The final

- Covers the material not covered on the midterm
- Not cumulative
- Multiple Choice. Bring a SCANTRON
- Should be just a little longer than the midterm

- At officially scheduled time next Wed at 4.
- These slides select things you should know, should have learned the second half of the class.
Images, Pixels, ART!!
White, Gray, Black

- You know that gray is just different degrees of white as the “light is turned down” till we get to black.

Black = [0, 0, 0] 0000 0000 0000 0000 0000 0000
Gray = [128, 128, 128] 1000 0000 1000 0000 1000 0000
White = [255, 255, 255] 1111 1111 1111 1111 1111

White-gray-black all have same values for RGB
Colors

Colors use different combinations of RGB

BLOOD RED is my favorite color
Positional Notation

- The RGB intensities are binary numbers.
- Binary numbers, like decimal numbers, use *place notation*.

\[ 1101 = 1 \times 1000 + 1 \times 100 + 0 \times 10 + 1 \times 1 \]

\[ = 1 \times 10^3 + 1 \times 10^2 + 0 \times 10^1 + 1 \times 10^0 \]

Except that the base is 2 not 10.

\[ 1101 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 \]

\[ = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \]

1101 in binary is 13 in decimal.
Positional Notation Logic: Base 10 vs. Base 2 (binary)

Recall that the place represents a power of the base value

\[ \begin{align*}
  d_7 \times 10^7 \\
  d_6 \times 10^6 \\
  d_5 \times 10^5 \\
  d_4 \times 10^4 \\
  d_3 \times 10^3 \\
  d_2 \times 10^2 \\
  d_1 \times 10^1 \\
  d_0 \times 10^0
\end{align*} \]

\[ \begin{align*}
  d_7 \times 2^7 \\
  d_6 \times 2^6 \\
  d_5 \times 2^5 \\
  d_4 \times 2^4 \\
  d_3 \times 2^3 \\
  d_2 \times 2^2 \\
  d_1 \times 2^1 \\
  d_0 \times 2^0
\end{align*} \]
Steganography:
Once things are encoded, can do some tricky things…
Why Hide Information?

- Most common reason to hide information is to avoid being caught with it
  - Military and spy documents
  - Repressive governments restricting news/info
  - Avoid others “snooping” – privacy

- Hiding is different than encryption ...
- Encryption you can see: You can tell that it doesn’t make sense
- Hiding uses the fact that the searcher doesn’t know it’s there
Illustrate A Way To Do It

- The Plan ...
  - hide “subversive” protest picture in “calendar art”
Step 1: Reduce Bits of Guest

- We don’t need all of the bits in RGB to get a decent picture.
Step 2: Replace Bits In Host

- Put guest bits into right 2 bits of host

```
1111 0100 1101 0011 1011 1101
1101 0011 1011 0100 1101 0011
0001 1100
1111 0110
1101 0011
1011 1100
```

```
1011 0100 1101 0011 0001 1100
1111 0100 1101 0011 1011 1100
1111 0110 1101 0011 1011 1100
```
More Pictures!
IMAGES AND PICTURES

- http://www.processing.org/learning/pixels/
- This tutorial is really fun.
Random Numbers

- Random numbers should be called random number sequences, because the definition requires that no matter how many numbers you already know in the sequence, it’s not possible to predict the next one. A non-random sequence is 2, 4, 6, 8, 10, …
- Computers cannot produce random numbers (because computers are completely predictable), but they can produce a sequence of numbers that passes all of the tests for randomness. These are called pseudo-random numbers, but everyone drops the “pseudo” part.

To generate a random number in Processing we write:
random(<smallest possible number>, <largest possible number>).
We get back a number – we can’t predict which – between the two limits, including the end points.
To generate a random number between 0 and 255, write random(0, 255).
To generate a number between 0 and 1, write random(0, 1).
What would we do in Processing? What does Random do?

```java
void draw() {
    mid_outer = lerpColor(outer, inner, .33);
    mid_inner = lerpColor(outer, inner, .66);
    fill(outer);
    rect(60, 60, 480, 480);
    fill(mid_outer);
    rect(100, 100, 400, 400);
    fill(mid_inner);
    rect(165, 200, 270, 260);
    fill(inner);
    rect(210, 260, 180, 160);
}

void mousePressed() {
    r = random(0,255);
    g = random(0,255);
    b = random(0,255);
    outer = color(r,g,b);
    r = random(0,255);
    g = random(0,255);
    b = random(0,255);
    inner = color(r,g,b);
}
```
void draw( ) {
    mid_outer = lerpColor(outer, inner, .33);
    mid_inner = lerpColor(outer, inner, .66);
    fill(outer);
    rect(60, 60, 480, 480);
    fill(mid_outer);
    rect(100, 100, 400, 400);
    fill(mid_inner);
    rect(165, 200, 270, 260);
    fill(inner);
    rect(210, 260, 180, 160);
}

void mousePressed() {
    r = random(0,255);
    g = random(0,255);
    b = random(0,255);
    outer = color(r,g,b);
    r = random(0,255);
    g = random(0,255);
    b = random(0,255);
    inner = color(r,g,b);
}
Important Processing Concepts (some review)
Processing Concepts you NEED TO KNOW!

- Variables and Declarations
- Assignments
- Expressions
- Repetition (looping) or For-statements
- Tests or If-statements
- Functions
Writing Programs

- Programs are given sequentially, the declarations at the top
- Braces {} are statement groupers ... they make a sequence of statements into \textbf{just one} thing, like the “true clause of an If-statement”
- All statements must end with a semicolon EXCEPT the grouping braces ... they don’t end with a semicolon
- Name your variables something meaningful
- Generally white space doesn’t matter; comment your code
Repetition, Another Picture

- Or how about a bullseye?

```python
int i;
size(200, 200);
background(0);
fill(255, 0, 0);
for (i = 0; i < 5; i = i + 1) {
    fill(180 + 20*i, 0, 0);
    ellipse(100, 100, 100-(20*i), 100-(20*i));
}
```

- Note the *loop variable* must be declared ... but could do it in loop itself like we did for pacman pills:
- `for (int i = 0; ...`
Layering: Building Functions out of Functions
Instructions Formed of Simpler Instructions

- Check out this screen shot of the Lightbot
- It is partway through an instruction ... its beacon is lit, but not the tile
- To a programmer the instruction is monolithic (one thing)
- To an agent each instruction is a series of steps

An Instruction *abstracts* those steps
Abstraction

- The word “abstraction” is used a lot in computing
- Remember: it was one of the 7 big ideas
- Abstraction is a way to understand and solve problems
- As a general definition, abstraction eliminates details to focus on essential properties
- The instruction example just given illustrates functional abstraction meaning that we have given a name to a series of operations that perform a coherent (and to us meaningful) activity; the name is the instruction, the series of operations are the bot’s actions to implement it
Abstracting

- Collecting the operations together and giving them a name is *functional abstraction*
  - The group of operations perform some function but we ignore all of the details
  - Giving it a name is *functional* abstraction
  - This is AMAZINGLY powerful
  - What makes it powerful, is we can forget about the operations and think only about the function they do; more about this later

- Let’s do some functional abstraction
Example: Abstraction in Everyday life

Get Dressed

Dress Bottom Half
- Put on Sox
- Put on Pants

Dress Top Half
- Put on Shirt
Homework 10: Functions

- void cell(int x, int y, int s, color tinto)
- void triple(int x, int y, int s, color tinto)
- Also functions for
  - block( )
  - row( )
  - cellarray( )

