

software studio

concept idioms

Daniel Jackson

what are idioms?

idioms

- › well-established concepts
- › used in many apps and contexts

reuse by designers

- › usually polished and well understood
- › risks and limitations known

familiar to users

- › helps ease of use

“near misses”

- › beware of almost using an idiom

format

problem

- › *a design problem to be solved*

solution

- › *one or more concepts*

examples

- › *in well known apps*

refinements

- › *more elaborate forms often used*

object model

- › *concepts & relationships*

idiom: multi-selection

problem

- › how to apply actions to sets of items

solution

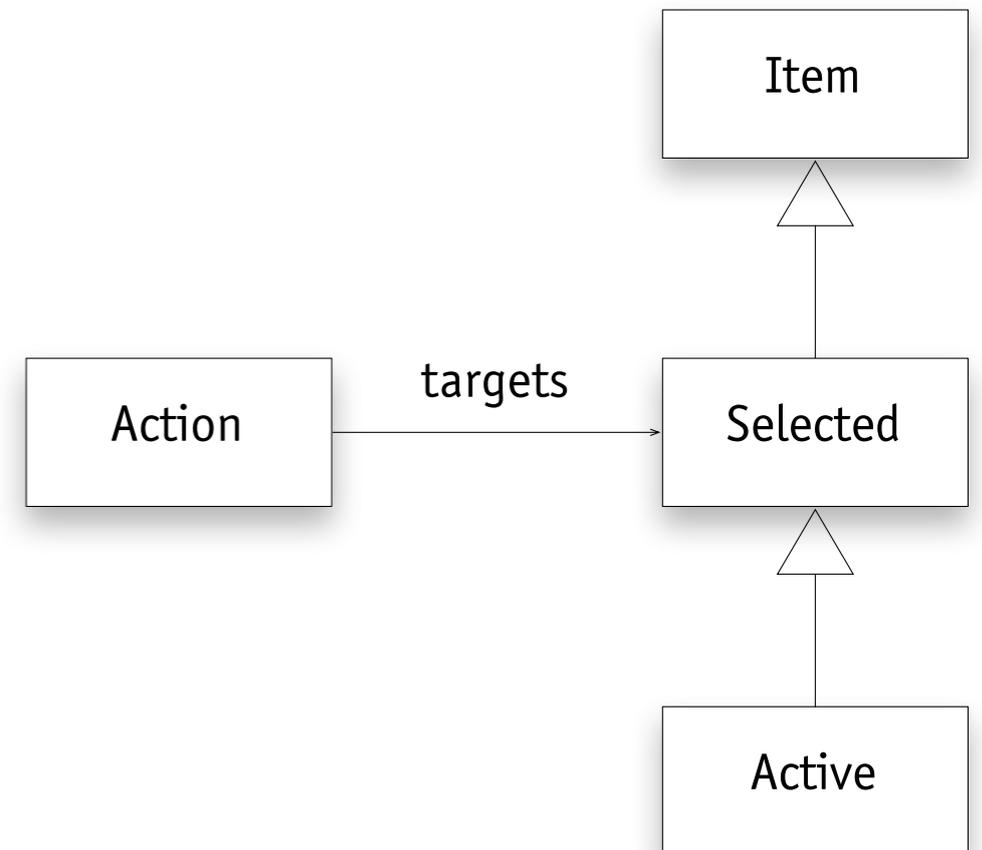
- › selection: a subset of items chosen
- › action: applied to all objects in selection

examples

- › characters in word processors
- › graphical objects in drawing apps
- › photos in cataloging apps

refinements

- › group/ungroup: form new composite object
- › active item: one element in selection plays special role
- › invert (and other set-theoretic operations)
- › partial selection (eg, of pixels in Photoshop)



case study: Lightroom selection



© Adobe Systems. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <http://ocw.mit.edu/fairuse>.

