Mini-Lecture 13

Lists (& Sequences)

Sequences: Lists of Values

String

- s = 'abc d'

0	1	2	3	4
a	b	c		d

- Put characters in quotes
  - Use \' for quote character
- Access characters with []
  - s[0] is 'a'
  - s[5] causes an error
  - s[0:2] is 'ab' (excludes c)
  - s[2:] is 'c d'

List

- x = [5, 6, 5, 9, 15, 23]

0	1	2	3	4	5
5	6	5	9	15	23

- Put values inside [ ]
  - Separate by commas
- Access **values** with []
  - x[0] is 5
  - x[6] causes an error
  - x[0:2] is [5, 6] (excludes 2nd 5)
  - x[3:] is [9, 15, 23]

Sequences: Lists of Values

String

- s = 'abc d'

0	1	2	3	4
a	b	c		d

- Put characters in quotes
  - Use \' for quote character
- Access cha
  - s[0] is 'a
  - s[5] causes ror
  - s[0:2] is 'ab' (excludes c)
  - s[2:] is 'c d'

List

- x = [5, 6, 5, 9, 15, 23]

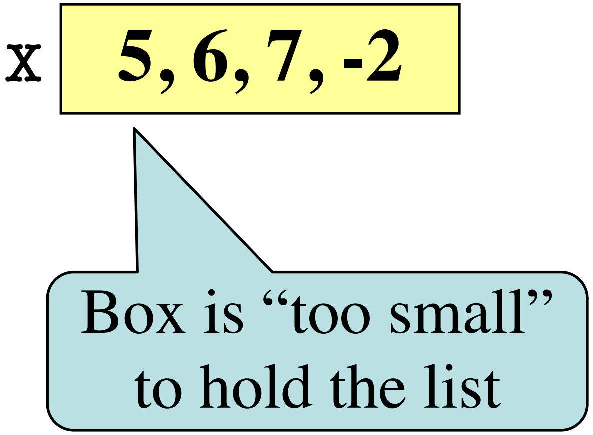
0	1	2	3	4	5
5	6	5	9	15	23

- Put values inside [ ]
  - mmas
- with []
  - [0] is 5
  - x[6] causes an error
  - x[0:2] is [5, 6] (excludes 2nd 5)
  - x[3:] is [9, 15, 23]