*Sudoku GUI*

The Sudoku board is a 9×9 array of cells. Alternating 3×3 cells are colored gray and strong lines separate all of the 3×3 cells, as follows. (More information at http://www.websudoku.com/.)
Functions In Processing: Result

- Functions that do something, but do not return a value, have `void` as their `<return type>`.
- Functions that return a value must say its type.

```java
void draw_a_box (int x_pos, int y_pos) {
    rect(x_pos, y_pos, 20, 20);
}

color pink () {
    return color(255, 200, 200);
}
```
Using Functions

- Once defined, functions can be called repeatedly ... it’s the point of writing them!

```cpp
void setup() {
    size(110, 100);
    background(0);
    noStroke();
}

void draw() {
    fill(255);
    hexa(20, 20);
    hexa(50, 20);
    hexa(80, 20);
}

void hexa(float xbase, float ybase) {
    rect(xbase, ybase+10, 20, 40);
    triangle(xbase, ybase+10, xbase+20, ybase+10, xbase+10, ybase);
    triangle(xbase, ybase+50, xbase+20, ybase+50, xbase+10, ybase+60);
}
```
Arguments Become Parameters

- Notice that if the DEFINITION has \( n \) parameters, the CALL needs \( n \) arguments
- The parameters and arguments correspond

```c
void draw( ) {
  fill(255);
  hexa(20, 40);
  hexa(50, 40);
  hexa(80, 40);
}

void hexa(float xbase, float ybase) {
  rect(xbase, ybase+10, 20, 40);
  triangle(xbase, ybase+10, xbase+20, ybase+10, xbase+10, ybase);
  triangle(xbase, ybase+50, xbase+20, ybase+50, xbase+10, ybase+60);
}
```

Inside of the function, the parameter, e.g. xbase, is declared and initialized to the corresponding argument, e.g. 80. Then, the definition uses it, e.g.

\[
\text{rect}(80, 40+10, 20, 40)
\]
Parameters

- Parameters are automatically declared (and initialized) on a call, and remain in existence as long as the function remains unfinished.
- When the function ends, the parameters vanish, only to be recreated on the next call.
- It is wise to choose parameter names that help you remember exactly what they mean.
  - colorFlag (Chao’s code),
  - dir (for direction)
Example: Clock Timer. In sample codes.

- Draw digital timer elements
- Assemble elements into digits
- Light digit segments to create numbers
- Select number based on a digit
Functional Abstraction: Layers of Functions

- Review What We Did

The computation is ONLY drawing triangles and rectangles, but we don’t think of it that way ... to us, it’s a timer
Binary Arithmetic; We’ve seen lots of bits and bytes already
With 8 places how many different letters?
White, Gray, Black

- You know that gray is just different degrees of white as the “light is turned down” till we get to black

  \[
  \text{Black} = [0, 0, 0] \quad 0000\ 0000\ 0000\ 0000\ 0000
  \]

  \[
  \text{Gray} = [128, 128, 128] \quad 1000\ 0000\ 1000\ 0000\ 1000\ 0000
  \]

  \[
  \text{White} = [255, 255, 255] \quad 1111\ 1111\ 1111\ 1111\ 1111\ 1111
  \]

White-gray-black all have same values for RGB
Colors

Colors use different combinations of RGB

Purple
Red=160
Green=76
Blue=230
Positional Notation: More after the midterm

- The RGB intensities are binary numbers
- Binary numbers, like decimal numbers, use place notation

\[ 1101 = 1 \times 1000 + 1 \times 100 + 0 \times 10 + 1 \times 1 \]

\[ = 1 \times 10^3 + 1 \times 10^2 + 0 \times 10^1 + 1 \times 10^0 \]

except that the base is 2 not 10

\[ 1101 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 \]

\[ = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \]

1101 in binary is 13 in decimal
Positional Notation Logic: Base 10 vs. Base 2 (binary)

Recall that the place represents a power of the base value

\[ d_7 \times 10^7 \]
\[ d_6 \times 10^6 \]
\[ d_5 \times 10^5 \]
\[ d_4 \times 10^4 \]
\[ d_3 \times 10^3 \]
\[ d_2 \times 10^2 \]
\[ d_1 \times 10^1 \]
\[ d_0 \times 10^0 \]

\[ d_7 \times 2^7 \]
\[ d_6 \times 2^6 \]
\[ d_5 \times 2^5 \]
\[ d_4 \times 2^4 \]
\[ d_3 \times 2^3 \]
\[ d_2 \times 2^2 \]
\[ d_1 \times 2^1 \]
\[ d_0 \times 2^0 \]
The Red of P As A Binary Number

Purple: Red=160, Green=76, Blue=230

Given a binary number, add up the powers of 2 corresponding to 1s

\[
\begin{align*}
1 \times 2^7 &= 1 \times 128 &= 128 \\
0 \times 2^6 &= 0 \times 64 &= 0 \\
1 \times 2^5 &= 1 \times 32 &= 32 \\
0 \times 2^4 &= 0 \times 16 &= 0 \\
0 \times 2^3 &= 0 \times 8 &= 0 \\
0 \times 2^2 &= 0 \times 4 &= 0 \\
0 \times 2^1 &= 0 \times 2 &= 0 \\
0 \times 2^0 &= 0 \times 1 &= 0 \\
\end{align*}
\]

= 160
Green of P As A Binary Number

Purple: Red=160, Green=76, Blue=230

Given a binary number, add up the powers of 2 corresponding to 1s

\[
\begin{align*}
0 \times 2^7 &= 1 \times 128 &= 0 \\
1 \times 2^6 &= 0 \times 64 &= 64 \\
0 \times 2^5 &= 1 \times 32 &= 0 \\
0 \times 2^4 &= 0 \times 16 &= 0 \\
1 \times 2^3 &= 0 \times 8 &= 8 \\
1 \times 2^2 &= 0 \times 4 &= 4 \\
0 \times 2^1 &= 0 \times 2 &= 0 \\
0 \times 2^0 &= 0 \times 1 &= 0 \\
\hline
&\text{=}76
\end{align*}
\]
Is It Really the Purple we want?

- So Purple is (160,76,230) which is

\[
\begin{align*}
1010 & 0000 \\
0100 & 1100 \\
1110 & 0110 \\
\end{align*}
\]

160 76 230

Suppose you decide it’s not “red” enough
- Increase the red by 16 = 1 0000

\[
\begin{align*}
1010 & 0000 \\
+ & 1 0000 \\
\hline
1011 & 0000 \\
\end{align*}
\]

Adding in binary is pretty much like adding in decimal
How do we make a Redder Purple

- **ADD 16 more**

  \[
  \begin{align*}
  00110 & 000 \\
  1011 & 0000 \\
  + & 10000 \\
  \text{1100} & 0000 \\
  & \text{Carries}
  \end{align*}
  \]

  The rule: When the “place sum” equals the radix (base) or more, subtract radix (base 2) & carry

  *Check it out online: searching* binary addition

  *hits 19M times, and all of the page one hits are good explanations*
Find Binary From Decimal: be able to do these

What is 230 (the Blue of P)? Fill in the Table:

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
<th>6</th>
<th>6</th>
<th>6</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Subtract. Remainder</td>
<td>102</td>
<td>38</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

3/11/12 © 2010 Larry Snyder, CSE
Guest Lecture: Sri Kurniawan
Announcements

- Today: Guest Lecture on Human Computer Interaction and Design for Special Populations by Prof. Sri Kurniawan
- Co-Director of the Assistive Technologies Lab
HTML, XML and Web Searching
What are you supposed to learn?