active element is highlighted
maintains selection when different folder selected
if selected images appear in that folder

idiom: cut-buffer

problem

- › how to move items between contexts

solution

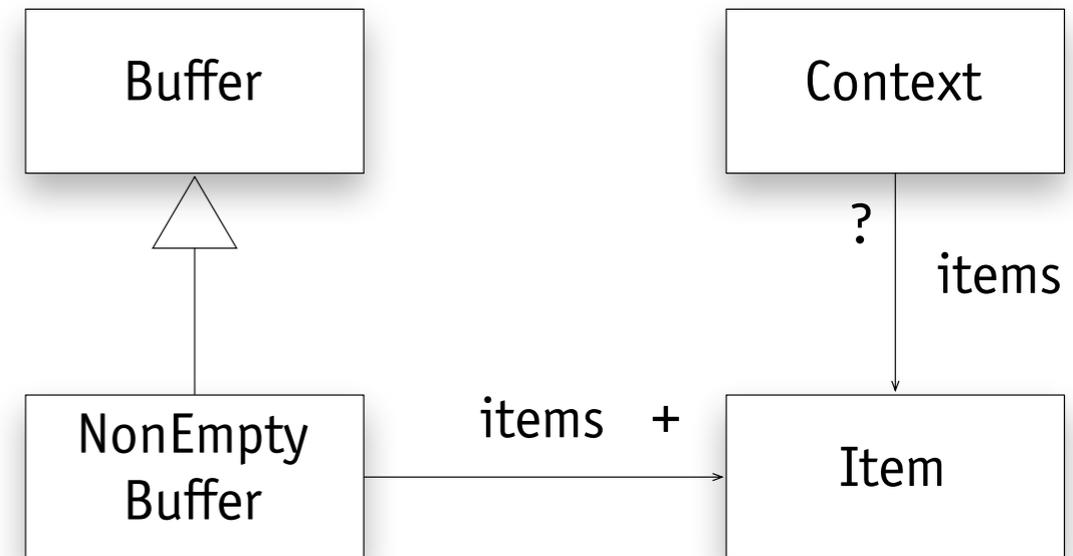
- › buffer: stores item to be transferred

examples

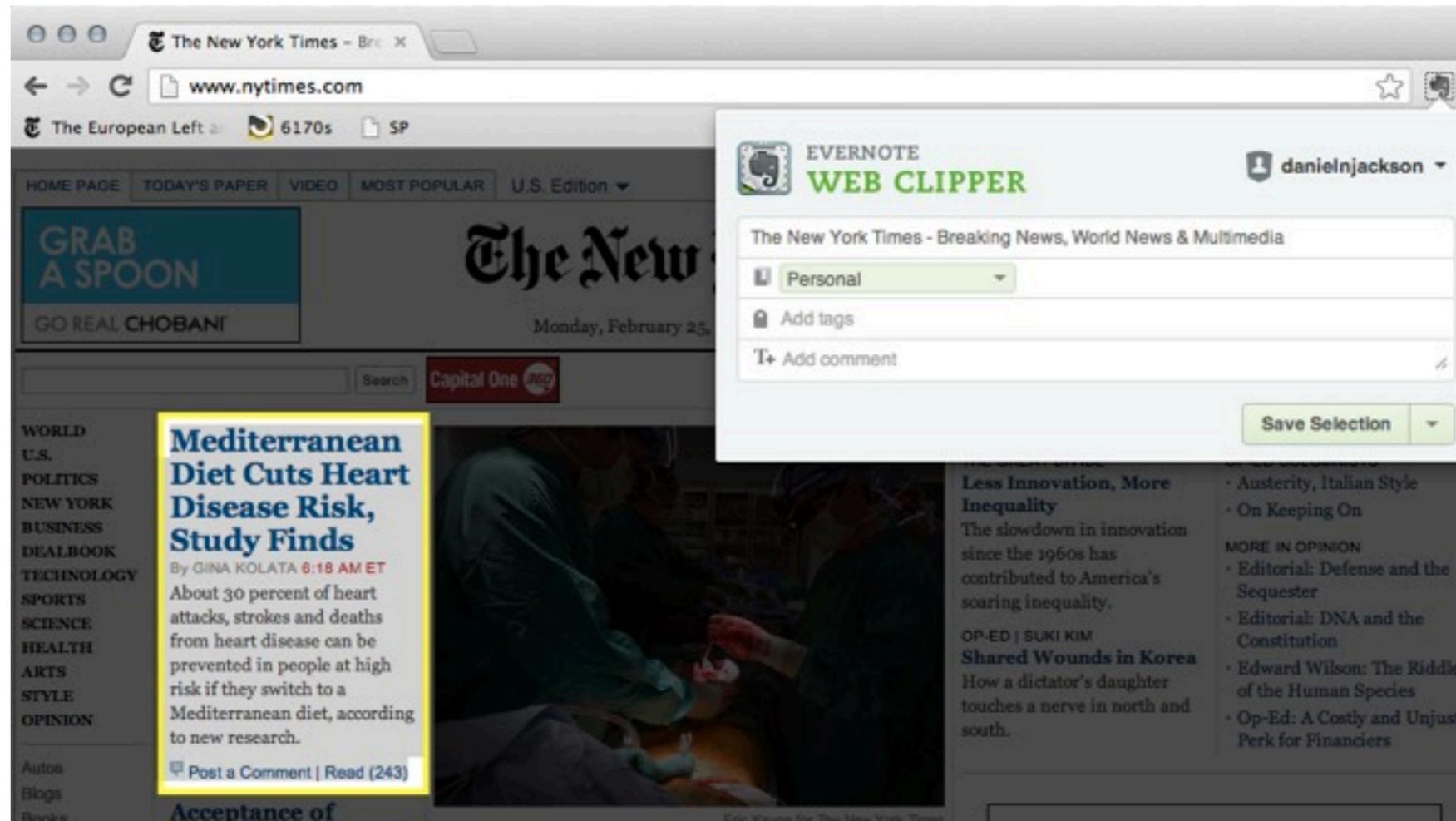
- › text editor cut-and-paste
- › moving files in Windows