Sequence is a name we give to both

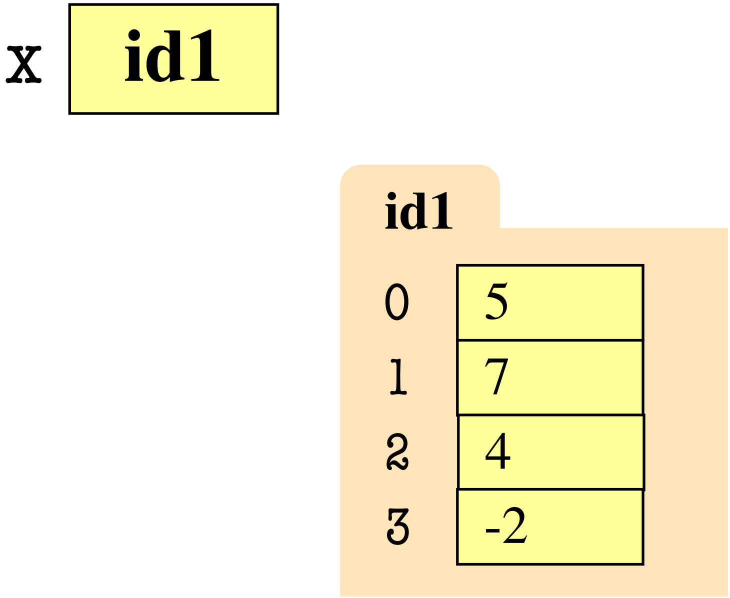
Representing Lists

Wrong



Box is “too small” to hold the list

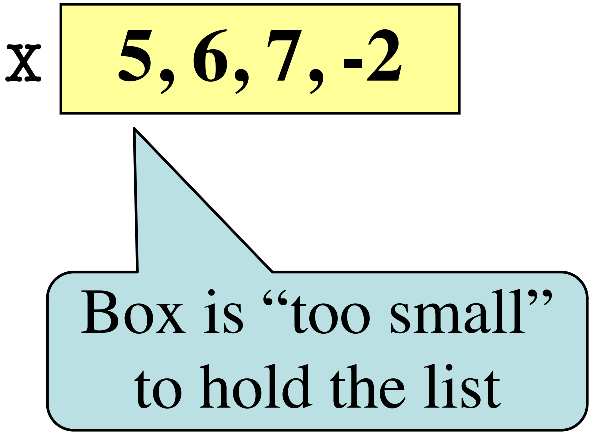
Correct



```
x = [5, 7, 4,-2]
```

Representing Lists

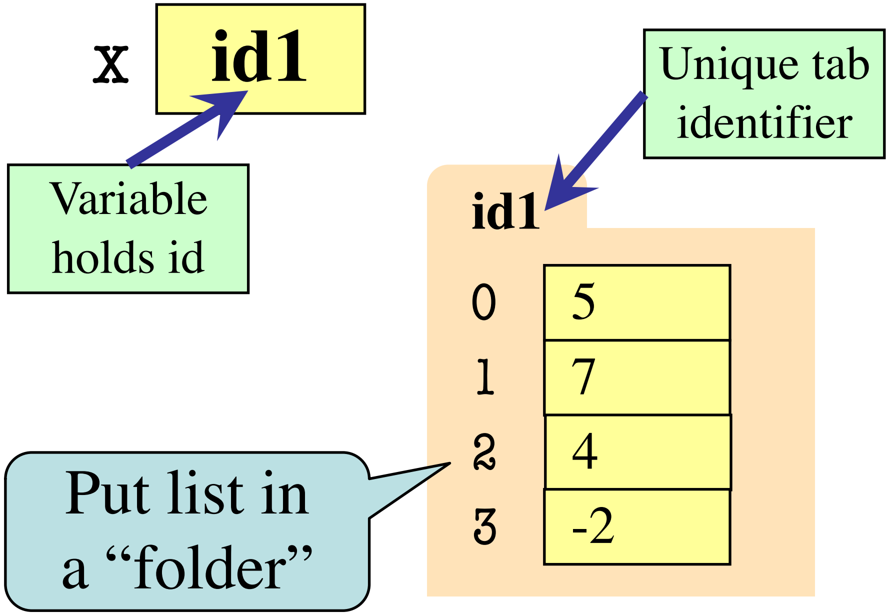
Wrong



x 5, 6, 7, -2

Box is “too small” to hold the list

Correct



x id1

Variable holds id

Unique tab identifier

id1

0	5
1	7
2	4
3	-2

Put list in a “folder”

```
x = [5, 7, 4,-2]
```

Modifying List Contents

- **List assignment**:

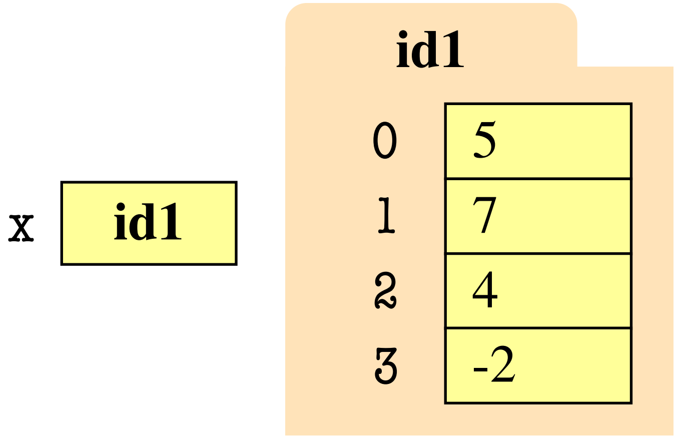
  `<var>[<index>] = <value>`

  - Reassign at index
  - Affects folder contents
  - Variable is unchanged
- Strings cannot do this
  - `s = 'Hello World!'`
  - `s[0] = 'J'` **ERROR**
  - String are **immutable**