- HTML lets you programmatically indicate how a particular content should be displayed.
- It can be served up by any HTTP server anywhere in the world.
- Typically uses UTF-8 encoding to guarantee being able to be shown.
- BUT XML separates the **DATA** from the **PROCESSING** of that data.
- This is a powerful idea.
- Processing => search it, display it etc.
DATA and SOFTWARE DOWNLOADS.

PROFESSIONAL EXPERIENCE

Professor of Computer Science, Natural Language and Dialogue Systems Lab, University of California, Santa Cruz, 2009 to present

Professor of Computer Science, Head of Cognitive Systems Group, University of Sheffield, Sheffield, England, 2003 to 2009

Principal Research Staff Member, ATT Labs - Research, Florham Park, N.J., Speech Processing Software and Technology Research, 1996 to 2003

Research Scientist, Mitsubishi Electric Research Laboratories, Cambridge, Ma., Interactive Learning and Entertainment, 1993 to 1996

Consultant, Hewlett Packard Laboratories, Bristol, England, on dialogue systems, speech technology, and personal information systems: 1989-1993

Researcher, Dialogue Modeling Department, Electrotechnical Laboratory, Tsukuba City, Japan: Summer 1991


EDUCATION


B.A. Computer and Information Science, With Honors, University of California Santa Cruz, 1984.
My Web page -- Metadata

- Title
- Font names
- Table Structure
- Where to put line breaks

<html>
<head>
<title>University of California Santa Cruz - Marilyn A. Walker</title>
<meta name="description" content="Marilyn Walker's Home Page">
<meta name="keywords" content="marilyn walker, darpa communicator, DARPA Communicator, HCI, CHI, Human Computer Interaction, PARADISE, paradise, machine learning, automatic adaptation in dialogue, adaptive dialog, user adaptation, natural language, research, dialog strategies, automated agents, agent personality, evaluation, spoken dialogue systems, reinforcement learning, boosting, rankboost, intelligent systems, voice recognition, speech recognition, spoken language generation, natural language generation, concept-to-speech generation, ripper">
</head>
<body>
<table border="0" cellspacing="0" cellpadding="5" width="100%">
<tr valign="top">
<td align="right" width="20%"><img src="lyn-v2.jpg" height="230" width="180"></td>
<td width="80%"><b><font face="verdana, helvetica, geneva"><font color="#000000"><font size="+2">Professor Marilyn Walker</font></font></font></b>
<br><b>Department of Computer Science</b>  
<br><b>University of California Santa Cruz</b>  
<br><b>1156 N. High, SOE-3</b>  
<br><b>Phone: 831 429 1058</b>  
<br><a href="http://nlds.soe.ucsc.edu">Natural Language and Dialogue Systems Lab</a>  
<br><a href="mailto:maw@soe.ucsc.edu">maw@soe.ucsc.edu</a>
</td>
</tr>
</table>
</body>
</html>
HTML and the Web

- The Web uses http:// protocol
- It's asking for a Web page, which usually means a page expressed in hyper-text markup language, or HTML
  - Hyper-text refers to text containing LINKS that allow you to leave the linear stream of text, see something else, and return to the place you left
  - Markup language is a notation to describe how a published document is supposed to look: what kinds of fonts, text color, headings, images, etc.
Basics of HTML #1

- Rule 1: Content is given directly; anything that is not content is given inside of tags
- Rule 2: Tags made of < and > and used this way:

\[
\text{Attribute&Value}
\]

\[
\text{Start} \quad \text{Content} \quad \text{End}
\]

Tag  

It produces: This is paragraph.

- Rule 3: Tags must be paired or “self terminated”
There are great resources out there

- http://www.w3schools.com/html/default.asp

HTML Tutorial

With HTML you can create your own Web site.

This tutorial teaches you everything about HTML.

HTML is easy to learn - You will enjoy it.

Examples in Each Chapter

This HTML tutorial contains hundreds of HTML examples.

With our online HTML editor, you can edit the HTML, and click on a button to view the result.

Example

```
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

Try it yourself

Click on the "Try it yourself" button to see how it works

Start learning HTML now!
Example: myfirst.html

<html>
<head>
<title>Fun in the Kitchen</title>
</head>
<body>
<h1>Making a Bake</h1>
<img src="cooking-ewan-isabel.jpg" alt="Kids Cooking" width="404" height="328" />
<p>When do we get to eat it?!</p>
</body>
</html>
Basics of HTML  #2

- Rule 4: An HTML file has this structure:
  
  ```html
  <html>
    <head><title>Name of Page</title></head>
    <body>
      Actual HTML page description goes here
    </body>
  </html>
  ```

- Rule 5: Tags must be properly nested
- Rule 6: White space is mostly ignored
- Rule 7: Attributes (style="color: red") preceded by space, name not quoted, value quoted
Basics of HTML #3

- To put in an image (.gif, .jpg, .png), use 1 tag
  
  `<img src="cooking-ewan-isabel.jpg" alt="Kids Cooking"/>

  Tag    Image Source    Alt Description    End

- To put in a link, use 2 tags
  
  `<a href=http://users.soe.ucsc.edu/~maw>Prof. Walker’s</a>`

  Hyper-text reference – the link    Anchor    End

- More on HTML (including good tutorials) at
  
  http://www.w3schools.com/html/default.asp

3/11/12  © 2011 Larry Snyder, CSE
In Processing: File Export

- Makes an index.html file. You browse it!
HTML Cheat Sheet: In Resources on Course page

Basic Tags
<html></html> Creates an HTML document
<head></head> Sets off the title and other information that isn’t displayed on the web page itself
<body></body> Sets off the visible portion of the document

Body Attributes
<body style="background-color:pink"> Sets the background color, using name or hex value
<body style="color:black"> Sets the text color, using name or hex value

Text Tags
<h1></h1> Creates the largest headline
<h6></h6> Creates the smallest headline
<b></b> Creates bold text
<i></i> Creates italic text
<tt></tt> Creates teletype, or typewriter-style text
<em></em> Emphasizes a word (with italic or bold)
<strong></strong> Emphasizes a word (with italic or bold)

Links
<a href="URL"></a> Creates a hyperlink; anchor between tags
<a href="URL"><img src="URL"></a> Creates hyperlink with image anchor

Formatting
<p></p> Creates a new paragraph
<p style="text-align:left"></p> Aligns a paragraph to the left (default), right, or center.
<br/> Inserts a line break
<blockquote></blockquote> Indents text from both sides
<hr /> Inserts a horizontal rule
<hr size="3" /> Sets size (height) of rule
<hr width="80%" /> Sets width of rule, in percentage or absolute value

Lists
<dl></dl> Creates a definition list
Tagging and Meta Data
Meta-data is data about data ... a description of what the data is

- Knowing what the data is, as in the OED, allows us to process it better for users
- Here’s an example: Search OED for def of “binary”
  - Without meta-data, get 8,311 hits ... which one is the definition?
  - With meta-data, get each definition in order ... how?

<e><hg><hw>binary</hw> ... </hg> ... <e>
Meta-Data Describes Data

- Meta-data is data about data ... a description of what the data is
  - Knowing what the data is, as in the OED, allows us to process it better for users
  - Here’s an example: Search OED for def of “binary”
    - Without meta-data, get 8,311 hits ... which one is the definition?
    - With meta-data, get each definition in order ... how?

\[
<e><hg><hw>binary</hw> \ldots </hg> \ldots <e>
\]

The Principle: We can program computers to better help us if we know what the content is
Meta-data Separation

- Improving on the meta-data of HTML
- Meta-data describes what the data is, but because the tags can be distinguished from the content, it separates itself from the content – that’s good
- But HTML combines “identifying content” with “saying how to process it”, i.e. display it
  - Big new idea (not part of HTML)

Separate the content and its tags entirely from the processing – produce a data-only file
The Advantage of Separating

- Separating the content (data) from the processing
  - You can do many different things with the content
    - Display it in multiple ways
    - Do calculations on it
  - You can maximize expertise
    - The content expert (you) puts the data together
    - The processing expert (some programmers) write the processing code based on the tags

Display App

[Image of a display app diagram]
The World of XML

- The Extensible Markup Language (XML) is a markup language in which YOU think up the tags ... it is a self-defining language!
  - The usual rules for tags apply
    - Enclose in < and >
    - Start tag `<mynewtag>` and End tag `</mynewtag>`
    - Tags must always be matched or self-terminated
    - Tags can have attributes (think those up, too) of form `attributename="valueInQuotes"`
    - Use `.xml` as the file extension
    - Always start with “standard text” (shown later)
Example of XML

- Suppose I want to record information about my cd collection using XML, I might write:

