

# Discussion 06:

# Iterators and Generators

---

TA: **Jerry Chen**

Email: **[jerry.c@berkeley.edu](mailto:jerry.c@berkeley.edu)**

TA Website: **[jerryjrchen.com/cs61a](http://jerryjrchen.com/cs61a)**

# Agenda

1. Feedback
2. Announcements
3. Iterators/Iterables
4. Generators

# Feedback

[tiny.cc/jerrydiscfb](http://tiny.cc/jerrydiscfb)

Pls

# Announcements

Exam prep OH

- Next week topics/times TBA

Manas' discussion is faster paced, and you will get through all problems on the worksheet as well as the extra problems. **Thurs 5-6:30 in 3105 Etcheverry**

Attendance will count as normal, so you can attend this section instead of your normal section

Labs 0-3, HW 1-3 and Hog all graded, double check OK and email me if there are any problems

# Announcements

Midterm scores released, submit regrades by March 5th

Hog Project composition revisions due Sunday, March 5th

Exam prep OH - TA will give you time to work through some problems, give an overview of the relevant topics, and walkthrough the problems.

- Next week topics TBA soon!

Homework 4 announced

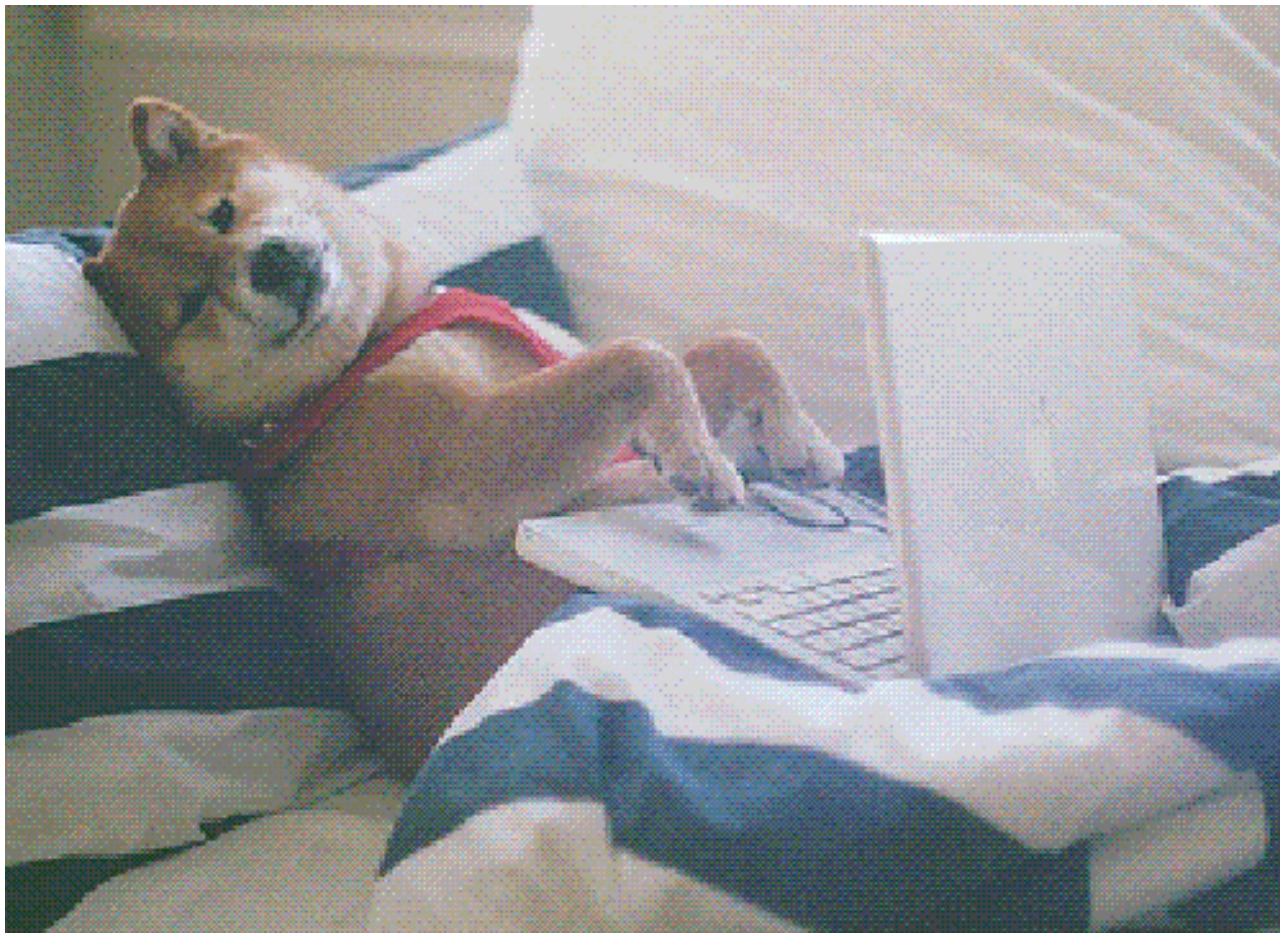
- Homework party Mon 3/6 6:30-8:30pm

Manas' discussion is faster paced, and you will get through all problems on the worksheet as well as the extra problems. Thurs 5-6:30 in 3105 Etcheverry. Attendance will count as normal, so you can attend this section instead of your normal section

Labs 0-3, HW 1-3 and Hog all graded, double check OK and email me if there are any problems

# Delayed Expressions

**"Lazy evaluation"**





# Real world application - streaming large data sets

```
ter • ls -lh
total 4.1G
-rw-r--r-- 1 jerrychen staff 1.4G Jun 16 2016 emails.csv
-rw-r--r-- 1 jerrychen staff 9.1K Feb 16 14:34 emails_top.csv
```

