Course	
Term	
Week	
Date	
Chapter. Topic	7. Lists and Tuples

Tuples

Siva R Jasthi

Computer Science and Cybersecurity

Metropolitan State University

Outline

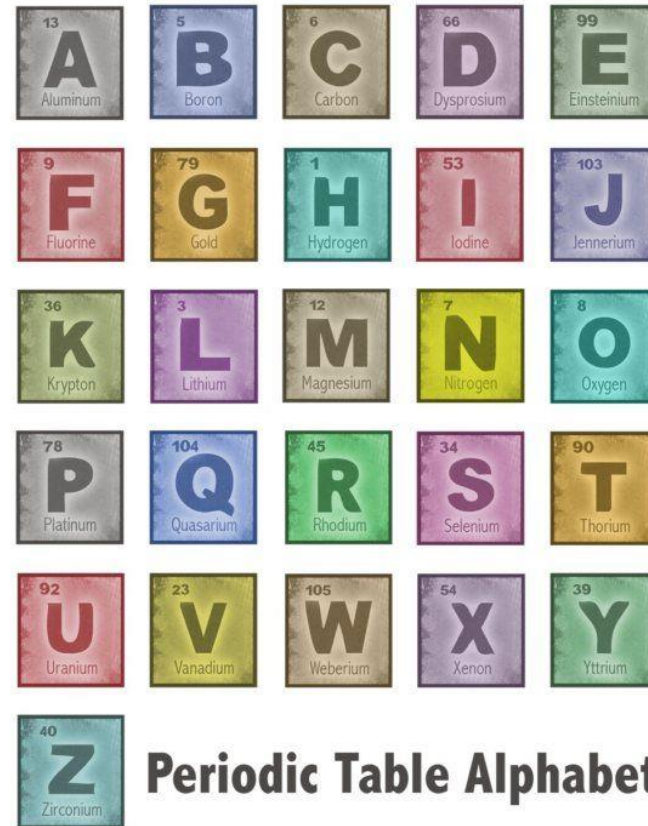- Lists
- Tuples
- Sets
- Dictionaries

are all collections

We used "Lists" so far.

We will cover "Tuples" today.

Lists vs Tuples

	Lists	Tuples	
Ordered	✔	✔	
Indexed	✔	✔	
Append or insert or Update items	✔	✗	Note: However, you can add two tuples together. You can recreate tuples
Can contain duplicates	✔	✔	
Uses	Square Brackets	Round Brackets	
	[ ]	**( )**	

When to use "tuples"?

When you have a collection that doesn't change, go with the "tuples".

"tuples" are faster compared to "lists"

Tuple

Tuple is a collection which is **ordered** and **unchangeable**. Allows **duplicate** members.

Once created, you can not add or delete items
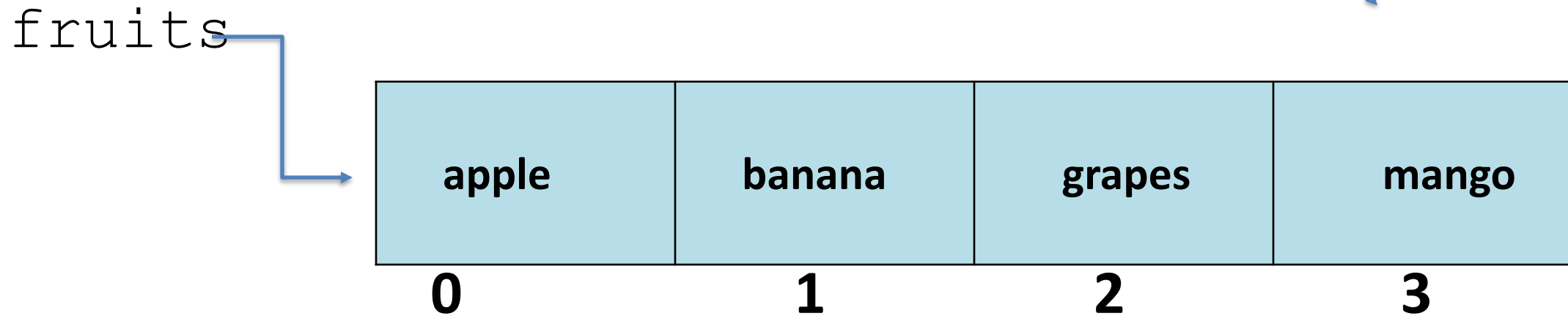
Tuples: An introduction
https://www.w3schools.com/python/python_tuples.asp

