### 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.

### Clickers: It Matters How It Works

“The search tools that help us find needles in the digital haystack have become the lenses through which we view the digital landscape. Businesses and governments use them to distort our picture of reality.”

A. TRUE
B. FALSE

### Clickers

- Google responds efficiently to queries by going out and searching the web in real time.

A. TRUE
B. FALSE

### Which steps happen in the background?

A. Gather information & Keep copies.
B. Gather information & Keep copies & Build an index.
C. Understand the query.
D. Determine the relevance of each possible result to the query & Determine the ranking of the relevant results.

### How does it find the pages to index?

A. Every website has to register with Google to get indexed
B. Every website has to pay Google to get indexed
C. Google knows which pages to index because of your digital footprint
D. Google has a list of ‘trusted pages’ and it just follows the links from them
Google indexes what percent of the pages in the world?

A. 100% It indexes everything.
B. About 50% on average but some days it’s 100% and some days it’s 30%
C. Less than 5%
D. Best estimates put it at 30% because of all the bots that keep crawlers out.

Clickers: What goes into the index?

A. Every word on the page
B. The words that other sites use when they point to (link to) this page.
C. Only the keywords on the page.
D. Google has a list of special keywords that all pages get indexed by.

How often does a page get visited?

A. Every page that gets indexed is visited every day.
B. It depends on the page, pages like whitehouse.gov get visited daily others rarely
C. Google decides by keeping track of how often pages change.
D. B & C
E. None of the above.

Web pages typically represent the present but they could be a few days out of date.

A. TRUE
B. FALSE
Clickers

* The federal government uses your tax dollars to guarantee that Google and other search engine providers like Microsoft (Bing) return the objectively best results of your query.

A. TRUE  
B. FALSE

How does the index get used at search time by default? (without advanced search)

A. Every website registers with Google exactly which terms to index by and which combinations  
B. Google uses "boolean" combinations. The index is made of single words. Google ANDS them together and finds which webpages (URLs) are in the intersection of all the terms  
C. Google indexes the pages individually for each person using your digital footprint  
D. All of the above.

Clickers

* Web search is free and democratic. Every web page has an equal chance of being indexed and coming out the top of the list.

A. TRUE  
B. FALSE

How does it rank the pages it finds?

* It uses what is called a “page rank” algorithm, that uses many different factors  
* It depends only on who is willing to pay the most.  
* The government tells it how to rank pages.  
* None of the above.

The information that Page Rank uses includes:

* Keywords in heading or titles and keywords in the body text  
* Information about whether the site is “trustworthy”  
* Whether the links on this page are to relevant pages  
* Whether the LINKS TO THIS PAGE are relevant  
* age of the page  
* quality of the text (e.g. absence of misspellings) It uses what is called a “page rank” algorithm, that uses many different factors

Search engine possible funding models:

A. Users could pay a subscription fee  
B. Web sites could pay for being indexed.  
C. The government could pay using taxes the same way they pay for roads or police.  
D. Advertisers could pay for having their ads featured in the side bar.  
E. All of the above are possible.
Search engine current funding model:
A. Users pay a subscription fee
B. Web sites pay for being indexed.
C. The government pays using taxes the same way they pay for roads or police.
D. Advertisers pay for having their ads featured in the side bar.
E. B&D

What makes search attractive to advertisers?
A. Advertisements can be targeted more precisely using your digital footprint.
B. Advertisers only pay when you click on their ad.
C. Millions of people use search every day.
D. All of the above.

Homework 10: Layering Functions
- Functions SUDOKU homework assigned next Tuesday for one week, start in advance if you can.

For loops (Repetition)
- Repeating commands is powerful:
  - Lightbot 2.0 used recursion, a function calling itself
  - Symbolic Lightbot prefixed a number, i.e. Step
- Processing uses a for loop:

```java
void setup() {
  size(500, 200);
  background(0);
  noStroke();
  smooth();
  fill(255);
  for (int i = 0; i < 16; i++) {
    ellipse(100 + 25 * i, 100, 15, 15);
  }
}
```

Using Functions
- Functions package up computation ... when do we use them? All the time.
- Write some simple code to achieve a goal ...
To put the rectangle in different places, we "parameterize" the position, that is, use input to the function to place the rectangle.

```processing
void setup() {
  size(400, 200);
  background(20);
  noStroke();
  frameRate(150);
  fill(0);
  digit(50, 30);
  rect(140, 90);
}
```

Now we quit thinking of drawing a rectangle, but now think of placing a 50x50 rectangle.

```processing
void draw() {
  pushMatrix();
  translate(125, 20, 0);
  for (let i = 0; i < 5; i++) {
    rect(150, 50, 50);
    fill(255, 0, 0);
    rect(150, 75, 50, 50);
    fill(255, 0, 0);
  }
  popMatrix();
}
```

We are going to make a Clock Timer.

