CS 200

Lecture 02

Structured Word Processing — Styles
Assigned reading

“The Non–Designer’s Design Book,” by Robin Williams (first half)


Please read and highlight BEFORE this week’s lab

Assignment 2 (due Tuesday September 27 at 9:00 am)

Notes for this lecture

Today and Thursday

[named] styles
the Poke pearl
backups (the backups assignment—A00— is now online)
The Poke Pearl

What happens if you
• double-click here?
• triple-click here?
• option-click here?
• control-click here?
• command-click here?
• etc...

What happens if you
• click and drag on this?
• option-click and drag on this?
• control-click and drag on this?
• command-click and drag on this?
• etc...
Assumptions

You have used a word processor before

You understand and use correctly the following WP tools

- Word Wrap (hard carriage return vs. soft carriage return)
- Cut/Copy and Paste
- Font usage (size, style and typeface)
- Rulers and Margins
- Proper use of tabs
- Headers and footers (page numbering)
- Footnotes
Things to Think About

- What are the data objects in a Word Processor?
- Is there more than one way to do any given task?
- What are deficiencies of the interface?
- What are efficiencies of the interface?
[Named] Styles

The idea

name a collection of paragraph or character attributes
you then apply the NAME to text instead of individually setting attributes

Hence changing a document’s appearance is

easier & faster, more accurate (⇒consistency, if used intelligently)

— you just change the definition of the name; everywhere it’s used, text changes

Does appearance matter?

legibility
understandability
professionalism

Terminology

we will refer to Helvetica, Palatino, bold, italic, underlined, red, 15 pt leading, .... as attributes

a style is a [specific] “bundle of attributes”

eg “24 pt Myriad Pro Light with 12 pts of Before Paragraph leading & 2 cm of left indent”
Compare this and the preceding slide

The idea name a collection of paragraph or character attributes you then apply the NAME to text instead of individually setting attributes Hence changing a document’s appearance is easier & faster more accurate (⇒ consistency) Does appearance matter? legibility understandability professionalism Terminology we will refer to Helvetica, Palatino, bold, italic, underlined, red, 15 pt leading, .... as attributes a style is a [specific] “bundle of attributes” eg “24 pt Myriad Pro Light with 12 pts of Before Paragraph leading & 2 cm of left indent” but sometimes “named style” for emphasis (and/or clarity)
Indirection

The underlying concept is indirection

• referring to something through something else

Indirection is useful in many contexts, including...

• drawing programs (object properties)
• HTML (tags in web pages; CSS)
• electronic mail (aliases, forwards)
• spreadsheets (named cells/ranges)

Remember the Model pearl?
### Demo 1

**The text being formatted**

**The style definitions (built-in & user-defined)**

<table>
<thead>
<tr>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block_Centered</td>
</tr>
<tr>
<td>Font: <em>(Default)</em> Times, 12 pt, English (US), Centered, Line spacing: single, Widow/Orphan control, No bullets or numbering</td>
</tr>
<tr>
<td>Default Paragraph Font</td>
</tr>
<tr>
<td><em>The font of the underlying paragraph style +</em></td>
</tr>
<tr>
<td>Emphasis_Slight</td>
</tr>
<tr>
<td>Default Paragraph Font + Font:Italic</td>
</tr>
<tr>
<td>Figure_Caption</td>
</tr>
<tr>
<td>Footer</td>
</tr>
<tr>
<td>Font: <em>(Default)</em> Times, 10 pt, English (US), Left, Line spacing: single, Widow/Orphan control, Tabs: 3.25&quot;, Centered + 6.5&quot;, Right</td>
</tr>
<tr>
<td>Footnote Reference</td>
</tr>
<tr>
<td>Default Paragraph Font + Superscript</td>
</tr>
<tr>
<td>Footnote Text</td>
</tr>
<tr>
<td>Header</td>
</tr>
<tr>
<td>Font: <em>(Default)</em> Times, 10 pt, English (US), Left, Line spacing: single, Widow/Orphan control, Tabs: 3.25&quot;, Centered + 6.5&quot;, Right</td>
</tr>
<tr>
<td>Heading 1</td>
</tr>
<tr>
<td>Heading 2</td>
</tr>
<tr>
<td>Instruction</td>
</tr>
</tbody>
</table>
Showing Style Usage in MS Word

In Draft view, the “style area width” controls the width of the style name column in the doc display; if it’s zero, the style name column vanishes.

