

#### **Announcements**

- Chapter 6 for today
- Chapter 7 for Friday



#### Announcement

- New Drop-In Lab added
  - \* Wednesdays, 1:30-2:30 (MGH 044)
  - \* All the other drop-in labs are in MGH 430:
    - Monday 4:30-5:20pm
    - Wednesday 10:30-12:20pm
  - \* See our course Web site's home page for more details
- Any student can go to anyone's office hours



#### Announcement

- New Drop-In Lab added
  - \* Wednesdays, 1:30-2:30 (MGH 044)
  - \* All the other drop-in labs are in MGH 430:
    - Monday 4:30-5:20pm
    - Wednesday 10:30-12:20pm
  - \* See our course Web site's home page for more details
- Any student can go to anyone's office hours



#### **PROJECT 1A**



#### Project 1—Details on Web

Topics can range from science fiction to the *National Inquirer!* 

- \* Another sphinx found amidst Lunar ruins of old civilization
- \* New human gene found
- \* Eiffel Tower moved to Seattle
- \* Dolphin dictionary developed and tested: first address by the Queen Dolphin
- \* 12-year-old quarterback leads Huskies to third national title!



## Project 1—Details on Web

- Subtlety? No!
  - \* Content should move from serious to absurd, ridiculous, or hilarious
    - First glance: serious
    - Anyone who reads it thoroughly should realize that the site is a hoax
- Label the site as a hoax or bogus
  - \* Logo
  - \* Disclaimer



#### Project 1

- Include copyright information for all photos and other content
  - \* See instructions for more details



#### Turn In Steps

- Publish your page by uploading to the Web server
- Copy your Project1A files into a Project1B folder
- Submit your Word or .txt with project URL in Catalyst Collect It
- Do not touch anything in your 1A folder after the deadline
   We will check
  - \* Until grading is complete the timestamps



Applied to Web publishing

## WWW AND INTERNET KNOWLEDGE



#### Questions

- Over the next six weeks, think about
  - \* What is Dante and where is it?
  - \* Where are your Web files stored?
  - \* What is the Students server?
  - \* How does Students relate to what is in your SSH File Transfer window?
  - \* What does "local" mean?
  - \* What does "remote" mean?



# Debugging & Troubleshooting

"To err is human, but it takes a computer to really foul things up"

© 2008 Lawrence Snyder



Precision: The High Standards of IT

- Precision in Everyday Life
  - \* Many people say "oh" rather than zero when giving a phone number
  - \* The listener makes the mental conversion because he knows phone numbers are numeric
  - \* Computer does not know that unless it has been programmed to know it



## Exactly How Accurate Is "Precise?"

- Modem or database software can be programmed to make "oh" to zero corrections automatically because all digits will always be numbers
- In e-mail programs, both letters and numbers are allowed, so the computer can't be programmed to auto-correct. Users have to be careful

7-13



#### **Lexical Structures**

- Field Inputs: Information entered into boxes/forms on screen
  - \* Governed by *lexical structures* (rules about the legal form, or syntax, for input fields)
    - May limit symbols that can appear in specific positions, length of entry, etc.
    - May also be *loose*, allowing any sequence of symbols of any length

7-14



## Debugging: What's the Problem?

- Debugging: The method of figuring out why a process or system doesn't work properly
  - \* Usually learned from experience
- Debugging in Everyday Life: People troubleshoot problems such as why their cars don't start
  - \* Usually involve working system with broken or worn-out part
- Debugging in Information Technology:
  - Working system might have wrong data or wrong configuration information or
  - \* There might be a logical design error
- We will always begin by assuming a correct, working system





Figure 7.1 The Harvard Mark II logbook noting "First actual case of bug being found."