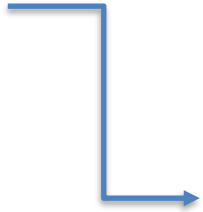# An example of a tuple

We use ROUND Brackets to indicate a Tuple.

```
fruits = ("apple", "banana", "grapes", "mango")
```

You can visualize the TUPLE as follows:

`fruits`

apple	banana	grapes	mango
0	1	2	3

tuple of integers

marks

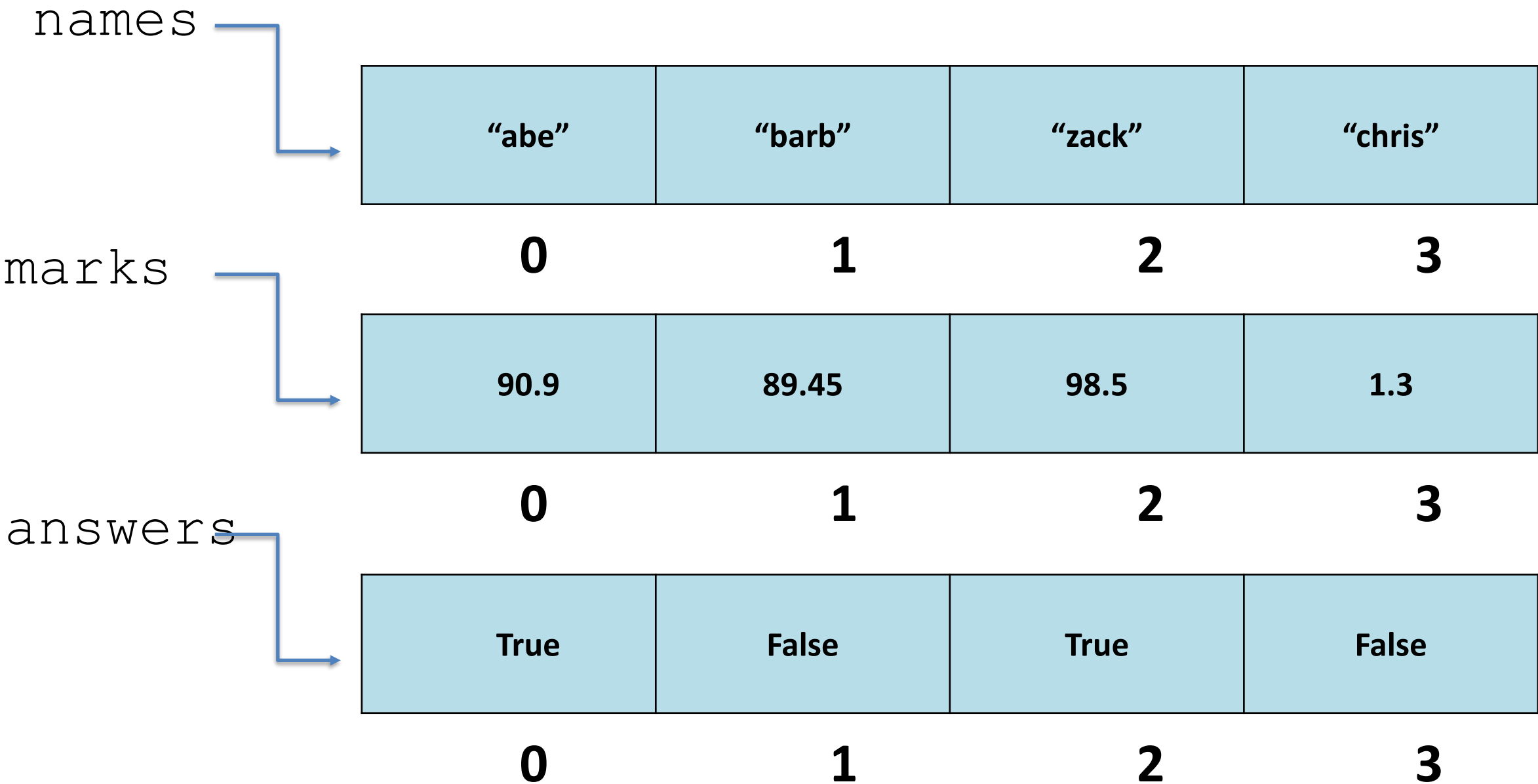| 10	8	7	0	9	11	10	10	4	5
0	1	2	3	4	5	6	7	8	9