In Draft view, View > Reveal Formatting to enable the (yellow) Formatting popup.

The text being formatted:

GEM initially displays two windows.

- The “cpu window” shows the contents of the cpu registers. On Unix and W machines, it contains a menu bar from which the emulator can be controlled.
- The “memory window” shows a scrollable display of the contents of memory.

File Menu (ROM Menu on the Macintosh)

- Load
  Use this command to load a relocatable object module into memory. (Such modules are created by the assembler gal.exe when it translates an assembly program language).
- Reload
  Use this command to reload into memory the contents of the last previous object module loaded. This provides a convenient, minimum-effort way to start a program.
- Quit
  Terminate execution of the emulator.

Execute Menu

Step
Execute a single instruction and stop. Clicking on the cpu window’s Step button is equivalent to selecting this menu item.

When you execute an instruction, certain mnemonic labels are displayed in memory windows to help you keep track of what’s happening:

- The letters P and N label the “previous” and “next” instructions. The previous instruction is the one that is, the one you executed just before you clicked on the cpu window’s Run button.
- The letter that is the next instruction to be executed.
- The letter that is the next instruction to be executed through the breakpoint.
- The letter that is the next instruction to be executed through the breakpoint.

Paragraph Formatting

Direct:

Font Formatting

Character Style: Direct:

Paragraph Style: Direct:

Paragraph Style: Font {Point Size: 12 pt, Do not check spelling or grammar, English (US), Direct:

Run

defered the instructions continuously, one right after the other. Clicking on the cpu window’s Run button is equivalent to selecting this menu item.
Hierarchical Styles

The idea

• style B is “everything style A is, except for . . .”
• style C is “everything style B is, except for . . .”
• etc.

- Style A
  - Helvetica, 14 pt, Bold

- Style B
  - Helvetica, 12 pt, Bold

- Style D
  - Times, 12 pt, Bold

- Style C
  - Helvetica, 14 pt, Italic

- Style E
  - Helvetica, 16 pt, Italic
Hierarchical Styles cont’d

When you change an attribute of A’s definition

- styles *based* on A change, too, unless that attribute has been explicitly set for the derived style

```
Style A
Palatino, 14 pt, Bold

Style B
Palatino, 12 pt, Bold

Style D
Times, 12 pt, Bold

Style C
Palatino, 14 pt, Italics

Style E
Palatino, 16 pt, Italics
```
Hierarchical Styles Relativity

There are choices in how to implement some aspects of this; eg, if I change B’s font size, is the new value

- absolute (MS Word), or relative to the size of A?
- does changing A’s size later automatically change B’s?
- by a fixed number of points, or by a percentage?

Ditto with paragraph indents

Etc.

Style A
Palatino, 18 pt, Bold

Style B
Palatino, ?? pt, Bold

Style D
Times, ?? pt, Bold

Style C
Palatino, ?? pt, Italics

Style E
Palatino, ?? pt, Italics
Hierarchical Styles — Comments

Comments
- this is a MODEL of named styles, though with parameters
- there’s no point to this unless parent and child styles
  * share some attributes
  * but not others
- you might have a forest instead of a tree
  (our previous example had no hierarchy at all — a forest of flat trees, so to speak)

In the “Files for Styles Lectures” under Week 2: Styles on Learn, compare
- User Manual Extract (Tree)
- User Manual Extract (Forest)
Block_Centered is a poorly-chosen name because it refers to the appearance of text tagged with this named style.

