

In []: `##1. What exactly is []?`
The empty list value, which **is** a list value that contains no items. This **is** similar

In []: `##2. In a list of values stored in a variable called spam, how would you assign the`
`spam[2] = 'hello'` (Notice that the third value **in** a list **is** at index **2** because the f

In [1]: `##3. What is the value of spam[int(int('3' * 2) / 11)]?`
`'d'` (Note that `'3' * 2` **is** the string `'33'`, which **is** passed to `int()` before being div

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-1-8310e7aa5d3a> in <module>
      1 ##3. What is the value of spam[int(int('3' * 2) / 11)]?
----> 2 val= spam[int(int('3' * 2) / 11)]
      3 print(val)

NameError: name 'spam' is not defined
```

In []: `##4. What is the value of spam[-1]?`
`'d'` (Negative indexes count **from** the end.)

In []: `##5. What is the value of spam[:2]?`
Let's pretend bacon has the list `[3.14, 'cat', 11, 'cat', True]` for the next three q

In []: `##6. What is the value of bacon.index('cat')?`
`1`

In []: `##7. How does bacon.append(99) change the look of the list value in bacon?`
`[3.14, 'cat', 11, 'cat', True, 99]`

In []: `##8. How does bacon.remove('cat') change the look of the list in bacon?`
`[3.14, 11, 'cat', True]`

In []: `##9. What are the list concatenation and list replication operators?`
The operator **for** list concatenation **is** `+`, while the operator **for** replication **is** `*`. (

In []: `##10. What is difference between the list methods append() and insert()?`
While `append()` will add values only to the end of a list, `insert()` can add them anyw

In []: `##11. What are the two methods for removing items from a list?`
The `del` statement **and** the `remove()` list method are two ways to remove values **from** a

In []: `##12. Describe how list values and string values are identical.`
Both lists **and** strings can be passed to `len()`, have indexes **and** slices, be used **in** f

In []: `##13. What's the difference between tuples and lists?`
Lists are mutable; they can have values added, removed, **or** changed. Tuples are immut

In []: *##14. How do you type a tuple value that only contains the integer 42?*
`(42,)` (The trailing comma **is** mandatory.)

In []: *##15. How do you get a list value's tuple form? How do you get a tuple value's list*
The `tuple()` **and** `list()` functions, respectively

In []: *##16. Variables that "contain" list values are not necessarily lists themselves. Ins*
They contain references to list values.

In []: *##17. How do you distinguish between `copy.copy()` and `copy.deepcopy()`?*
The `copy.copy()` function will do a shallow copy of a list, **while** the `copy.deepcopy()`