refinements

- › set: buffer can contain multiple items
- › object embedding: item is shared, not copied
- › multiple buffers



case study: evernote web clipper



© Evernote. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <http://ocw.mit.edu/fairuse>.

copies to server
have to sync to see in local app

idiom: stylesheet

problem

- › maintain consistent styling
- › make global changes easy

solution

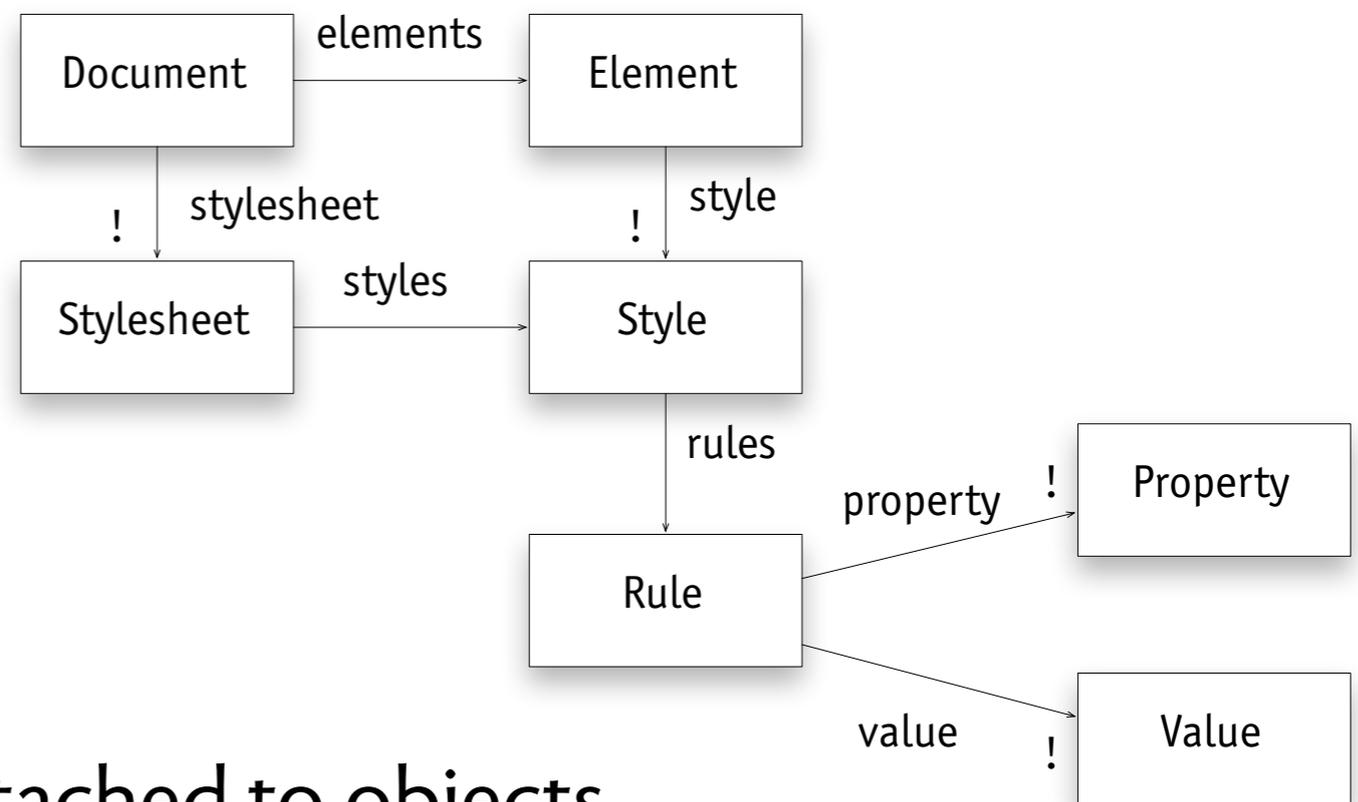
- › style: holds formatting rules, attached to objects

