More topics for Lab 2

UNIVERSITY OF MOUNT UNION

The `draw()` function

- Code that responds to keyboard or mouse input must run continuously
- That's what code in the draw() block does: runs from top to bottom, then repeats until you hit "stop" or close the window
- Each trip through draw() is called a frame
  - The framerate is 60 frames per second by default, but this can be changed.

The `setup()` function

- This function runs just ONCE when the program starts
- Typically, you can use this to define starting values
  - `size()` comes first, followed by `fill()`, `stroke()`, etc.

Global Variables

- A variable created inside `setup()` can't be accessed inside `draw()`, and vice versa
- We can declare variables first so they can be used anywhere in the program
- These are called **global variables**

```
int x = 280;
int y = -100;
int diameter = 380;

void setup() {
  size(480, 120);
  fill(102);
}

void draw() {
  background(204);
  ellipse(x, y, diameter, diameter);
}
```

Data types: Booleans

- Booleans: true or false
- Examples:
  - `mousePressed`
  - `keyPressed`
- Can be used like this:

Here, we just have `if (keyPressed)` and Processing interprets that to mean "if `keyPressed` is `true`," meaning "if a key has been pressed"

```
void setup() {
  size(240, 120);
}

void draw() {
  background(204);
  line(20, 20, 220, 100);
  if (keyPressed) {
    line(220, 20, 20, 100);
  }
}
```

- Or like this:

```
void setup() {
  size(240, 120);
  strokeWeight(30);
}

void draw() {
  background(204);
  stroke(102);
  line(40, 0, 70, height);
  if (mousePressed == true) {
    stroke(0);
  }
  line(0, 70, width, 50);
}
```

Here, we have `if mousePressed == true` and Processing interprets that to mean "if `mousePressed` is `true`" meaning "if the mouse button has been pressed"

Either method can be used for **`mousePressed`**, **`keyPressed`**, or any Boolean variable!

Data types: characters and strings

- `char` (short for character) stores any single character (letter, number, symbol)
- Specified by single quotes
- Example:

```
char c = 'A';  // Declares and assigns 'A' to the variable c
```

And these attempts will cause an error:

```
char c = "A";  // Error! Can't assign a String to a char
char h = A;    // Error! Missing the single quotes from 'A'
```

- `String` stores text data (can be many characters)
- Specified by double quotes
- Example:

```
String message = "You have won the game!";

println(message);
```

```
You have won the game!
```

Some shortcuts: +=, -=, *=, /=, ++, --

- `x=x+2;` → `x+=2;`
- `x=x-2;` → `x-=2;`
- `x=x*2;` → `x*=2;`
- `x=x/2;` → `x/=2;`
- `x=x+1;` → `x++;`
- `x=x-1;` → `x--;`

Comparisons

- Logical AND (&&) operator: both sides must be true for this condition to be true.

```
int x = 10;

  if(x >= 5 && x <= 20) {
    println("x is between 5 and 20");
  }
  else {
    println("x is not between 5 and 20");
  }
```

```
x is between 5 and 20
```

- Logical OR (||) operator: at least one side must be true for this condition to be true
  - Unlike in English where “or” is used to mean just one (“exclusive or”), the || logical operator is still true if both sides are true!

```
int x = 10;

  if(x >= 5 || x <= 20) {
    println("the first OR condition is true");
  }
  else {
    println("the first OR condition is false");
  }
   if (x < 100 || x > 50) {
     println("the second OR condition is true ");
   }
   else {
     println("the second OR condition is false");
  }
```

```
the first OR condition is true
the second OR condition is true
```