Intro to Programming

Computers are *dumb*.

We have to give them instructions to make them do things.

We do this by...

- Storing data in variables (or constants)
- Writing functions that complete specific tasks
- Using logic to make decisions
- Using loops to repeat tasks over and over

What is a variable?

Everything in the world around us has attributes we can identify.

Example

This whiteboard has...

- a width and height
- a color
- no grid lines (but some do)

We can store these attributes or qualities in variables.

Variables are like storage containers.

Example:

```
width = 24                # inches (integer or float)
height = 36               # inches (integer or float)
color = “white”           # (string)
grid_lines = false        # (boolean)
```

These (values) are called data types.

We can use data types and variables to pass information around our programs more easily.

We have variables down, so let’s get into functions!

We write a function to complete one specific task.

Example: If my job is to say "hello"

```
function instructor() {
  // say "hello"
}
```

Ignore syntax, this is pseudo code!

We can also pass information into functions using variables.

We can make this function say hello to a specific person.

```
function instructor(person) {
  // say "hello" to person
}

function instructor("Dylan") {
  // say "hello" to "Dylan"
}
```

The "person" is passed into the function as a parameter and then used in the sentence.

A real world example of function

If my function is to throw whatever someone passes to me, it might look something like this

```
function instructor(thing) {
  // throw thing
}
```

Demonstration! Pass me something.

Now let’s get into logic and making decisions

Say I want to write some logic to determine whether or not I should eat.

A logic statement will equate to a boolean value

`true` or `false`

```
if (instructor == “hungry”) {
  // Eat food
} else {
 // Don’t eat food
}
```

If I’m hungry, “logically” I should eat some food, if not, then I don’t.

We *all* know that sometimes we eat when we’re not hungry, but bored!

```
if (instructor == “hungry”) || (instructor == “bored”) {
  // Eat food
} else {
  // Don’t eat food
}
```

What happened there?

We used some operators to help our logical statement make a decision.

There are two different types of operators

- Comparison operators
- Logical operators

Comparison Operators

```
>   # greater than
<   # less than
==  # equal to (2 equal signs)
!=  # not equal to
>=  # greater than OR equal to
<=  # less than OR equal to
```

`!` *is called a "bang" and means "not"*

Logical Operators

```
&&  # and
||  # or (double pipe)
```

Let's look at our logical statement again

```
if (instructor == “hungry”) || (instructor == “bored”) {
  // Eat food
} else {
  // Don’t eat food
}
```

If "hungry" OR (double pipe) "bored", eat.

Combine operators to find the boolean value of a statement

```
true && true     # true
true && false    # false
true || false    # true
true || true     # true
false || false   # false
false && false   # false
```

Now let’s get into loops!

Repeat a certain task over and over

- "while” a certain condition is true
- “until” a certain condition is true
- for a designated number of times

Examples

```ruby
while students == clapping do
  # jumping jacks
end

until instructor == tired do
  # jumping jacks
end

10.times do
  # jumping jack
end
```

That's it!

All programs use these basic principles to give computers instructions.

Each *programming language* will have slightly different syntax, but do similar things!