CS21: INTRODUCTION TO COMPUTER SCIENCE

Prof. Mathieson

Fall 2018

Swarthmore College

Outline Nov 30:

- Quiz 5: first 30 min
- Examples of abstraction/encapsulation
- baseball.py example
- ord/chr practice (cipher.py)

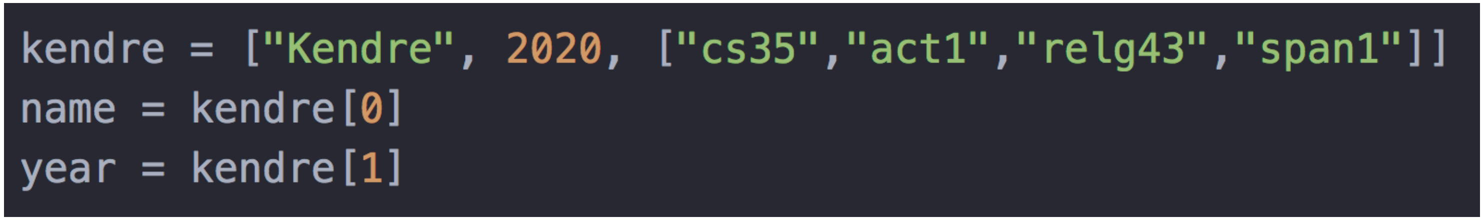
Notes

•Lab 10 due **MONDAY** (no office hours Monday)
•Office hours **today! 3-5pm**
•Ninja session tonight!

Abstraction/Encapsulation

Student example

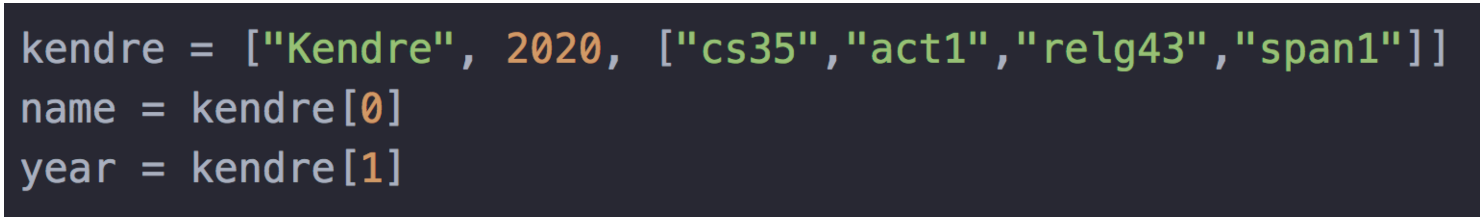
- Encapsulated (student is represented as one thing, a list), but not abstract



```
kendre = ["Kendre", 2020, ["cs35","act1","relg43","span1"]]
name = kendre[0]
year = kendre[1]
```

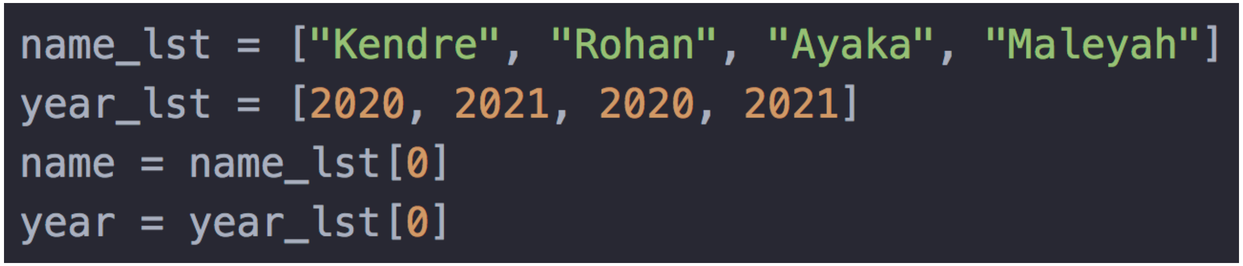
Student example

- Encapsulated (student is represented as one thing, a list), but not abstract



```
kendre = ["Kendre", 2020, ["cs35","act1","relg43","span1"]]
name = kendre[0]
year = kendre[1]
```

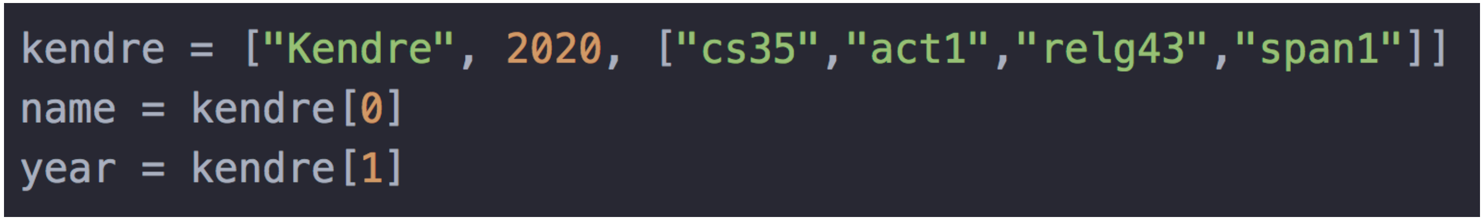
- Neither encapsulated (data for one student is spread over multiple objects), nor abstract



```
name_lst = ["Kendre", "Rohan", "Ayaka", "Maleyah"]
year_lst = [2020, 2021, 2020, 2021]
name = name_lst[0]
year = year_lst[0]
```