The Paragraph Styles Tree for User Manual Extract
A Paragraph Style Sheet for UME.doc Using Multiple Trees

**Built-in styles**

*The names are built-in, however the*

**Default Paragraph Font**
- The font of the underlying paragraph style +

**Footer**
- Normal + Font: 10 pt, Right: 0.25", Tabs: 3.25", Centered + 6.5", Right

**Footer Reference**
- Default Paragraph Font + Superscript

**Footer Text**
- Normal + Font: 10 pt, Justified, Line spacing: exactly 10 pt, Space Before: 4 pt

**Header**
- Normal + Tabs: 3", Centered + 6", Right

**Heading 1**
- Style for Next Paragraph: Block
  - Heading_Abstract + Font: Bold, Centered, Space Before: 14 pt, Keep with next, Level 1

**Heading 2**
- Style for Next Paragraph: Block
  - Heading_Abstract + Font: Bold, Space Before: 7 pt, Keep with next, Level 2

**No List**
- No List +

**Normal**
- Font: (Default) Times New Roman, 12 pt, English (US), Left, Line spacing: single, Space Before: 3 pt, Widow/Orphan control

**Page Number**
- Default Paragraph Font +

**Table Normal**
- Font: (Default) Times New Roman, 10 pt, Left, Line spacing: single, Widow/Orphan control

**User-defined styles**

**Block**
- Body_Abstract + Justified, Space Before: 6 pt

**Block_Centered**
- Block + Centered, Numbered

**Body_Abstract**
- Normal +

**Emphasis_Slight**
- Default Paragraph Font + Font: Italic

**Figure_Caption**

**Heading_Abstract**
- Normal +

**Instruction**
- Style for Next Paragraph: Instruction_Expl
  - Body_Abstract + Font: Courier, 11 pt, Indent: Left: 0.2", Space Before: 6 pt, Keep with next

**Instruction_Expl**
- Body_Abstract + Indent: Left: 0.44", Numbered

**Instruction_Item**
- Style for Next Paragraph: Instruction_Expl
  - Body_Abstract + Font: 10 pt, Indent: Left: 0.63", Hanging: 0.19", Line spacing: exactly 12 pt

**Item**
- Body_Abstract + Indent: Left: 0.19", Hanging: 0.19", Space Before: 6 pt

**Menuitem**

**Menuitem_Expl**
- Body_Abstract + Indent: Left: 0.44"

**Operand**
- Body_Abstract + Indent: Left: 0.25"

**Operand_Expl**
- Body_Abstract + Indent: Left: 0.56", Space Before: 6 pt

**Demo 2!**
Character Styles

Should character attributes be part of a paragraph style definition?

Are character styles hierarchical?

- yes in MS Word
- no in FrameMaker
- yes in Nisus Writer Express & Nisus Writer Pro
- yes in Adobe InDesign

Are character styles used to specify the (default) character attributes of paragraph styles?

- no in MS Word
- optionally in Nisus Writer Express & Pro

Incidentally, if you own a Mac, Nisus Writer Pro is a very nice $39 US (ed) word processor (www.nisus.com).
Naming Named Styles

Style names should reflect function, not appearance

“Quotation” not “Indented Paragraph”
“Emphasis” not “Bold”
“List Item” not “Bulleted Paragraph”

Style names

should be chosen logically

— NOT based on their appearance

so if you change the appearance, you don’t have to change the name to avoid confusion
Tables-of-Contents (TOCs)

What’s our model of a TOC? (example on the next slide)

Steps to create a table-of-contents

• identify paragraph styles from which to build the TOC
  eg Heading 1, Heading 2, Heading 3, etc, in Word
• specify a TOC style for each TOC level
  TOC 1, TOC 2, TOC 3, etc, in Word
• build the TOC (most word processors do this for you)
  copy paragraphs with specified paragraph style tags into the TOC
  apply the corresponding TOC style to each
  *Heading 1 to TOC 1, Heading 2 to TOC 2, Heading 3 to TOC 3, etc.*
  append a tab character & page number to each
• modify the TOC styles to get the desired layout
• rebuild the TOC whenever appropriate

Note the application of two distinct styles to each piece of text
Menu Commands in the Machine Emulator (GEM)

GEM initially displays two windows.