How many students are there in the class?
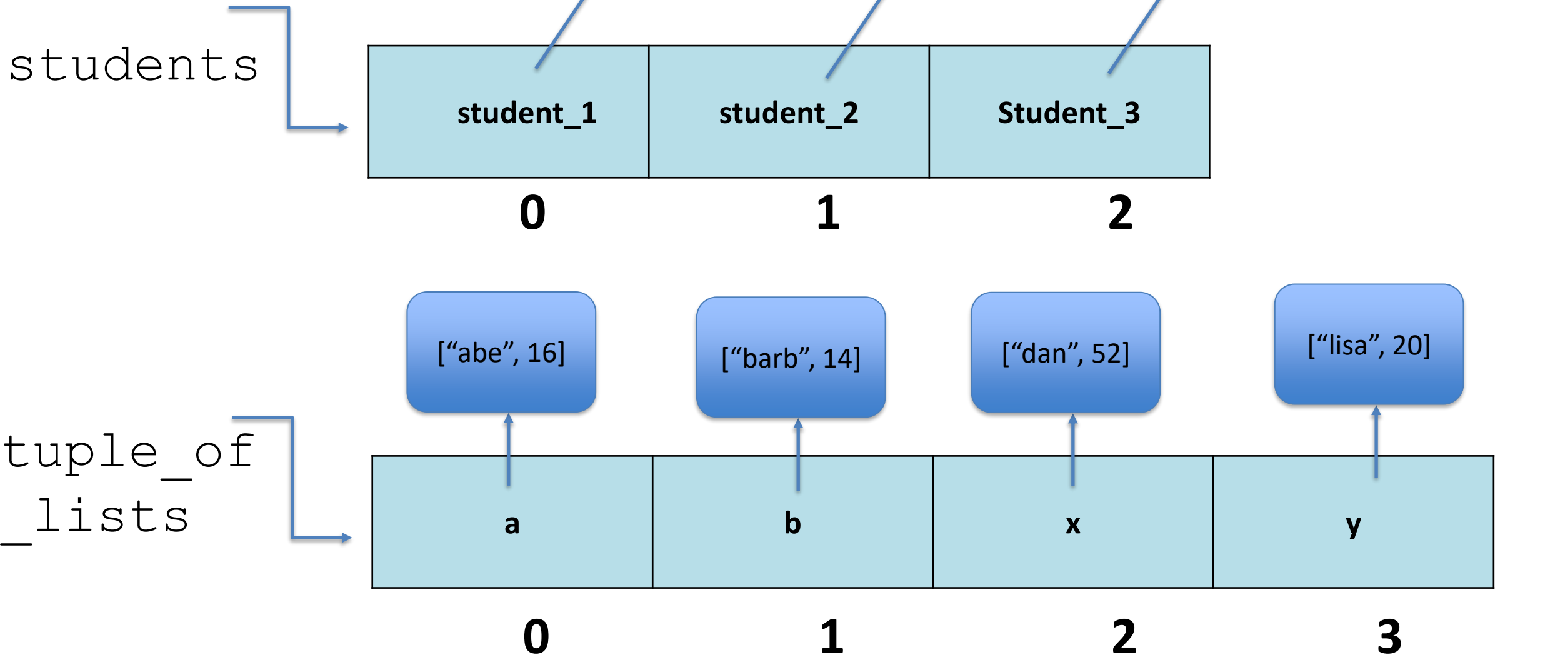What is the total marks earned by all the students together?
What is the highest score?
What is the lowest score?

tuples of (integers, strings, floats, booleans)

names

"abe"	"barb"	"zack"	"chris"
0	**1**	**2**	**3**

marks

90.9	89.45	98.5	1.3
0	**1**	**2**	**3**

answers

True	False	True	False
0	**1**	**2**	**3**

Tuples (objects, lists)

students

student_1	student_2	Student_3
0	**1**	**2**

tuple_of
_lists

["abe", 16] ["barb", 14] ["dan", 52] ["lisa", 20]

