

# ECS10

10/8

## Last time

- Boolean data – True or False
- Boolean operators:  
and, or, not
- Use to check for valid user input:

```
reply = raw_input('Enter R or D: ')
if not ((reply == 'R') or (reply == 'D')):
    print 'Not one of the choices.'
```

## Ask until we get one of the inputs we understand

- May have to ask an arbitrary number of times.
- Use a **while** loop.

**while Boolean is True:**

```
Do statements in block
```

- **while** loop **iterates** over block – does block over and over, until Boolean becomes False somehow.

## A loop

```
while not ((choice == 'D') or (choice == 'R')):
    print 'Not a major party.'
    choice = raw_input('R or D: ')
print 'Thank you.'
```

- Like **if**, the block under **while** is done when its Boolean test is **True**.
- Unlike **if**, after doing the block, the program goes back up to the **while** statement.

## Infinite loop

- One of the classic programming bugs
- Get out of it using CTRL-c (hold down control key and type c)
- Repeat after me: CTRL-c



## Unintended infinite loop

```
while ((choice != 'D') or (choice != 'R')):
    print 'Not a major party.'
    choice = raw_input('R or D: ')
```

- Infinite loop because the Boolean condition is always True!

## Converting types

```
int('5')/2
```

- Converts the string '5' to the integer 5

## Converting Data

- Convert an integer to a string:

```
x = str(10)
```

- x is the string '10'

```
y = float('8.3333')
```

- y is the floating point number 8.3333

## Getting numbers from the user

```
reply = raw_input('Feet: ')
feet = int(reply)
reply = raw_input('Inches: ')
inches = int(reply)
```

- Notice we use the variable reply twice.
- When we put the second string into reply, the first one gets written over. It's gone!

## Body mass index

Kilograms / (meters<sup>2</sup>)

- Or, in 'English Imperial Units'

(Pounds \* 703) / (inches<sup>2</sup>)

- 25 or greater is overweight.

## Yao Ming



7 feet, 6 inches. 310 pounds.

## Next assignment: compound interest program

- Original debt is \$10,000
- Every month, make a payment and pay interest
- Loop until debt is paid off