

DS2000

~~Apr~~ 2/3 - Fri ☁️

Admin

- OH online
- HW3 late deadline 2/6 aprn (no late penalty)
- mini v'z assignment due 3/3

Agenda

1. lists ~ value/position
2. Python

1. Lists - value and position

A list is a data structure

↳ one label,
multiple values

Lists are mutable

↳ they can be changed

last time...

op1
tot-score
opp-score
...
~ tot



3 lists

- opponents
- tot.scores
- opp-scores

opponents = ["Ars", "Man", "West"] values
 0 1 2 pos

tot_scores = [10, 9, 8] values
 0 1 2 pos

opp_scores = [0, 0, 1] values
 0 1 2 pos

lists correspond
with each
other

↓
Same position in
the 3 lists is
same game

Learn any new data structure...

1. Create lst = []
2. Add lst.append(w) → value
3. Look at one thing lst[-1] → position

4. Look at all things

Last time...

lst = [2, 3, 4]

for item in lst:
 ~~variable~~ ~~list~~
 item = item * 2
 print(item)

for loop

→ shortcut for...

item = 2

item = ~~2~~ item * 2

item = 3

item = item * 2

item = 4

item = item * 2

print:
4
6
8 } doubles item

Iterate by value

for item in lst:
do a thing

plusses:

- Mean, short
- default loop!

downside:

- doesn't modify the list
- can't do 2 lists at once

Less elegant
more messy
but... can do these!

Iterate by position

Second choice for loop

lst = [⁰2, ¹3, ²4]

for i in range(len(lst)):

variable
position in
the list

length of the list
len(lst)

3

Python vocab

for: repeat

in: all positions for i

range: 0 to end of list

ex: print out one game at a time

```
for i in range(len(opponents)):
```

```
    print(opponents[i], tot-scores[i], opp-scores[i])
```

Short cut for:

```
i = 0
```

```
opp>[i], tot-scores[i], opp-scores[i]
```

```
i = 0 1
```

```
opp>[i], tot-scores[i], opp-scores[i]
```

```
i = 2
```

```
opp>[i], tot-scores[i], opp-scores[i]
```

*i is a position,
never a value!*