- `x = [5, 7,4,-2]`

0	1	2	3
5	7	4	-2

- `x[1] = 8`



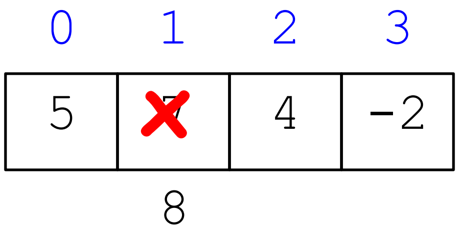
Modifying List Contents

- **List assignment**:

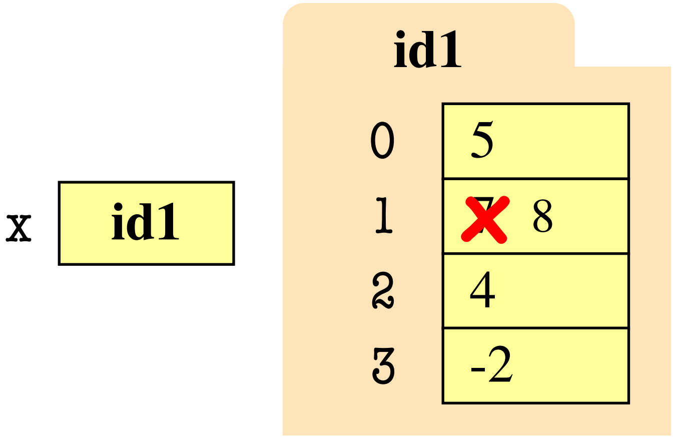
  <var>[<index>] = <value>

  - Reassign at index
  - Affects folder contents
  - Variable is unchanged
- Strings cannot do this
  - s = 'Hello World!'
  - s[0] = 'J' **ERROR**
  - String are **immutable**

- x = [5, 7,4,-2]



- x[1] = 8



Exercise: List Assignment

- Assignment copies id into y

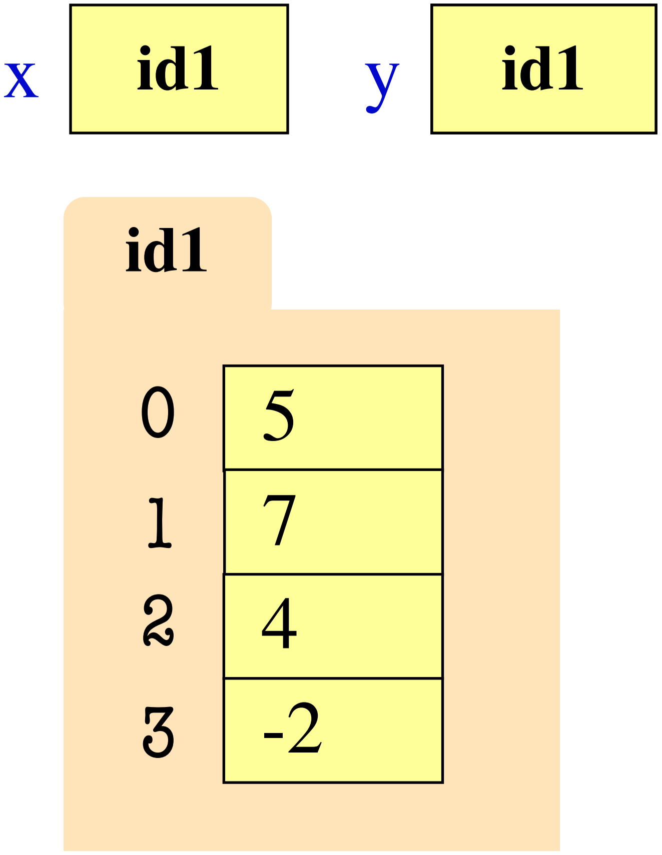
```
>>> x = [5, 7, 4, -2]
>>> y = x
```

- Execute the assignments:

```
>>> x[2] = 8
>>> y[2] = 3
```

- What is value of x[2]?