7-16



#### Animation

- Debugging HTML
  - \* http://courses.washington.edu/benefit/ FIT100/Lessons/Lesson3/graphics/htmld eBuq.swf



#### Whose Problem Is It?

- When we debug an information system, we are always part of the problem
  - \* We give the commands and the input
    - Operator or pilot error
  - \* Only other possible cause is broken system
- People don't knowingly make errors
  - \* We may think we did everything right because we misunderstand the system



# Using the Computer to Debug

- · A computer can't debug itself
- · We can't debug it directly, either
- Error is internal to the computer
  - \* To get information about the error, we have to ask the computer what data it stored, etc.
- With faulty software we cannot fix, try to bypass error with alternative approach (workaround)

7-19



## Dialog About Debugging

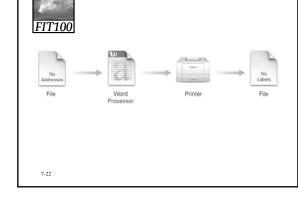
- Debugging is solving a mystery
  - \* Whatdunit vs. whodunit
- Ask purposeful questions like:
  - \* Do I need more clues?
  - \* Are my clues reliable?
  - \* What is a theory to explain the problem?
- Better than aimlessly "trying stuff"



### Debugging (cont'd)

- · Steps to follow:
- 1. Check that the error is reproducible
- 2. Make sure you know what the problem is
  - If no mailing labels come out of printer, problem may be with printer or with program sending labels to printer or file containing addresses
- 3. Check all the "obvious" sources of error
- 4. Isolate the problem
  - Divide operation into those parts that are working and those that are not working
     Think objectively—ask yourself if you've made a wrong assumption, etc.

-21





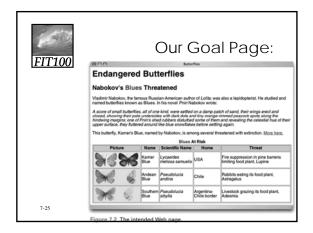
## Debugging Recap

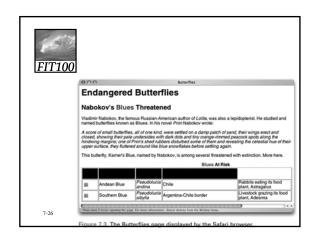
- · Make sure you can reproduce the error
- Determine exactly what the problem is
- Eliminate "obvious" causes (Is it plugged in?)
- Divide the process, separating out the parts that work from the part that doesn't (isolate the problem)
- When you reach a dead end, reassess your information; then step through the process again
- As you work through, make predictions about what should happen and verify that they are
   7-23 fulfilled

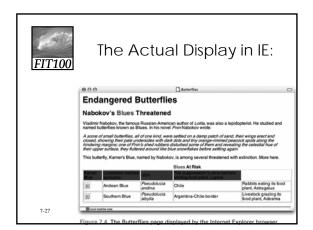


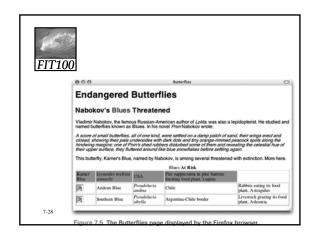
Butterflies And Bugs: A Case Study

- Imagine we've developed a simple HTML page
- Following is our goal page:











# Butterflies and Bugs (cont'd)

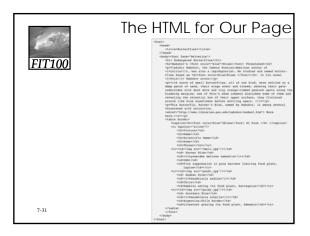
- Displaying the page in Safari and Mozilla also produces errors
  - \* Test in multiple browsers!
- We can study the HTML very closely and "brain out" where the error is
- We will use the debugging strategy

7-2



# Butterflies and Bugs (cont'd)

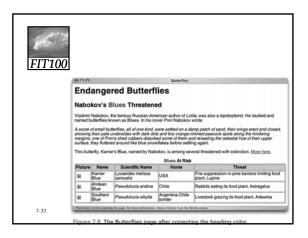
- First, be sure we can re-create the error
  - \* Reload the page
  - \* In this case, the results are the same
- · Next, determine the problem exactly
  - \* Look at the displayed page and determine where the errors occur
  - Since there are multiple errors on different parts of the page, assume they are caused by different mistakes in the HTML
    - Try to fix first/top error on page first; save and refresh browser