- The “cpu window” shows the contents of the cpu registers. On Unix and Windows machines, it contains a menu bar from which the emulator can be controlled. (On Macintoshes, these menus are appended to the menu bar at the top of the screen.)
- the “memory window” shows a scrollable display of the contents of memory.

File Menu (ROM Menu on the Macintosh)

Load
Use this command to load a relocatable object module into memory. (Such “rom files” are created by the assembler gal.exe when it translates an assembly program into machine language).

Reload
Use this command to reload the contents of the last previous relocatable object module loaded. This provides a convenient, minimum-effort way to re-execute a program.

Quit
Terminate execution of the emulator.

Execute Menu

Step
Execute a single instruction and stop. Clicking on the cpu window’s Step button is equivalent to selecting this menu item.

When you execute an instruction, certain mnemonic labels are displayed in the cpu and memory windows to help you keep track of what’s happening:

- The letters P and N label the “previous” and “next” instructions. The previous instruction is the instruction you have just executed. The next instruction is the instruction that will be executed next, that is, the instruction whose address is in the program counter. These labels are particular useful when you have just executed a jump instruction.
- The letter S labels any register or memory word that supplied a source operand for the instruction just executed.
- The letter D labels the register or memory word, if any, into which a result value was stored by the instruction just executed.
- The letter I will label a register if the instruction just executed referenced an operand indirectly through it.

Run
Execute program instructions continuously, one right after the other. Clicking on the cpu window’s Run button is equivalent to selecting this menu item.

Table of Contents

Table of Contents

Menu Commands in the Machine Emulator (GEM) ........................................... 2
File Menu (ROM Menu on the Macintosh) ..................................................... 2
Execute Menu ............................................................................................... 2
Display Menu ............................................................................................... 3
GEM’s Architecture ....................................................................................... 4
Program Counter ......................................................................................... 4
Instruction Register .................................................................................... 4
Memory ........................................................................................................ 4
CPU Registers ............................................................................................. 4
Condition Code Register ........................................................................... 4

The Instruction Set and Assembly Syntax ................................................... 5
Data Movement ............................................................................................. 6
Arithmetic / Logical Instructions .................................................................. 7
Flow-of-Control ............................................................................................ 9
Pseudo-Instructions .................................................................................... 10

The Trap Instruction ...................................................................................... 12
Trap #1 — Reading an Integer ................................................................. 12
Trap #2 — Write an integer to the listing and to the display, base 10 ........ 12
Trap #3 — Write an integer to the listing, base 10 .................................. 12
Trap #4 — Write an integer to the listing and to the display, in some radix . 13
Trap #5 — Write an integer to the listing, in some radix ............................ 13
Trap #6 — Read a string ........................................................................... 13
Trap #7 — Write a string to the display and to the listing ....................... 13
Trap #8 — Write a string to the display and to the listing ....................... 14
Comments ................................................................................................. 14
Conditional Text

Display or not display
Print or not print

• in MS Word these are separate attributes
• is that a good idea?

Look up “Hidden Text” in Word’s help index

Word only implements one class of conditional text

• but it makes sense to have more
• you can simulate having more than one class of conditional text using named styles
The Format > Style… Dialog in Word

Format → Style...

Interaction techniques

A preview checkbox would be better. & everywhere—not just here.
The Format > Style... > Modify Sub-Dialog

Recall the Model Pearl (interaction)

- Nested dialog boxes
  + radio buttons, check boxes, etc.
- Which is better? Should you have both?
The Format > Style... > Organizer Sub-Dialog

Body_Abstract + Justified, Space before 6 pt
Not all paragraph attributes are available in this dialog
e.g. bullets, borders...
although they belong here

when/why “Exactly?”
Widows and Orphans

**Widow**

stranded line at top of page

The seven-part Harry Potter series of fantasy novels was written by English author J. K. Rowling about an adolescent boy wizard named Harry Potter and his best friends Ron Weasley and Hermione Granger.

**Orphan**

stranded line at bottom of page

The seven-part Harry Potter series of fantasy novels was written by English author J. K. Rowling about an adolescent boy wizard named Harry Potter and his best friends Ron Weasley and Hermione Granger.

