conceptual design of software

Daniel Jackson · Accenture/CSAIL Design Hackathon · January 24, 2015

approaches to improving design

process

optimize organization & task structure

Taylor (1880s), Toyota JIT (1980s), CMM (1990s), agile (2000s)

"new" activities

brainstorming

user studies

prototypes

style

Dieter Rams, 1958
pocket transistor radio T3

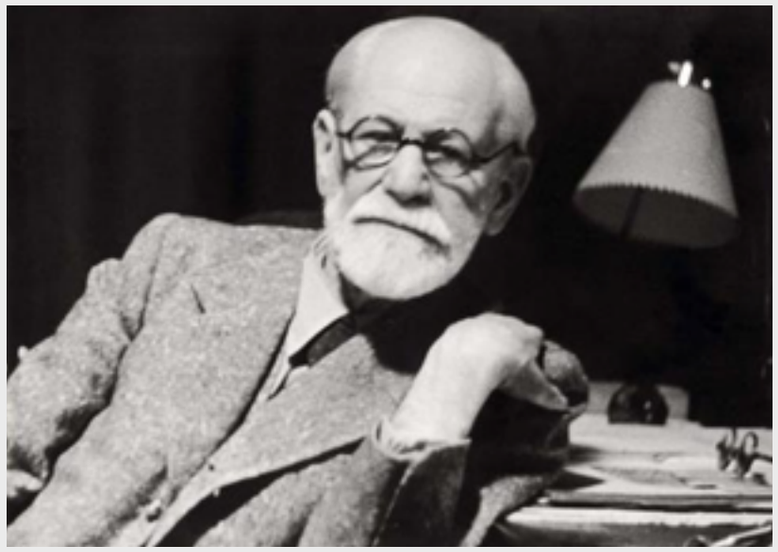
Henry Hobson Richardson, 1877
Trinity Church, Boston

Alfonso Bialetti, 1933
La Moka coffee maker

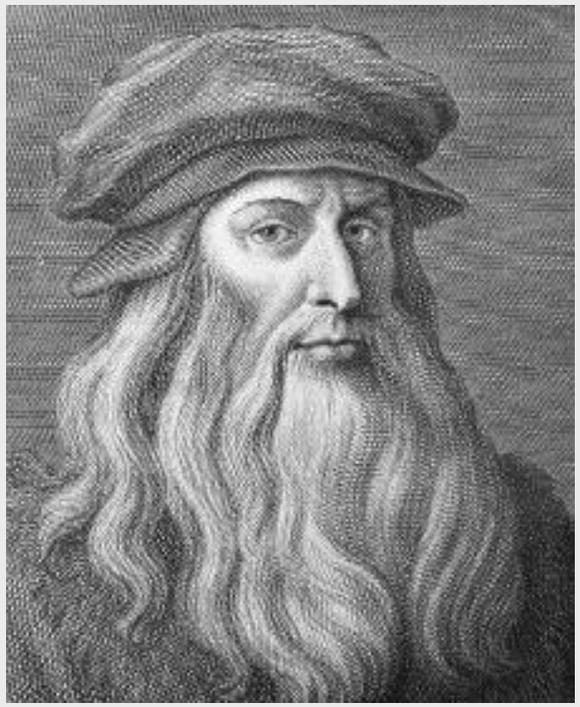
Leica, 1953
M3 rangefinder camera

broadening

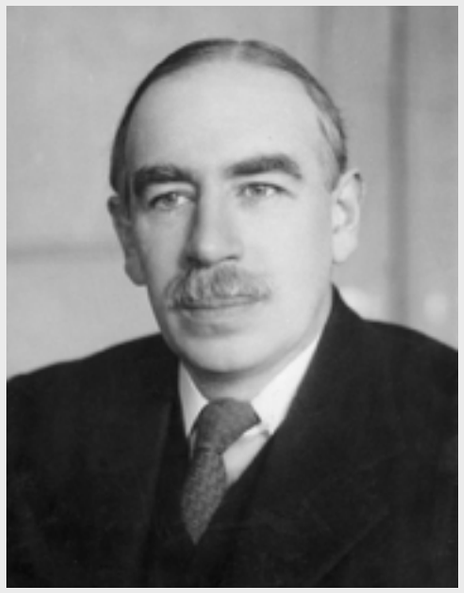
anthropology



psychology



art



economics

seeking the essence

designer vs engineer

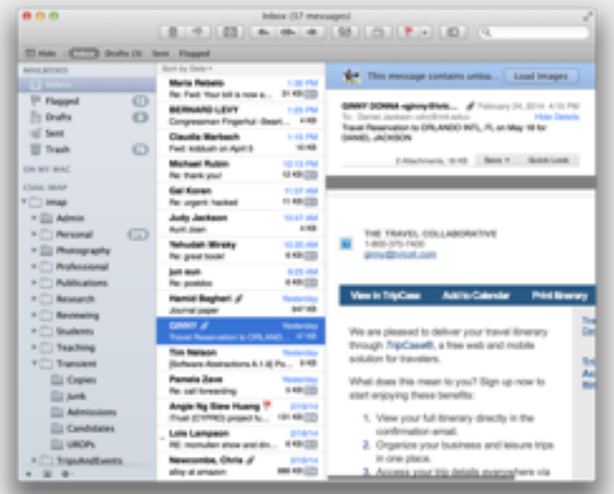
	designer	engineer
building	facade, entry flow, access, light	beam, wall structural integrity
software	purpose, concept fitness for purpose	function, datatype decoupling, localization

concepts & software

what characterizes an app?

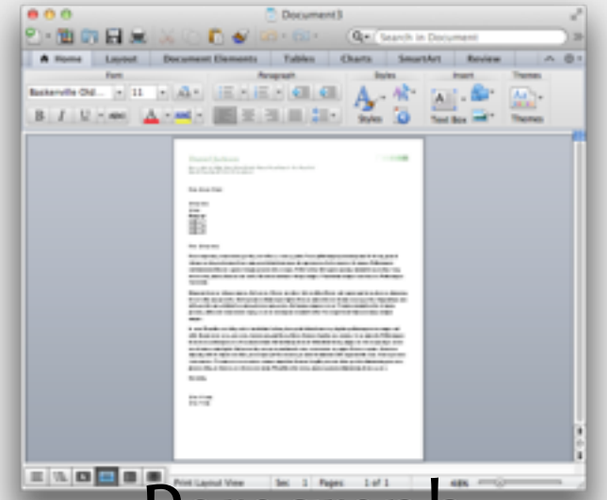
concepts!

Apple Mail



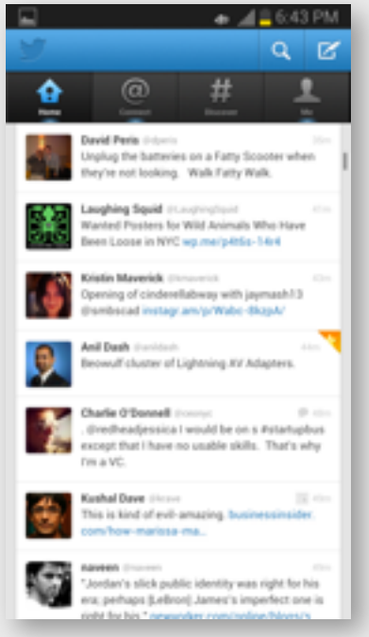
EmailAddress
Message
Folder or Label

Microsoft Word



Paragraph
Format
Style

Twitter



Tweet
Hashtag
Following

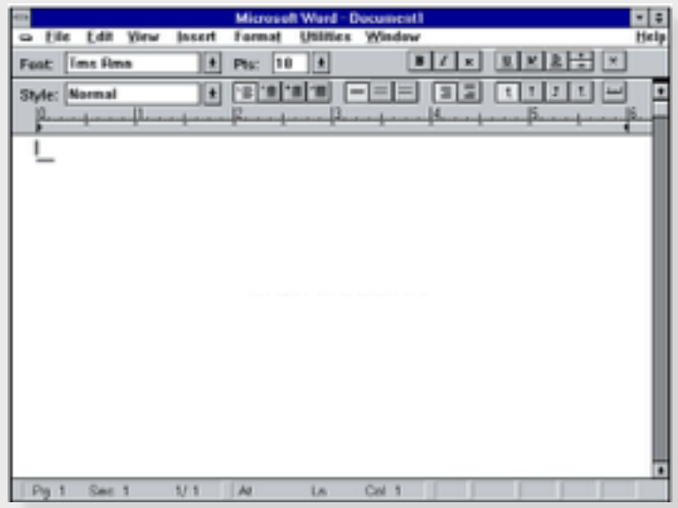
Photoshop

PixelMap
Layer/Mask
Adjustment

concepts differentiate classes

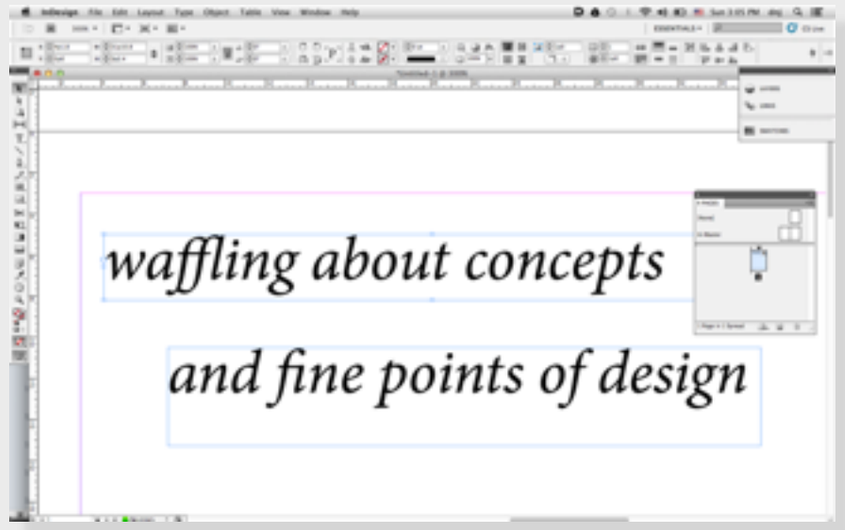
text editor

line, buffer,
character set



word processor

paragraph,
format, style



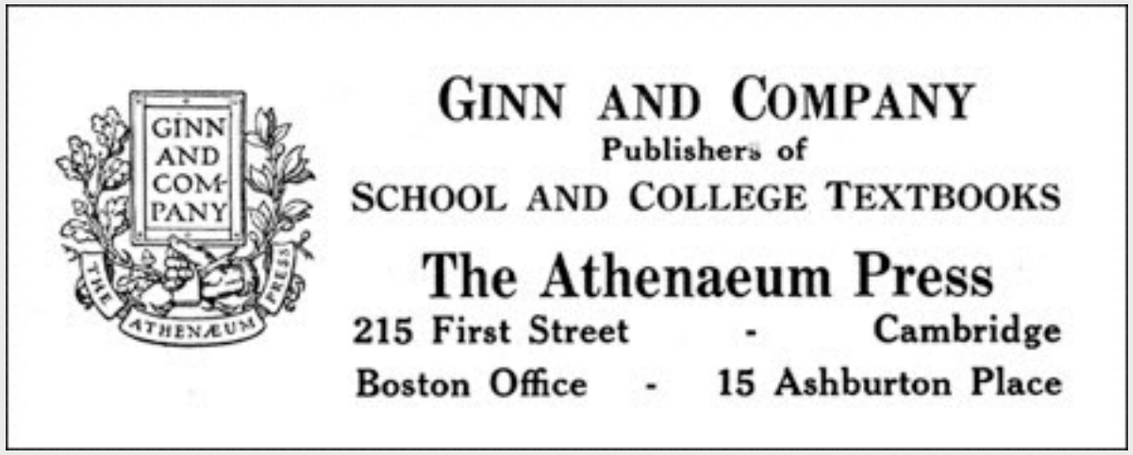
desktop publishing app

text flow, link,
page template

where are Word's concepts from?

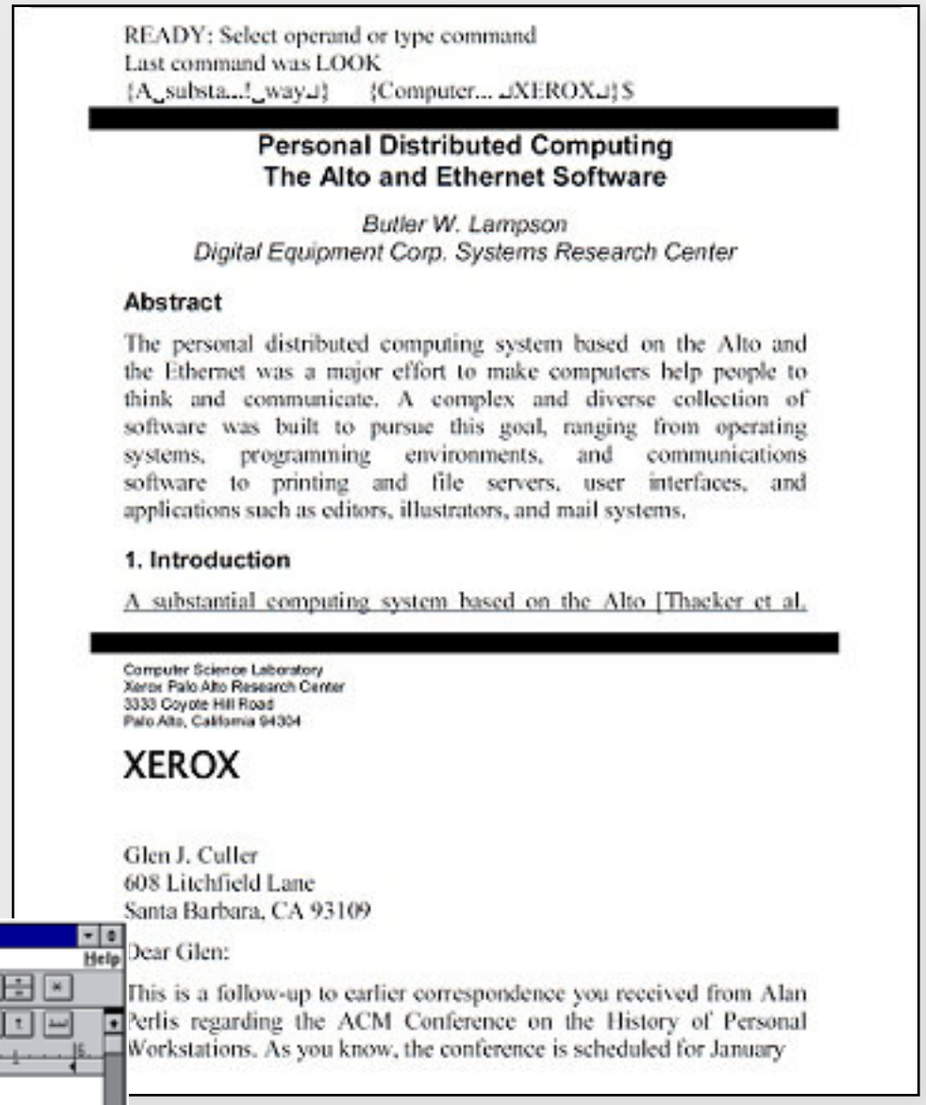
Charles Simonyi: brought key concepts to Word from Xerox PARC

rich concepts have long journeys

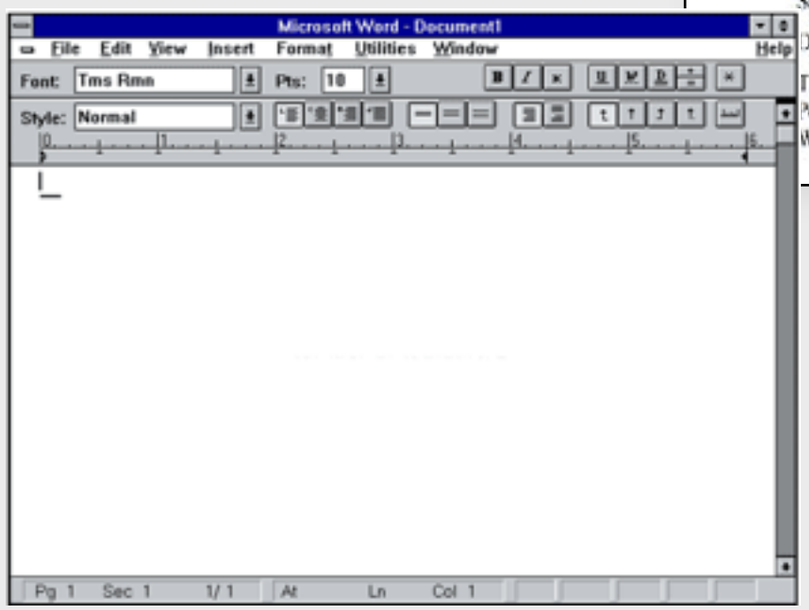


Ginn & Co, since 1868

Apple Pages, 2005



Bravo, 1974



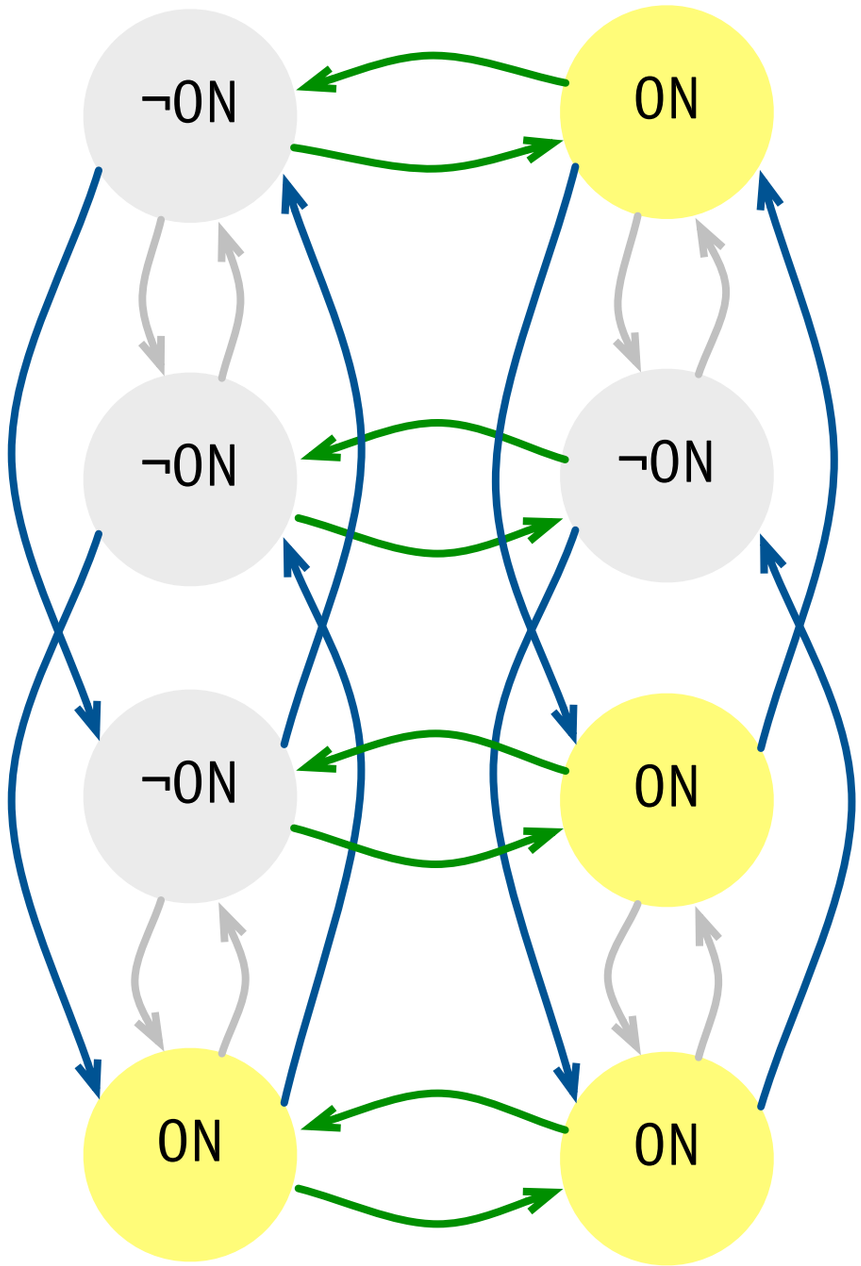
Microsoft Word, 1983

where do concepts come from?