### **Butterflies and Bugs** (cont'd)

- · Next, eliminate the obvious
  - \* In HTML, most common error is forgetting to close a tag
    - For example: <b> with no closing </b> tage
  - \* Make sure all quotes match (open and close)
    - In this case, the word **Blues** appears in red on the page. Why?
      - <caption><b><font color=blue">Blues</font>
    - Missing opening quotation mark around the attribute value "blue"
  - \* Why is the "More here" link not highlighted?
  - Space missing between <a and href

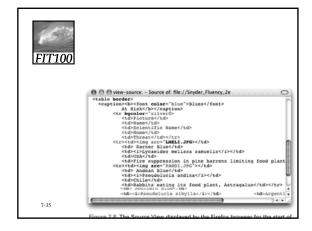




## **Butterflies and Bugs** (cont'd)

- Divide up the process
  - \* Separate the parts of the system that work from the parts that do not
    - · First row of the table
    - Browsers and debugging systems for HTML can give us a color-coded version of the HTML, showing how it's being interpreted (for example, Notepad++)
      - View (page) source
    - We see the entire heading line is colored blue as if it's an attribute

The problem turns out to be a fancy quote that the browser cannot interpret





## **Butterflies and Bugs** (cont'd)

- Reassess
  - \* The pictures are still not displaying
  - \* We find that the image tags are well structured and the browser is interpreting them correctly
  - \* What is the problem?
    - To check that the images are ok, display in another document
    - To check that the JPEG specification is ok, put another image file in the document
    - Check the exact file name for the image, including capitalization
    - In this case, we find the filename extension is .jpeg, not .jpg as we have been using



### **Butterflies and Bugs** (cont'd)

- Unnecessary changes:
  - \* During the debugging process, we made some unnecessary changes due to wrong conjectures about the error
  - \* Making unnecessary changes is typical in debugging
    - Sometimes we even make the situation worse by introducing new errors



### **Butterflies and Bugs** (cont'd)

- · Hiding other errors:
  - \* At first we thought we had three errors—bad caption, missing link, busted table
  - \* Because there were two things wrong with the table (messed up heading line and wrong file names specified) there were actually four errors
  - \* Because it is common for one error to hide another, always suspect there is more than one error



## **Butterflies and Bugs** (cont'd)

- · Viewing the source:
  - \* Most effective technique in our debugging was the View Source feature
  - \* In general, one of the most powerful debugging techniques is to find ways for the computer to tell us the meaning of the information it stores or the commands it executes
- · Little errors, big problems:
  - \* The errors in the HTML code were tiny, but they had serious effects
  - \* We must be extremely precise



#### No Printer Output: A Classic Scenario

- · Most systems we use, we don't create
- The software is very complicated. How do we troubleshoot a system we don't understand?
- Generally, software has been extensively tested before we come in contact with it
  - \* Standard operations are likely to be bug-free
- To illustrate debugging a system without understanding it, consider a common problem: You try to print a document and nothing comes out of the printer



## Applying the Debugging Strategy

- · Apply the debugging strategy:
  - \* Reproduce the error, understand the problem, check the obvious causes
  - \* Check the printer's control panel, the paper, the cartridges, cable connection, file to be printed, installation of printer driver, whether others can print (if it's a shared printer) and whether you can print a different document
  - \* If this does not solve the problem, press on



#### Pressing On

- Try to isolate the problem:
  - \* Configuration is correct because you've printed before
  - \* Try printing a simple document Same problem
  - \* Think through the probable process
    - If the computer couldn't send the data to the printer, wouldn't it give a message to plug in the printer?

    - Look around for the stranded files
       In the printing monitor's files, you find a list of files you've tried to print recently
       Start > Printers and Faxes



7-43

#### The Print Queue

- You've found the print queue
- Computer's settings may tell it to "queue" converted files rather than printing them immediately
  - \* Pause Printing
- You may never know how this occurred, but you can still fix it by re-configuring the driver
  - \* Unclick Pause Printing or click Resume Printing

FITT100

| State PFF | State P