A: 8
B: 3
C: **id1**
D: I don't know



x **id1** y **id1**

id1

0	5
1	7
2	4
3	-2

Exercise: List Assignment

- Assignment copies id into y

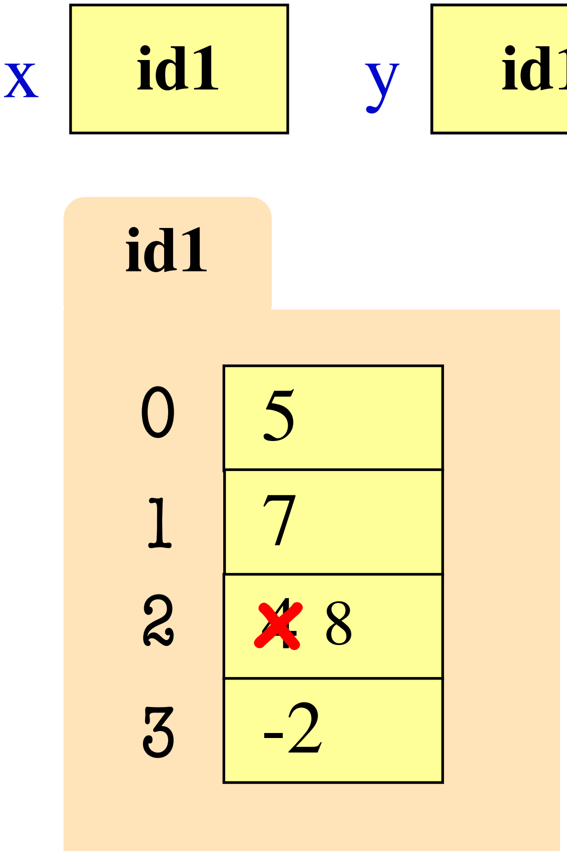
```
>>> x = [5, 7, 4, -2]
>>> y = x
```

- Execute the assignments:

```
>>> x[2] = 8
>>> y[2] = 3
```

- What is value of x[2]?

A: 8
B: 3 **CORRECT**
C: **id1**
D: I don’t know



Exercise: List Assignment

- Assignment copies id into y

```
>>> x = [5, 7, 4, -2]
>>> y = x
```

- Execute the assignments:

```
>>> x[2] = 8
>>> y[2] = 3
```

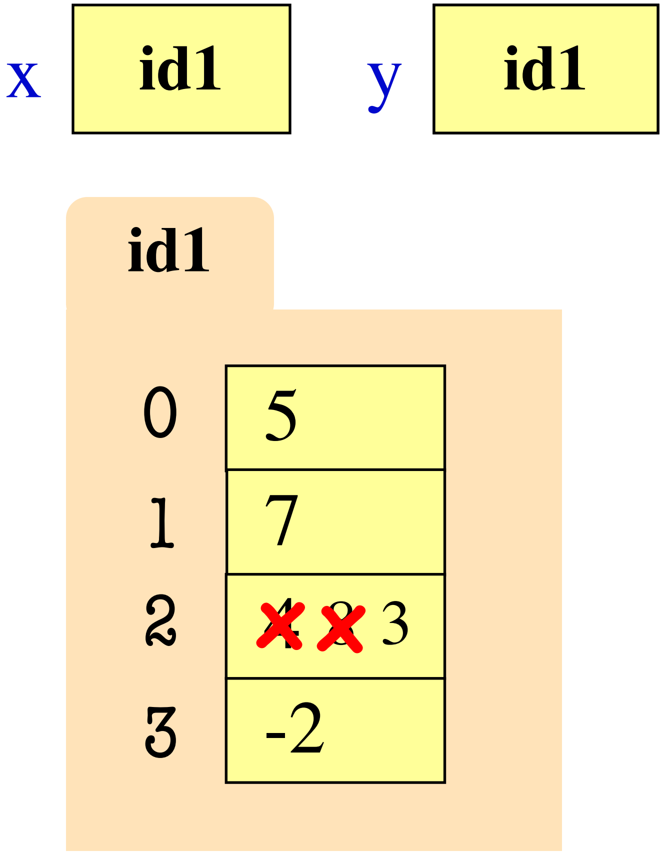
- What is value of x[2]?