```xml
<catalog>
  <cd>
    <title>Maggie May</title>
    <artist>Rod Stewart</artist>
    <country>UK</country>
    <company>Pickwick</company>
    <price>8.50</price>
    <year>1990</year>
  </cd>
  <cd>
    <title>When a man loves a woman</title>
    <artist>Percy Sledge</artist>
    <country>USA</country>
    <company>Atlantic</company>
    <price>8.70</price>
    <year>1987</year>
  </cd>
  .......
</catalog>
```

I invent the tags; they make sense to me, and I can write a program to process such descriptions.
Learning XML

- Since we think up the tags ourselves, it's the easiest language in the world to learn, right?
- Tags can serve in three roles ...
  - **Identity** – tag it so you know what it is
    
    ```xml
    <name>George Washington</name>
    ```
  - **Affinity** – all properties of a thing should be collected together
    
    ```xml
    <personal>
      <name>George Washington</name>
      <height>6’2”</height>
      <teeth>Wooden</teeth>
      <home>Mount Vernon</home>
    </personal>
    ```
Ways To Use Tags

- **Identity** – tag it so you know what it is
  
  `<name>George Washington</name>`

- **Affinity** – all properties of a thing should be collected together

  `<personal>
   
   `<name>George Washington</name>
   `<height>6’ 2”</height>
   `<teeth>Wooden</teeth>
   `<home>Mount Vernon</home>
   
  </personal>`
Ways To Use Tags (continued)

- **Collection** – enclose a group of items of the same type in a collective tag

  <presidents>
  
  <prez num="1"> <personal> <name> George ... 
  <prez num="2"> <personal> <name> John ... 
  <prez num="3"> <personal> <name> Thomas ... 
  ...
  <prez> num="44"> <personal> <name> Barack ... 
  
  </presidents>

- These uses become intuitive quickly
Example: Classify The Uses

<travels>
  <visit>
    <sight> Orangatan Reserve. Borneo </sight>
    <action> flag = "orangutan.jpg"
    I got invited to give a talk at a workshop in Singapore, and we all went and popped over to Borneo before the workshop and went to an Orangutan Reserve. Fascinating!
  </action>
  <year>2008</year>
</visit>

<visit>
  <sight> Kyoto Japan </sight>
  <action> flag = "isabel-koyasan.jpg"
  I got invited to give the keynote at SIGDIAL 2010 in Tokyo and we all went and visited Kyoto and outlying areas like the holy mounta in Koya-San.
  </action>
  <year>2010</year>
</visit>

<visit>
  <sight> Roatan Honduras </sight>
  <action> flag = "snorkeling-kids.jpg"
  My kids hadn't been snorkeling since they were about 5. We heard Roatan was good and went for a week last year. Rays! Turtles!
  </action>
  <year>2011</year>
</visit>

-uu=:**-F1 travellog.xml Top L6 (XML)-----------------------------
Display XML with Browser

- We can see the structure of our XML (and check that it is well formed) by displaying it in Firefox.

- Introduce an error in a tag and see the error message when you browse it.

This page contains the following errors:

error on line 3 at column 51: Opening and ending tag mismatch: sight line 0 and sighte

Below is a rendering of the page up to the first error.
A Diary Of Travels

- One XML database could have entries for trips
- Kinds of info:
  - Where you went
  - Year
  - Pictures
- Also need a title

Some recent fun trips

2008
I got invited to give a talk at a workshop in Singapore, and we all went and popped over to Borneo before the workshop and went to an Orangutan Reserve. On the last morning we saw a mom and baby in the wild. You can kind of see it here. Borneo was the most interesting place I’ve ever been.

2010
I got invited to give the keynote at SIGDIAL 2010 in Tokyo and we all went and visited Kyoto and the Holy mountain Koya-San. We stayed at a Buddhist temple and feasted on umeboshi.

2011
My kids hadn’t been snorkeling since they were about 5. We heard Roatan was good and went for a week last year. Rays! Turtles!
Example: Building The XML File

<travels>
  <visit>
    <sight> Orangatan Reserve. Borneo </sight>
    <action pic = "orangutan.jpg">
    I got invited to give a talk at a workshop in Singapore, and we all went and popped over to Borneo before the workshop and went to an Orangutan Reserve. On the last morning we saw a mom and baby in the wild. You can kind of see it here. Borneo is the most interesting place I've ever been.
  </action>
  <year>2008</year>
  </visit>
  <visit>
    <sight> Kyoto Japan </sight>
    <action pic="isabel-koyasan.jpg">
    I got invited to give the keynote at SIGDIAL 2010 in Tokyo and we all went and visited Kyoto and the Holy mountain Koya-San. We stayed at a Buddhist temple and feasted on umeboshi.
  </action>
  <year>2010</year>
  </visit>
  <visit>
    <sight> Roatan Honduras </sight>
    <action pic="snorkeling-kids.jpg">
  </visit>
</travels>
Big Picture of How This Works

- Here are the players in this application

```xml
<travel>
  <visit>
    <sight>Orangutan Reserve, Borneo</sight>
    <action>flag = "orangutan.jpg"</action>
    I got invited to give a talk at a workshop in
    all went and popped over to Borneo before the talk
    Orangutan Reserve. Fascinating!
  </visit>
  <visit>
    <sight>Isode, Japan</sight>
    <action>flag = "isode-japan.jpg">My kids hadn’t been snorkeling since they were about 5. We heard it
    was good and went for a week last year. Rays! Turtles!
  </visit>
  <visit>
    <sight>Roatan, Honduras</sight>
    <action>flag = "snorkling-kids.jpg"</action>
    Rayo was good and went for a week last year. Rays! Turtles!
  </visit>
</travel>
```

Any Browser

Travels Page

XSL file

XML file
XSL: Extensible Markup Language

- XSL is “processing” markup language for XML ... and (of course) it’s written in XML
- Let’s take a look
- This is the top of the file

```xml
<?xml version="1.0" encoding="utf-8" ?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="travels">
    <html>
      <head><title>My Fun Trips</title>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
        <style type="text/css">
          body {background-color : #E6DCD6; color : #4C4A47;
            font-family : helvetica; width:800px; padding-left:70px}
        </style>
        </head>
        <body>
          <h2>Some recent fun trips</h2>
          <p style="text-align:center">This is my page about some recent family trips</p>
          <hr width="40%"/>
          <xsl:apply-templates/>
        </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```
XSL: Extensible Markup Language

