Name:
Date:

// Basic Operators

Often when you are programming you will need to do simple (and sometimes not so simple) mathematical operations. The signs used to do this vary from very simple to confusing if you've never seen them before. Below is a table of definitions as well as some examples:

Arithmetic Operators

Arithmetic operators are your standard mathematical signs

+ (addition)
- (subtraction)
* (multiplication)
/ (division)
% (modulus)
= (assignment)

Relational Operators

Relational operators are used to compare values and variables

== (equality)
!= (inequality)
> (greater-than)
< (less-than)
>= (greater than or equal to)
<= (less than or equal to)

Pay attention to = and ==.
= is used to assign variable values,
== to compare values.

Logical Operators

Logical operators are used to join two or more conditional statements together

! (NOT)
&& (AND)
|| (OR)

Relational Operator Example

```
if (x!=7){
//loop body code here
}
```

Compares x to the number 7,
executes code inside body loop
if the value of x **does not equal 7**

Logical Operator Example:

```
if ((x==7)||(x==9)){
//loop body code here
}
```

Compares x to the number 7 and 9,
executes code inside body loop
if the value of x equals 7 **or** 9

Name:
Date:

// Comments

As you use code other people have written you will notice //, /* and */ symbols. These are used to “comment” lines out so they do not affect the code. This way people who write code can add comments to help you understand what the code does. Good code has comments that explain what each block of code (functions, classes, etc.) does but does not explain simpler portions of the code as this would be a waste of time. Commenting lines out is also a very useful tool when you are writing code yourself. If you have a section of code you are working on, but isn’t quite finished or doesn’t work, you can comment it out so it does not effect the rest of your code when you compile or upload it.

//

This is used to comment out a single line

```
//commented out line
```

/*

This is used to start a section of commented lines

```
/*comments start here
```

*/

This is used to end or close a section of commented lines

```
comments end here*/
```

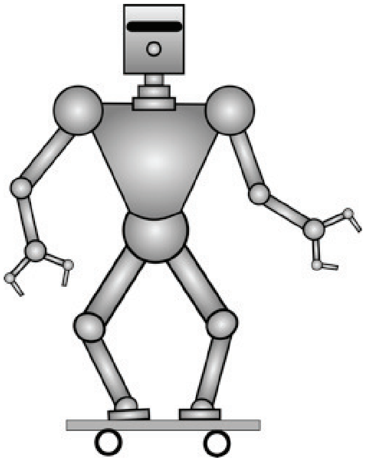
Name:
Date:

// Vocabulary: Variable, Boolean, Integer, Character, Value

Variables are one of the most important concepts in computer programming. But what exactly are **variables**? **Variables** are like baskets that hold pieces of information. There are a couple different kinds of **variables** depending on what kind of information you need to keep track of. You have probably already heard of most of the different kinds of **variables**. Here are the definitions of three different kinds of **variables**. There are more types of variables, but, let's start with these.

- **Boolean variable:** A boolean variable can be true or false (one or zero).
- **Integer variable:** An integer variable can be any whole number between –32768 and 32767.
- **Character variable:** A character variable can be any one letter (or punctuation or symbol).

Below is a robot, answer the questions to the right of the robot and be as silly as you want. Then write the type of **variable** you would use to store this information. For a **boolean** write "boolean", for an **integer** write "int" and for a **character** write "char".



Is this robot good at skateboarding? ___________________ Variable type:______________

How old is this robot?_______________________________ Variable type:______________

What is the first letter of this robot's name?_____________ Variable type:______________

How many years has it been skateboarding?____________ Variable type:______________

Is it wearing pants? _________________________________ Variable type:______________

What is the first letter of the robot's dog's name? ________ Variable type:______________

Is the robot going to crash? ___________________________ Variable type:______________

How many feet of air has this robot gotten? _____________ Variable type:______________

The number, or character, you put into a **variable** is called its **value**. Once you have created a **variable** you can change the **value** whenever you need to. For example, if we decided the robot is 1000 years old, in a year we need to be able to change its age to 1001. First we need to create a **variable** to keep track of its age. We can name the **variable** whatever we want, but "age" makes sense so we'll go with that. Then we need to put a **value** into the **variable**. The first **value** was 1000, but a year later we delete that **value** and replace it with the new **value**, 1001. Pretty easy, huh? If we wanted to keep track of how old the robot used to be when we met it we could create a new **variable** called "ageWeMet". That way when we have to change the "age" **variable** we can keep track of how old the robot was when we met it in the other **variable** "ageWeMet". You may have noticed that there are no spaces in the name of this second **variable**. That is because **variable** names can't have any spaces.

Circle the **variable** in the sentences below and put a box around the **value**.

The robot's favorite letter is Q. The robot's height is 100 ft.

The robot's power is on.

Name:
Date:

// Vocabulary: Boolean, Declare, Assign

OK! You're ready to start programming your first **boolean** variable. Anytime you see *italics like this* it is an example of how you would write something in the Arduino language.

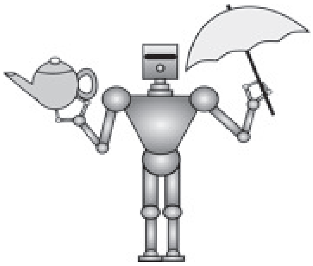
- A **Boolean** variable is the simplest kind of variable, it is either true or false.
- True is represented by a one or HIGH and false is represented by a zero or LOW.
- HIGH can be used as true, but it means there is electricity flowing through a circuit.
- LOW can be used as false, but it means there is no electricity flowing through a circuit.
- To create a **Boolean** variable you type the following: *boolean variableName;*
- Creating a variable is called "**declaring**" a variable.
- The variableName can be anything you like, but it should make sense to you.

For example you could **declare** a **Boolean** variable named *dayLight(boolean dayLight;)* that represents whether it is daytime or not. Once you have **declared** your variable it is not equal to anything, it is empty and waiting for you to set it equal to true or false. To do this you type the following: *dayLight = true;* or *dayLight = 1;*. (Don't forget the ; at the end, it's very important! It is called a semicolon and it tells the computer that you are finished doing something.)

This means that dayLight is true, and you can see the sun. Setting a variable equal to a value is called "**assigning**". **Declare** three **Boolean** variables about the robot on this page in the spaces below and then **assign** them values of true or false (or one or zero). Remember, you can name the variables whatever you want! They're your variables, it's up to you. Look at the example above if you are unsure of how to **declare** and **assign**. (Don't forget the semicolons at the end of each line, they're important!)

Declare:

Assign:



List three of the silliest things you can think of that you might keep track of with a **boolean** variable. Examples: Do I have peanut butter in my ear? Are penguins good to use as dodgeballs?

____

____

____

____

Now pick one of the silly ideas above. In the space below **declare** your silly variable and then **assign** it a value. For example: *boolean peanutButter; peanutButter = true;* This means that I do have peanut butter in my ear... maybe I am saving it for lunch.

____

____

____

____