Memory Management

You must release an object
if and only if
you got it from alloc

Retain count

- alloc gives objects with retain count 1
- release decrements retain count by 1
- Objects are deallocated (freed) when their retain count reaches 0

Other cases

```
// a class called “Widget”
// with an ivar called “name”

- (void)setName:(NSString *)aString {
  name = aString;
}
```

Other cases

```
Widget *w = [[Widget alloc] init];
NSString *s = [[NSString alloc] initWithFormat: ...];

[w setName:s];

// put w in an instance variable

// follow the rule
[s release];

// now what about that string?
```

Other cases

- Widget needs that string to not be freed
- This is what retain is for
- retain increments retain count by 1

Other cases

```
// a class called “Widget”
// with an ivar called “name”

- (void)setName:(NSString *)aString {
  [aString retain];
  name = aString;
}

- (void)setName:(NSString *)aString {
  name = [aString retain];
}
```

Other cases

- [NSString alloc], retain count 1
- [aString retain], retain count 2
- [s release], retain count 1
- When will it reach 0?

Other cases

```
- (void)setName:(NSString *)aString {
    name = [aString retain];
}
```

Other cases

- [name release];
- This “balances” the retain call

Other cases

```
- (void)setName:(NSString *)aString {
  [name release];
  name = [aString retain];
}
```

This is still not right!

Deallocating

- - (void)dealloc
- When an object is freed
- Release instance variables
- Then [super dealloc]

Deallocating

```
- (void)dealloc {
    [name release];
    [super dealloc];
}
```

If I ever see you call dealloc on anything other than super

I will kill you

You must release an object
if and only if
you got it from alloc or copy
or you retained it

A special case

```
- (Widget *)someWidget {
  Widget *result = [[Widget alloc] init];

  // do stuff to result

  return result;
}
```

Autorelease

- Calls release on an object “later”
- “Later” does not happen within code you write
- Really: at the end of the run loop (event loop)

A special case

```
- (Widget *)someWidget {
  Widget *result = [[Widget alloc] init];

  // do stuff to result

  return [result autorelease];
}
```

You must release or autorelease an object if and only if you got it from alloc or copy or you retained it

Properties

- @property(assign) just assigns the new value
- @property(retain) retains new value and releases the old one
- @property(copy) copies the new value and releases the old one