Out[225]: [('kosta', 45), ('maria', 36), ('sotiri', 40)]
```

```
In [228]: min(a)
```

```
Out[228]: ('kosta', 45)
```

```
In [231]: min(a)[0]
```

```
Out[231]: 'kosta'
```

```
In [232]: a
```

```
Out[232]: [('kosta', 45), ('maria', 36), ('sotiri', 40)]
```

```
In [236]: pick()
```

```
Out[236]: 'Elsa'
```

```
In [237]: sorted([[3, 4, 5], [3, 4, 4], [3, 4, 4, 5]])
```

```
Out[237]: [[3, 4, 4], [3, 4, 4.5], [3, 4, 5]]
```

```
In [ ]:
```