Microsoft Excel - melano38\_3614

Geneva 9 B I U

AB1 = CRC1634

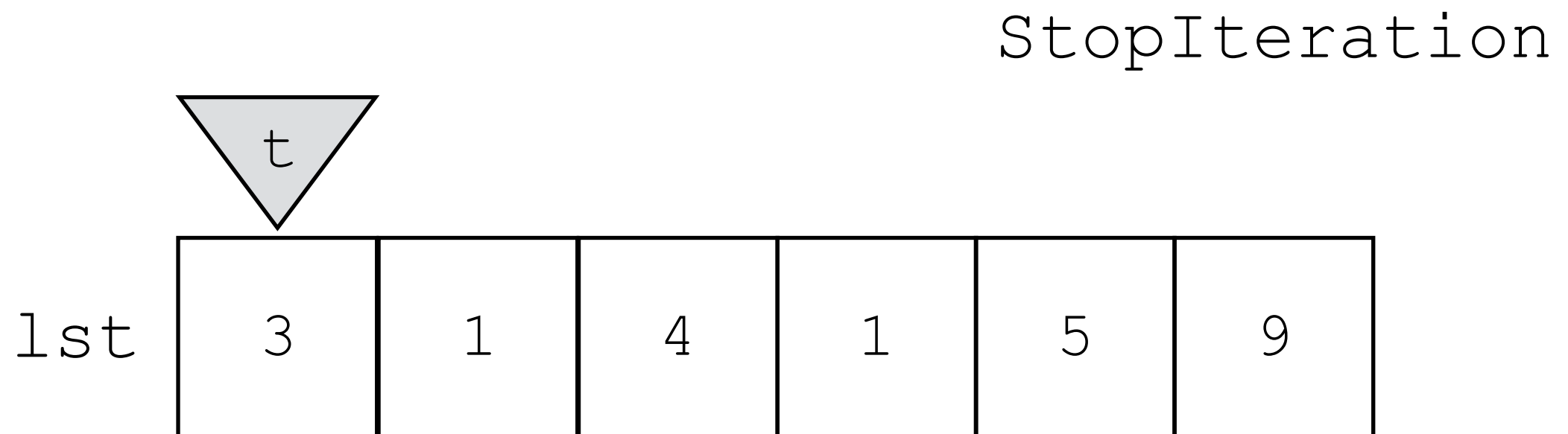
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	CloneID	UACC383	KA	UACC3093	UACC2534	M92-001	A-375	UACC502	M91-054	UACC1256	UACC091	UACC1273	M93-007	HA-A	MCF10A	UACC1012	SR83	TC-MA	UACC1022	TD-1720	TD-1638	TD-1730	TC-1376-3	TD-1376-3	TC-HA	SR85
2	183950	0.06	0.12	0.09	0.15	0.14	0.1	1.2	0.11	0.37	0.14	0.19	0.11	0.24	0.11	0.06	0.4	0.1	0.09	0.11	0.17	0.13	0.07	0.22	0.24	0.97
3	486844	0.06	0.05	0.05	0.1	0.07	0.12	0.05	0.06	0.07	0.05	0.09	0.06	0.07	0.11	0.05	0.23	0.12	0.07	0.05	0.06	0.16	0.05	0.09	0.64	0.31
4	30966	0.09	0.05	0.07	0.14	0.11	0.15	0.05	0.05	0.05	0.08	0.07	0.05	0.06	0.17	0.05	0.72	0.54	0.06	0.1	0.14	0.08	0.05	0.19	0.24	1.98
5	244147	0.25	0.24	0.24	0.25	0.25	0.26	0.18	0.22	0.16	0.17	0.21	0.22	0.35	0.13	0.21	0.37	0.2	0.25	0.15	0.24	0.2	0.22	0.5	0.38	0.28
6	31687	0.14	0.41	0.38	0.36	0.53	0.99	0.34	0.25	0.34	0.23	0.28	0.25	0.33	0.31	0.27	0.53	0.27	0.18	0.32	0.38	0.31	0.33	0.6	0.71	0.91
7	813823	0.31	0.63	0.05	0.84	0.15	0.52	0.05	0.11	0.38	0.57	0.6	0.07	0.07	0.11	0.12	0.54	0.47	0.08	0.06	0.11	0.15	0.05	0.2	0.11	0.92
8	810603	0.14	0.59	0.29	0.27	0.19	0.26	0.82	0.51	0.58	0.47	0.59	0.29	0.72	0.14	0.14	0.55	0.54	0.4	0.26	0.3	0.35	0.71	0.42	0.79	0.92
9	179711	0.46	0.23	0.5	0.71	0.67	0.15	0.47	0.49	0.41	0.63	0.41	0.67	0.76	0.18	0.6	0.46	0.39	0.2	0.22	0.16	0.33	0.33	0.38	0.34	0.76
10	32609	0.18	0.36	0.08	0.42	0.31	0.15	0.84	0.17	0.69	0.13	0.25	0.44	1.44	0.13	0.05	0.52	0.51	0.16	0.16	0.17	0.2	0.11	0.12	0.29	0.51
11	114116	0.2	0.5	0.22	0.36	0.44	0.29	0.63	0.35	0.65	0.38	0.31	0.54	0.65	0.28	0.16	0.61	0.15	0.27	0.14	0.22	0.56	0.47	0.54	0.68	0.3