- Draw digital timer elements
- Assemble elements into digits
- Light digit segments to create numbers
- Select number based on a digit
Clicker: if we didn’t have hexa

A. The draw function would be exactly like it is.
B. The code for hexa would be inside the draw function one time exactly as it appears here.
C. The code for hexa would be inside the draw function three times exactly as it appears here.
D. The code for hexa would be inside the draw function three times with different values given as arguments to rect and triangle.
Count In Lights: a function for each number
- Light up the digit for each number
- Count_in_lights.pde

Count in Lights. Layers of Functions

Define hexa() and rexa()
- Parameterize the functions by a consistent position – upper left corner is good

Clicker: I have hexa() and rexa(), now what?
A. I can make fun little shapes
B. It obvious how I can make digits given these two shapes
C. I have to define at least two more functions before I can make digits
D. I think I can make digits but I just can’t see how.

Clicker: What does digit help you do?
A. You can re-use it over and over again
B. It tells you what the coordinates are for the various bits of the digit
C. If you pass the right arguments for xbase and ybase to digit you can make every digit
D. A & B
E. None of the above

Use hexa, rexa to make a Digit
- hexa_rexa_makeadigit.pde
Let There Be Light (and Dark)

- Define the illumination of the digit
- hexa_rexa_makeadigit_addlight.pde
  - Must declare two color variables, initialize to proper colors, use them in fill, and then check them

Clicker: What does the lighted digit help you do?

A. You can re-use it over and over again
B. It tells you what the coordinates are for the various bits of the digit
C. If you pass the right arguments for \( x_{\text{base}} \) and \( y_{\text{base}} \) to \text{digit} you can make every digit
D. A & B
E. None of the above

Count In Lights: a function for each number

- Light up the digit for each number
- Count_in_lights.pde

Clicker: why would you want Count_in_lights_three_level?

A. I wouldn’t want it, it just makes the code even more confusing
B. Adding numbers to \( y_{\text{base}} \) and \( x_{\text{base}} \) for \text{rexa} and \text{hexa} can cause silly errors.
C. It makes the code more self-documenting.
D. B & C

Clicker: this code for writing the digit 4 is …

```
void four (float \( x_{\text{base}} \), float \( y_{\text{base}} \)) {
  hexa(\( x_{\text{base}} \), \( y_{\text{base}} \)); //left upper
  hexa(\( x_{\text{base}}+60 \), \( y_{\text{base}} \)); //right upper
  rexa(\( x_{\text{base}}+10 \), \( y_{\text{base}}+10 \)); //mid horizontal
  hexa(\( x_{\text{base}}+60 \), \( y_{\text{base}}+60 \)); //right lower
}
```

A. Perfect just as it is
B. Has a bug for right upper
C. Has a bug for mid horizontal
D. Actually makes a five
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

- The computation is ONLY drawing triangles and rectangles, but we don’t think of it that way ... to us, it’s a timer

Select A Number To Display

- Given an integer, display it in lights

Create a 3 Digit Display

```c
void three_digit(int n, float base, float gap) {
  fill(dark);  
  digit(50, 50);  
  digit(100, 50);  
  fill(white);  
  rect(base+105, gap*125, 15, 15);  
  setf(n,0);  
  self(n,1);  
  setf(n,2);  
  self(n,3);  
  setf(n,4);  
  self(n,5);  
  setf(n,6);  
  self(n,7);  
  setf(n,8);  
  self(n,9);  
}
```

Count up At The Frame Rate

What is time doing?

- A. Nothing. It has a bug in it.
- B. It is counting to infinity.
- C. It is counting to 999 then starting at zero again
- D. It counts up to 1000 then starts at 1.
What is time doing?

```c
void draw_j()
{
    lite = color(255,185,0);
dark = color(64, 48, 0);
    if (step == 1) {
        fill(dark);
digit(50,90);
digit(140, 90);
digit(200, 90);
        fill(lite);
rect(255, 155, 15, 15);
time(time+1)%8000;
    sel(time%20, 250, 900);
    sel((time/10)%8, 140, 90);
    sel(time/100, 50, 90);
}
```

Functional Abstraction: Layers of 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()`

Homework 10: Functions