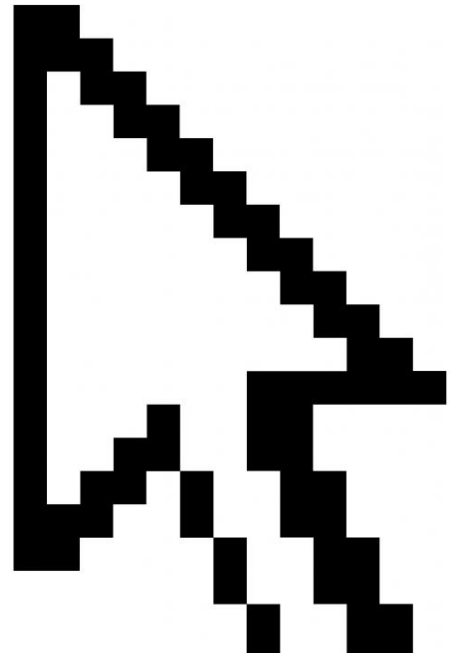


Day 3

aat kqssssssssss

cl i ck cl i ck

l nput s



Agenda

1. Recap: Conditionals
2. Debugging
3. Input: Keyboard
4. Input: Mouse

Recap: Conditionals

If I am hungry, then I will eat food.
Otherwise I will not eat.

```
// code
```

Recap: Conditionals

If I am hungry, then I will eat food.
Otherwise I will not eat.

```
if (hungry) {  
    EAT FOOD;  
}  
else {  
    DO NOT EAT;  
}
```

Recap: Conditionals

If I am hungry **and thirsty**, then I will eat food. Otherwise I will not eat.

```
if (hungry) {  
    EAT FOOD;  
}  
else {  
    DO NOT EAT;  
}
```

Recap: Conditionals

If I am hungry and thirsty, then I will eat food. Otherwise I will not eat.

```
if (hungry && thirsty) {  
    EAT FOOD;  
}  
else {  
    DO NOT EAT;  
}
```

Recap: Conditionals

If I am hungry **or** thirsty, then I will eat food. Otherwise I will not eat.

```
if (hungry && thirsty) {  
    EAT FOOD;  
}  
else {  
    DO NOT EAT;  
}
```

Recap: Conditionals

If I am hungry or thirsty, then I will eat food. Otherwise I will not eat.

```
if (hungry || thirsty) {  
    EAT FOOD;  
}  
else {  
    DO NOT EAT;  
}
```


Recap: Conditionals

If I am hungry and the temperature is 70° F, then I will eat food. Otherwise I will not eat.

```
if (hungry || thirsty) {  
    EAT FOOD;  
}  
else {  
    DO NOT EAT;  
}
```

Recap: Conditionals

If I am hungry and the temperature is 70° F, then I will eat food. Otherwise I will not eat.

```
int temp = 71;

if (hungry && temp==70) {
    EAT FOOD;
}
else {
    DO NOT EAT;
}
```

Recap: Conditionals

If I am hungry and the temperature is 70° F or below, then I will eat food.
Otherwise I will not eat.

```
i n t t e m p = 71;

i f ( h u n g r y && t e m p == 70) {
    E A T F O O D;
}
e l s e {
    D O N O T E A T;
}
```

Recap: Conditionals

If I am hungry and the temperature is 70° F *or below*, then I will eat food.
Otherwise I will not eat.

```
int temp = 71;

if (hungry && temp<=70) {
    EAT FOOD;
}
else {
    DO NOT EAT;
}
```

Debugging

when you go (boot) camping, be careful of bugs.

9/9

0800 Andam started
 1000 " stopped - andam ✓
 13^{sec} (032) MP - MC ~~1.982647000~~ { 1.2700 9.037847025
 (033) PRO 2 2.130476415 (2) 4.615925059(-2)
 condt 2.130676415
 Relays 6-2 in 033 failed special speed test
 in relay " 10,000 test -

1100 Started ^{Relays changed} Cosine Tapc (Sine check)
 1525 Started Multy Adder Test.

1545



Relay #70 Panel F
 (moth) in relay.

First actual case of bug being found.
~~1630~~ 1630 andam started.
 1700 closed down.