examples

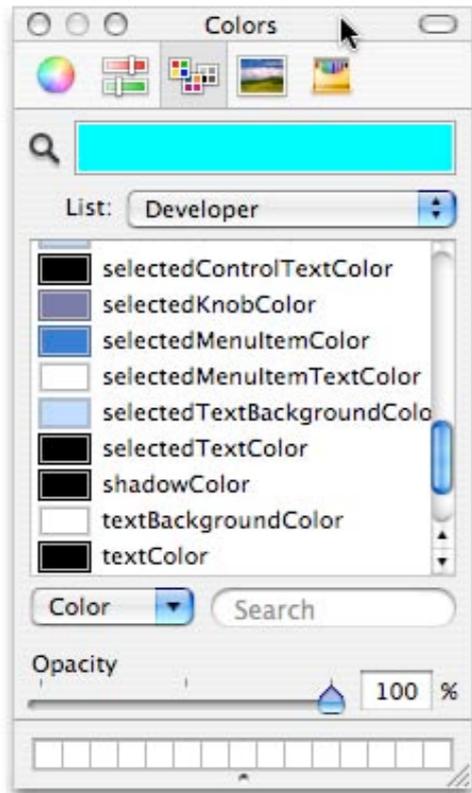
- › paragraph and character styles in Word, Indesign, Pages
- › themes in Powerpoint
- › cascading style sheets for HTML

refinements

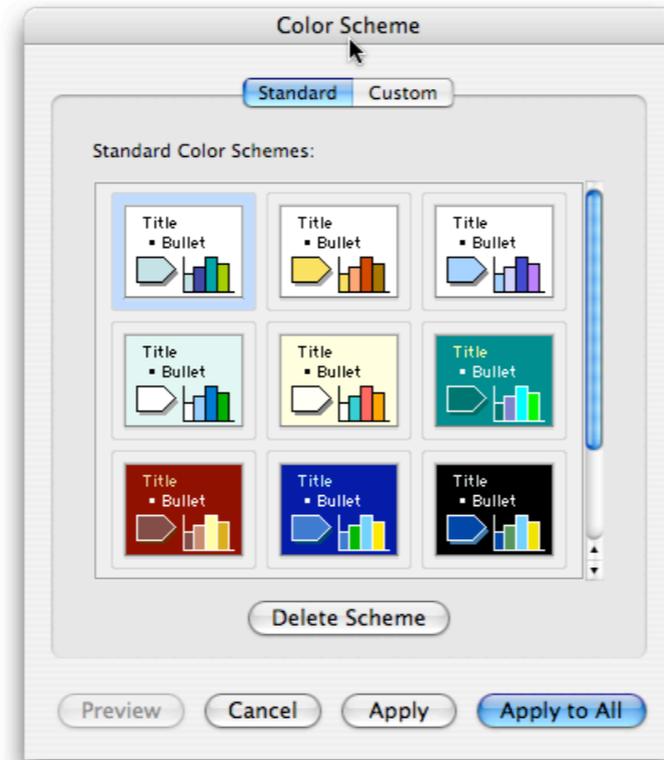
- › stylesheets: reuse and separation of sets of styles
- › selectors: match objects implicitly
- › inheritance: of properties from style to style
- › overlapping styles: eg, character and paragraph styles