- Just looking at XSL, it seems very complicated because it mixes its own tags with HTML tags

```xml
<?xml version="1.0" encoding="utf-8" ?>
<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="travels">
    <html>
      <head><title>My Fun Trips</title>
      <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
      <style type="text/css">
        body {background-color: #E6D6D6; color: #4C4A47;
           font-family: helvetica; width:800px; padding-left:70px;}
      </style>
      </head>
      <body>
        <h2>Some recent fun trips</h2>
        <p style="text-align:center">This is my page about some recent family trips</p>
        <hr width="40%"/>
        <xsl:apply-templates/>
      </body>
    </html>
</xsl:template>
```

Standard header text ... must be first in XSL file

Here’s how to say what XML tag this definition is for

HTML in here

Important tag, to be explained next
The “root tag” needs to style the main page

```
<html>
<head>
    <title>My Fun Trips</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <style type="text/css">
        body {background-color: #E6D6D6; color: #4C4A47; 
            font-family: helvetica; width:800px; padding-left:70px}
    </style>
</head>
<body>
    <h2>Some recent fun trips</h2>
    <i>This is my page about some recent family trips</i></p>
    <hr width="40%"/>
</body>
</html>
```
Continue For Other Tags

```xml
<xsl:template match="visit">
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match="sight">
  <h3>
    <xsl:apply-templates/>
  </h3>
</xsl:template>

<xsl:template match="sight">
  <tr>
    <td>
      <h3>
        <xsl:apply-templates/>
      </h3>
    </td>
  </tr>
</xsl:template>
```
Linking to XSL

- To make this all happen, we go to the XML and link it to the XSL file
  - Add two “boiler plate” encoding lines (just copy)
  - Make href point to the XSL file

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="mytravel-v2.xsl"?>
<travels>
  <visit>
    <sight> Orangatan Reserve. Borneo </sight>
    <action pic = "orangutan.jpg">
      I got invited to give a talk at a workshop in Singapore, and
      ll went and popped over to Borneo before the workshop and went to
      n Orangutan Reserve. On the last morning we saw a mom and baby
      he wild. You can kind of see it here. Borneo is the most interes
      g place I’ve ever been.
    </action>
  </visit>
</travels>
```
Ta Dah!

We have an app! ...

open XML file with Firefox

Some recent fun trips

This is my page about some recent family trips

2008
I got invited to give a talk at a workshop in Singapore, and we all went and popped over to Borneo before the workshop and went to an Orangutan Reserve. On the last morning we saw a mom and baby in the wild. You can kind of see it here. Borneo was the most interesting place I’ve ever been.

2010
I got invited to give the keynote at SIGDIAL 2010 in Tokyo and we all went and visited Kyoto and the Holy mountain Koya-San. We stayed at a Buddhist temple and feasted on umeboshi.

2011
My kids hadn't been snorkeling since they were about 5. We heard Roatan was good and went for a week last year. Rays! Turtles!
Now, Add to Data As Needed

- The Travel Log app is done; we don’t expect to write more XSL, except to restyle something
- Just add to XML file