12	755663	0.1	0.09	0.07	0.33	0.1	0.09	0.44	0.34	0.4	0.34	0.29	0.48	0.45	0.07	0.12	0.7	0.71	0.11	0.17	0.08	0.3	0.87	0.47	0.14	1.18
13	357374	0.15	0.15	0.18	0.23	0.12	0.27	0.5	0.36	0.74	0.33	0.59	0.36	0.45	0.09	0.17	0.67	1.57	0.13	0.19	0.09	0.31	0.78	0.43	0.16	1.15
14	366830	0.22	0.28	0.29	0.31	0.22	0.32	0.87	0.49	0.46	0.43	0.65	0.7	0.85	0.4	0.79	1.35	0.45	0.48	0.59	0.3	0.37	0.64	0.7	0.37	2.13
15	71101	0.51	0.12	0.21	0.24	0.12	0.33	0.79	0.23	0.48	0.05	0.41	0.46	1.22	0.3	0.15	0.72	0.64	0.18	0.09	0.16	0.24	0.68	0.51	0.11	0.8
16	233721	0.45	0.07	0.06	0.07	0.68	0.19	0.77	0.13	0.37	0.33	0.62	0.63	0.49	0.08	0.28	0.06	0.29	0.11	0.1	0.15	0.41	0.07	0.13	0.65	0.2
17	760224	0.09	0.06	0.06	0.13	0.09	0.11	0.22	0.19	0.18	0.29	0.18	0.25	0.1	0.11	0.07	0.06	0.17	0.06	0.06	0.05	0.08	0.05	0.05	0.05	0.09
18	759873	0.05	0.06	0.05	0.05	0.08	0.05	0.19	0.21	0.2	0.22	0.33	0.35	0.09	0.07	0.05	0.09	0.21	0.44	0.06	0.06	0.19	0.05	0.08	0.07	0.09
19	200814	0.06	0.05	0.05	0.06	0.05	0.07	0.05	0.46	0.12	0.22	0.49	0.07	0.05	0.36	0.05	0.05	0.05	0.05	0.06	0.05	0.05	0.08	0.05	0.06	0.09
20	122159	0.05	0.05	0.07	0.07	0.06	0.07	0.05	0.05	0.05	0.06	0.05	0.05	0.05	0.09	0.05	0.05	0.49	0.06	0.05	0.05	0.05	0.05	0.09	0.1	0.05
21	358168	0.09	0.05	0.05	0.05	0.07	0.08	0.05	0.05	0.07	0.06	0.09	0.08	0.05	0.1	0.06	0.05	0.78	0.16	0.06	0.12	0.06	0.05	0.11	0.26	0.05
22	75009	0.41	0.43	0.58	0.44	0.45	0.39	0.63	0.56	0.61	0.97	0.53	0.87	0.56	0.85	0.51	0.41	1.19	0.79	0.48	0.72	0.63	0.47	0.72	0.68	0.66
23	108837	0.05	0.05	0.05	0.05	0.05	0.05	0.07	0.05	0.05	0.05	0.05	0.05	0.05	0.08	0.3	0.33	0.05	0.05	0.05	0.06	0.05	0.05	0.05	0.15	0.74