domain concepts: exist in problem domain	analogic concepts: based on known notions	synthetic concepts: invented for software
photo, movie, song	blog post, email, tweet	friend, follower
account, balance	desktop, folder, file	relative reference
social security number	layer, mask, stacking	selection, buffer
typeface, ligature	cart, order, item	permission, capability

so what exactly
is a concept?

a behavioral model



a conceptual model

```
on: bool
time: Slot
schedule: set Slot

inv on = (time ∈ schedule)

tick ≜  time := next(time)

toggle (s: Slot) ≜
    if s ∉ schedule then schedule := schedule ∪ {s}
    else schedule := schedule \ {s}
```

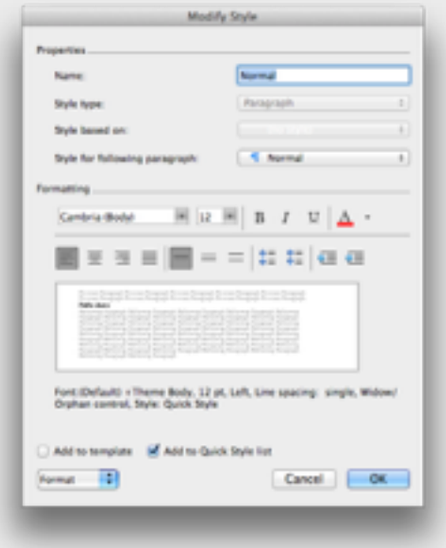
concepts as explanatory state

a concept is
part of the internal state
modifiable by user actions
indirectly affecting the external world

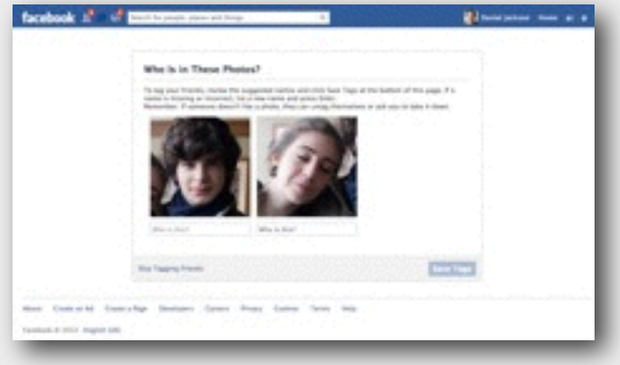
examples
style (Word): affects formatting
tag (Facebook): affects view permission
selection (Finder): affects result of delete, move, etc

concept story

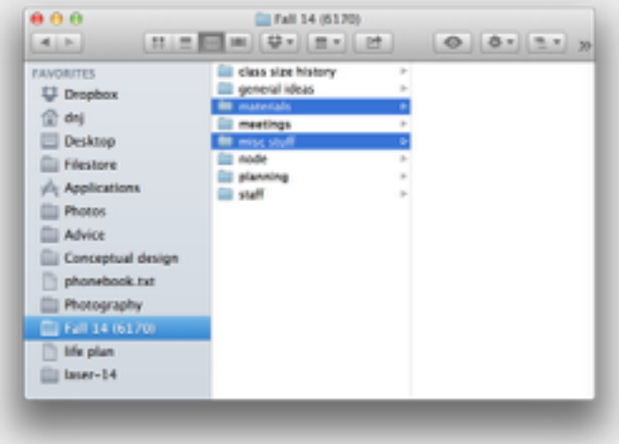
if you use the concept like this, the outcome will be...



"**if** you change a style's format, **then** all paragraphs of that style will change format accordingly"



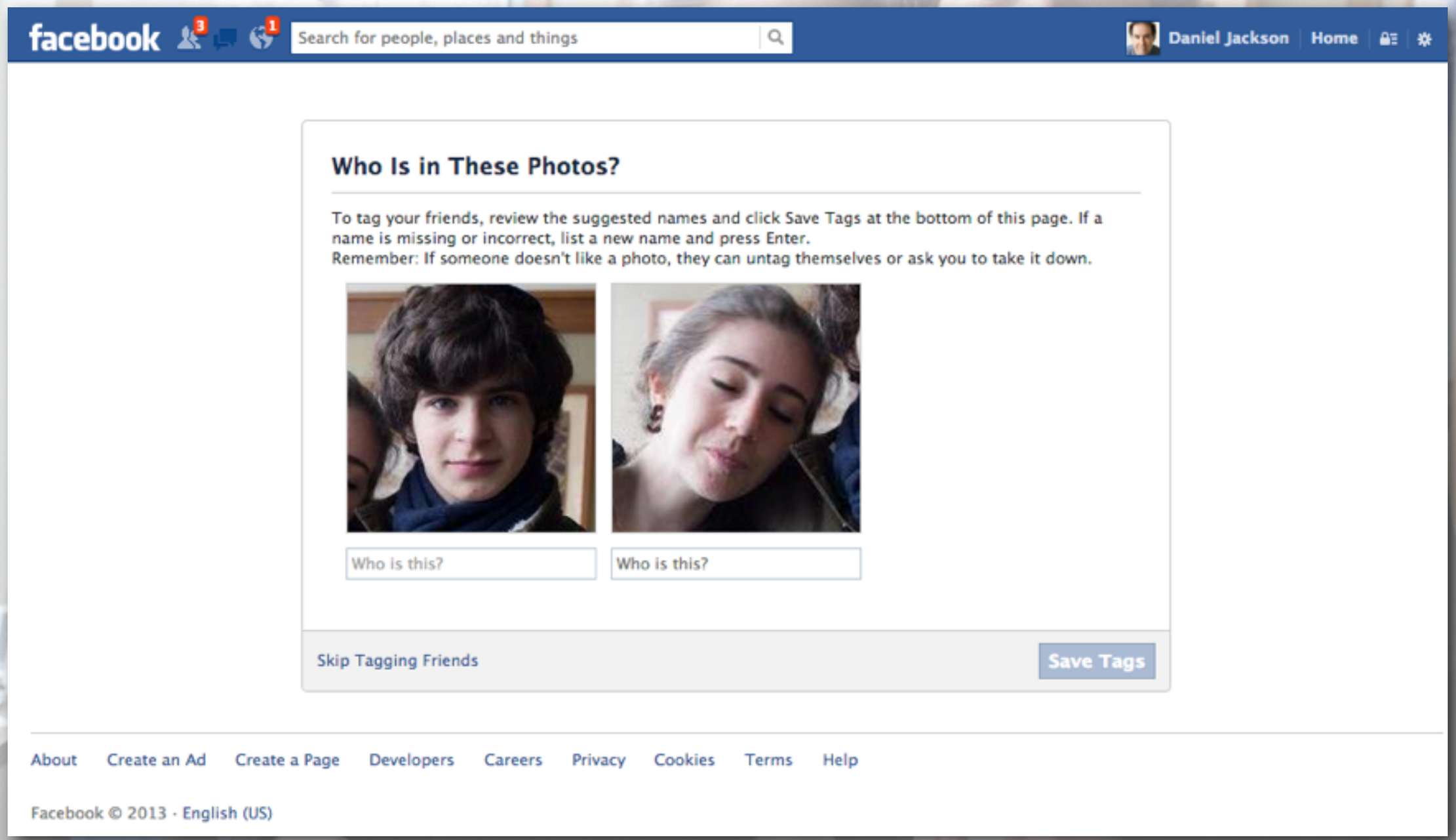
"**if** you tag a photo, **then** all friends of the person tagged will be able to see the photo"



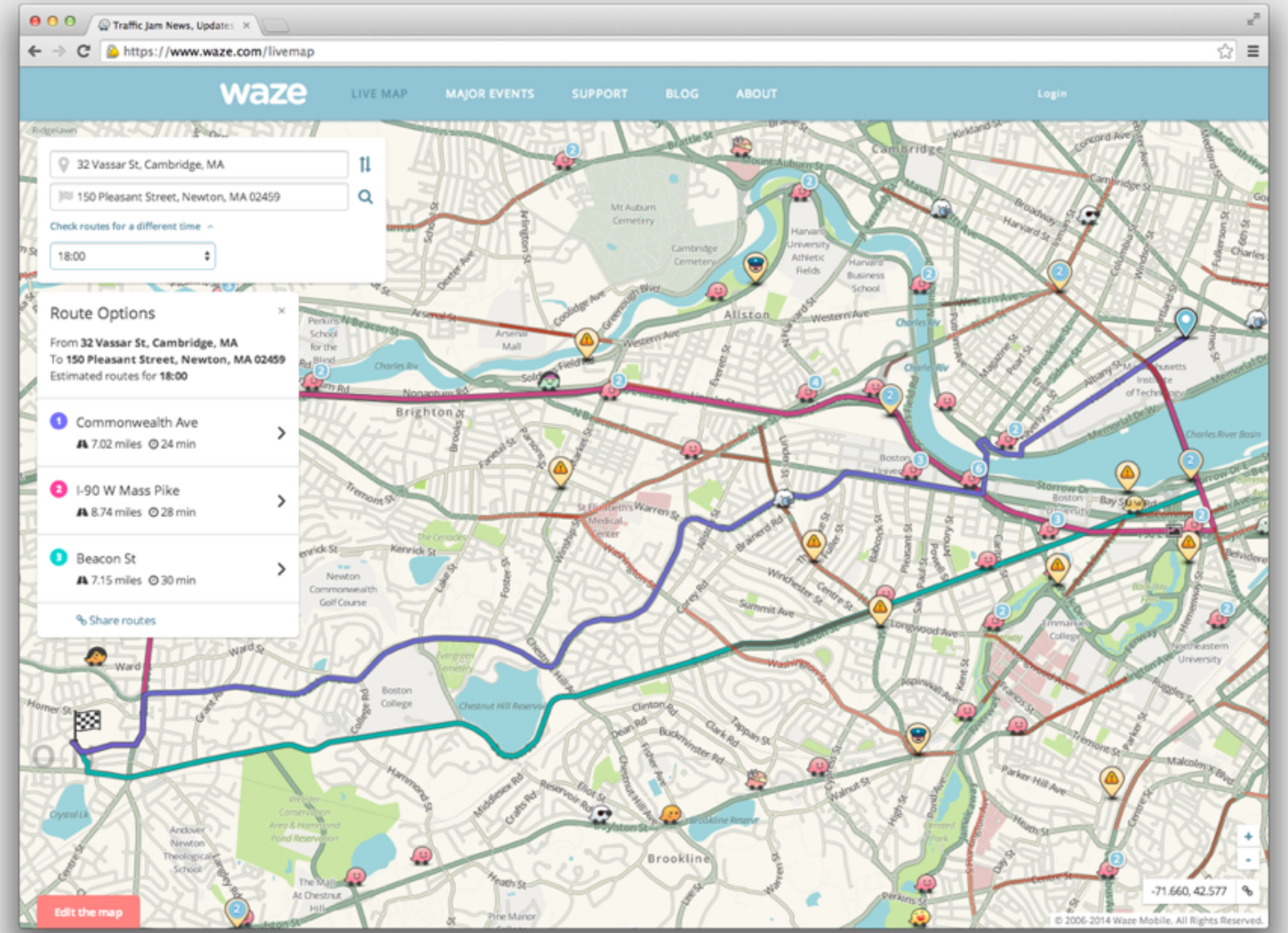
"**if** you select a file, **then** pressing delete will move the file to the trash"

exercise:
concepts

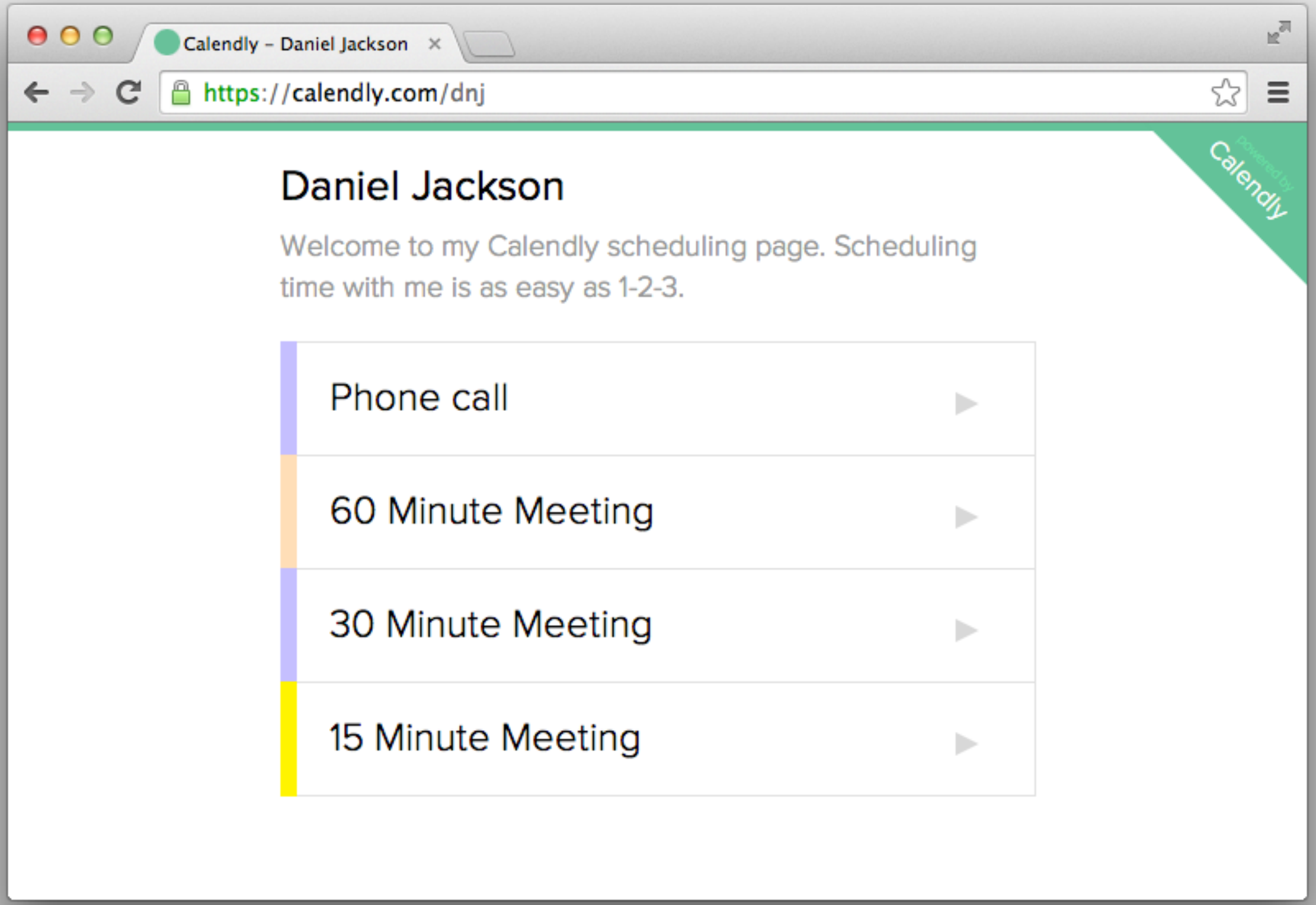
what are the key concepts?



what are the key concepts?