```xml
<visit>
  <sight> Big Sur </sight>
  <action pic="bigsur.jpg"> Always good for a short break. </action>
  <year>2012</year>
</visit>
```
Adding More KINDS of Data

- What about the pictures???
- Add an XML picture tag ... like HTML, it will be self-terminating; needs styling XSL, too

- `<map name="borneo-map.jpg" />` goes in XML
- `<xsl:template match="map">` <img src="{@name}" alt="Country Map" width="500"/> `</xsl:template>`
- Place `<map>` tags inside of `<action>` tags
What are you supposed to learn?

- HTML let's you programmatically indicate how a particular content should be displayed.
- It can be served up by any HTTP server anywhere in the world.
- Typically uses UTF-8 encoding to guarantee being able to be shown.
- BUT XML separates the DATA from the PROCESSING of that data.
- This is a powerful idea.
- Processing => search it, display it etc.
And now for something completely different!!

Computability and correctness of a computer program.
Thinking About Computing

- Computers do things quickly ... except when they don’t
  - Usually we don’t know why
  - It could just be congestion on the internet
  - Or, when saving large files, like movies, we’re waiting for the hard disk to copy everything

- Describing Computability
  - *Time proportional to X*

- Sometimes the time a computer takes is *linearly* proportional to how much data there is ...
Linear = Time Proportional To N

Each item in a list that is N items long

- Problems whose work (computation time) is proportional to n are called n-time or linear time problems
  - Making an image lighter in your photo software
  - Adding a column of numbers in a spreadsheet
  - Crawling the Internet looking for links
  - ... many more ... linear problems are common

- Some problems are not ...
Algorithms

- Def. An algorithm is a precise, systematic process for an agent to produce a specified result

- Programs are algorithms

- Five properties characterize algorithms
  - **Input specified** – tell form and amount of input required
  - **Output specified** – tell form and amount of output produced
  - **Definiteness** – say explicitly what to do & in what order
  - **Effectiveness** – operations within agent’s abilities
  - **Finiteness** – will stop and give an answer or say “none”
People try to design algorithms e.g. to sort things, to run quickly
Linear = Time Proportional To N

Each item in a list that is N items long

- Problems whose work (computation time) is proportional to n are called n-time or linear time problems
  - Making an image lighter in your photo software
  - Adding a column of numbers in a spreadsheet
  - Crawling the Internet looking for links
  - ... many more ... linear problems are common

- Some problems are not ...
Linear Search

- Pick a number between 1 and 1000. How many guesses will I need?

- What about Google’s index with 25 million entries?

- The search happens in memory, but the list of URLs associated with the “term” will likely be on disk.
Binary Search

- Pick a number between 1 and 1000. How many guesses will I need?

- Web Search: What about that index with 25 million entries?
Sorting

- Putting a sequence of items into alphabetical or numerical order
- First: let’s try exchanging them as we go along (bubbling them along)
  
  *walrus seal whale gull clam*

- Algorithm: compare to all following items, reorder if needed
- Other ways to sort we will talk about in a minute
How Long To Sort w/ Bubble Sort?

- The pattern is, for n items
  - n-1 focus on first item in the list
  - n-2 focus on second item
  - n-3 focus on third item
  ... 
  - 1 on next to last
- n-1 rows in list, we just want a good estimate
- average of each row n/2,
- so (n-1) times (each row)
- Multiply by average = n/2
  = (n^2 – n)/2
- Computing time = Time proportional to n^2
Time proportional to $n^2 = \text{Polynomial time}$

- Other computations have running time
  - proportional to $n^3$ – matrix multiplication
  - proportional to $n^4$
  - ...

- All of them are lumped together as “polynomial time computations”
  - Considered to be realistic ... a person can wait
  - Polynomial, but not linear
  - Many algorithms you would learn in 12B, 101
There are Different Algorithms

- Is there a better way to do sorting?

- **QUICKSORT**
  - Fastest known sorting algorithm in practice
  - Average case: $O(N \log N)$ (we don’t prove it)
  - Worst case: $O(N^2)$
    - But, the worst case seldom happens.
  - A divide-and-conquer recursive algorithm
Quicksort is the best: Divide and Conquer

- **Divide step:**
  - Pick any element *(pivot)* \(v\) in \(S\)
  - Partition \(S - \{v\}\) into two disjoint groups
    \[S1 = \{x \in S - \{v\} | x \leq v\}\]
    \[S2 = \{x \in S - \{v\} | x \geq v\}\]

- **Conquer step:** recursively sort \(S1\) and \(S2\)

- **Combine step:** the sorted \(S1\) *(by the time returned from recursion)*, followed by \(v\), followed by the sorted \(S2\) *(i.e., nothing extra needs to be done)*

To simplify, we may assume that we don’t have repetitive elements,
So to ignore the ‘equality’ case!
Different sorting algorithms and animations

This one is the parameterizable web page that has the same code and you can give different parameters for number of things to sort, speed of animation etc.


- This one is one of the ‘dancing animations’ we looked at both bubble sort and quick sort. This shows the “recursion” in a better way

- Hungarian Folk Dance Quick Sort http://www.youtube.com/watch?v=ywWBry6J5gz8
Also CORRECTNESS!

- How do we know that the algorithms work?
  - Developing algorithms is not just thinking them up
  - It is also reasoning through why they work ... you need to know *why* explicitly enough to tell someone else

- What is involved in doing that?
- Usually some kind of “proof”,
  - by considering different possible cases
  - By imagining it didn’t work and proving a contradiction
Why Does Bubble Sort Work?

- Why do you think it sorts?
  - There are several passes through the data with leading item fixed (marked with lines)

- Notice this property: After each pass, the leading item must be the smallest of all processed on the pass

- The leading item is first the first item, then the second etc.

- Proof by induction
Summary: Proving Correctness

- It is not sufficient to think up a clever algorithm ... you need to know why it works
- It's usually not tough, because the logic of your method typically translates into an explanation of why it works.
- This is what you are often doing when you are trying to fix your program.
To Infinity And Beyond

- There are more complex computations ...
  - Suppose you want to visit 28 cities in the US (for a rock concert?) and you want to minimize your how much you pay for airplane tickets
  - You could select an ordering of cities (SEA → PDX → SFO → LAX ...) and compute the ticket price.
  - Then pick another ordering (SEA → SFO → LAX → PDX ...), compute this ticket price and compare to the previous one
  - Always keep the cheapest itinerary
- This seems very dumb ... is there a better way?
Traveling Salesman Problem

- Actually, no one knows a way to solve this problem significantly faster than checking all routes and picking the cheapest ...
- Not polynomial time ... so we are guessing that there is no polynomial solution (Non Polynomial = NP)
- This is what is called an NP-Complete problem
  - Many many related problems ... the best solution is “generate and check”
    - Best way to pack a container ship
    - Most efficient scheduling for high school students’ classes
    - Least fuel to deliver UPS packages in Washington
    - Fewest public alert broadcast stations for US
Summary: Computational Complexity

- Many computations have time proportional to $n$
- Many, like sort, have running time proportional to $n^2$
- Others have running time proportional to $n^3, n^4, \ldots$
- Some computations are computable in principle but not in practice: \textit{NP-complete}
- Some things cannot be computed at all, such as the \textit{Halting Problem}
Needles in the Haystack: Google and Other Brokers in the Bits Bazaar

Blown To Bits Chapter 4
Important questions: What you should learn

- How can a search engine respond so fast?
- Does it find every relevant link?
- How does a search engine decide what gets listed first?
- If you try another search engine will you get the same result? If so, which is right? Which is better? Which is more authoritative?
- Are sponsored links better than “organic” links? Is the advertising necessary?
- What is the role of government? What should it be?
“For the user, search is the power to find things, and for whoever controls the engine, search is the power to shape what you see.”

Blown to Bits pg 112
Web search: It Matters How It Works

1. Gather information.
2. Keep copies.
3. Build an index.
4. Understand the query.
5. Determine the relevance of each possible result to the query.
6. Determine the ranking of the relevant results.
7. Present the results.
Its all free?? : Well no. Who Pays for What?

- Users could pay a subscription fee (early AOL and CompuServe)
- Web sites could pay for being indexed.
- The government could pay (taxes?).
- Advertisers could pay.

- And it matters who pays cause it affects how it works
Page Rank Algorithms

- The “crown jewels” of search engines lie in their page rank algorithms.
- Factors include:
  - keywords in heading or titles
  - keyword only in the body text
  - site is “trustworthy”
  - links on this page are to relevant pages
  - links to this page are relevant
  - age of the page
  - quality of the text (e.g. absence of misspellings)
Search Engines

No one controls what’s published on the WWW ... it is totally decentralized

To find out, search engines crawl Web

- Two parts
  - *Crawler* visits Web pages building an *index* of the content (stored in a database)
  - *Query processor* checks user requests against the index, reports on known pages [You use this!]

Only a fraction of the Web’s content is crawled
I. Gather Information

- Spiders or web crawlers wander the web building indices
- Estimates range from .02% to 3% of information is indexed
- How often does a page get visited?
  - some frequently (daily whitehouse.gov), others rarely
  - Crawler keeps track of which pages change frequently
- How does the crawler find its way and not go in circles?
- Logins keep bots/crawlers out.
2. Keep Copies

- Spider downloads the page as part of the “visit” in order to create the index.
- Search engine may “cache” the copy.
- Is this legal? What about copyright?
- But wait, browsing requires copying as well.

“(AFP) – Sep 15, 2011

NEW YORK — Google and publishers told a US judge Thursday they are close to settling a lawsuit over the Internet giant's controversial book-scanning project…”
3. Build an Index

- list of terms and for each term a list of where it appeared
- more than just the terms
  - terms in bigger font might be more important
  - terms in the title might be more important
- must be very fast to lookup
- could be millions of entries (not just words, but names, special numbers, etc.) requiring Gigabytes of memory
- must fit in the computers memory (see next slide)
4. Understand the Query

- Steps 1-3 happen in “the background”
- Not much “understanding” in today’s search engines but that could change soon.
- Advanced search engine features help

Cardinal’s beat Rangers
Vs.
Ranger’s beat Cardinal’s

What about a business called “THE”?
Make A Query

- When Google gets the query

![Google Logo](https://via.placeholder.com/150)

- It “ands” the two lists together, finding URLs that are on both lists
- It counts them up, records time, shows 10 hits
5. Determine Relevance

- “Recall” - what percentage of relevant documents are returned by the search?
- Simple relevance calculation -
  - count the number of times each search word appears in the document, add them all up
- Long documents get higher scores.
- Uninteresting words like “the” contribute to the score.
- All word occurrences are not equal (title words should count more).
6. Determine Ranking

- Which of the relevant documents should be displayed first?
- Simple solution - put one with highest relevance score first.
- What if many have the same score?
- Are ones with the highest relevance score really the most important? What about the source of the document (e.g. NY Times vs some random blog post).
Page Rank Algorithms

- The "crown jewels" of search engines lie in their page rank algorithms.

Factors include:
- keywords in heading or titles
- keyword only in the body text
- site is "trustworthy"
- links on this page are to relevant pages
- links to this page are relevant
- age of the page
- quality of the text (e.g. absence of misspellings)
Google Query Algorithm

In a first paper Larry Page and Sergey Brin gave their algorithm for processing a Google query:

1. Parse the query.
2. Convert words into wordIDs.
3. Seek to the start of the doclist in the short barrel for every word.
4. Scan through the doclists until there is a document that matches all the search terms.
5. Compute the rank of that document for the query.
6. If we are in the short barrels and at the end of any doclist, seek to the start of the doclist in the full barrel for every word and go to step 4.
7. If we are not at the end of any doclist go to step 4.
8. Sort the documents that have matched by rank and return the top k.

This algorithm is understandable to readers.
Google Short Index & Long Index

- Known as the short barrels and the long barrels..
- Short index:
  - store the words in link texts that point to a page (inbound links!!)
  - the words in a page’s title, and one or two other special things.
- The link text words are attributed to the target page, and not to the page that the link is on.
  - In other words, if my page links to your page, using the link text “Miami hotels”, then the words “Miami” and “hotels” are stored in the short index as though they appeared in your page, but they belong to my page. If 100 pages link to your page, using those same words as link text, then your page will have a lot of entries in the short index for those particular words.