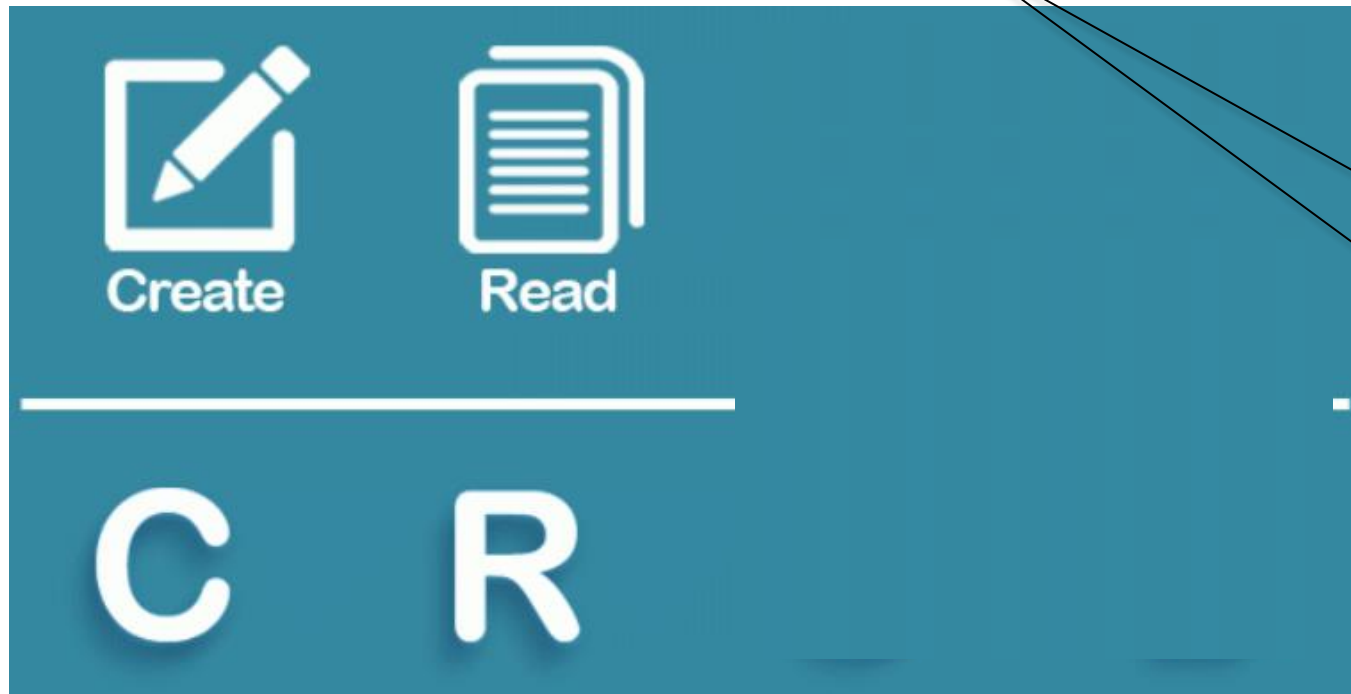| a	b	x	y
0	**1**	**2**	**3**

CRUD of Tuples

C – Create (with an initial list)

R – Read (Query, Traversal, Find, Search)

