VARIABLES, CONDITIONALS & BOOLEANS

Day 2

RECAP:

- using ints or floats to set positions
- use the “color” datatype to save colors
- use an int to set individual RGB components that color

organize and name things in a way that makes sense to you.

Conditionals

If I am hungry, then I will eat food.

Otherwise, I will not eat.

"If" I am hungry, **"then"** I will eat food.

Otherwise (**"else"**), I will not eat.

If **I am hungry**, then I will eat food.

Otherwise, I will not eat.

If **I am hungry**, then I will eat food.

Otherwise (**"else"**), I will not eat.

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

If I am thirsty and I am hot, I will drink cold water.

If I am thirsty **and** I am hot, I will drink cold water.

```
if (thirsty && hot){
    //if "thirsty" AND "hot" are both true, do the following:
    DRINK COLD WATER;
}
if (thirsty && cold){
    //if "thirsty" AND "cold" are both true, do the following:
    DRINK HOT TEA;
}
```

//Note: if one is true and the other is false, then the if statement will not run

If I am tired or it is late, I will go to sleep.

If I am tired **or** it is late, I will go to sleep.

```
if (tired || late){
    //if “tired” is true or “late” is true, then do the following:
    GO TO SLEEP;
}
```

== vs. =

==
(Test)

Double equal sign compares two values and returns true if they are equal

Asks a question

```
if (x == 10) {
    do this
}
```

=
(Assign)

Single equal sign sets a variable equal to a value.

Does not ask a question

```
x = 32;
```

CORRECT

```
if(x == 10){
     do this;
}
```

INCORRECT

```
if(x = 10){
     do this;
}
```

Other ways to compare...

Operator	Meaning	Example
<	“less than”	if (x < 10) { //do something}
<=	“less than or equal to”	if (x >= 15) { //do something}
>	“greater than”	if (x > 3) { //do something}
>=	“greater than or equal to”	if (x >= 7) { //do something}
!=	“not equal to”	if (x != 100) { //do something}

```
grade = 86;

if(grade >= 90){
    //“Your grade is an A”;
}
else if (grade >= 80) {
    //“Your grade is a B”;
}
else if (grade >= 70) {
    //“Your grade is a C”;
}
else if (grade >= 60) {
    //“Your grade is a D”;
}
else {
    //“Fail”;
}
```

```
grade = 98;

if(grade >= 60){
    //“Your grade is an D”;
}
else if (grade > 70) {
    //“Your grade is a C”;
}
else if (grade > 80) {
    //“Your grade is a B”;
}
else if (grade > 90) {
    //“Your grade is a A”;
}
else {
    //“You are a failure”;
}
```

```
grade = 98;

if(grade >= 60 && grade < 69){
      //“Your grade is an D”;
}
else if (grade >= 70 && grade < 79) {
      //“Your grade is a C”;
}
else if (grade >= 80 && grade < 89) {
      //“Your grade is a B”;
}
else if (grade >= 90 && grade <=100) {
      //“Your grade is a A”;
}
else {
      //“You are a failure”;
}
```

Boolean variables

Can only hold one of two possible values:

true

or

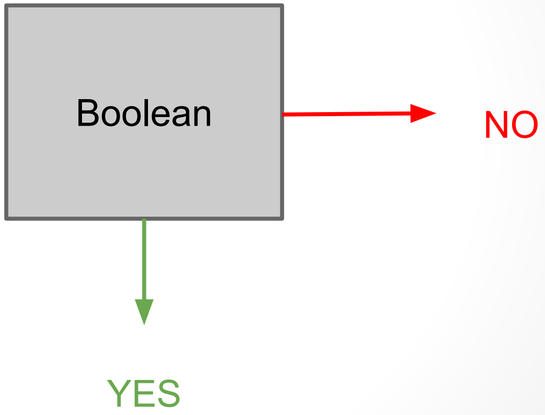
false

```
int x = 10;

boolean isEqualTen = (x == 10);  // check
//boolean isEqualTen = TRUE;           // value

if ( isEqualTen ) {
     //do this
}
```

```
boolean isEqualTen = (x == 10);
boolean b = true;

if ( isEqualTen && b ){
     //do this
}

// same as...
if (isEqualTen == true && b == true) {
     //do this
}
```

```
boolean b = true;
b = !b;

println(b);
```

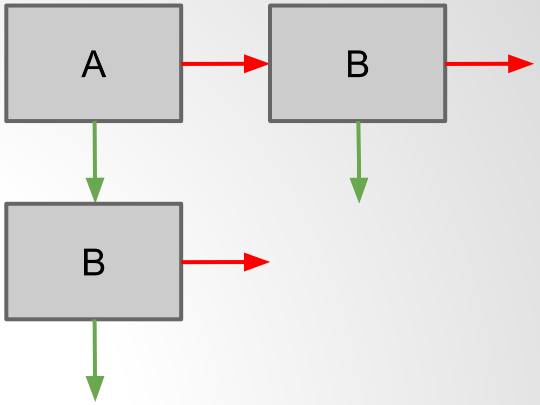
```
boolean a;
boolean b;

void setup() {

 if (a == true) {
        if (b == true){
            //    a && b
        } else {
            // a && !b
        }
 } else {
        if (b == true){
            //    !a && b
        } else {
            // !a && !b
        }
 }
}
```



Boolean

NO

YES



```
boolean a;
boolean b;

void setup() {

  if (a) {
        if (b){
              //     1: a && b
        } else {
              // 2: a && !b
        }
  } else {
        if (b){
              //     3: !a && b
        } else {
              // 4: !a && !b
        }
  }
}
```

Homework

CAPTURE A PROCEDURAL IMAGE!

Ideas:

- Parameterize your previous HW to make it interactive and/or animated
- Use variables, operators and conditionals to change the sketch in some way over time
- Try using system variables too

Try not to plan out the result - instead let the look happen naturally with exploration and screenshot something that you think is cool!