- The long index is used to store all the other words on a page – its actual content.
Processing a query: short and long indices

- First try to get enough results from the short index.
- If you can’t get enough results, then use the long index to add to what they have.
  - It means that, if they can get enough results from the short index – that’s the index that contains words in link texts and page titles – then they don’t even look in the long index where the actual contents of pages are stored. Page content isn’t even considered if they can get enough results from the link texts and titles index – the short index.

- Thus: link texts are very powerful for Google rankings.
  - Much more powerful than page titles, because a page can have the words from only one title in the short index, but it can have the words from a great many link texts in there.
  - Page titles and meta descriptions were the second most powerful ranking factors, because they are stored in the short index.
What does Google use to search?

- An inbound link is simply a hyperlink that can also have an href description: the anchor text
- Last Thurs: To put in a link, use 2 tags

\[ \text{<a href=\text{http://users.soe.ucsc.edu/~maw}>Prof. Walker’s</a>} \]

- Inbound links are important because of the way that Google stores a page’s data, and the way that they process a search query.
Make A Query: [http://www.google.ca/advanced_search](http://www.google.ca/advanced_search)

- When Google gets the query
- It “ands” the two lists together, finding URLs that are on both lists
- It counts them up, records time, shows 10 hits
Tarta de chocolate de mi amiga Carol Kotkin y otra del NYT

Carol Kotkin es una maestra de cocina de Miami con quien cogí clases de cocina allá y luego cuando abrí Kitchen World la tuvimos como maestra invitada. Carol es una señora encantadora, la pasamos muy bien cuando estuvo aquí. Hace muchos años que no sé de ella. Hace unos años escribió un libro de cocina, el cual me mandó de regalo, donde resalta no tan solo la cocina judía/american, sino la gran influencia latina que ha tenido esa cocina en el Sur de la Florida. Hizo muy buenas recetas en Kitchen World, pero para mí la mejor fue ésta.

Es una tarta, no un bizcocho porque en vez de harina, lleva nueces. Carol es judía y siempre pensó que esta receta era judía. Pero en el New York Times Magazine del domingo pasado, encontré otra receta similar que suena buenísima también. Se las voy a traducir aquí, aunque todavía no la he hecho pero tengo la intención de hacerla este fin de semana.

El artículo está muy interesante pues habla del libro Chocolate Decadence de Janice Feuer el cual compré hace muchísimos años en un viaje a Williamsburgh, Virginia. Allí almorzamos en un restaurante y de postre Patricia, por supuesto, pidió el bizcocho de chocolate que se llamaba "Death by Chocolate". Era tan chocalatoso que mi esposo le dijo que si no iba a tener un "chocolate overdose" a lo que ella respondió: "you can never have a chocolate overdose"... ahí entendimos la fascinación de
Artificial Intelligence as Question Answering.
IBM Watson on Jeopardy
Summary: Search Engines => QA

- Search engines offer unprecedented access to information.
- Search engines place the power to shape what we see into the hands of a few companies.
- Search engines continue to evolve.
  - Recently adding in Google Plus
  - Startups on indexing twitter etc
- Question Answering is the next new thing!
Watson: In Winter 2011 it won at Jeopardy!

- Let’s try it.
What is the underlying technology called? Question Answering (QA)

This is a Natural Language Processing Technology
How long have people been working on it?

- TREC = Text REtrieval Conferences
  - Series of annual evaluations, started in 1992
  - Organized into “tracks”

- Test collections are formed by “pooling”
  - Gather results from all participants
  - Corpus/topics/judgments can be reused

- TREC has had a QA Track since 1999.
Information Retrieval (IR)

- The first versions of search engines
- IR systems
  - Use statistical methods
  - Rely on frequency of words in query, document, collection
  - Retrieve complete documents
  - Return ranked lists of “hits” based on relevance
- Limitations
  - Answers questions indirectly (with a ranked list of documents that may or may not have the answer)
  - Does not attempt to understand the “meaning” of user’s query or documents in the collection
Information retrieval technology was the start for web search

Q/A is what web search might be like in 5 years (possibly less. You heard it here first!)
Contrast IR/Search with Asking an Expert
What does Jeopardy Host do?

Informed Decision Making

**Decision Maker**
- Asks NL Question
- Considers Answer & Evidence

**Expert**
- Understands Question
- Produces Possible Answers & Evidence
- Analyzes Evidence, Computes Confidence
- Delivers Response, Evidence & Confidence

**Computer**
- Understands Question
- Produces Possible Answers & Evidence
- Analyzes Evidence, Computes Confidence
- Delivers Response, Evidence & Confidence

**Jeopardy! Host**
- Asks NL Question
- Judges Answer Correct
Information Extraction (IE)

- IE systems (*usually... but recent advances*)
  - Identify documents of a specific type
  - Extract information according to pre-defined templates
  - Place the information into frame-like database records

<table>
<thead>
<tr>
<th>Weather disaster:</th>
<th>Type</th>
<th>Damage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Date</td>
<td>Deaths</td>
</tr>
<tr>
<td></td>
<td>Location</td>
<td>...</td>
</tr>
</tbody>
</table>

- Templates = pre-defined questions
- Extracted information = answers
- Limitations
  - Templates are domain dependent and not easily portable
Types of Question Answering


- **Factoid**
  - Who discovered oxygen?
  - When did Hawaii become a state?
  - Where is Ayers Rock?
  - What team won the World Series in 1992?

- **List**
  - What countries export oil?
  - Name U.S. cities that have a “Shubert” theater.

- **Definition**
  - Who is Aaron Copland?
  - What is a quasar?
Central Idea of Factoid QA

- Determine the semantic type of the expected answer
  “Who won the Nobel Peace Prize in 1991?” is looking for a PERSON

- Retrieve documents that have keywords from the question
  Retrieve documents that have the keywords “won”, “Nobel Peace Prize”, and “1991”

- Look for named-entities of the proper type near keywords
  Look for a PERSON near the keywords “won”, “Nobel Peace Prize”, and “1991”
Extracting Named Entities

Person: Mr. Hubert J. Smith, Adm. McInnes, Grace Chan
Title: Chairman, Vice President of Technology, Secretary of State
Country: USSR, France, Haiti, Haitian Republic
City: New York, Rome, Paris, Birmingham, Seneca Falls
Province: Kansas, Yorkshire, Uttar Pradesh
Business: GTE Corporation, FreeMarkets Inc., Acme
University: Bryn Mawr College, University of Iowa
Organization: Red Cross, Boys and Girls Club
An Example

Who won the Nobel Peace Prize in 1991?

But many foreign investors remain sceptical, and western governments are withholding aid because of the Slorc's dismal human rights record and the continued detention of Ms Aung San Suu Kyi, the opposition leader who won the Nobel Peace Prize in 1991.

The military junta took power in 1988 as pro-democracy demonstrations were sweeping the country. It held elections in 1990, but has ignored their result. It has kept the 1991 Nobel peace prize winner, Aung San Suu Kyi - leader of the opposition party which won a landslide victory in the poll - under house arrest since July 1989.

The regime, which is also engaged in a battle with insurgents near its eastern border with Thailand, ignored a 1990 election victory by an opposition party and is detaining its leader, Ms Aung San Suu Kyi, who was awarded the 1991 Nobel Peace Prize. According to the British Red Cross, 5,000 or more refugees, mainly the elderly and women and children, are crossing into Bangladesh each day.
Answer Type Hierarchy
The Turing Test

- Turing in 1950 published a philosophical paper designed to stop people arguing about whether or not machines could think.
- He proposed that the question be replaced with a test, which is what is now called the Turing Test.
In 2007: Jeopardy seemed possible

The Jeopardy! Challenge
A palpable, compelling and notable way to drive the technology of Question Answering along 5 Key Dimensions

- Broad/Open Domain
- Complex Language
- High Precision
- Accurate Confidence
- High Speed

- $200
  If you’re standing, it’s the direction you should look to check out the wainscoting

- $400
  A round type of this structure, with the silo inside it, was popular in the 1900s; it’s rare now

- $1000
  Of the 4 countries in the world that the U.S. does not have diplomatic relations with, the one that’s farthest north

- $600
  1948: Johns Hopkins scientists find that this antihistamine alleviates motion sickness

- $800
  This title character was the crusty & tough city editor of the Los Angeles Tribune
Enabling Technologies – The Time Was Right

**Natural Knowledge**
- Large volumes natural language electronic text (e.g., news, wikis, reference, web, etc.)
- Encodes knowledge and greater linguistic contexts to better resolve intended meaning

**Semi-Structured Knowledge**
- Large volumes of Thesauri, Dictionaries, Folksonomies, Linked Data, and the Semantic Web
- Rapid, community-based construction
- Across many domains – Specialized and General

**NLP (Text Analysis)**
- Entity and Relation Detection, Syntactic & Semantic Parsing
- Statistical NLP - Broader coverage, lower cost Information Extraction
- Statistical Paraphrasing: Learn ways to express same meaning

**Compute Power**
- Massive parallel compute power
- 1000s of compute cores working simultaneously
- TBs of globally addressable main memory
Answer Types: Cannot Anticipate!!

Broad Domain

We do NOT attempt to anticipate all questions and build specialized databases.

In a random sample of 20,000 questions we found 2,500 distinct types*. The most frequent occurring <3% of the time. The distribution has a very long tail.

And for each these types 1000’s of different things may be asked.

Even going for the head of the tail will barely make a dent

*13% are non-distinct (e.g. it, this, these or NA)

Our Focus is on reusable NLP technology for analyzing vast volumes of as-is text. Structured sources (DBs and KBs) provide background knowledge for interpreting the text.
Interesting tradeoff: Knowledge, Precision, Open Domain

- **Structured KB** approach delivers high confidence if questions can be precisely mapped to existing & reliable sources. Turns out to be rarely the case and confidence and accuracy drop off quickly.

- Basic **Text Search** approach **never** delivers high confidences but quickly reaches and maintains a peak accuracy at about 30%.

- Must combine deep and shallow semantic analysis over structured & unstructured content to drive up precision, recall and confidence.
Does Wordnet know about Chocolate Cake?

WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: chocolate cake

Display Options: (Select option to change) [ ] Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations
Display options for sense: (gloss) "an example sentence"

Noun

• **S: (n) chocolate cake** (cake containing chocolate)
  • **direct hyponym** / **full hyponym**
  • **S: (n) devil's food, devil's food cake** (very dark chocolate cake)
  • **direct hypernym** / **inherited hypernym** / **sister term**
Chocolate Decadence Cake?: The Web knows all!

chocolate decadence recipe

About 1,990,000 results (0.13 seconds)

**Chocolate Decadence Cake I Recipe - Allrecipes.com**

![Image of chocolate decadence cake](image)

6★ ★★★★ 65 reviews - 425 cal

This is the richest chocolate cake ever! Any questions?

Recipes like this - Recipe Reviews - Photos of this recipe

**Chocolate Decadence Recipe - Allrecipes.com**

allrecipes.com/recipe/chocolate-decadence/

5★ ★★★★ 10 reviews - 4 hrs 45 mins - 250 cal

Six dainty cups of deeply rich chocolate are gently baked and served cooled to a small grateful crowd.

**Chocolate Decadence Cake III Recipe - Allrecipes.com**

allrecipes.com/recipe/chocolate-decadence-cake-iii/

4★ ★★★☆ 6 reviews - 1 hr - 324 cal

This is a very rich cake, a small piece goes a long way. The best way to describe it is that it's a cross between a moist brownie and fudge. Be sure to use good ...
How good does it have to be to win?

Our Analysis Reveals the Winner’s Cloud

Each dot – actual historical human Jeopardy! games

Winning Human Performance

In 2007, we committed to making a Huge Leap!

Grand Champion Human Performance

Computers? Not So Good.

2007 QA Computer System

More Confident

Less Confident

% Answered

Precision
A Few Guiding Principles

- Specific large hand-crafted models won’t cut it
  - Too slow, too narrow, too brittle, too biased
  - Need to acquire and analyze information from as-is knowledge sources

- Intelligence from many diverse methods
  - Many diverse algorithms must be combined: No single one is expected to solve the whole problem. Each addressing different weaknesses.
  - Relative impact of many overlapping methods must be learned

- Massive Parallelism is a Key Enabler
  - Pursue many competing independent hypotheses over large data
  - Efficiency will demand simultaneous threads of evidence evaluation
DeepQA: The Technology Behind Watson

Massively Parallel Probabilistic Evidence-Based Architecture
Generates and scores many hypotheses using a combination of 100’s of Natural Language Processing, Information Retrieval, Machine Learning and Reasoning Algorithms. These gather, evaluate, weigh and balance different types of evidence to deliver the answer with the best support it can find.
Evaluating possibilities and their Evidence

In cell division, mitosis splits the nucleus & cytokinesis splits this **liquid cushioning** the nucleus.

- Many candidate answers (CAs) are generated from many different searches
- Each possibility is evaluated according to **different dimensions of evidence**.
- **Just One** piece of evidence is if the CA is of the right type. In this case a “liquid”.