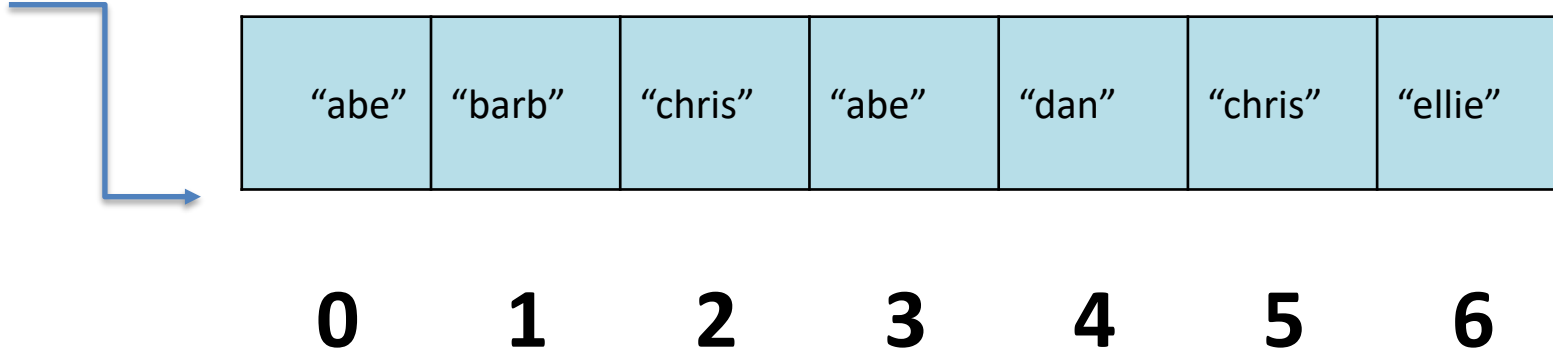~~U – Update (Modify, Change, Edit)~~

~~D – Delete (Remove, Empty)~~

You can delete the entire tuple, but you can not delete items in the tuple
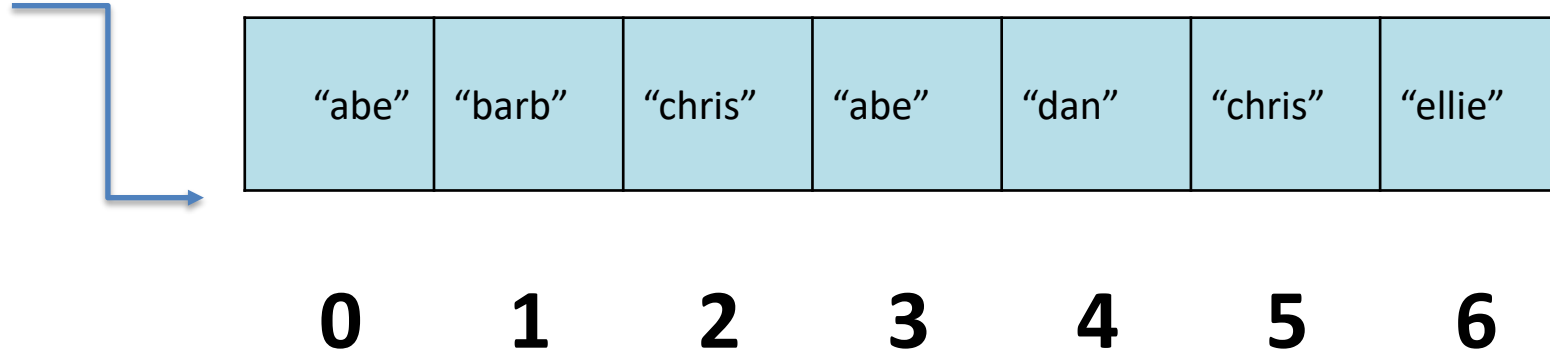
Creating a tuple

`students`

"abe"	"barb"	"chris"	"abe"	"dan"	"chris"	"ellie"
0	1	2	3	4	5	6

# tuple of students
students = ("abe", "barb", "chris", "abe", "dan", "chris", "ellie")

Reading a tuple

`students`

"abe"	"barb"	"chris"	"abe"	"dan"	"chris"	"ellie"

   0   1   2   3   4   5   6

We use **subscript** notation to access an element.

The first element sits at the index 0
The second element sits at the index 1

student_1 = students[0]
student_5 = students[6]

You can not edit a tuple.


```
Python 3.6
(known limitations)

1  list = [10,20,30]
2  list.append(40)
3
4  tuple = (10, 20, 30)
5  tuple.append(40)

Edit this code
```

