

CS21: INTRODUCTION TO COMPUTER SCIENCE

Prof. Mathieson

Fall 2017

Swarthmore College

Outline Dec 1:

- Quiz 5: first 30 min
- ord/chr (cipher.py)
- Overloading other operators in classes

Notes

- Lab 10 due **Sunday**
- Office hours **today!! 3-5pm (in 240)**
- **Ninja session tonight!**

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)

Author: Sara Mathieson
Date: 12/1/17
"""

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)
```

Example of using ord/chr to create a cipher

```
def encode(message):  
    """Return an encoded version (list of integers) of the message (string)."""  
    encoded = []  
    for ch in message:  
        encoded.append(ord(ch)) # use built-in ord(..) function  
    return encoded  
  
def decode(encrypted):  
    """Return the decoded message (string) from a list of integers."""  
    message = ""  
    for num in encrypted:  
        message += chr(num) # use built-in chr(..) function  
    return message
```

Overloading operators in classes

Example of `len(..)` and `in` operators

```
# overload the len(..) operator
def __len__(self):
    return len(self.players)

# overload the "in" operator
def __contains__(self, player):
    return player in self.players
```

```
# now we can use the built-in len(..) function
num_players = len(red_sox)
print(num_players)

# now we can use the "in" operator
player = red_sox.getRandomPlayer()
print(player)
if player in red_sox:
    print("in team")
else:
    print("not in team!")
```