24	768638	0.07	0.05	0.05	0.12	0.1	0.11	0.11	0.1	0.08	0.11	0.1	0.08	0.08	0.12	0.28	0.27	0.05	0.05	0.05	0.06	0.07	0.05	0.46	0.1	0.71
25	768561	0.05	0.05	0.05	0.72	0.15	0.1	0.15	0.13	0.11	0.11	0.1	0.09	0.05	0.08	0.25	0.31	0.05	0.05	0.05	0.08	0.1	0.07	0.09	0.13	0.83
26	199945	0.09	0.06	0.05	0.09	0.12	0.12	0.11	0.08	0.1	0.26	0.09	0.19	0.09	0.14	0.6	0.34	0.05	0.07	0.05	0.05	0.07	0.07	0.22	0.07	0.23
27	36060	0.4	0.26	0.42	0.24	0.52	0.4	0.7	0.53	0.38	0.39	0.4	0.38	0.17	0.47	0.59	0.66	0.43	0.5	0.56	0.45	1	0.21	0.31	0.34	0.6
28	71434	0.3	0.26	0.44	0.46	0.56	0.48	0.69	0.57	0.51	0.71	0.54	0.75	0.25	0.26	0.51	0.65	0.36	0.46	0.45	0.42	0.99	0.29	0.47	0.48	0.6
29	36717	0.54	0.43	0.62	0.61	0.55	0.6	0.72	0.67	0.49	0.78	0.68	0.53	0.4	0.62	0.6	0.76	0.47	0.41	0.6	0.5	0.77	0.46	0.39	0.57	0.77
30	44477	0.09	0.31	0.28	0.08	0.09	0.16	0.18	0.26	0.37	0.28	0.35	0.36	0.16	0.14	0.2	0.31	0.1	0.18	0.16	0.16	0.44	0.11	0.32	0.3	0.44
31	44722	0.23	0.36	0.32	0.26	0.23	0.51	0.63	0.39	0.4	0.67	0.7	0.46	0.29	0.26	1.01	0.58	0.25	0.33	0.41	0.37	0.38	0.21	0.26	0.3	0.86
32	815542	0.06	0.13	0.28	0.05	0.05	0.07	0.24	0.27	0.14	0.23	0.22	0.26	0.11	0.07	0.1	0.16	0.07	0.44	0.24	0.09	0.42	0.07	0.22	0.06	0.17
33	813673	0.52	0.37	0.39	0.25	0.28	0.39	0.59	0.79	0.38	0.63	0.58	0.58	0.45	0.71	0.84	0.75	0.53	0.61	0.42	0.55	0.71	0.52	0.43	0.61	0.44
34	376574	0.05	0.08	0.13	0.07	0.07	0.12	1.65	0.1	0.43	0.26	0.28	0.48	0.12	0.1	0.08	0.34	0.05	0.08	0.05	0.05	0.18	0.05	0.09	0.08	0.24
35	712668	0.16	0.26	0.21	0.19	0.19	0.24	0.54	0.47	0.6	0.67	0.69	0.66	0.27	0.4	0.26	0.44	0.31	0.25	0.24	0.27	0.36	0.18	0.31	0.14	0.2
36	151767	0.09	0.08	0.22	0.12	0.22	0.25	0.37	0.42	0.59	0.36	0.73	0.65	0.28	0.66	0.28	0.38	0.07	0.19	0.24	0.21	0.44	0.05	0.11	0.11	0.37