➡ line that just executed
➡ next line to execute

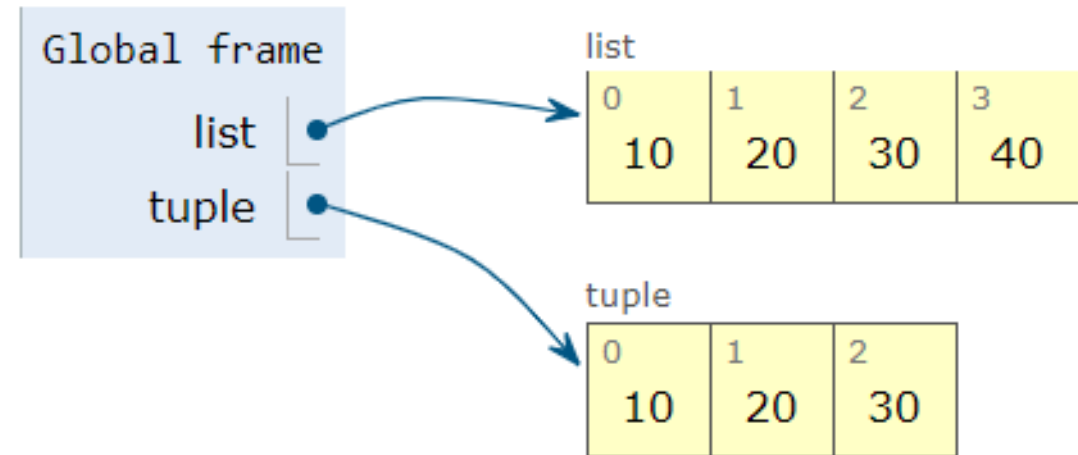<< First   < Prev   Next >   Last >>

Done running (4 steps)

AttributeError: 'tuple' object has no attribute 'append'
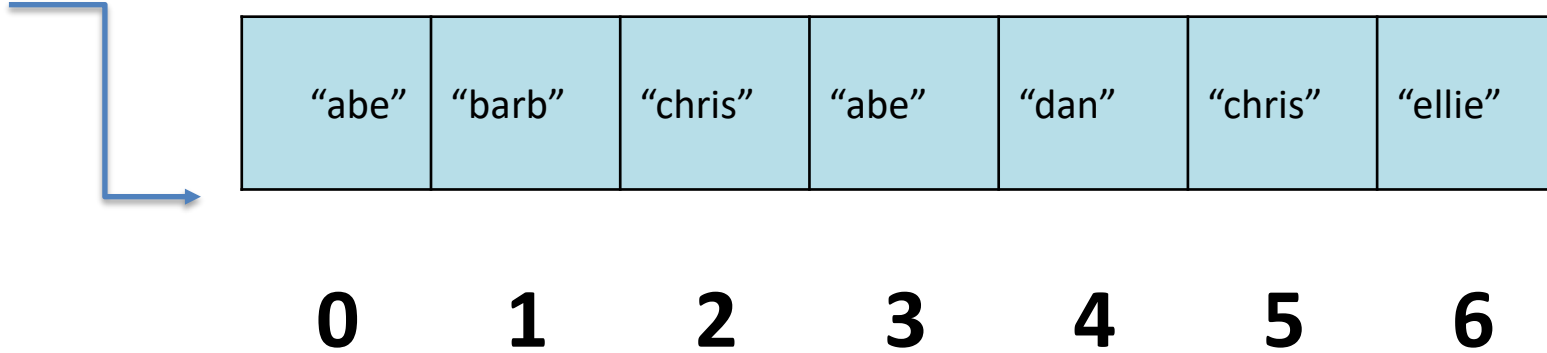(see UNSUPPORTED FEATURES)

Frames

Objects

Global frame

list ●
tuple ●

list
0	1	2	3
10	20	30	40

tuple
0	1	2
10	20	30

Tuples are unchangeable
(also called "**immutable**")

How to update/change a value? We can't!

students

"abe"	"barb"	"chris"	"abe"	"dan"	"chris"	"ellie"
0	1	2	3	4	5	6

Python 3.6
(known limitations)

```
1  tuple = ("one","three" )
2  tuple[1] = "two"
```

Edit this code

➡ line that just executed
➡ next line to execute

For example, how do we change
the name of 4th student from "abe"
to "abraham"?