The story is mostly set at Hogwarts School of Witchcraft and Wizardry, a school for young wizards and witches, and focuses on Harry Potter's fight against the evil wizard Lord Voldemort, who killed Harry's parents as part of his plan to take over the wizarding world.
Word’s Format > Font… (ie Character) Dialog Box

![Font Dialog Box]

Font:
- Times
- Techno
- Textile
- Times
- Times CE
- Times New Roman

Font style:
- Bold
- Regular
- Italic
- Bold Italic

Size:
- 10
- 11
- 12
- 14
- 16
- 18

Font color:
- Automatic
- (none)
- Automatic

Underline style:
- Automatic
- (none)

Underline color:
- Automatic

Effects:
- Strikethrough
- Shadow
- Small caps
- Double strikethrough
- Outline
- All caps
- Superscript
- Emboss
- Hidden
- Subscript
- Engrave

Preview:
- CS 200

Scale: 100%

Spacing: Normal

Position: Normal

Kerning for fonts: On

Points and above

Preview:
- CS 200
Style Templates

Store style definitions in a separate “template” or “style sheet” file

Use templates to keep styles consistent across multiple documents

• each such document is linked to the template
• when the template is changed, the appearance of every linked document
  * changes automatically? (optional in MS Word)
  * or do you have to request an update? (Keynote)

Another data model...

Tools → Templates and Add-ins…

![Templates and Add-ins dialog box](image)
Style Templates in MS Word

Every document is based on a template (“Normal” by default)

- apparently in the “~ / Documents / Microsoft User Data / “ folder...
  (instead of ~ / Library / Application Support / Microsoft ... sigh)
- whose styles are copied to the document

To use a different base template

- use the “Attach” button in the Tools → Templates and Addins... dialog box

To have Word reload the template’s styles every time you open the document

- check “Automatically Update Document Styles” in the Tools → Templates and Addins... dialog box

To cause a style change applied in a document to update its template definition

- Use the “Add to template” button in the Modify Styles dialog box, or change it directly in the template
“Invisible” (aka “non-printing”) characters

See Word’s Preferences - View dialog

Books by Dick Francis
- The Sport of Queens (1957)
- Dead Cert (1962)
- Nerve (1964)
- For Kicks (1965)
- Odds Against (1965)
- Flying Finish (1966)
- Blood Sport (1967)
- Forfeit (1968)
- Enquiry (1969)
- Rat Race (1970)
- Bonecrack (1971)
- Smokescreen (1972)
- Slayride (1973)

...
Things That May Confuse You in MS Word

Character attributes in paragraph styles

• are a matter of convenience
• when done, one set of char attributes is a property of the paragraph AS A WHOLE

The Style Column

• is visible only in Normal View

Some paragraph attributes

• are in “auxiliary” dialog boxes instead of the paragraph dialog box even though they ARE paragraph attributes
• e.g. borders, bullets, numbering, shading

Paragraph attributes

• are “stored in” the ¶ at the end of each paragraph
• are automatically copied to a new paragraph if you press RETURN
• you can copy/paste this character to transfer its attributes to another paragraph
The Case For Styles

It’s easier / quicker to change

• the appearance of an existing document
• determine the appearance of a conforming document

They make it easier to achieve consistent appearance

• within a document
• using templates, across documents

You can switch media much more easily
The Case Against

It takes longer to get started
In well-designed applications you can do everything via

- menu items
- dialog boxes opened by a menu item
- a toolbar opened from a menu item
  — Why?

Often there are other ways of doing things

- typically faster but more obscure
  — Why?

Are there other ways to

- DEFINE named styles in Word?
- APPLY named styles In Word?

Explore the Styles pull-down in the Formatting toolbar

- (The Poke pearl)
Styles & Style Templates (aka “Style Sheets”) Elsewhere

These ideas are applicable \textit{anywhere you have objects with attributes}, though the term may not be used

- graphics applications
- printing
- web pages

So look for them!
Next week

Pixel Graphics

Have a digital photo ready for lab next week.