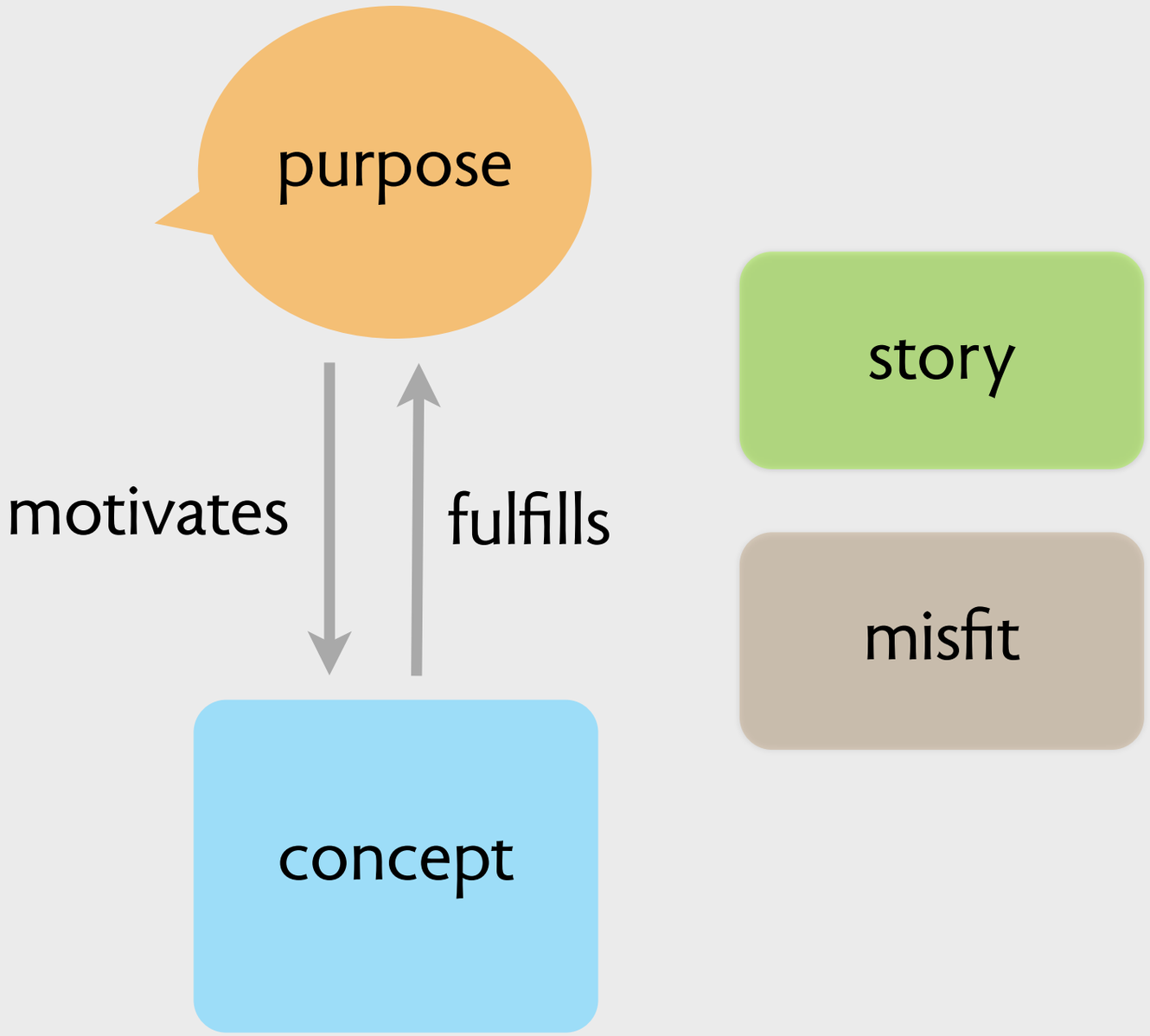


what are the key concepts?

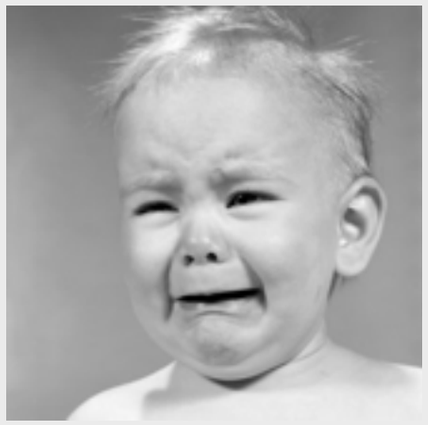


purposes

designing for a purpose



an example



undo deletions

purpose

motivates

fulfills

concept

trashcan

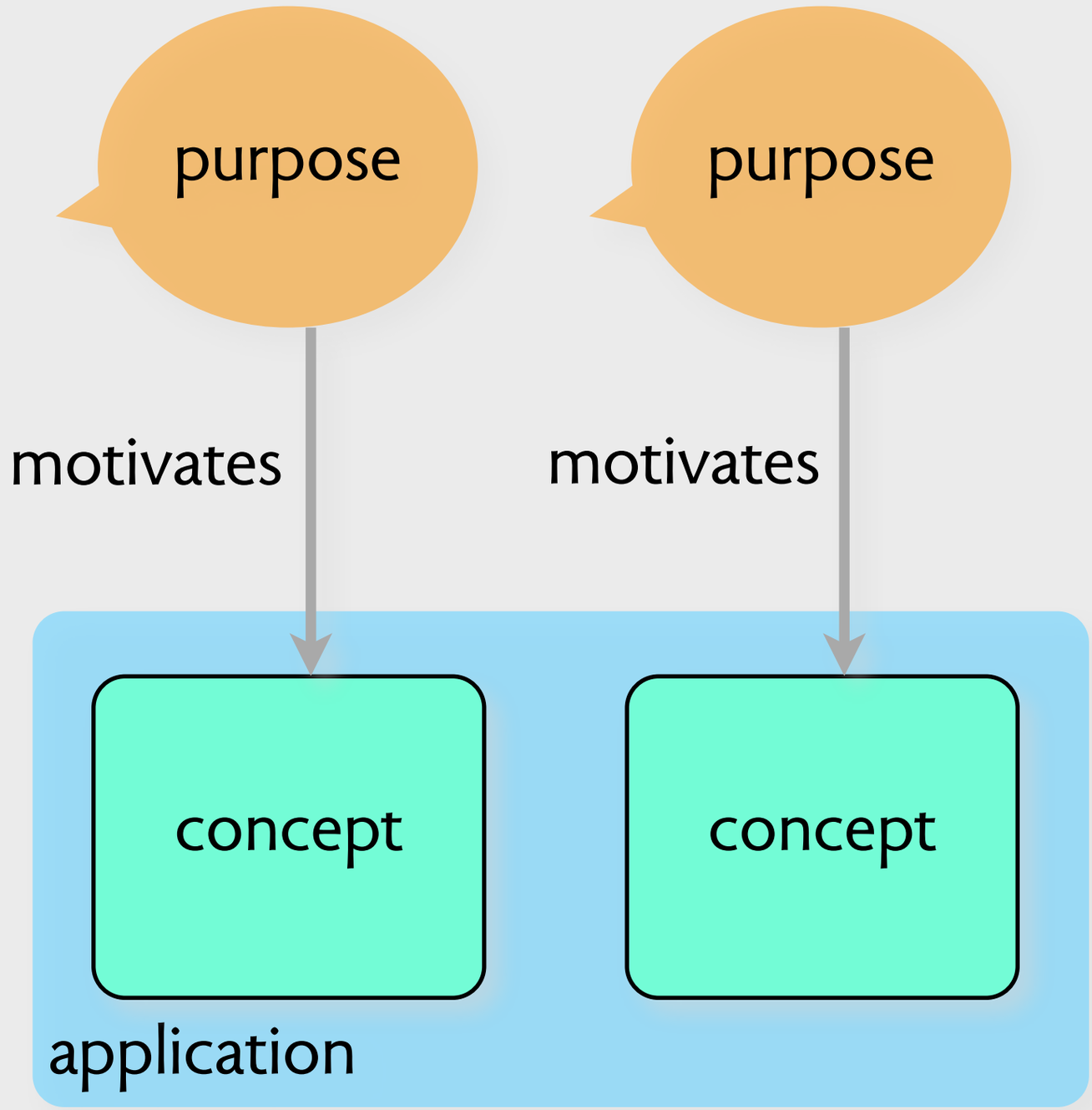
if you select an object and delete it, it will be in the trash; if you empty the trash, it's gone, but if not you can move it out

story

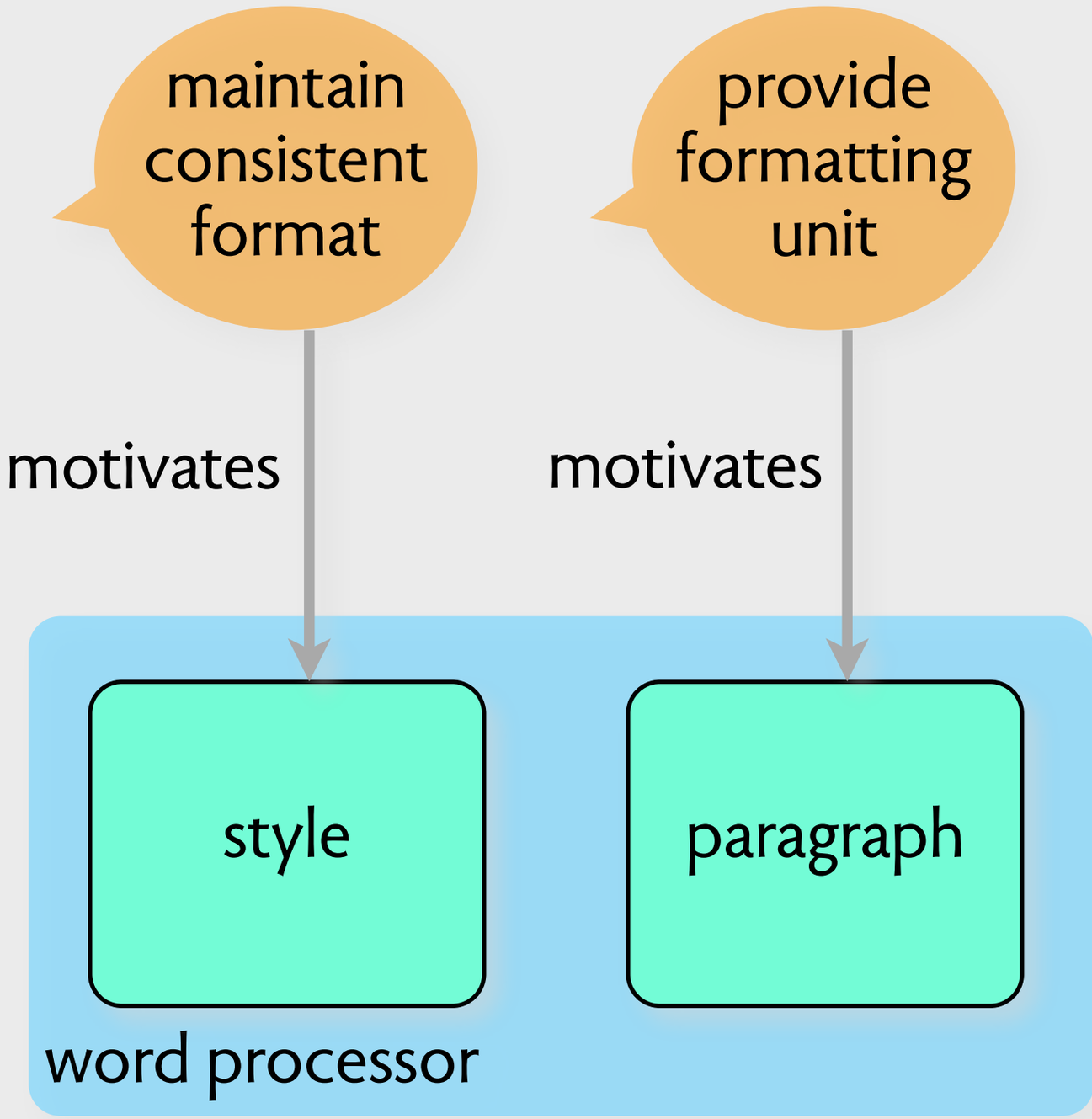
misfit

if there are many objects in the trash, and you delete an object and don't remember it's name, you won't be able to find it again