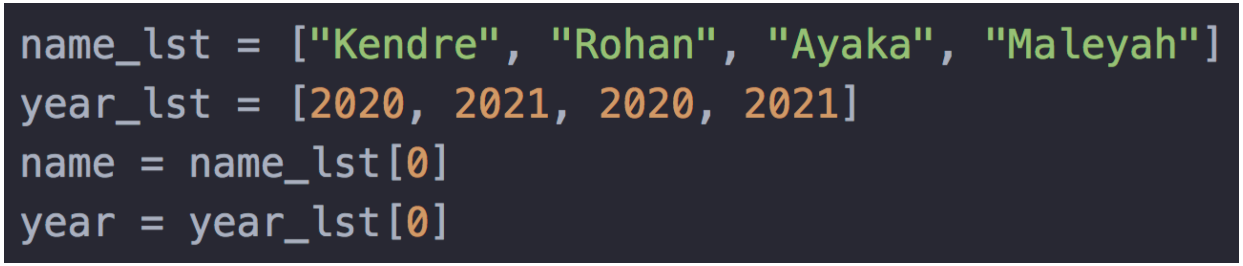
Student example

- Encapsulated (student is represented as one thing, a list), but not abstract



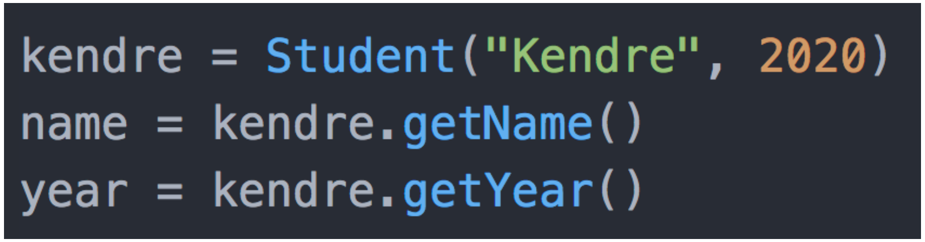
```
kendre = ["Kendre", 2020, ["cs35","act1","relg43","span1"]]
name = kendre[0]
year = kendre[1]
```

- Neither encapsulated (data for one student is spread over multiple objects), nor abstract



```
name_lst = ["Kendre", "Rohan", "Ayaka", "Maleyah"]
year_lst = [2020, 2021, 2020, 2021]
name = name_lst[0]
year = year_lst[0]
```

- Both abstract and encapsulated



```
kendre = Student("Kendre", 2020)
name = kendre.getName()
year = kendre.getYear()
```

Advantages of encapsulation/abstraction

- Interface (how you interact with something) is consistent even if the internal details change.

  - 1) If you change the engine in your car, you still drive it the same way – don’t need to know how the engine works.

Advantages of encapsulation/abstraction

- Interface (how you interact with something) is consistent even if the internal details change.
  - 1) If you change the engine in your car, you still drive it the same way – don’t need to know how the engine works.
  - 2) In online shopping you have a “Cart”, which is an abstract concept and is roughly the same across sites. Probably represented as a list underneath but user doesn’t need to know.

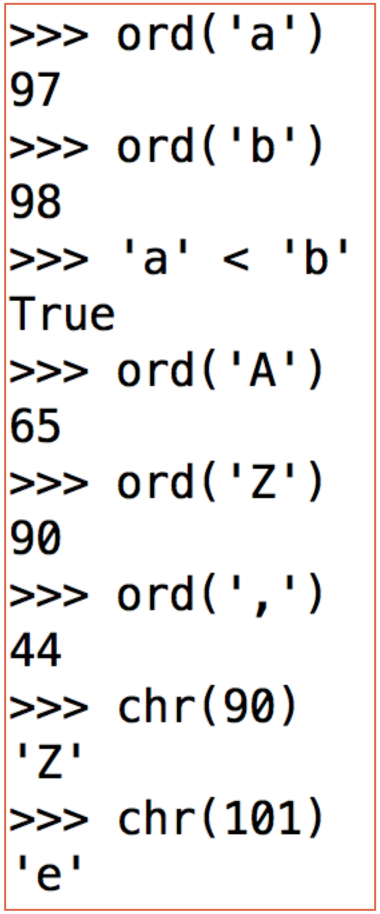
Ord(..) and Chr(..)

Using ord(..) and chr(..)