Ready NUM

```
lst = [3, 1, 4, 1, 5, 9]
```

```
t = iter(lst)
```

*# Now, repeatedly call print(next(t))*





# Iterators/Iterables

**Iterable** — returns an iterator using `iter()`

**Iterator** — get next item in iterable using `next()`

- `next()` likely **modifies some state**

# Iterators/Iterables

## Miscellaneous

- Signal end of an iterator's sequence by raising a `StopIteration` exception
- `iter()` on an iterator usually gives you the same iterator back. Why?

# Iterators/Iterables

"The iterable is a book, and the iterator is a bookmark"

If something is **iterable**, we can get its **iterator** using `iter()` and examine all its elements by repeatedly calling `next()` on that iterator.

Keep in mind that iterators are usually **one-time use**. Stepping through a sequence again means calling `iter()` again, on the original iterable.

# Generators

Generator functions return a generator, which is a special **iterator**

- **next** will cause us to run until the next **yield**
- Return the expression at the yield, and **pause**

# Check your understanding

```
class Naturals():  
    def __iter__(self):  
        current = 0  
        while True:  
            yield current  
            current += 1
```



```
class Naturals():  
    def __init__(self):  
        self.current = 0  
  
    def __iter__(self):  
        while True:  
            yield self.current  
            self.current += 1
```

```
>>> nats = Naturals()  
>>> nats_iterator1 = iter(nats)  
>>> nats_iterator2 = iter(nats)  
>>> next(nats_iterator1)  
?  
>>> next(nats_iterator1)  
?  
>>> next(nats_iterator2)  
?
```

```
class Naturals():  
    def __init__(self):  
        self.current = 0  
  
    def __iter__(self):  
        while True:  
            yield self.current  
            self.current += 1
```

```
>>> nats = Naturals()  
>>> nats_iterator1 = iter(nats)  
>>> nats_iterator2 = iter(nats)  
>>> next(nats_iterator1)  
0  
>>> next(nats_iterator1)  
1  
>>> next(nats_iterator2)  
1
```