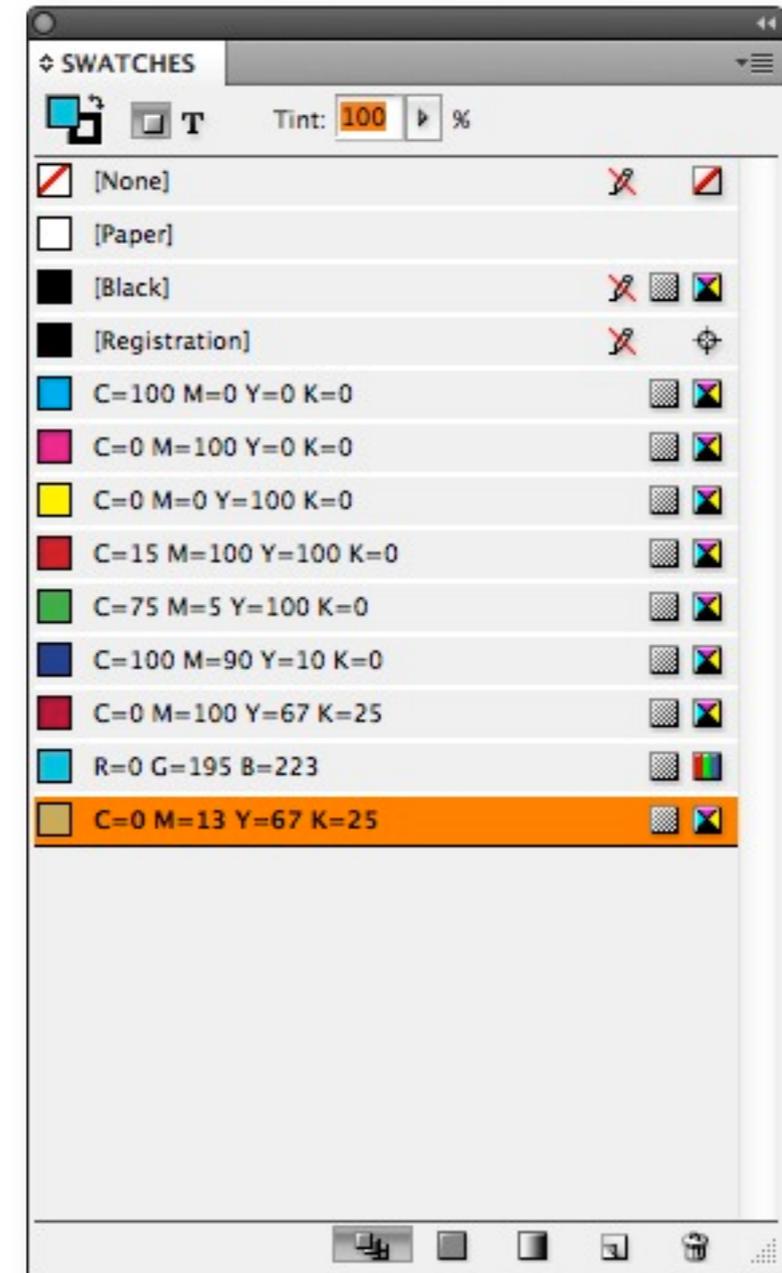
case study: color palettes



OS X palette
not real styles



Powerpoint themes

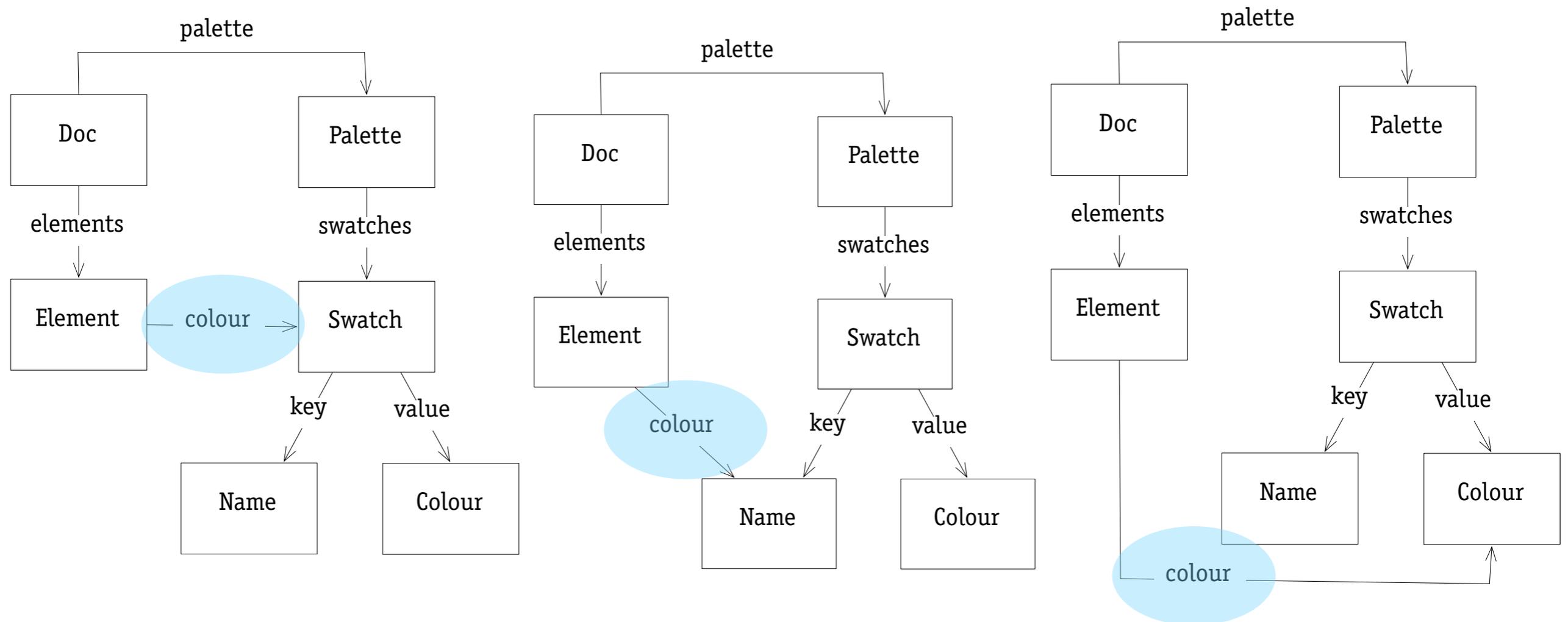


Indesign swatches

palette object models

three subtly different approaches

- › what happens when palette is modified?