students[3] = "abraham"
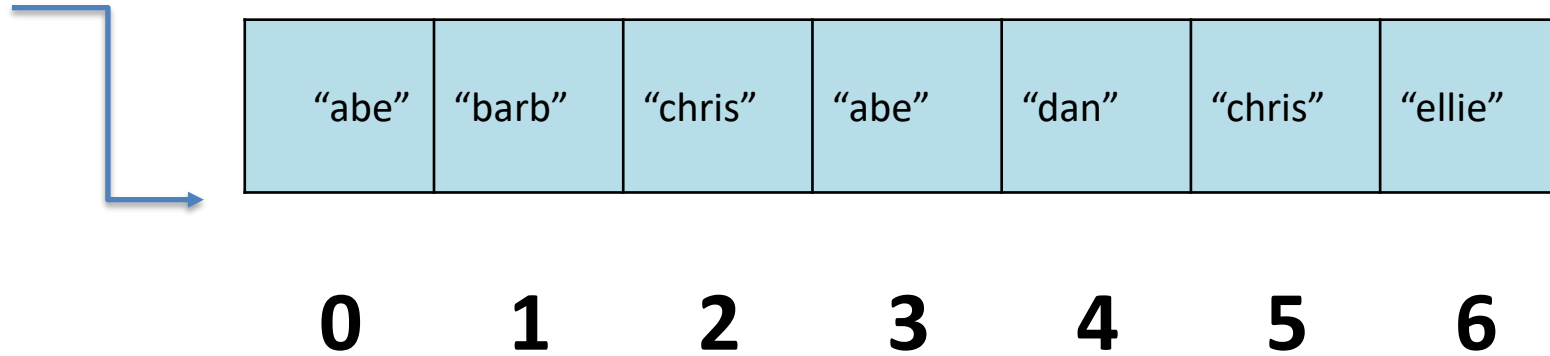
| << First | < Prev | Next > | Last >> |

Done running (2 steps)

TypeError: 'tuple' object does not support item assignment

You can reassign or delete the entire tuple

students

"abe"	"barb"	"chris"	"abe"	"dan"	"chris"	"ellie"

0   1   2   3   4   5   6

```
1  students = ("abe", "barb", "chris")
2
3  students = ("abe", "barb", "chris", "dan")
4
5  del students
6  |
```

Tuple having only one value

```python
#@title Tuples having only one item
a = 7
a = (7)
a = 3 + 4
a = (3 + 4)
print(type(a), a)

b = [5, 10]
print(type(b), b)

c = [9]
print(type(c), c)

d = (8, 4)
print(type(d), d)

e = (3)
print(type(e), e)

f = (88, )
print(type(f), f)
```
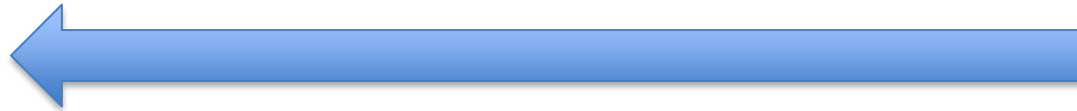
Iterating the tuple

`students`

"abe"	"barb"	"chris"	"abe"	"dan"	"chris"	"ellie"

0   1   2   3   4   5   6
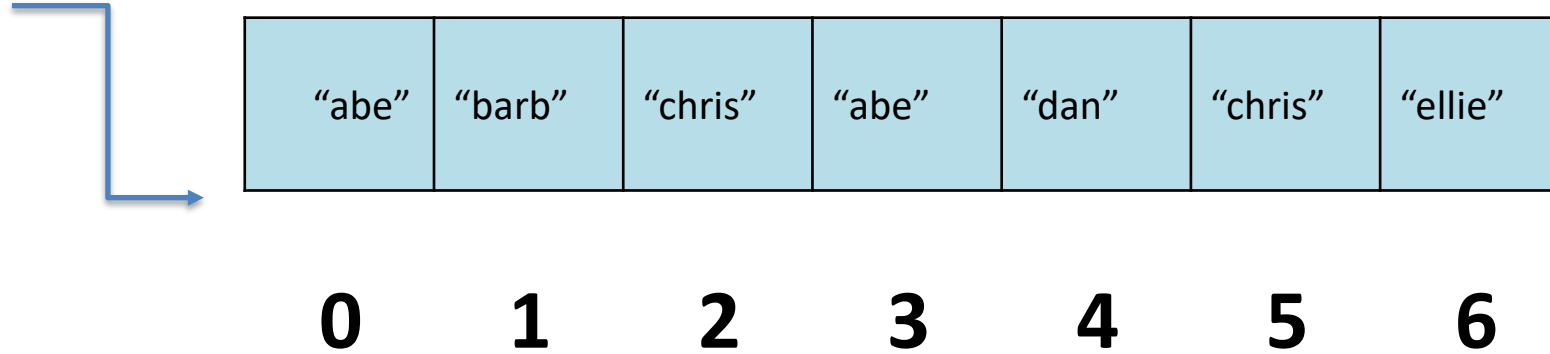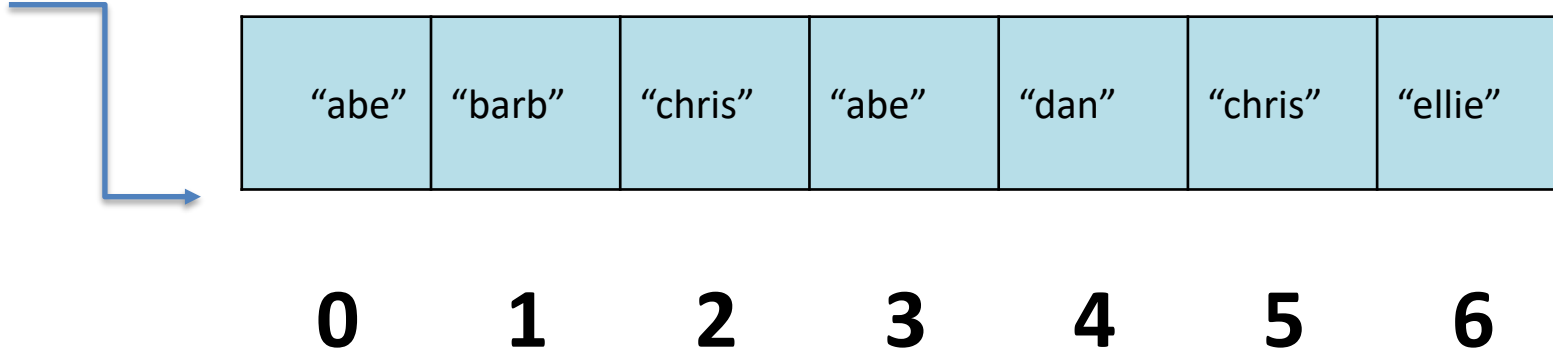
Iteration = Loop
Iterating the list = Traversing the list = Visiting each and every element of the liste fi.