SOURCE

americanhistory.si.edu/
 collections/search/
 object/nmah_334663

Debugging

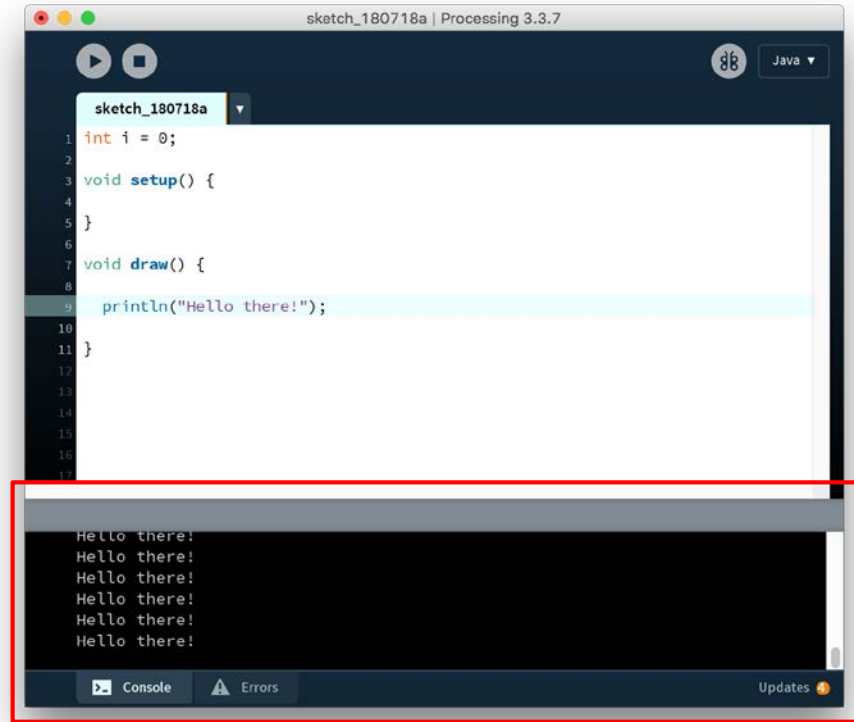
1. Bugs are errors in computer programs
2. Common problems: forgot a semicolon; or bracket }); spelling errors, ...
3. Bugs happen all the time and it's important to get good at finding and fixing them. Otherwise coding can become frustrating quickly.

Debugging

```
println("Hello there!");
```

```
// OR
```

```
int a = 4;  
print(a);
```



Exercise: Using println();

Create a simple sketch and see what messages you get when you add errors, (like adding typos or leaving out important characters)

Declare a variable that changes while the program is running and use `println()` to print its values to the console. Or use `println()` with a condition to test when it is true or false.

Debugging

When your code doesn't work, don't randomly guess!

- Read the error message in the console! It's trying to help you.
 - Check the line number if there is one.
 - Check the message: it tells you what's wrong
 - Can't understand it? Google It! StackOverflow usually has the right answer
- Debug your program
 - Don't try to fix everything at once. Separate the code into pieces (Line by line if necessary) and figure out where it breaks.
 - Use `/* comments */` to de-activate code blocks
 - Use the console to check if variables contain the right content; or if conditions are executed at the right time.

Inputs: Keyboard

situation-dependent speech,
prepared speech.



keyPressed

boolean: true or false

Use in an `if` statement in the `draw()`

```
void draw() {  
    if (keyPressed == true) {  
        fill(0);  
    } else {  
        fill(255);  
    }  
    rect(25, 25, 50, 50);  
}
```

keyPressed + key

key is the currently-pressed key

```
if (keyPressed == true) {  
    if(key == 's' || key ==  
'S'){  
        //doSome thing  
    }  
    else {  
        //doNot hi ng  
    }  
}
```

keyPressed()

runs when `keyPressed == true`

```
int value = 0;

void draw(){
    fill (value);
    ellipse(25, 25, 50, 50);
}

void keyPressed(){
    if(value == 0){
        value = 255;
    } else {
        value = 0;
    }
}
```

keyPressed() + keyCode

`keyCode` is a special ID number for every (or most) key.