**“Every problem in computer science can be solved by introducing another level of indirection”
-- David Wheeler**

idiom: directory

problem

- › organize many items into navigable space

solution

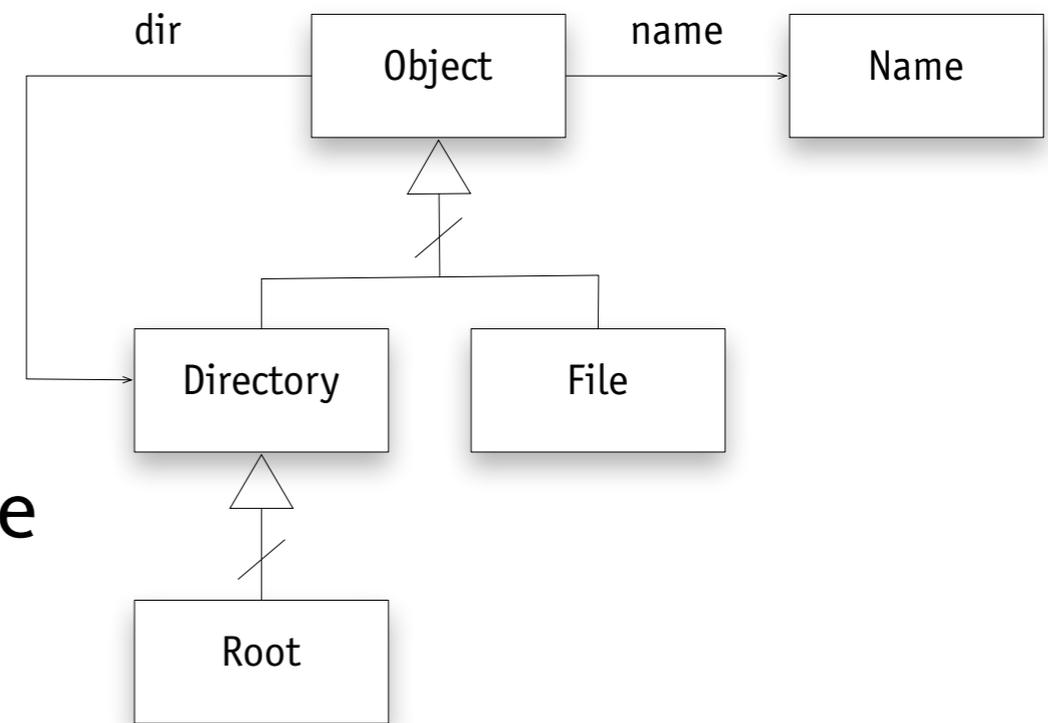
- › tree of directories, each its own namespace