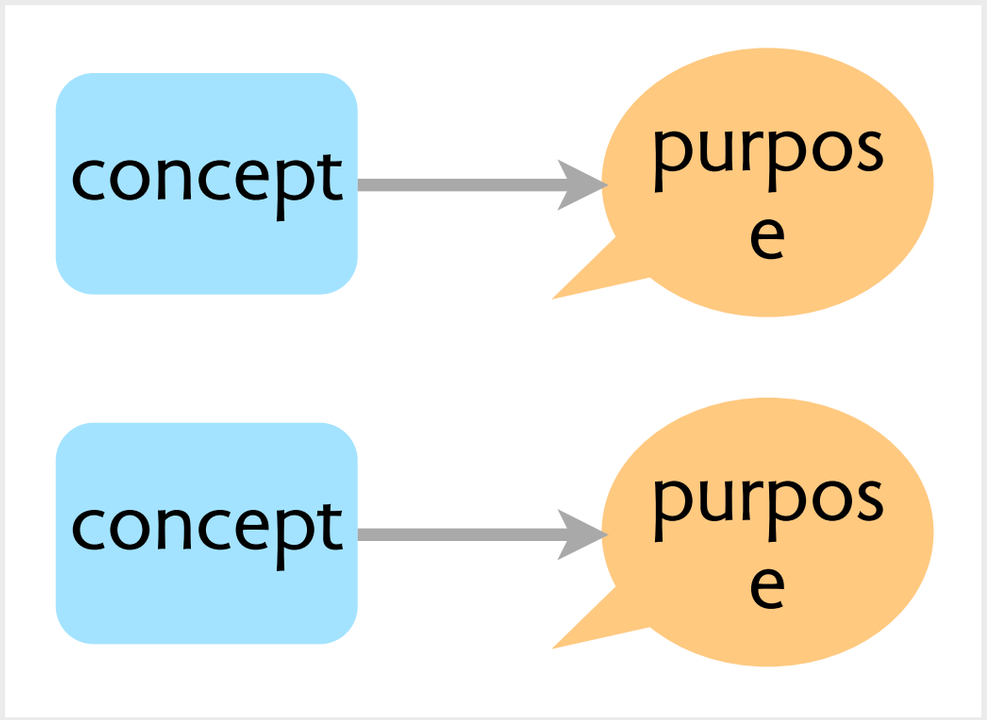
one purpose/concept

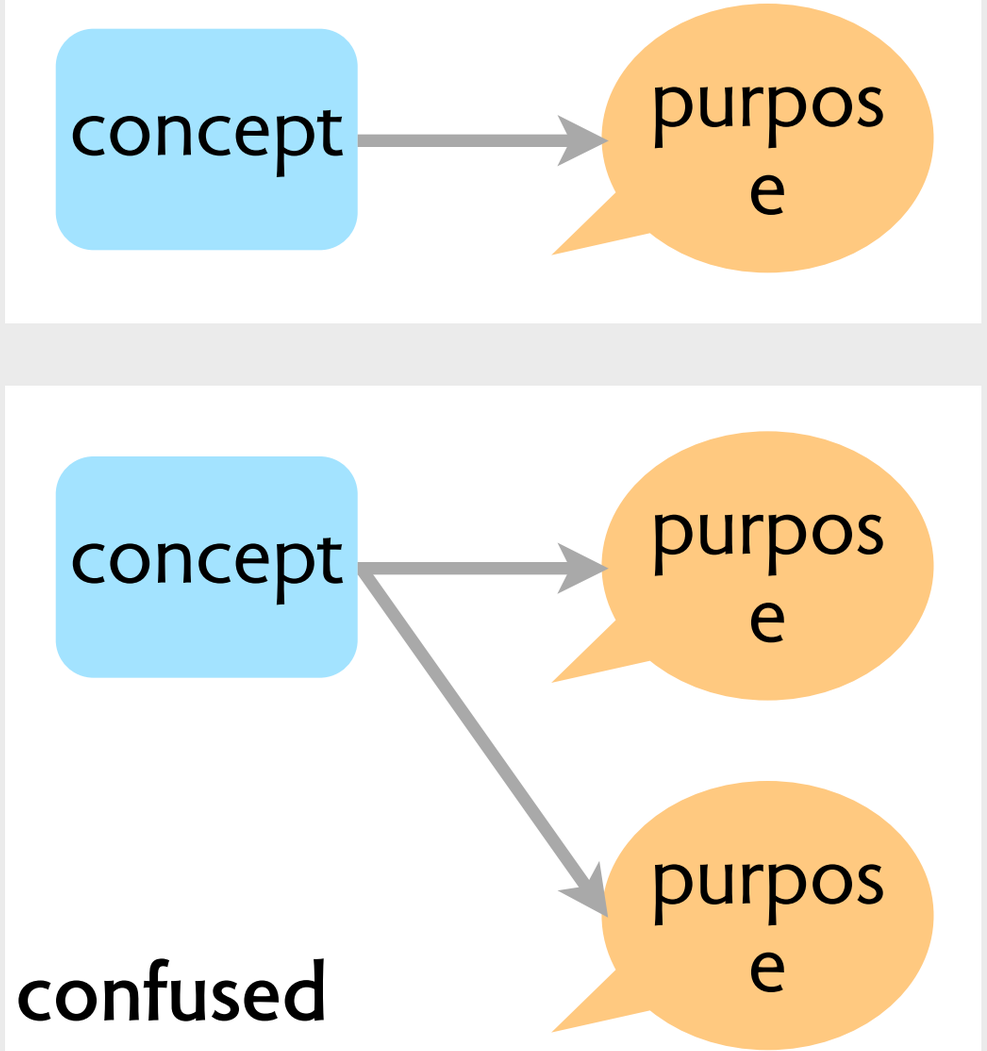


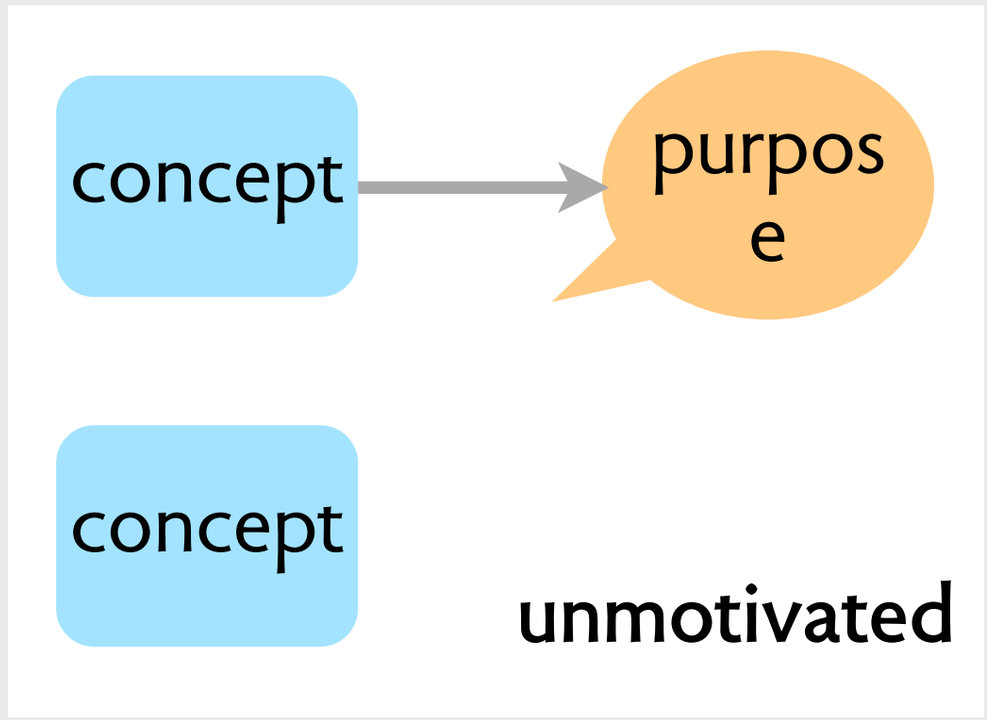
example

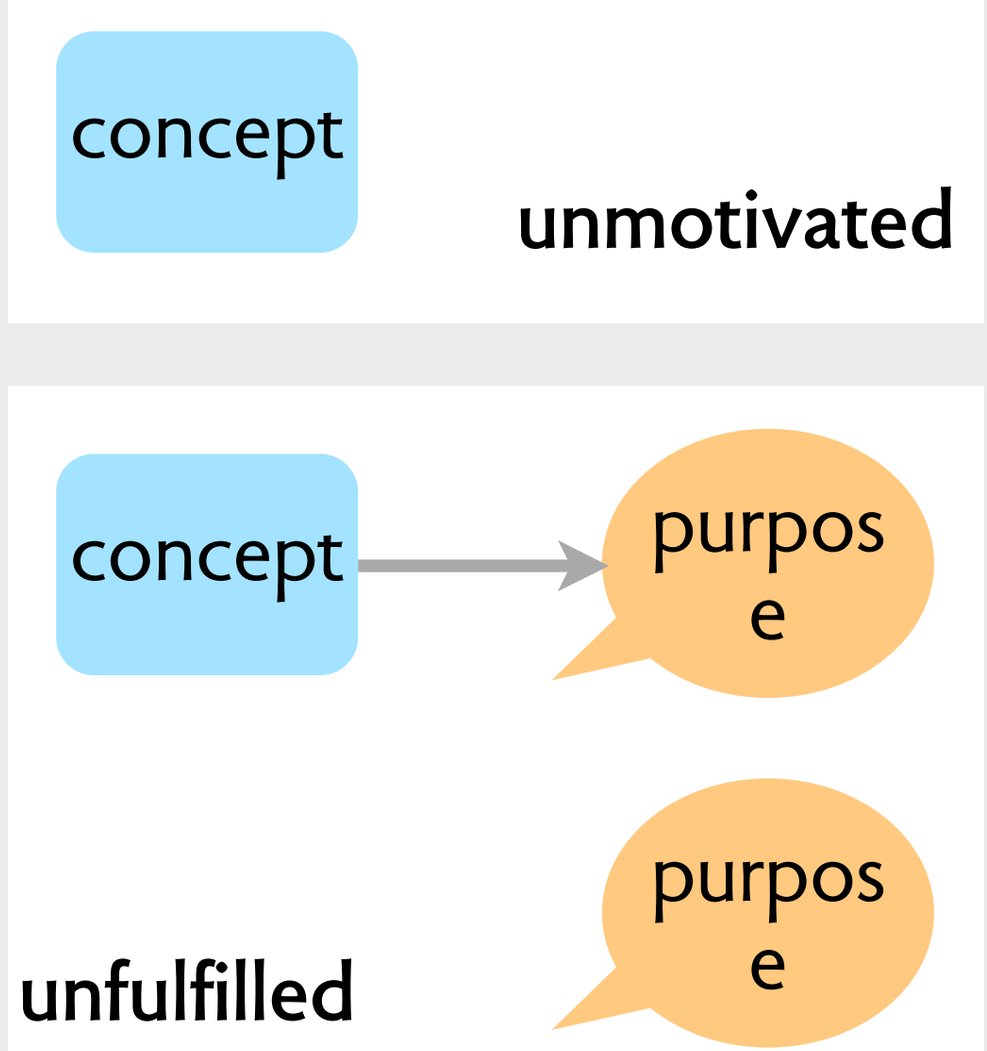

maintain consistent format
provide formatting unit
motivates
motivates
style
paragraph
word processor

concept-purpose map









exercise:

purposes

what's the purpose of ...?

these apps?

Dropbox

Photoshop

WhatsApp

these concepts?

tagging (in Facebook)

buffers (in emacs)

relative references (in Excel)

concept

criteria

concept criteria

compelling:clear story, one purpose

controllable: fulfills purpose without interference

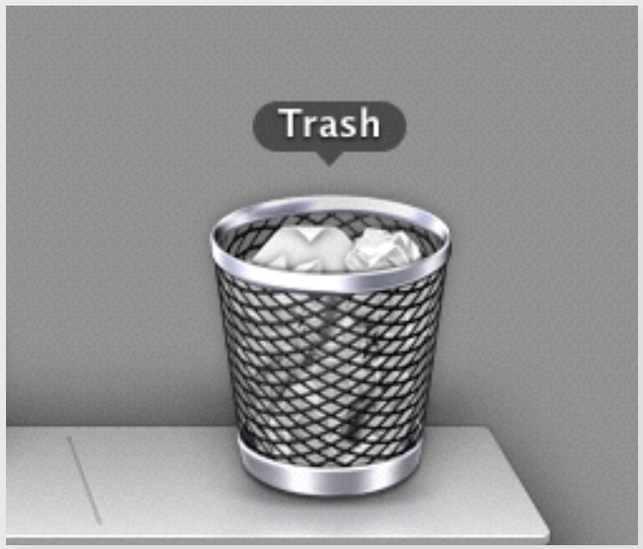
complete: general and uniform over concept's scope

consistent: represented consistently in the user interface

conventional: reuses existing concepts when applicable

compelling

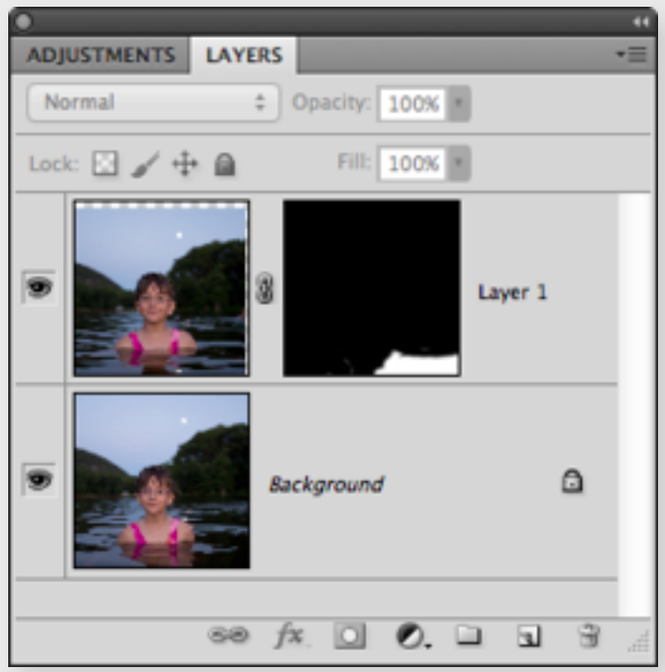
simple story & motivated by one purpose



trash (OS X)

story: delete moves to special folder; emptying it removes contents for good

purpose: allow undo of deletions



layer (Photoshop)

story: resulting pixel map is background map with composition of functions given by layers

purpose: non-destructive edits

faucets

compelling

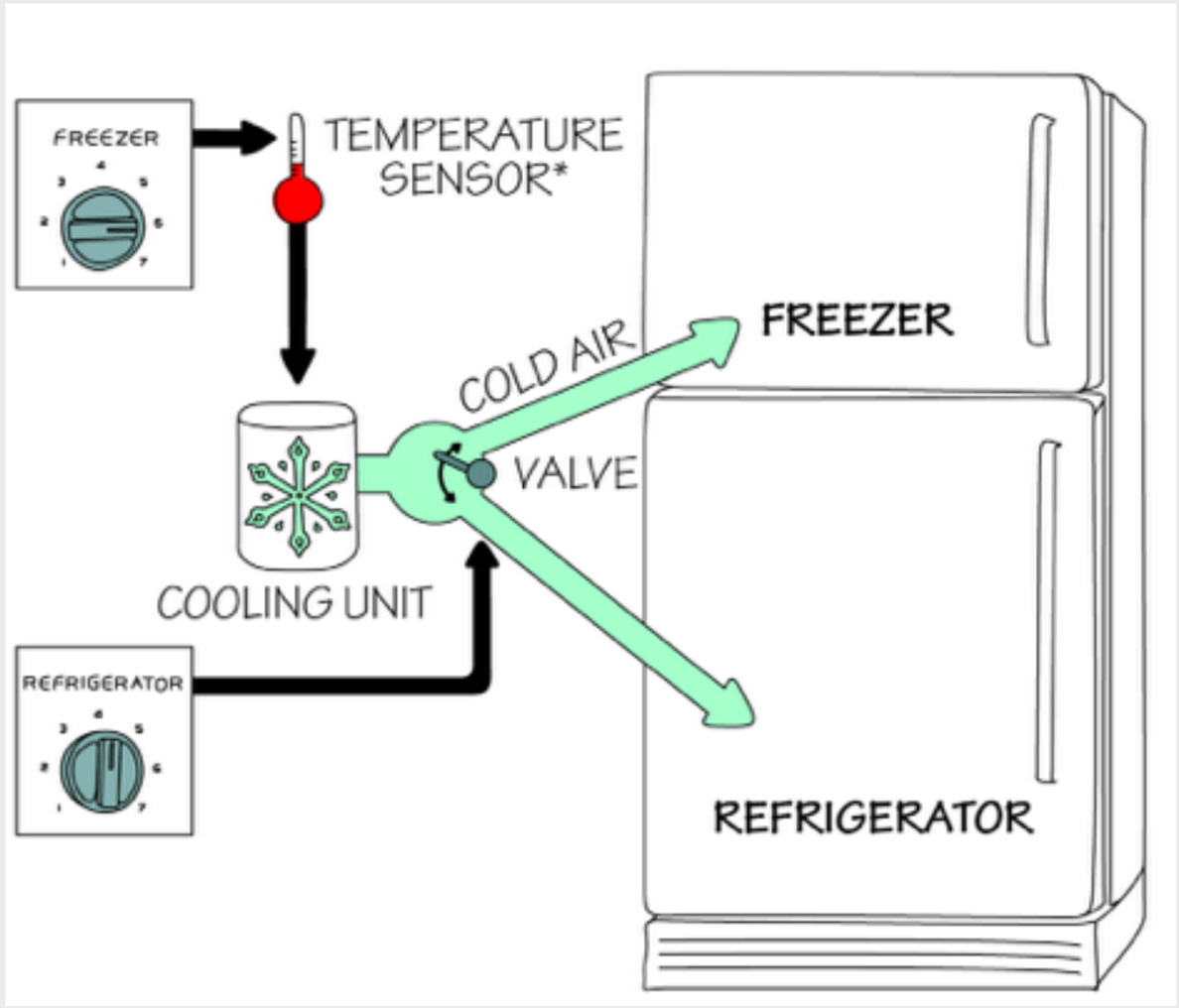
not compelling

refrigerator controls

Don Norman, Design of Everyday Things, 1988 (2013)

controls

internal mechanism



more uncompelling concepts

no single clear purpose

Git's staging area

JPEG quality and resolution

Gmail's categories (vs labels)

no simple operational principle

CSS's dimensions (inches, pixels)

Tumblr's answers vs comments

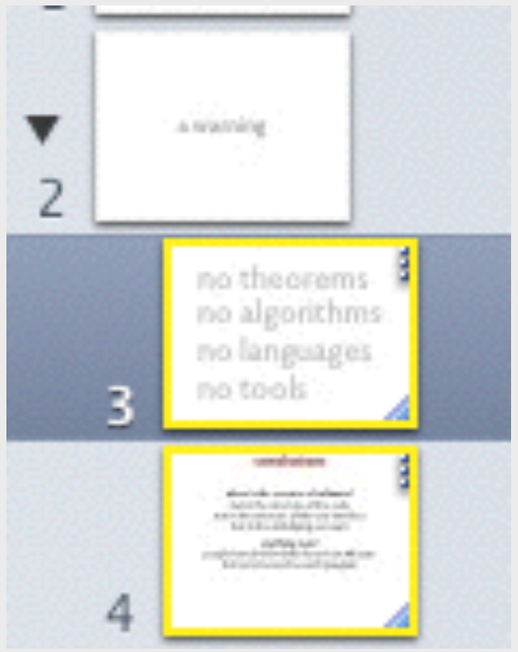
serves someone else's purpose

direct flight (legs share flight number)

controllable

fulfills purpose without interference from other concepts

skip slide



make parent

make parent and skip

skipping and hierarchy do not interfere in Apple Keynote

gmail labels & conversations

from Eunsuk Kang

Eunsuk Kang <eskang@csail.mit.edu>
To: Daniel Jackson <dnj@MIT.EDU> , Santiago Perez De Rosso <sperezde@csail.mit.edu>
an example of bad coupling in gmail?

December 11, 2014 2:27 AM
Hide Details
Conceptual Design

Hi Daniel, Santiago,

I am not sure whether this falls into one of the conceptual design issues that you have been analyzing, but here's a little annoyance that I just ran into using labels in Gmail.

This evening, I made two separate reservations with Enterprise (one in Phoenix and another one in Vegas), and the company sent me a confirmation e-mail for each one of them. Usually, I tag travel-related e-mails using a label (e.g. "Vegas Trip 2014") so that I can easily find all the relevant e-mails for a particular trip. Now, the problem is that Gmail automatically grouped the two Enterprise emails into a single conversation. It turns out that in Gmail, a label applied to an e-mail in a conversation is applied to every other e-mail in the same conversation. This means I can't assign separate labels to the two car rentals! Uggh!

(By the way, is it possible to take an e-mail out of a conversation? Either this is impossible or the Gmail UI does not make this easy).

Eunsuk

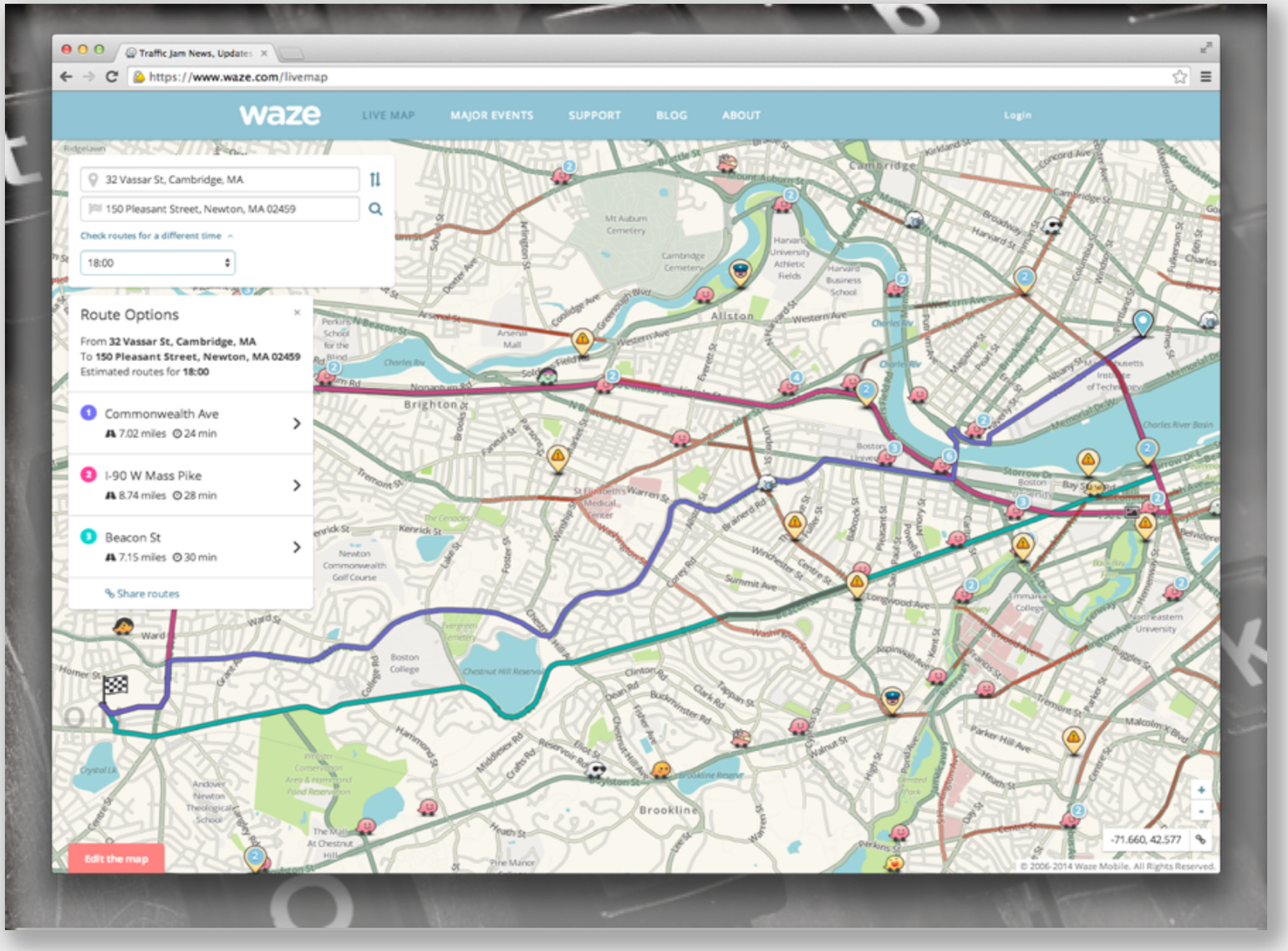
conversations interfere with labels

aspect ratio fuji x100s

image quality setting

aspect ratio

“image size” setting

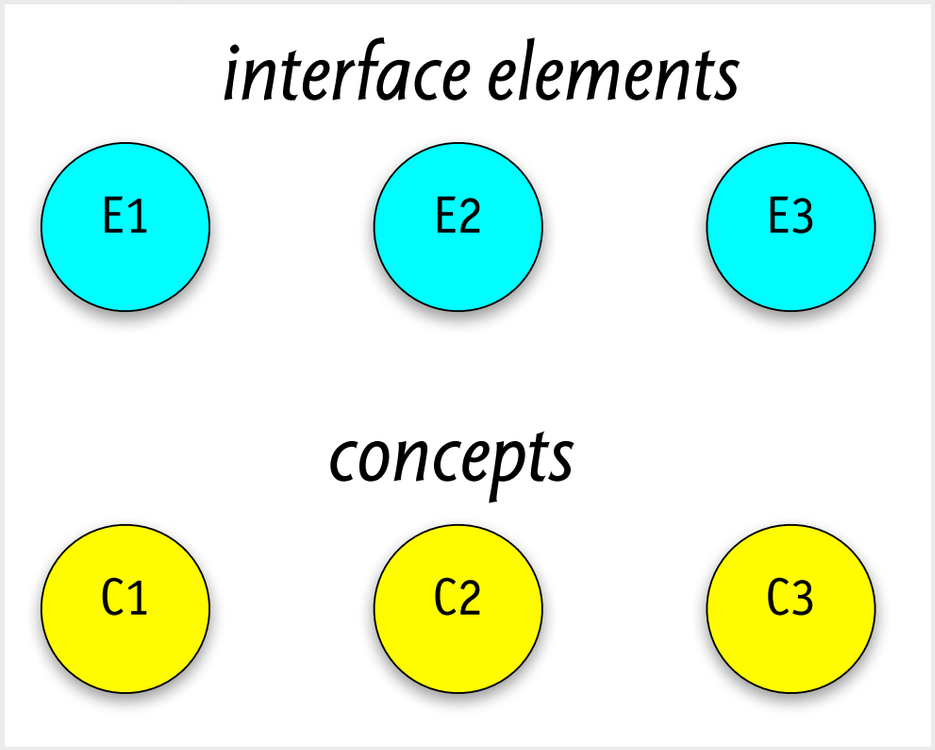


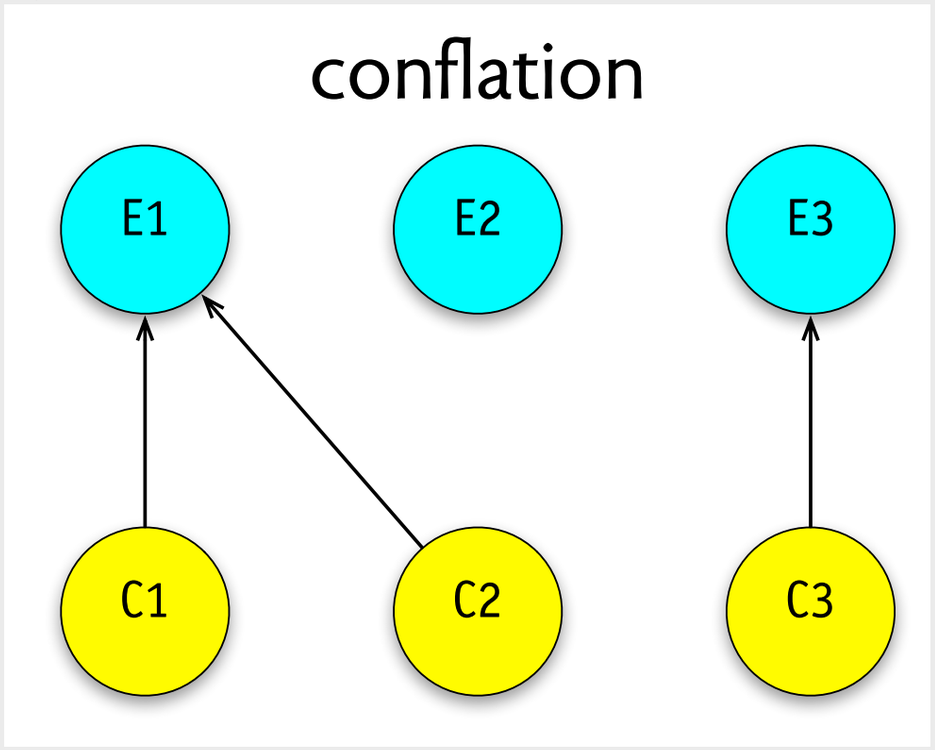
non-standard ratio + RAW?

