

CPSC 231 Tutorial #23

michael-hung.ca/teaching

Reminders

TODAY

Last Quiz!

THURSDAY

Last Quiz Review

Final Exam Review

FRIDAY

Assignment 6 Paired Component Due

Deque

For Assignment 6 Problem 3, we want you to use a deque (pronounced "deck").

- from the collections module → high efficiency
- means "double-ended queue"
- can "pop" from right or left ends (pop vs popleft)
 - popping removes the rightmost or leftmost element and returns its contents
 - If using popleft, all the elements to the right get their indices updated
- can append to right or left ends (append vs appendleft)

Deque

To instantiate a deque (assuming it's imported), call:

```
d = deque()           to initialize an empty deque; or
```

```
d = deque(i)         for some iterable i
```

Examples of iterables are strings, arrays, and range objects.

You can get an element's value by its index just like an array.

Circular Buffer

```
from collections import deque  
d = deque([0 for i in range(5)])
```

Circular Buffer

```
from collections import deque  
d = deque([0 for i in range(5)])
```

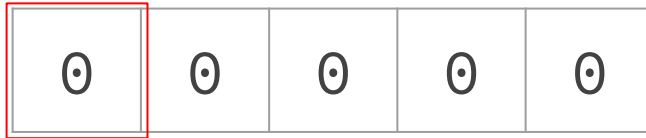
d



Circular Buffer

```
from collections import deque  
d = deque([0 for i in range(5)])
```

d



Circular Buffer

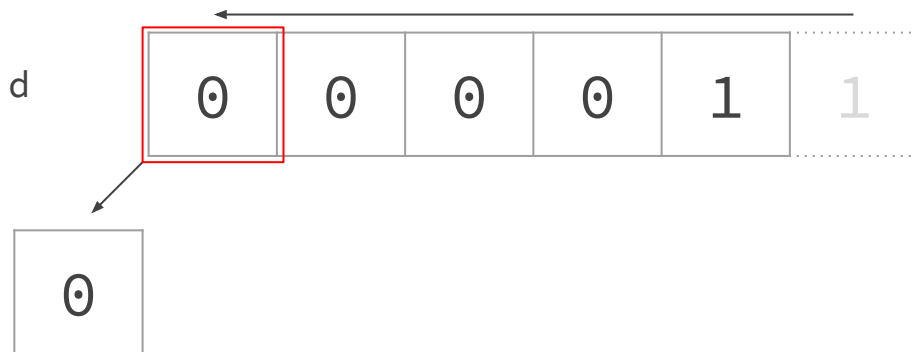
```
from collections import deque
d = deque([0 for i in range(6)])
d.append(1) # Add a new value
```

d



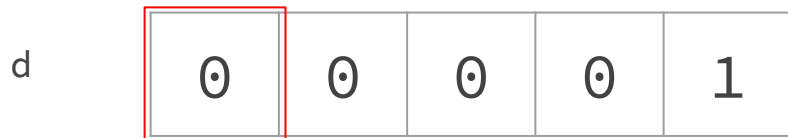
Circular Buffer

```
from collections import deque
d = deque([0 for i in range(6)])
d.append(1) # Add a new value
d.popleft() # remove leftmost element and return it
```

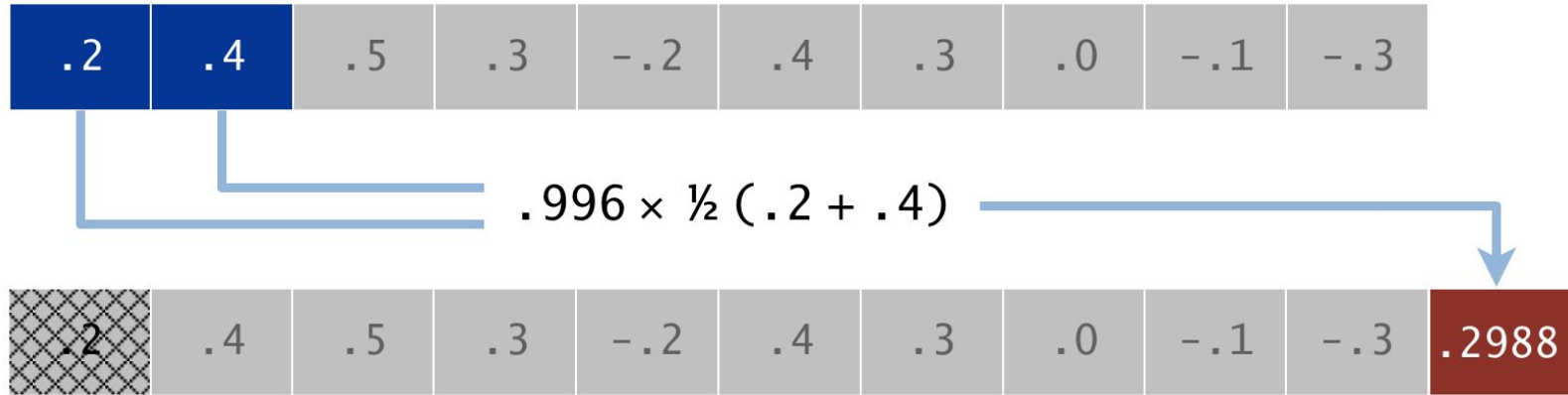


Circular Buffer

```
from collections import deque
d = deque([0 for i in range(6)])
d.append(1) # Add a new value
d.popleft() # remove leftmost element and return it
```



Karplus-Strong Algorithm



Notes

- Use **playSample** not **playSamples**
- The argument is a **single number**, not an array or the deque
- This single number will be the **output of tick()**, which runs Karplus-Strong *once* (one append and pop)
- To blend multiple strings, use playSample on the **sum** of all your GuitarStrings.tick()

Protips

1. Use a loop to calculate frequencies for strings
2. Instead of making a bunch of GuitarString objects to map to keys:
 - a. Make a string, keys = "q2we4r5...", i.e. the keyboard
 - b. Put GuitarStrings in an array so that index(nextKeyTyped) gives you the same index for the corresponding GuitarString

You might want to limit the number of possible keys and GuitarStrings if you don't have a powerful machine.

**email me topics you want me
to review for Thursday!**