A: 8

B: 3 **CORRECT**

C: **id1**

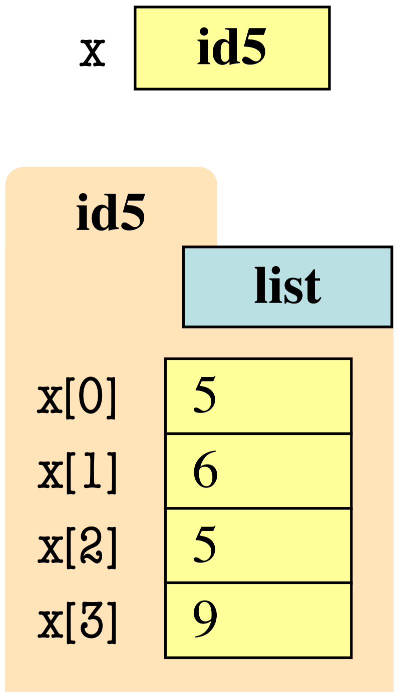
D: I don't know

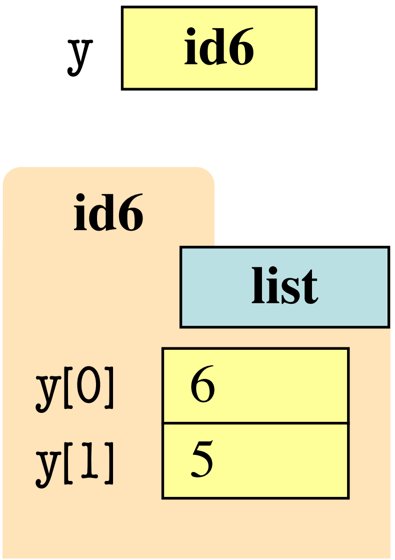


List Slices Make Copies

```
x = [5, 6, 5, 9]
```

```
y = x[1:3]
```

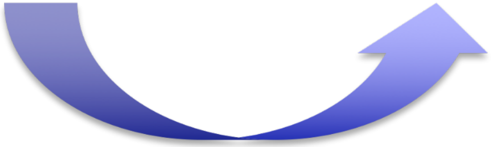




copy = new folder

Advanced List Features: Method Calls

- Function call with a “list in front”
  - Usage: ***list*.*method*(*x*,*y*…)**



The value list is an *implicit* argument

- Example: count()
  - x = [1,3,5,3,2]
  - x.count(3) == 2
  - x.count(0) == 0

Searching Lists

- x.index(y)
  - Position of the first instance of y in x
- s_1.count(s_2)
  - Number of times s_2 appears inside of s_1
- s.strip()
- Like index_str in introcs

- x = [1,3,5,3,2]
- s.index('a') == 0
- s.index('rac') == 2
- s.count('a') == 5
- s.count('b') == 2
- s.count('x') == 0
- ' a b '.strip() == 'a b'

Methods Can Alter the List

x = [5, 6, 5, 9]

- x.append(value)
  - A **procedure method**, not a fruitful method
  - Adds a new value to the end of list
  - x.append(-1) *changes* the list to [5, 6, 5, 9, -1]
- x.insert(index, value)
  - Put the value into list at index; shift rest of list right
  - x.insert(2,-1) changes the list to [5, 6, -1, 5, 9,]