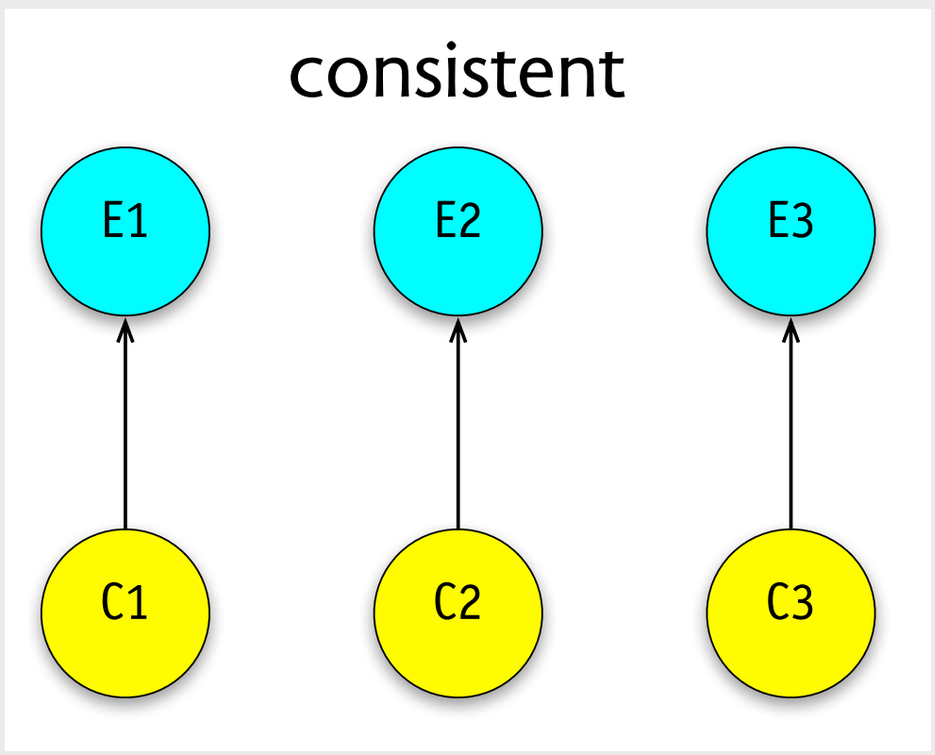
image quality interferes with image size

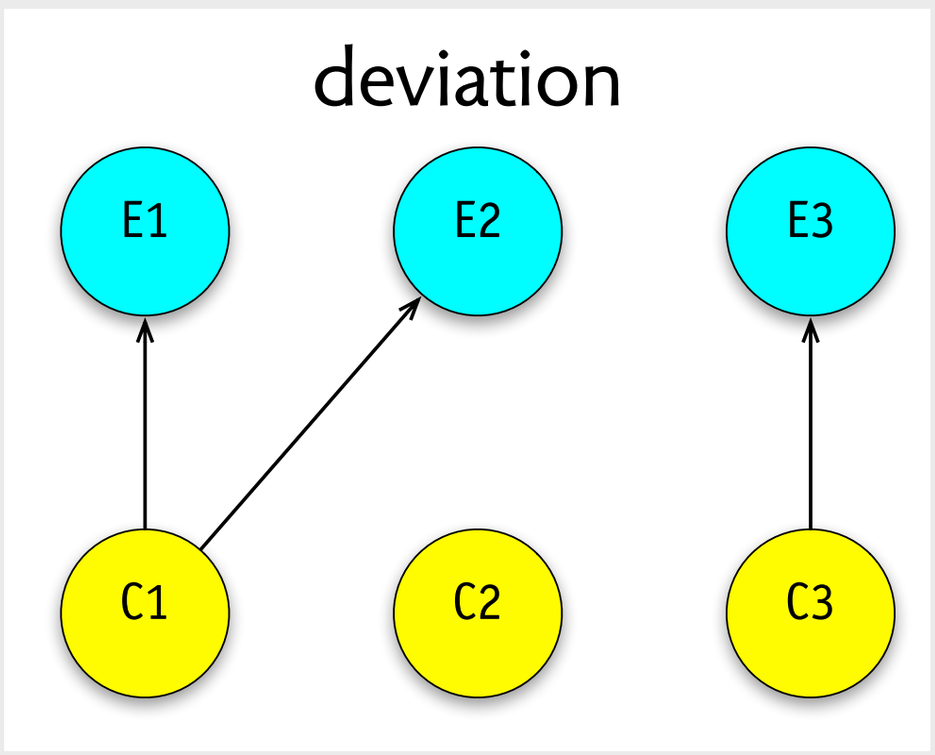
consistent

represented consistently in the user interface





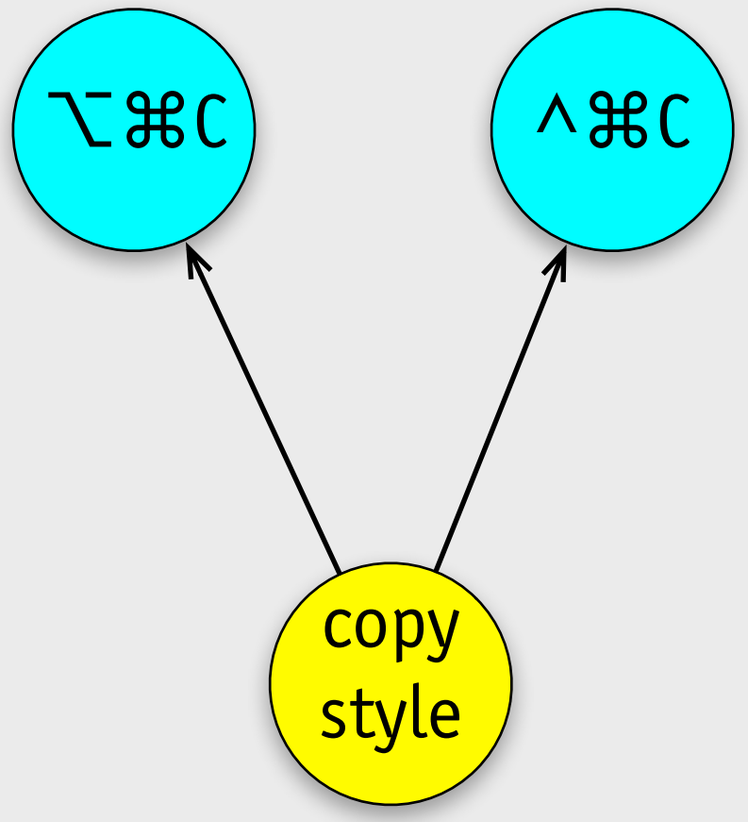




deviation in OS X apps

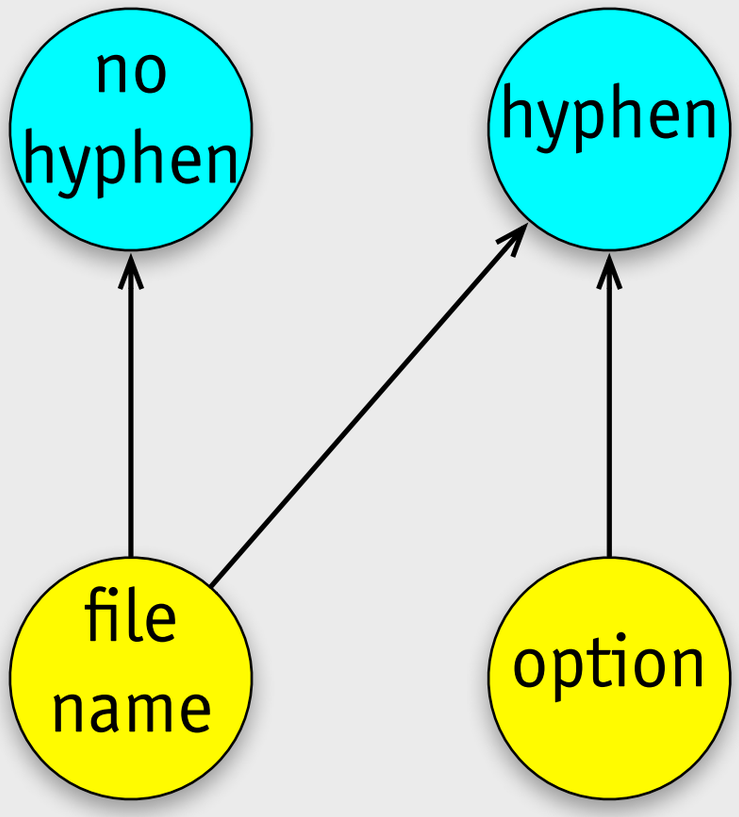
OmniGraffle:
control key

Apple apps:
option key



conflation in unix

```
> touch readme
> ls
readme
> mv readme -README
> ls
-README
> rm -README
rm: illegal option -- E
usage: rm [-f | -i] [-dPRrvW] file ...
```



a sad dropbox tale

Quora

Search

Dropbox: Edit

Someone accidentally deleted thousands of files in my company Dropbox: how can I quickly undelete them? Edit

Add Question Details

Comment · Share · Report · Options

STORAGE & NETWORKING file sharing, dropbox

Friends don't let friends delete shared Dropbox items

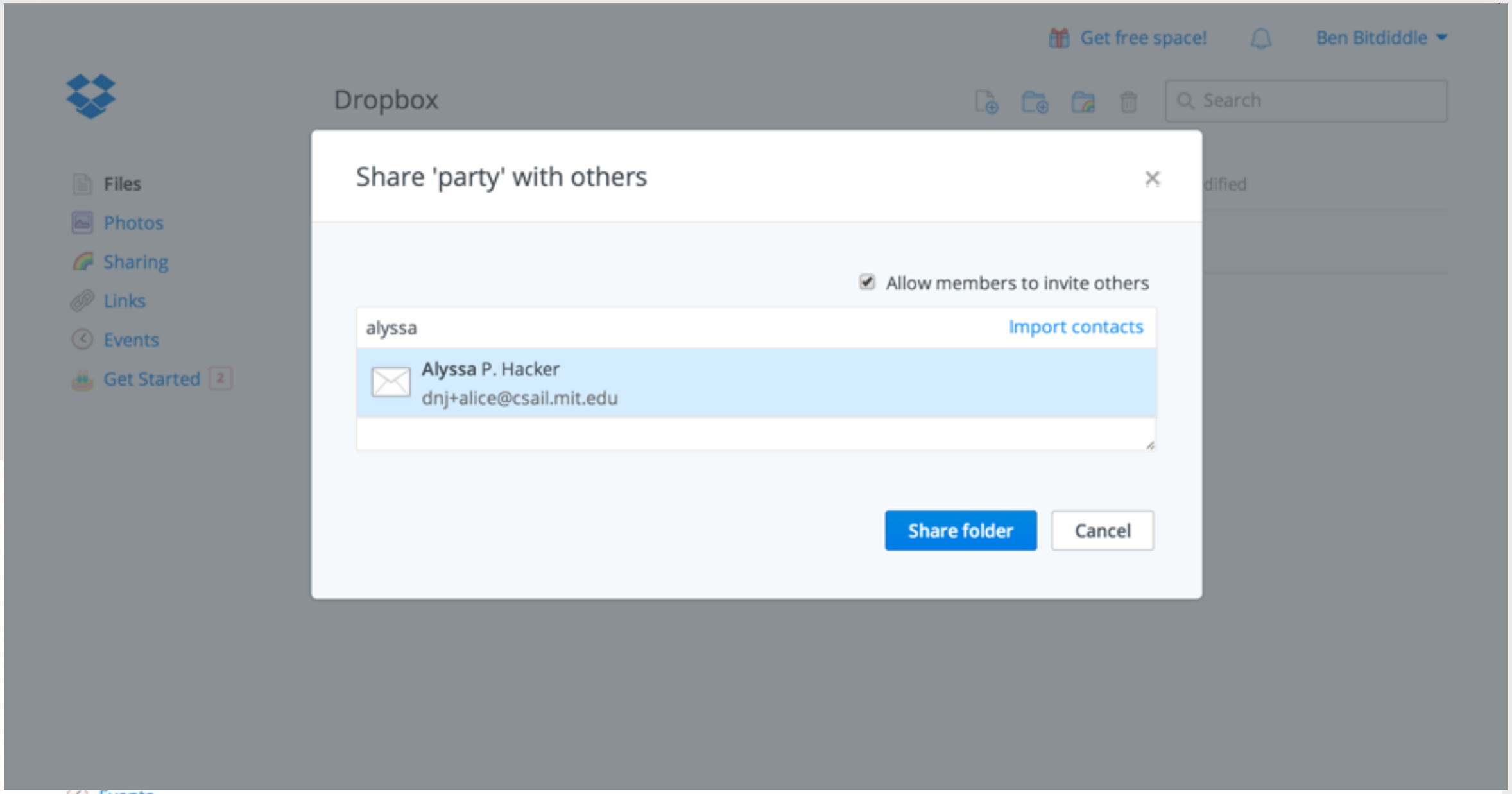
Christopher Breen
@BodyofBreen

Sep 9, 2013 5:00 AM

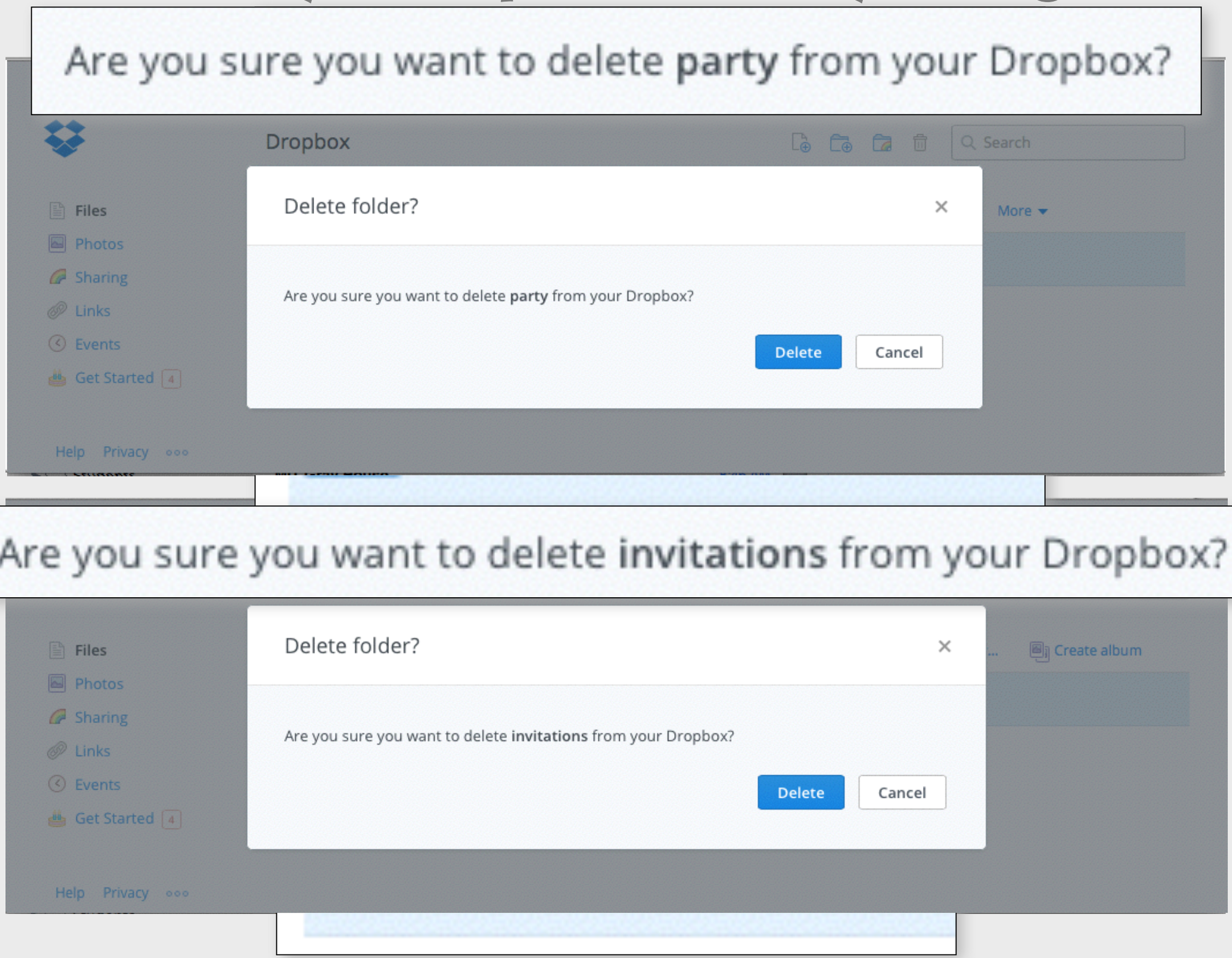
Reader Paul Cramblett has a problem with others who just don't know how to share. He writes:

