R Notebook

I'm so confused by the ellipses argument, so let's try playing with it a little because it would be very useful for my matrix fns if I didn't have to define the default in the parent and then pass everything down. And everything I look up that might be relevant is defined in an object-orientated programming format that I'm struggling to follow.

Let's say my example is something to do with rep and print.

```r
foo <- function( num, times = 10 ){
  fubar <- bar( num = num, times = times  )
  print( fubar )
}
bar <- function( num, times ){
  rep( num, times = times  )
}
```

So this is how my functions are currently built. And technically, it works.

```r
foo( 5 )
```

```
##  [1] 5 5 5 5 5 5 5 5 5 5
```

```r
foo( 7, times = 2 )
```

```
## [1] 7 7
```

But it's clunky.

So what about this way?

```r
foo2 <- function( ... ){
  fubar <- bar2( ...  )
  print( fubar )
}
bar2 <- function( num, times ){
  rep( num, times = times  )
}

foo2( 5 )
```

```
## [1] 5
```

```r
foo2( 7, times = 2 )
```

```
## [1] 7 7
```

Okay, that makes sense. What about with a default value? Does it still work then?

```r
foo2 <- function( ... ){
  fubar <- bar2( ...  )
  print( fubar )
}
bar2 <- function( num, times = 10 ){
  rep( num, times = times  )
```

```
}

foo2( 5 )
```

```
##  [1] 5 5 5 5 5 5 5 5 5 5 5
```

```
foo2( 7, times = 2 )
```

```
## [1] 7 7
```

It does! I think I just needed to try it a few ways to get it.

Okay, another scenario, we need the var earlier but don't want to mention it every time. If it exists in the parent will you bring it in?

```
foo3 <- function( num, ... ){
  print( num )
  fubar <- bar3( ...  )
  print( fubar )
}
bar3 <- function( times = 10, ... ){
  rep( num, times = times  )
}

foo3( 5 )
foo3( 7, times = 2 )
```

Nope. So you do need to define it in the function, at the very least. Do you need to say num = num though?

```
foo3 <- function( num, ... ){
  print( num )
  fubar <- bar3( ...  )
  print( fubar )
}
bar3 <- function( num, times = 10 ){
  rep( num, times = times  )
}

foo3( 5 )
foo3( 7, times = 2 )
```

I mean there's the positional option here but ugh. Yeah, no real way to get it to look at the parent, which is unfortunate.

Also looks like online recommends just defining a default if a bool is optional and you don't need a third option - optional args are for more complicated things.