Checking if `key == coded` makes sure that there is actually a `keyCode` for the one that's currently pressed

```
color value = color(125);

void keyPressed() {

    if (key == CODED) {
        if (keyCode == UP) {
            value += 10;
        } else if (keyCode == DOWN)
        {
            value -= 10;
        }
    } else {
        value = 125;
    }

}
```

keyReleased()

runs when `keyPressed` switches
from `true` to `false`

```
int value = 0;

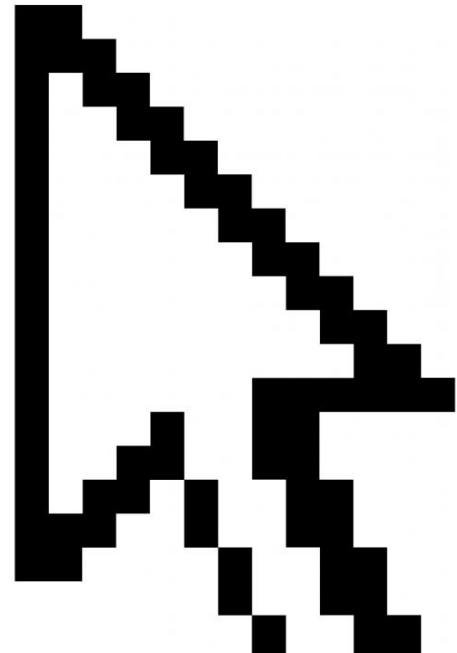
void draw() {
    fill(value);
    ellipse(25, 25, 50);
}

void keyReleased() {
    if (value == 0) {
        value = 255;
    } else {
        value = 0;
    }
}
```



```
String break = " 10 Minutes";
```

Inputs: Mouse



mouseButton and mousePressed

```
if (mouseButton == LEFT) ... // or CENTER or RIGHT
```

```
if (mousePressed == TRUE) ... // or CENTER or RIGHT
```

mouseButton and mousePressed

```
void draw() {  
    if (mousePressed && (mouseButton == LEFT)) {  
        fill(0);  
    } else if (mousePressed && (mouseButton == RIGHT)) {  
        fill(255);  
    } else {  
        fill(125);  
    }  
    ellipse(width/2, height/2, 25, 25);  
}
```

mousePressed()

runs once when the mouse is pressed

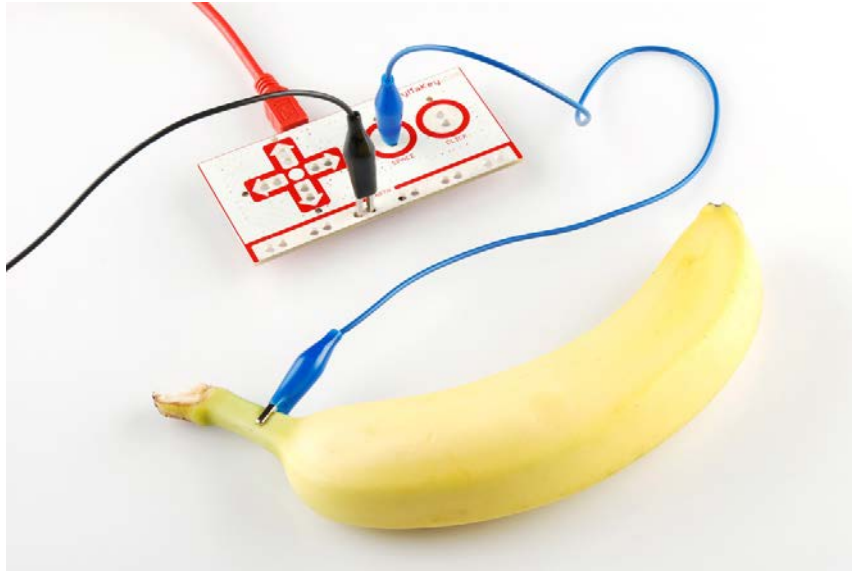
```
void draw(){  
    ellipse(width/2, height/2, 25, 25);  
    fill(125);  
}
```

```
void mousePressed(){  
    if(mouseButton == LEFT) {  
        fill(0);  
    } else if (mouseButton == RIGHT) {  
        fill(255);  
    }  
}
```

mouseX and mouseY

Integer variables that contain the X and Y position of the mouse cursor

```
void draw() {  
  background( 255);  
  fill( 0);  
  ellipse( mouseX, mouseY, 10, 10);  
}
```



Makey Makey



Leap Motion

Homework (or start now)

Create an interactive sketch using inputs like `mousePressed`, `keyPressed`, `keyCoded`, ...

Use conditionals to add even more control. For example, combine `mousePressed` and `mouseX`, `mouseY` to make a drawing app