> *I maintain a Dropbox folder that I use to share files with a select group of friends. I've tried to explain how Dropbox works to these people but someone invariably drags all the files out of the folder, which means they're no longer available to the rest of us. Is there some way to prevent files from being removed by someone who doesn't understand the difference between "copy" and "move"?*

alyssa & ben plan a party



alyssa spoils everything

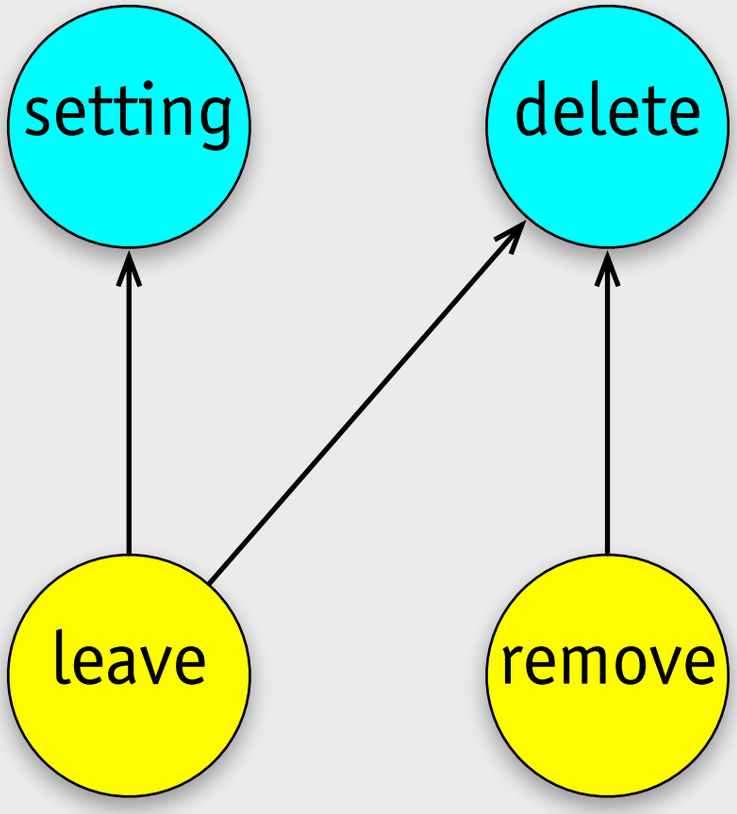


conflation in dropbox

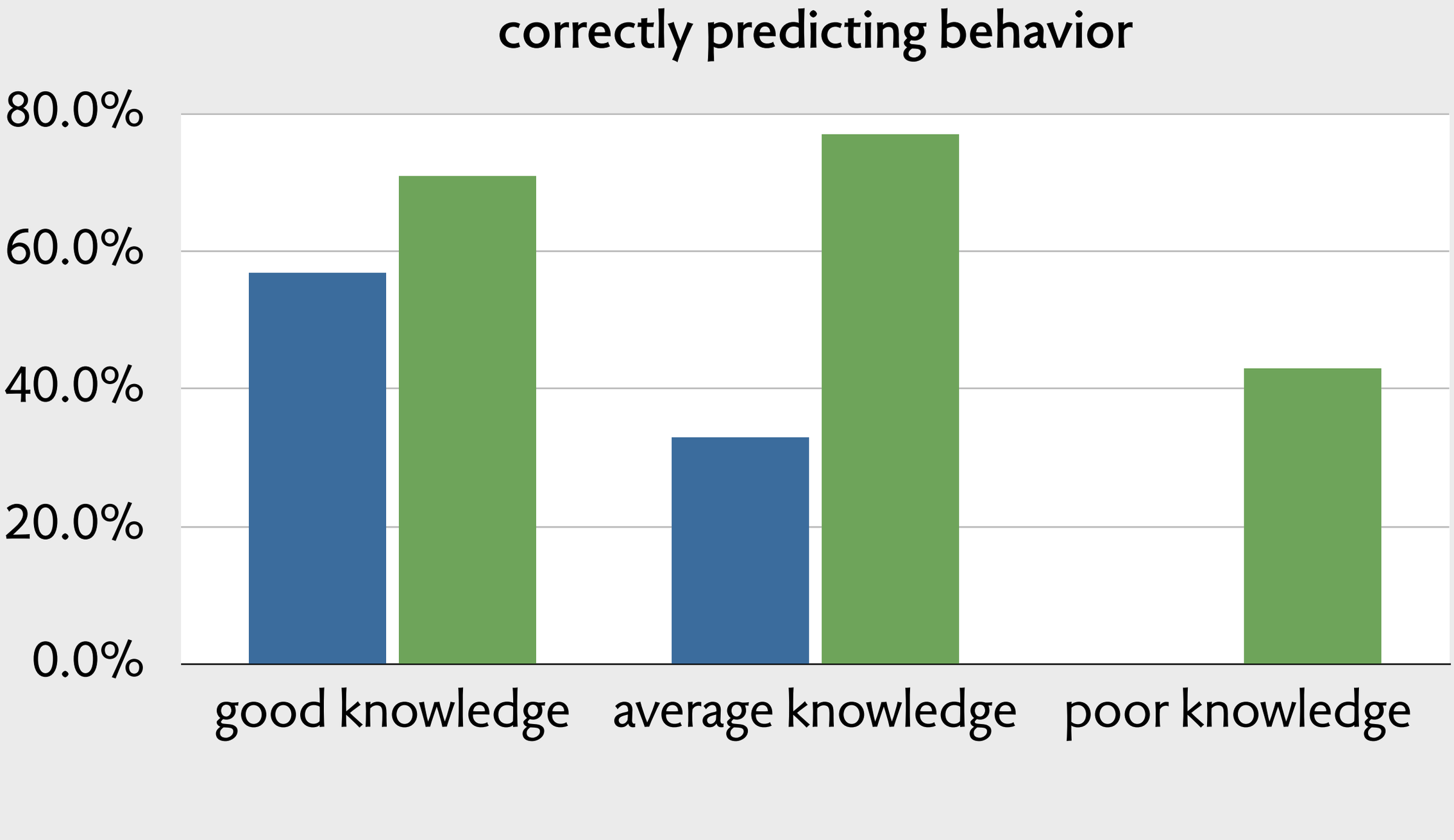
How do I leave a shared folder?

“by changing the folder’s settings or by deleting it”

You can leave a shared folder at any time by changing the folder's settings on the Dropbox website or deleting it. If you change your mind or leave a folder accidentally, you can rejoin it later.