```
I(s(“Cytoplasm”, “liquid”) = 0.2↑
I(s(“organelle”, “liquid”) = 0.1
I(s(“vacuole”, “liquid”) = 0.2
I(s(“plasma”, “liquid”) = 0.7
```

“Cytoplasm is a **fluid surrounding** the nucleus…”

Wordnet → Is_a(Fluid, Liquid) → ?

Learned → Is_a(Fluid, Liquid) → yes.
In May 1898 Portugal celebrated the 400th anniversary of this explorer’s arrival in India.

In May, Gary arrived in India after he celebrated his anniversary in Portugal.

Evidence suggests “Gary” is the answer but the system must learn that keyword matching may be weak relative to other types of evidence.
Clue: In 1698, this comet discoverer took a ship called the Paramour Pink on the first purely scientific sea voyage.

Peter Sellers comes into the picture because he played a character whose wife is a paramour of the Phantom in "The Pink Panther"
Potential Business Applications

- **Healthcare / Life Sciences**: Diagnostic Assistance, Evidence-Based, Collaborative Medicine
- **Tech Support**: Help-desk, Contact Centers
- **Enterprise Knowledge Management and Business Intelligence**
- **Government**: Improved Information Sharing and Security
Summary

- Search engines offer unprecedented access to information.
- Search engines place the power to shape what we see into the hands of a few companies.
- Search engines continue to evolve.
  - Recently adding in Google Plus
  - Startups on indexing twitter etc
- Question Answering is the next new thing!
Guest Lecture: Narrative and Games
Guest Lecture: Online Reputation on Wikipedia