- We can use **ord(ch)** to obtain a numerical representation of a character

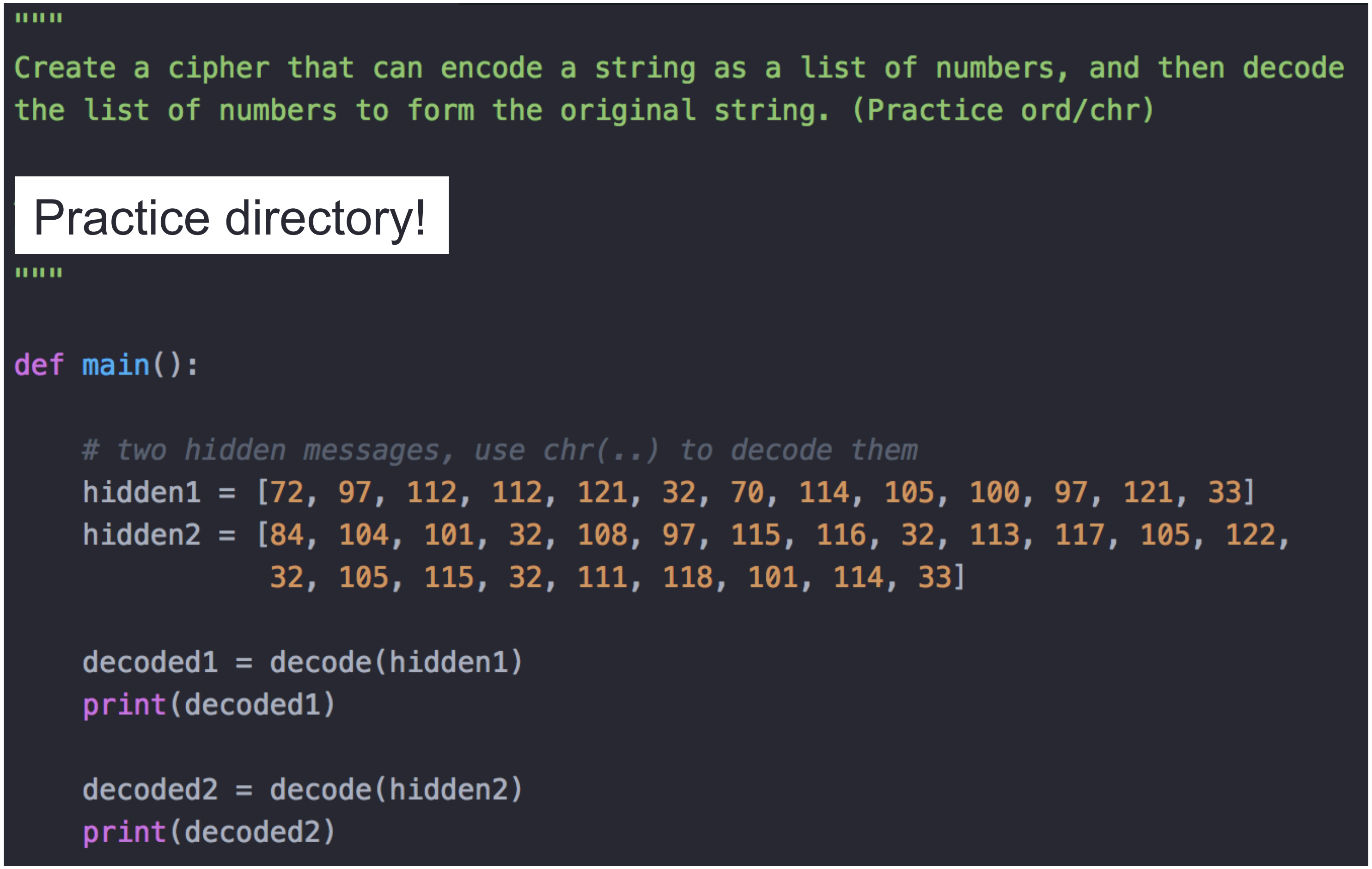
- When we compare characters (when searching or sorting, for example), we are really comparing their numerical representations

- We can use **chr(int)** to convert an integer back into a character



```
>>> ord('a')
97
>>> ord('b')
98
>>> 'a' < 'b'
True
>>> ord('A')
65
>>> ord('Z')
90
>>> ord(',')
44
>>> chr(90)
'Z'
>>> chr(101)
'e'
```

Example of using ord/chr to create a cipher



```
"""
Create a cipher that can encode a string as a list of numbers, and then decode
the list of numbers to form the original string. (Practice ord/chr)
```

Practice directory!

```
"""


def main():

    # two hidden messages, use chr(..) to decode them
    hidden1 = [72, 97, 112, 112, 121, 32, 70, 114, 105, 100, 97, 121, 33]
    hidden2 = [84, 104, 101, 32, 108, 97, 115, 116, 32, 113, 117, 105, 122,
               32, 105, 115, 32, 111, 118, 101, 114, 33]

    decoded1 = decode(hidden1)
    print(decoded1)

    decoded2 = decode(hidden2)
    print(decoded2)
```

Back to baseball example