small survey of MIT dropbox users



study by Kelly Zhang

conventional

reuses existing concepts when applicable

concept idioms: across apps

style: separate formatting from element

selection: select objects to apply action to set

subtle issues & their solution

reapply in different context

eg, partial styles

some apps are just cliches

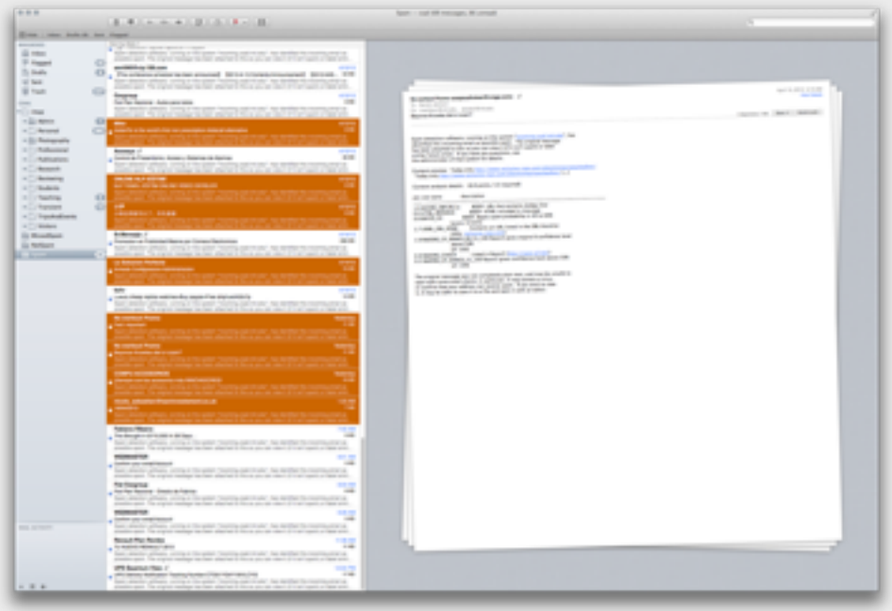
collection of idiomatic concepts

social app = {profile, post, friend, notify}

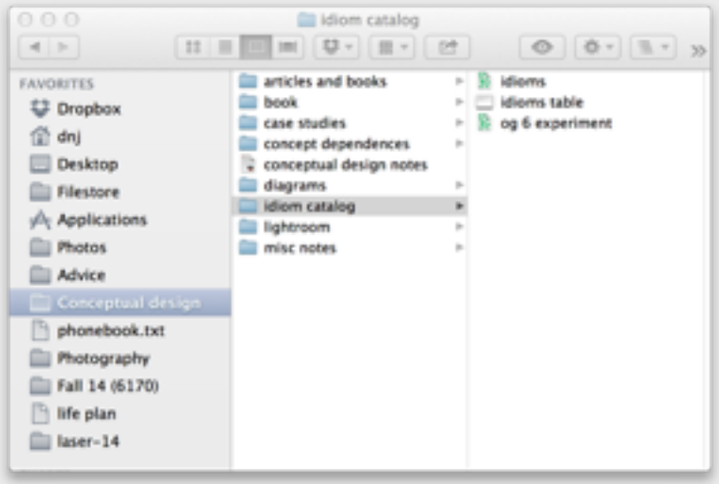
concept idiom selection

slides in Keynote

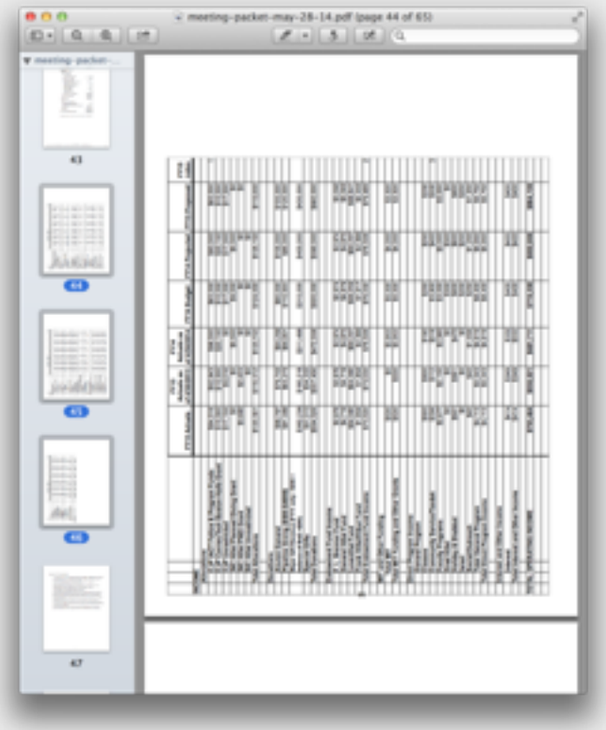
photos in Adobe Lightroom



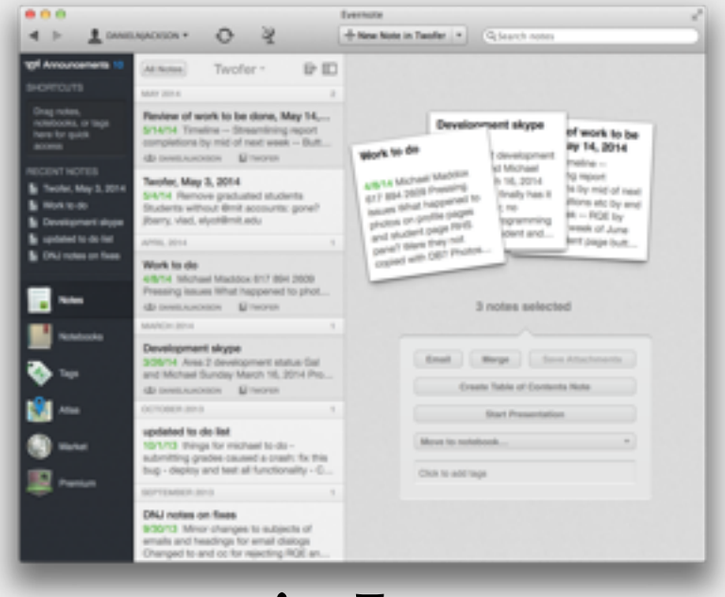
messages in Apple Mail



objects in OS X Finder

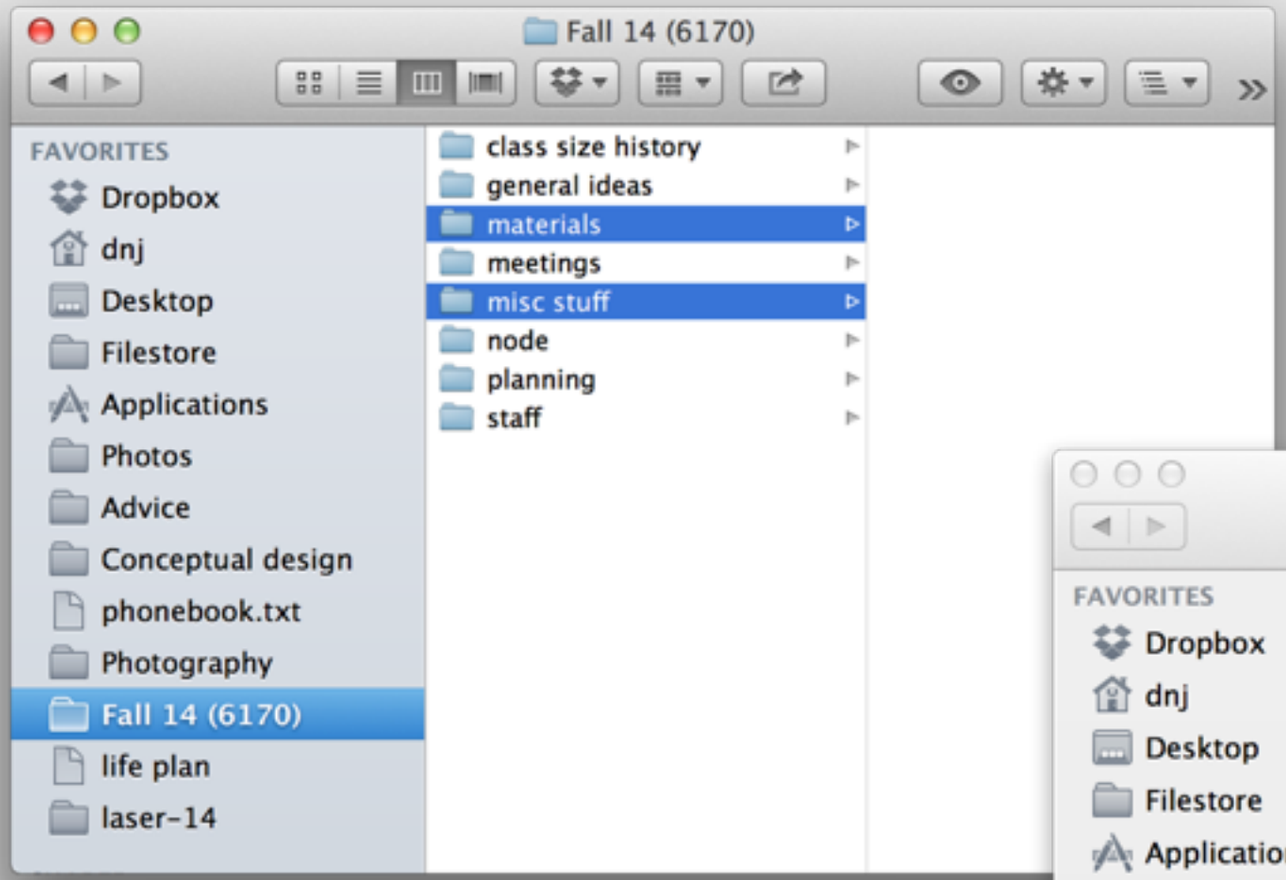


thumbnails in Preview

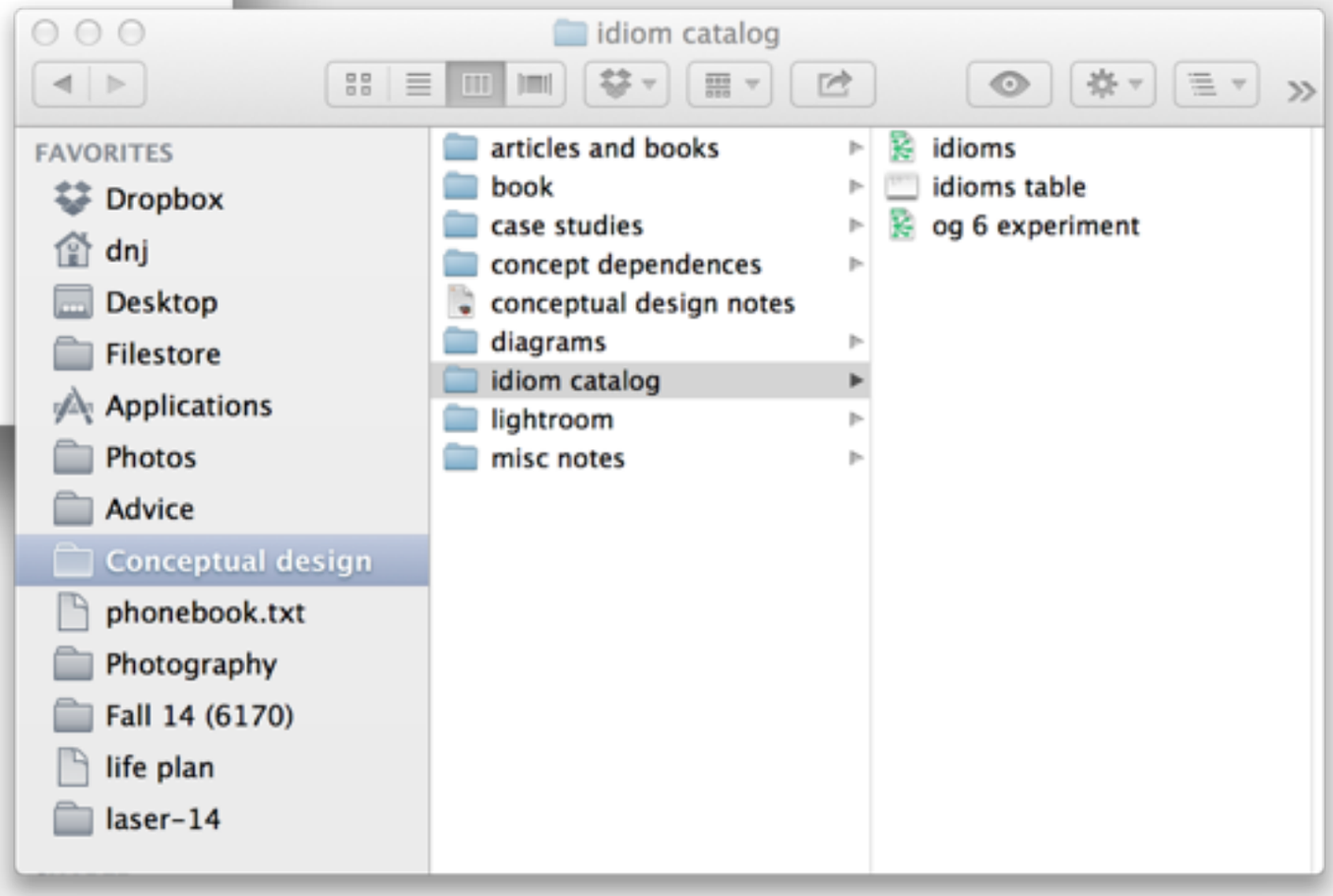


notes in Evernote

subtlety selection scope



subset of selection in scope

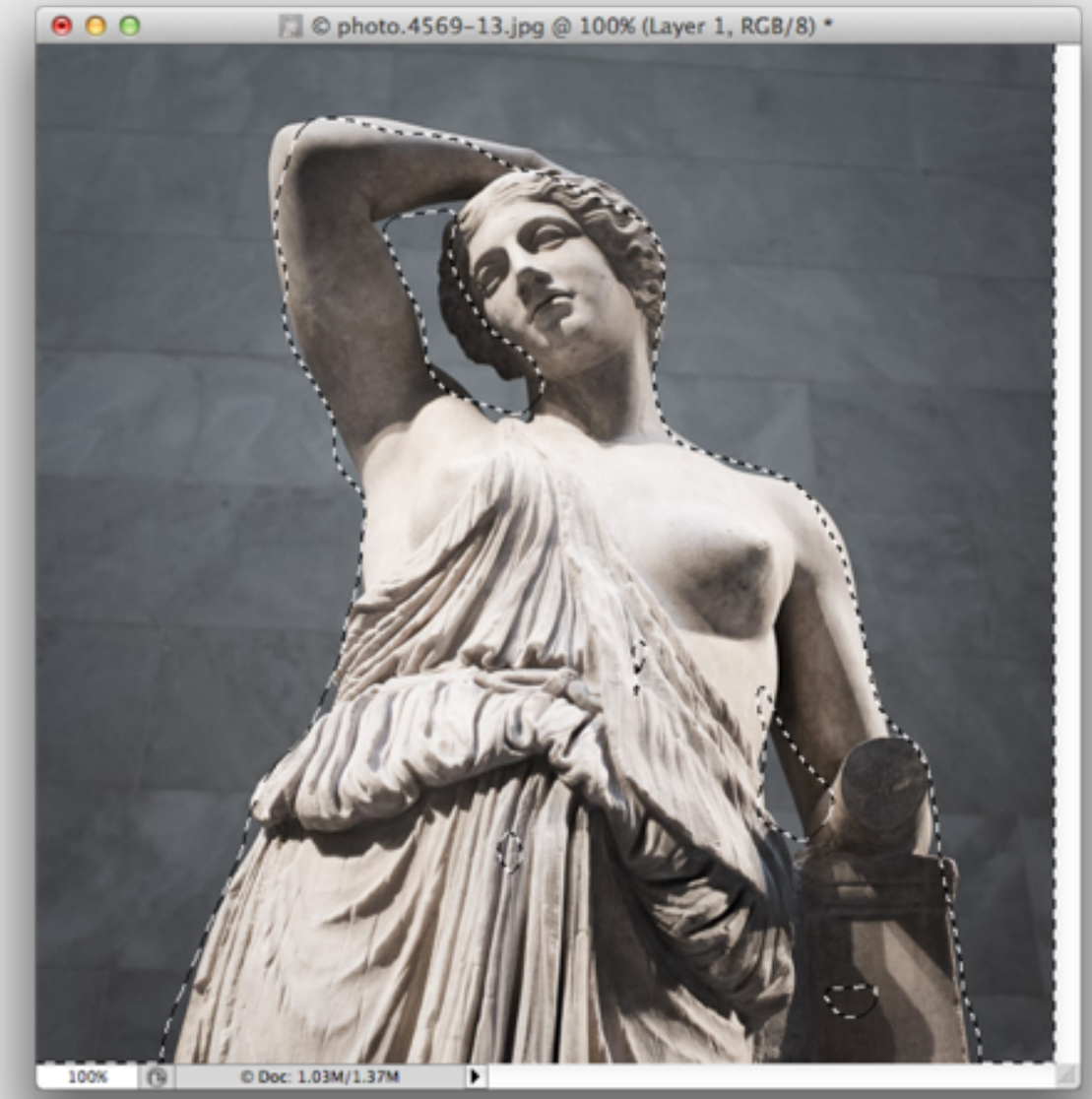


subset of selection out of scope

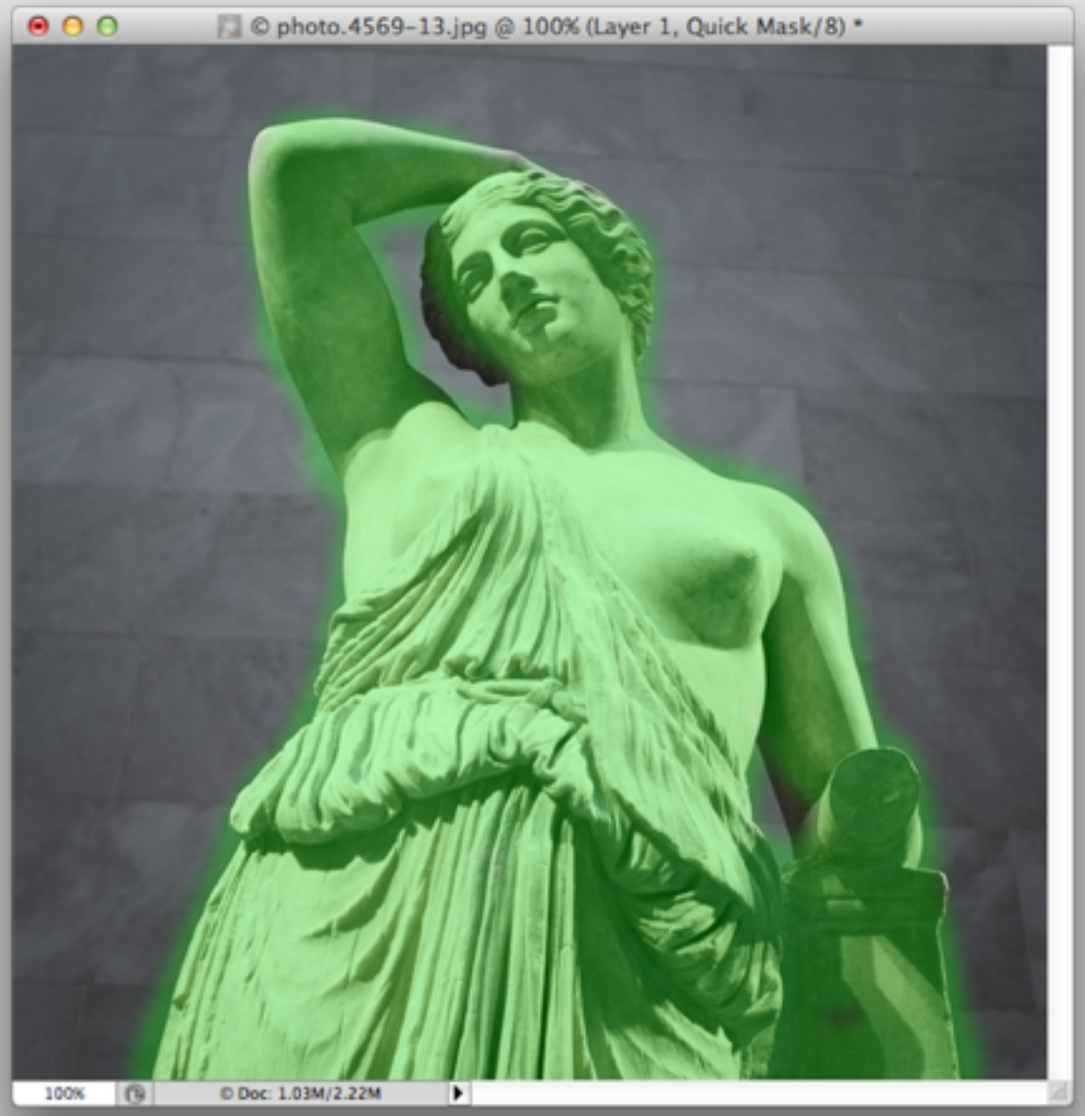
subtlety active element

Adobe Lightroom: brightest thumbnail is the “active photo”

subtlety continuous selection

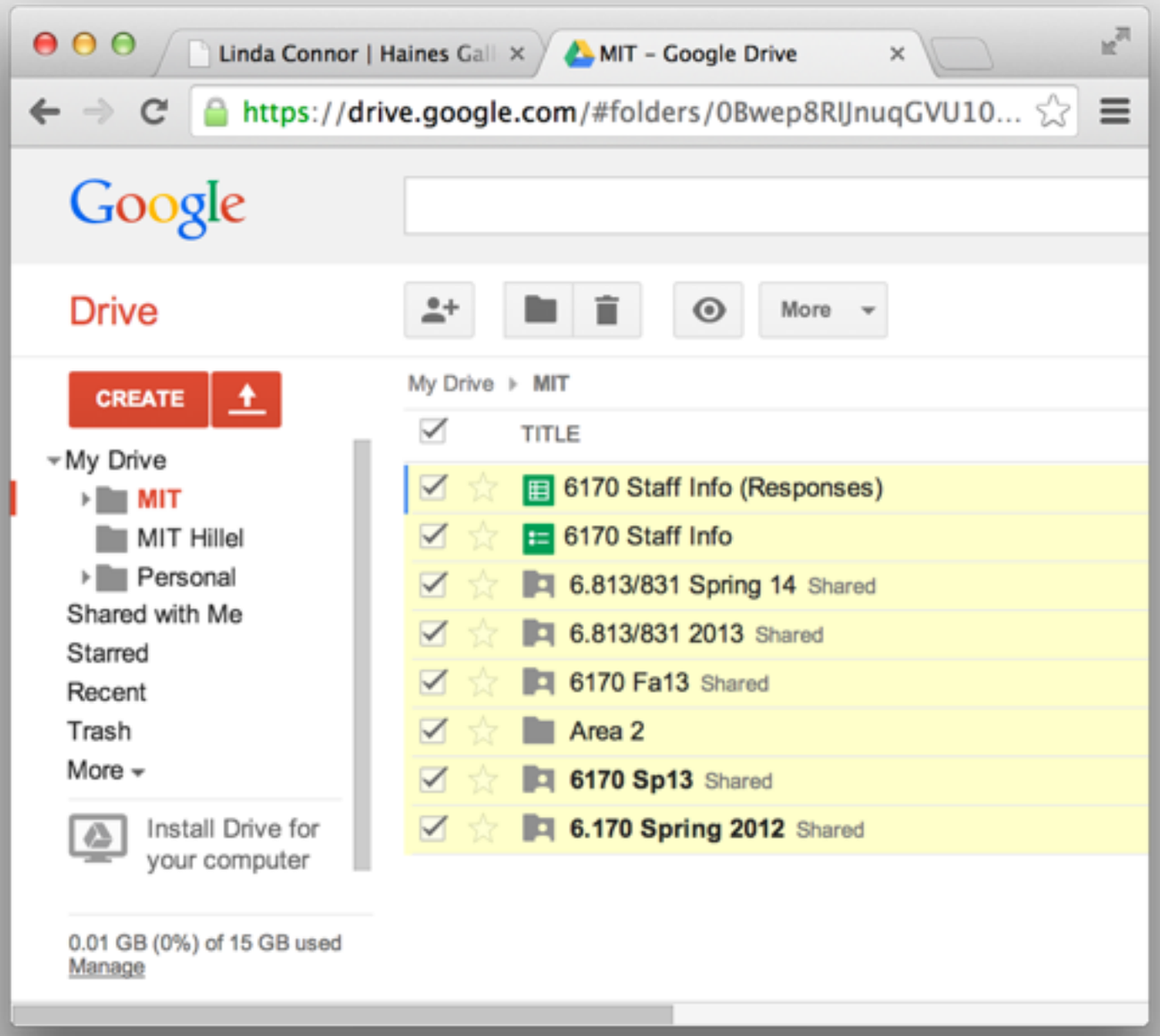


Photoshop: outline shown with “marching ants”