examples

- › mailboxes in email clients
- › folders in OS X, directories in Unix
- › domains in DNS
- › package structuring in Java

refinements

- › links: allows arbitrary cross-connections
- › encapsulation (eg, OS X packages)



case study: IMAP mailboxes

some issues

- › many email clients allow '/' in mailbox names: trouble
- › in some servers, mailboxes cannot contain messages & mailboxes

idiom: save-as

problem

- › allow updating of multiple files on disk

solution

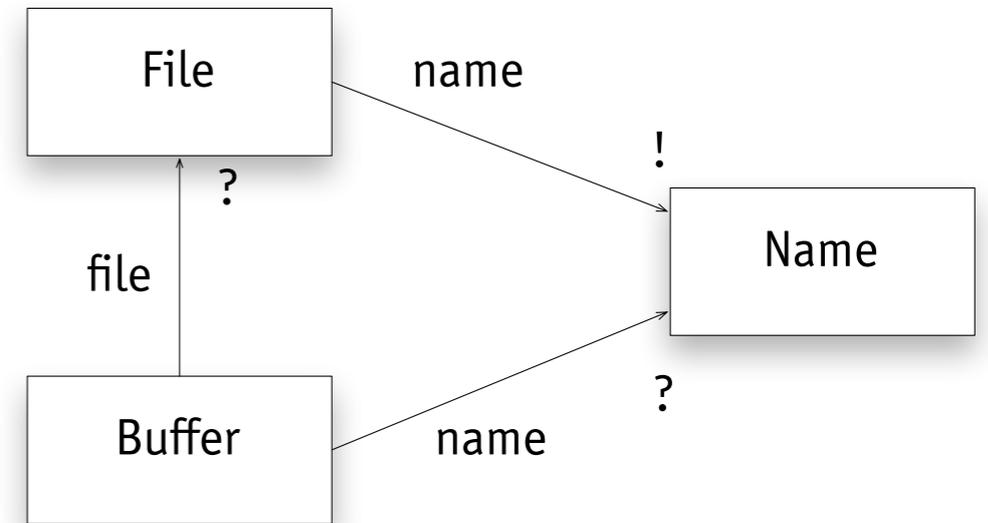
- › buffer, with explicit commands to load from and write to disk

examples

- › almost all desktop applications

refinements

- › export: separate out saving different types, especially if lossy
- › propagation of renaming (from file to buffer)
- › versions: track all previous versions