For example, how can we print the first three characters of each element in the students list?

for x in students:

Iterating the tuple (Contd.)

students
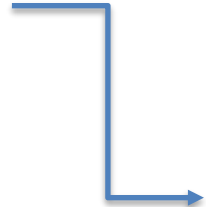
"abe"	"barb"	"chris"	"abe"	"dan"	"chris"	"ellie"
0	1	2	3	4	5	6

How can we print the first three characters of each element in the students list?

```
for x in students:
    print(x[0:2])
```

Checking for the memberships? **in**

`students`

"abe"	"barb"	"chris"	"abe"	"dan"	"chris"	"ellie"

0   **1**   **2**   **3**   **4**   **5**   **6**

Is "barb" in the tuple?

"barb" **in** students

```
>>> if ("barb" in students):
        print("Yes! Barb is registered")

Yes! Barb is registered
```

Python's Built-in Functions

		Built-in Functions		
abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	

Some functions are valid for lists.

I highlighted some.

Can you find other functions that are valid on tuples?

https://docs.python.org/3/library/functions.html

tuple Methods (only two methods are valid)

Method	Purpose
~~append(x)~~	~~Add x to the end of the list~~
~~extend(list_x)~~	~~Add all items from list_x at the end of the list~~
~~insert(,x)~~	~~Inserts an item at a given position. The first argument is the index of the element before which to insert. For example, a.insert(0, x) inserts at the front of the list.~~
~~remove(x)~~	~~Removes the first item x (note: there can be multiple items x in the list)~~
~~pop()~~	~~Removes the last item and returns the item~~
~~pop([i]~~	~~Removes the first item~~
~~clear( )~~	~~Removed all elements in the list. Empties the list.~~
index(x)	Returns the index of the first item x.
count(x)	Counts the number of times x is appearing in the list
~~sort( )~~	~~Sorts the elements in ascending order. sort(reverse=True) sorts the elements in descending order~~
~~reverse( )~~	~~Reverses a list~~
~~copy( )~~	~~Returns a copy of the list. You can also use "list" built-in function for the same purpose.~~

https://www.w3schools.com/python/python_ref_tuple.asp

Tuple summary

Do you have a list that is fixed?

Do you have a list that doesn't need to change?

Go for the "tuples".

Lists vs Tuples

	Lists	Tuples
Ordered	✔	✔
Indexed	✔	✔
Add or Update items	✔	✖
Can contain duplicates	✔	✔
Uses	Square Brackets	Round Brackets
	[ ]	**( )**