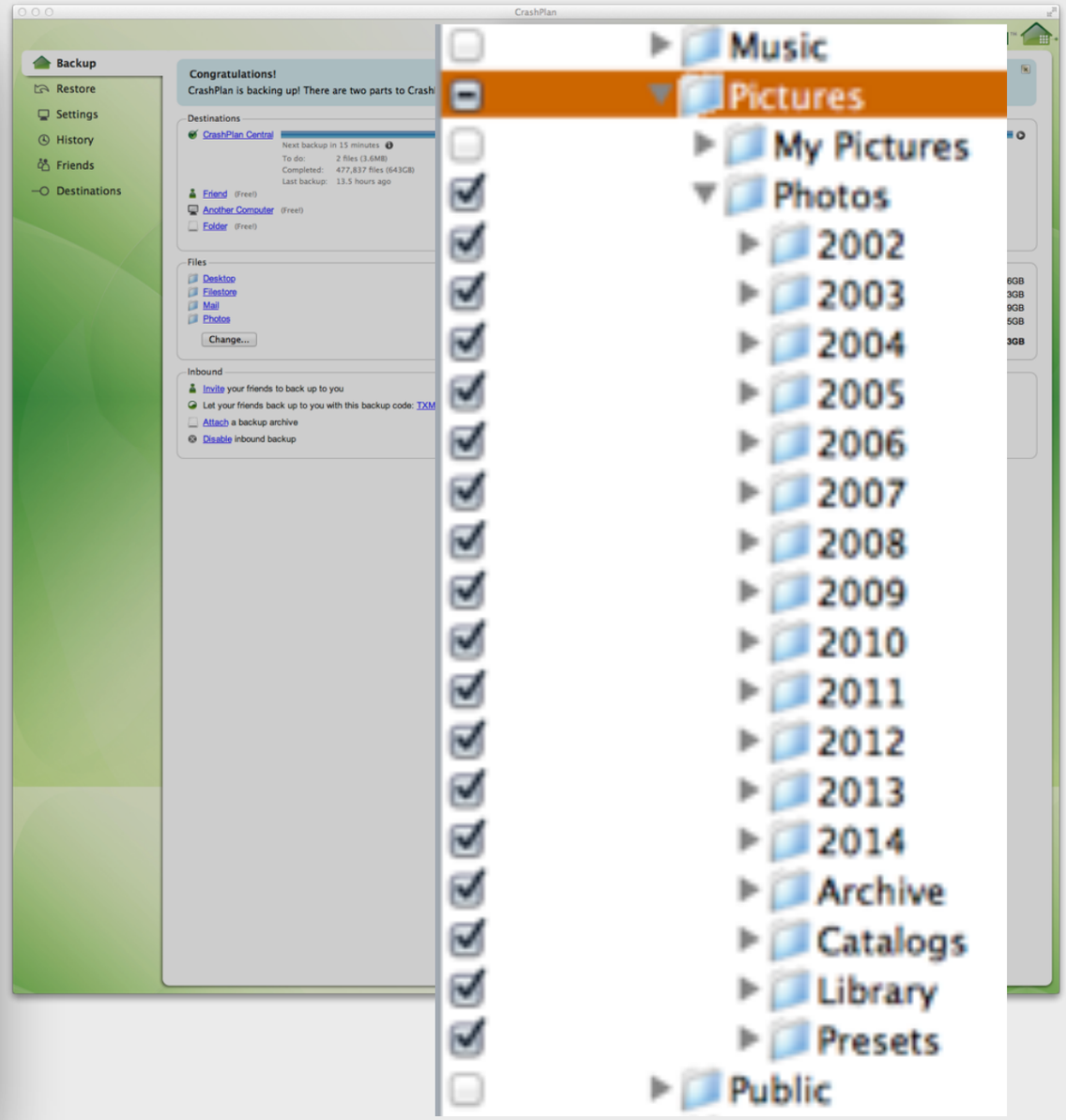


Photoshop: selection shown in Quick Mask mode

subtlety folder selection

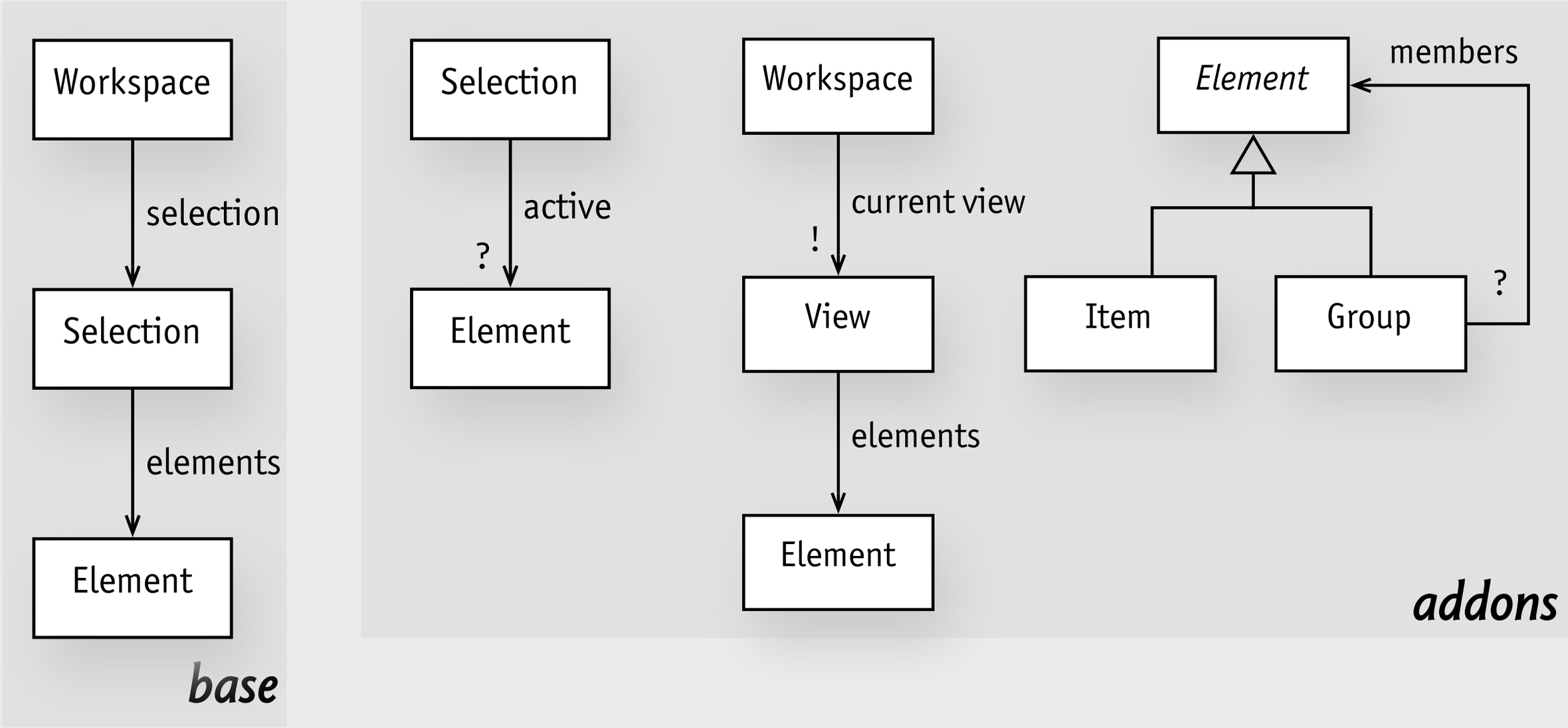


Google Drive: selecting folder = selecting children



CrashPlan: selecting folder = selecting all future children

concept idiom selection



Purposes: apply action in aggregate to many items at once

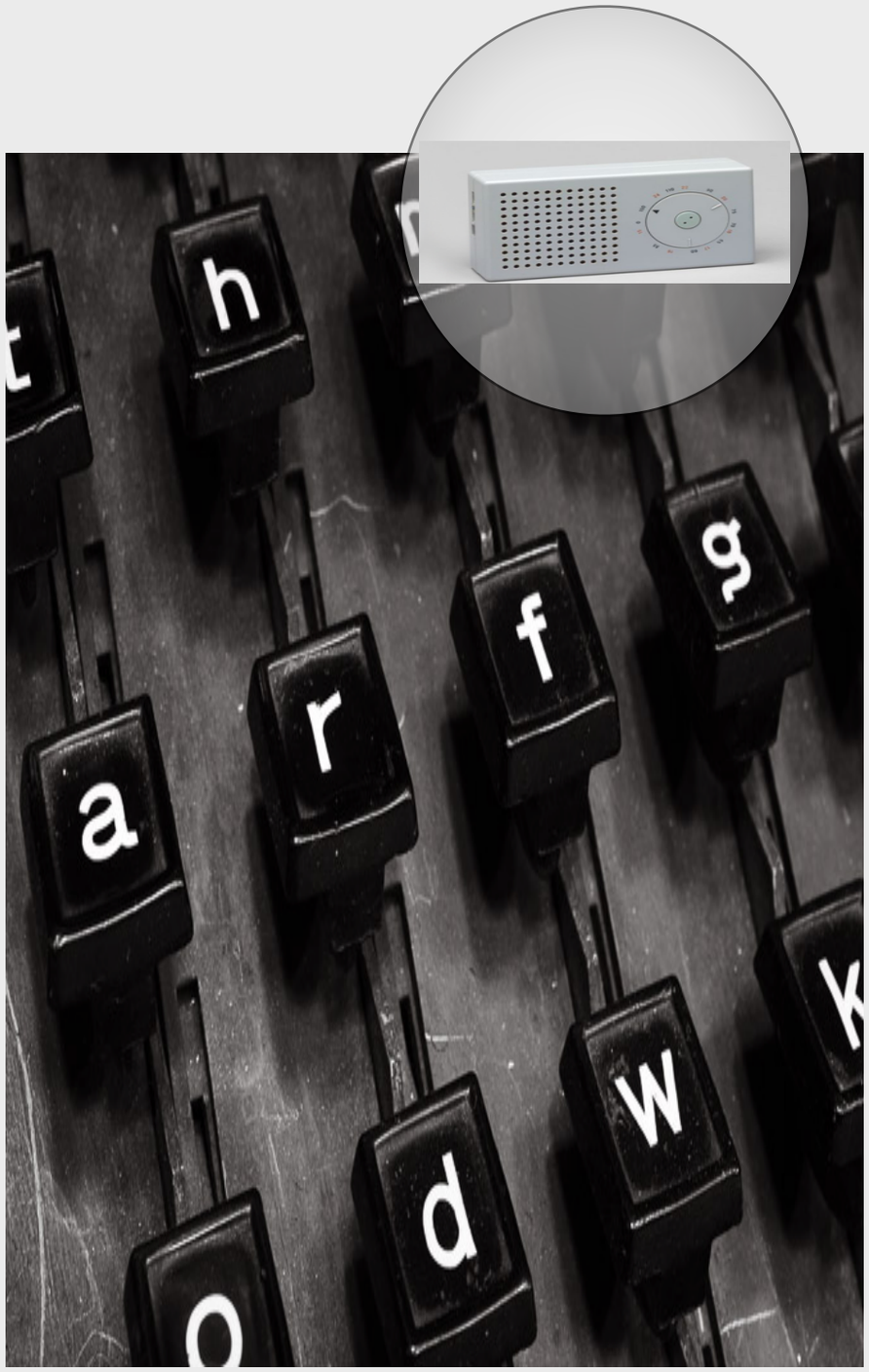
idiom catalog (so far)

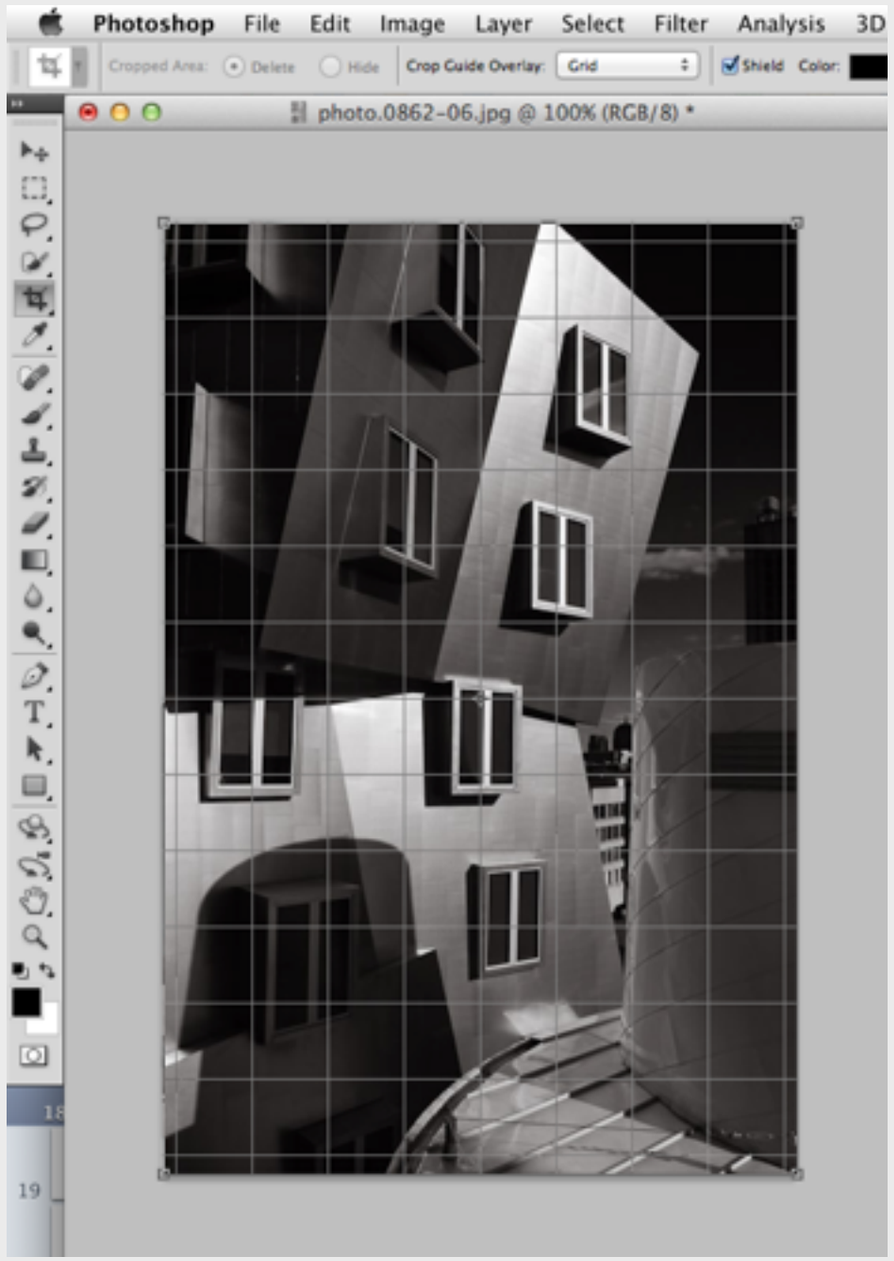
instantiate	*organize*	*relate*	*resource*	*save*	*communicate*	*personalize*
stylesheet	selection	friend	access token	history	message	account
master	folder	clique	notification	buffer	posting	karma
stencil	group	invitation	reservation	cursor		OOBA
style buffer	label		REST	sync		rating
	layer		cart	export		status
	stack		subscription			
	alias		purchase order			
	preset		RMA			
	cursor		coupon			
	filter		catalog			
	property					
	metadata					

exercise: criteria

cropping in photoshop

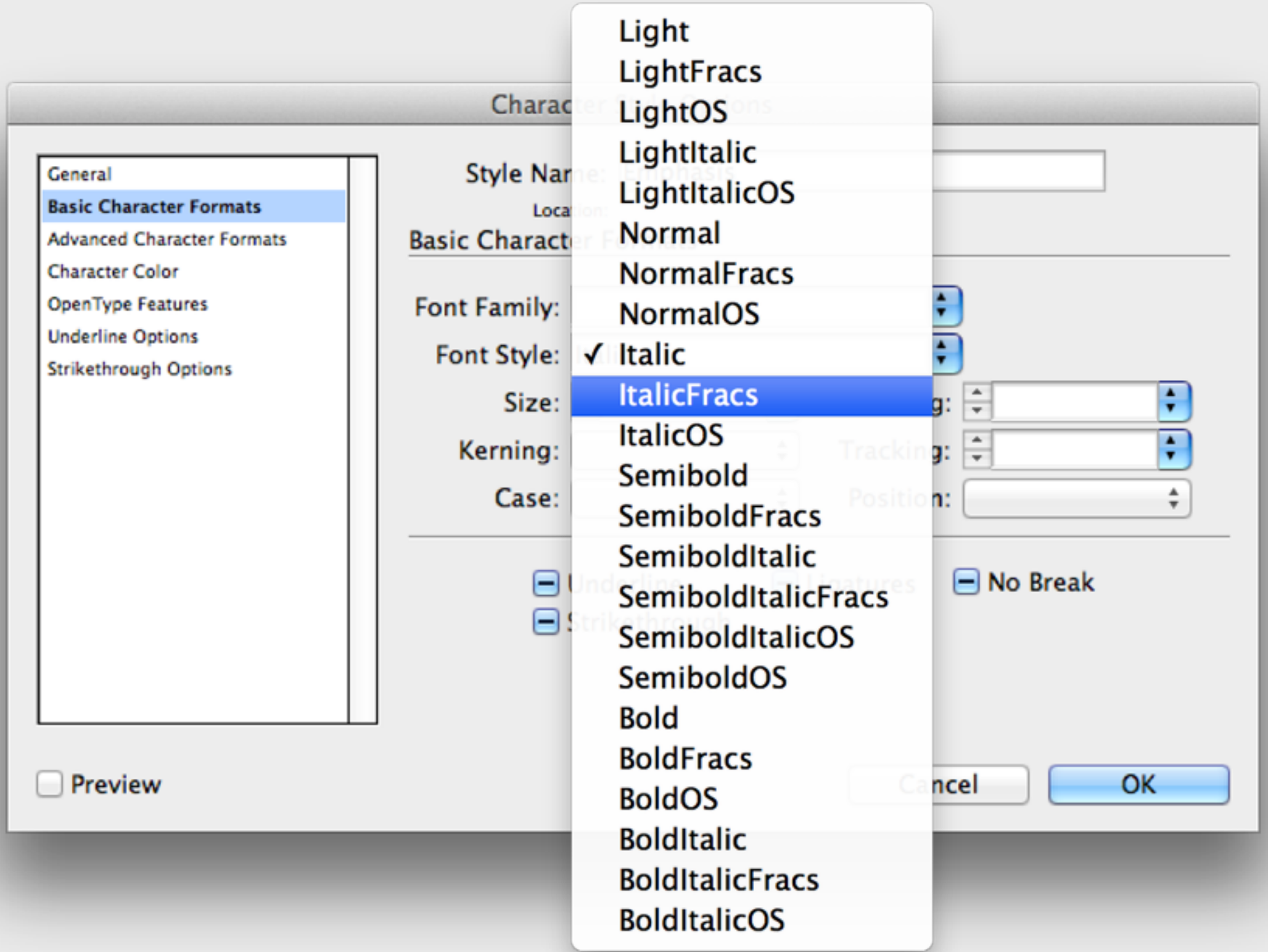
cropping in photoshop (part 2)





what criterion is violated here?

what's the misfit story?



applying to your project

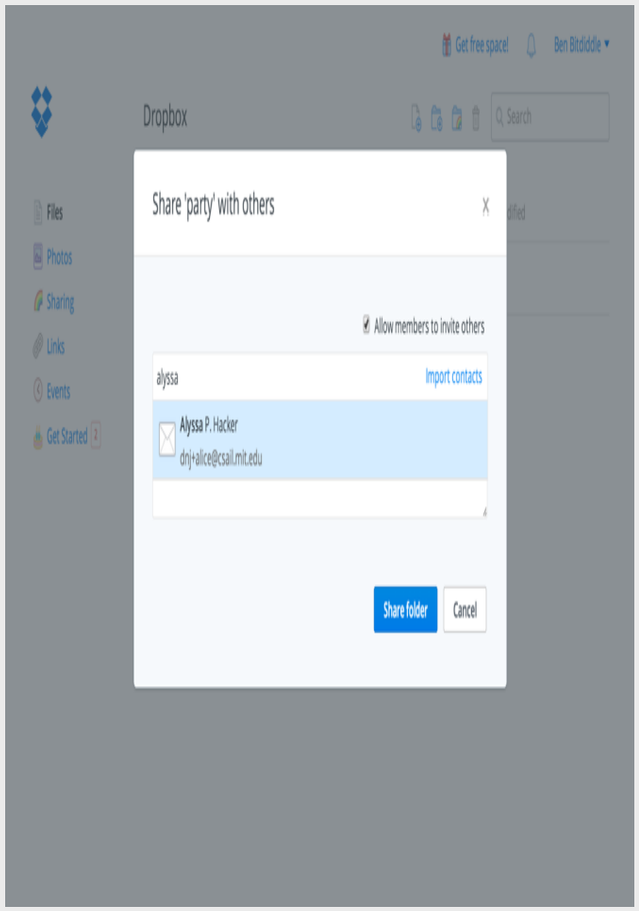
purpose
what is the purpose of your app?
express it as an advertising slogan:
“track your favorite blogs!”

concepts
what are they?
give them good names
which are conventional? novel?
which are essential?

stories
what’s the purpose of each concept?
a (tweet-length) story to explain how?

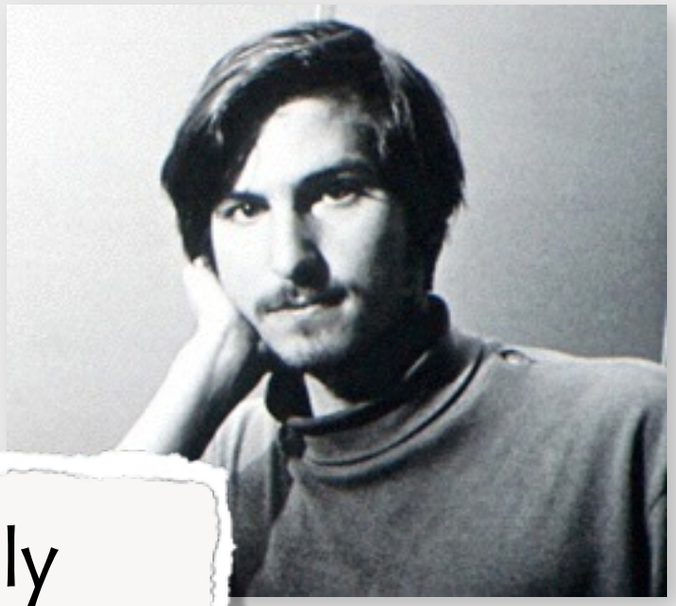
background

mitch kapor on designers



Architects work in the medium of buildings, graphic designers work in paper and other print media, industrial designers on mass-produced manufactured goods, and software designers on software. **The software designer should be the person with overall responsibility for the conception and realization of the program.**

A Software Design Manifesto, 1996



To design something really well, you have to **get it**. You have to really grok what it's all **about**. It takes a passionate commitment to really thoroughly **understand** something, chew it up, not just quickly swallow it. Most people don't take the time to do that.

a metaphor

software design:
smoothing rough & sharp edges
big picture by attention to small details

"That's quite obsessive, isn't it?"
Jonathan Ive

https://www.youtube.com/watch?v=zaLMOSWAwdw