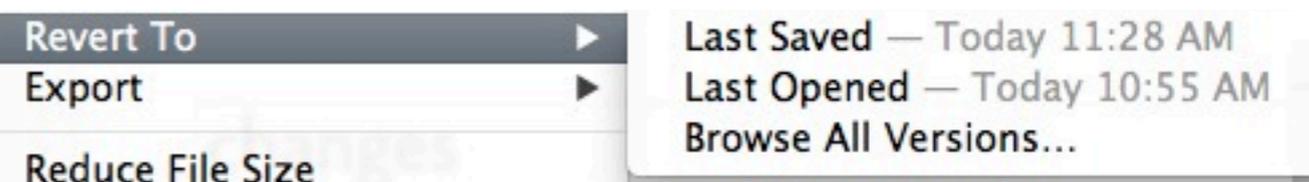
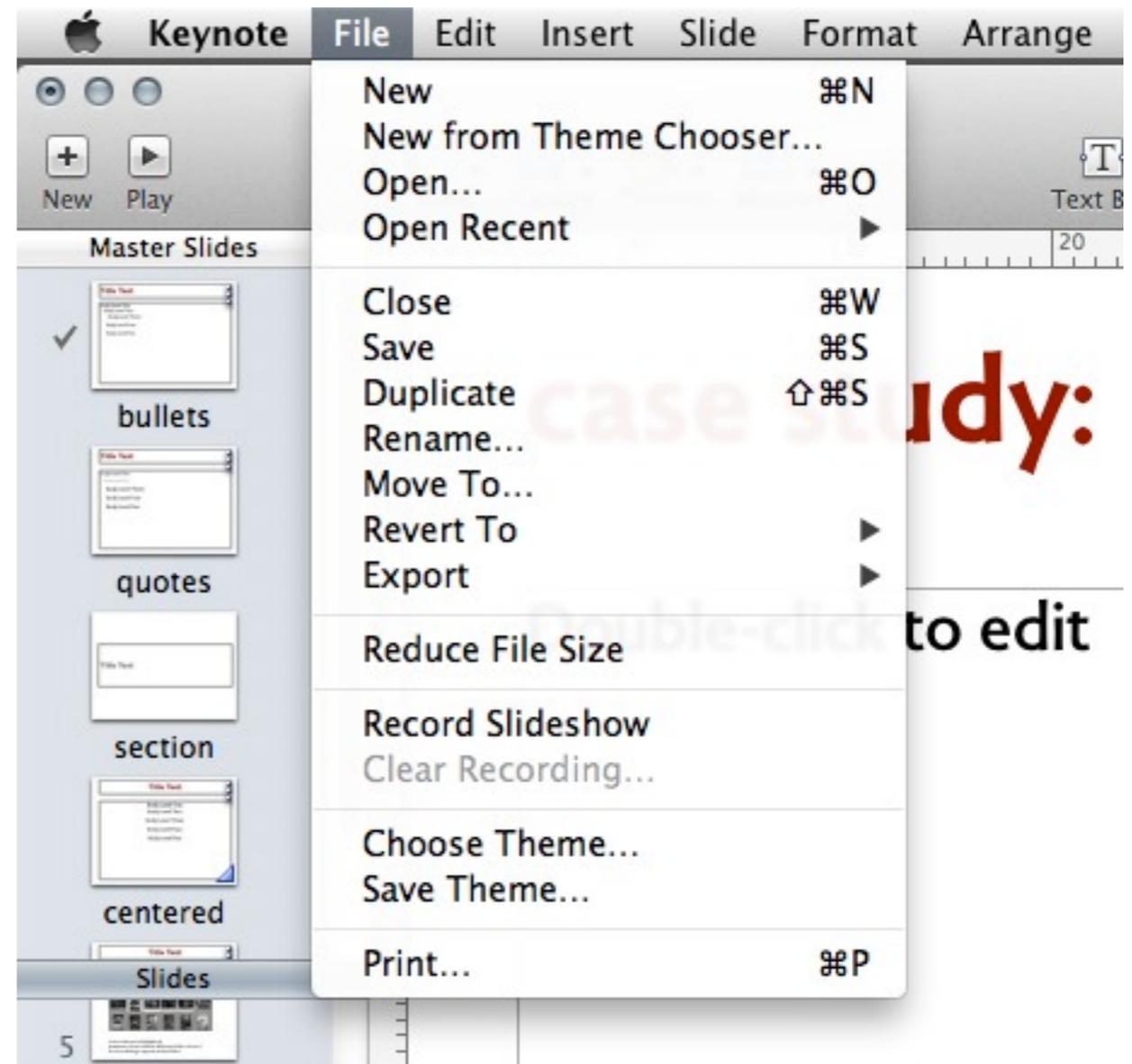


case study: OS X Lion rethinking

changes

- › save as replaced by duplicate
- › rename, move to added
- › revert to any version

why did they do this?



MIT OpenCourseWare
<http://ocw.mit.edu>

6.170 Software Studio
Spring 2013

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.