

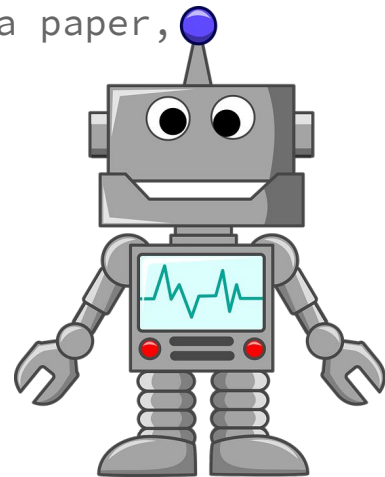
# 2016 GIRLS' CODING CAMP

**Beginner Section**

DAY 1

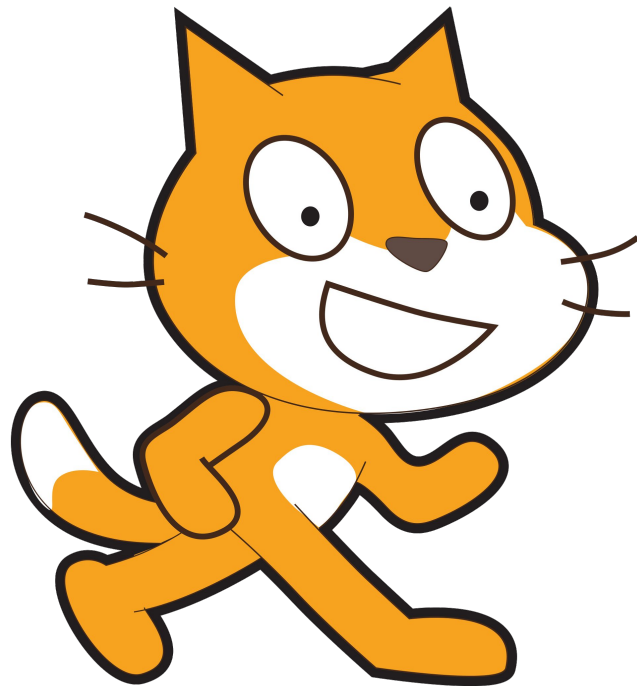
# INTRODUCTION TO PROGRAMMING: JELLY SANDWICH

- Split into groups of at most 5
- Check the supplies given
- Task: Program a “robot” to make a jelly sandwich
  - Provide step-by-step instructions on how to make the jelly sandwich given the available supplies
  - Have ONE person in your group write the instructions on a paper, write the names of the people in your group on the paper
  - Once you’re done, turn in your paper to the front
  - The “robot” will execute your “programs”



# SCRATCH

- Go to <https://scratch.mit.edu/help/videos/> and try out the tutorials
- Let us know when you finish!



# YOUR 1ST PROGRAM: HELLO WORLD

- Create new Project folder
- File -> New -> Java Class (Name it HelloWorld)
- Main function: tells computer where to start running
  - `public static void main(String[] args)`

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

# DATA TYPES

- `int` - integers (... , -2, -1, 0, 1, 2, ...)
- `boolean` - `true/false`
- `double` - decimal numbers (Ex: 3.14, 2.5678)
- `char` - character ('a', '\$', '5')
- `String` - words/sentences ("Hello", "How are you today")
  - You "string" together "chars" to create words and sentences

\* IMPORTANT: `char` - single quote, `String` - double quotes

# PROGRAM: MATH FUN1 & MATHFUN2

- Write a program that adds 7 and 9
  - Store 7 in a variable
  - Store 9 in a variable
  - Store the result in a different variable and print out answer
- Write a program that divides 46 by 4
  - Hint: follow similar instructions as last program
  - Remember: 4 doesn't divide 46 evenly, so what type should the numbers be?

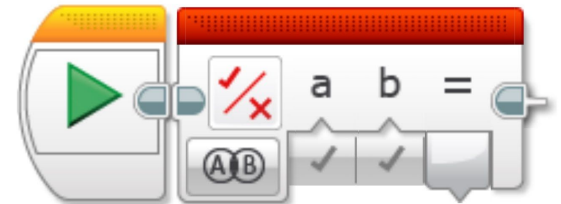
# LOGIC

```
if (argument)
{
    Code to run
}
```

- Can also have an else (otherwise) case and/or multiple if cases

```
if(argument1)
{
    Code to run
}
else if (argument2)
{
    Code to run
}
...
else
{
    Code to run
}
```

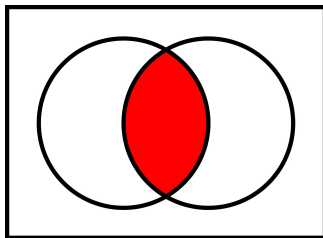
## Logic Operations



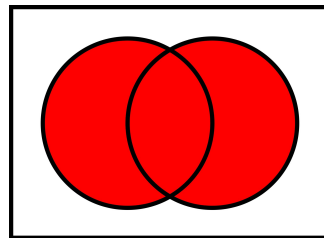


# LOGIC ARGUMENTS

- Check if variable 'a' is equal to a 'value'
  - `a == value`
  - Remember: '`a = value`' assigns value to variable a, '`a == value`' checks equality
- `if (argument1 && argument2)`
  - Both argument1 AND argument2 need to be true to run code in this case
- `if (argument1 || argument2)`
  - Either argument1 OR argument2 need to be true to run code in this case

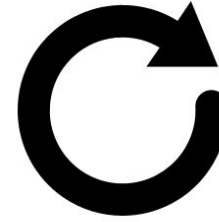


`&&`



`||`

# LOOPS



LOOPS REPEAT  
ACTIONS...  
SO YOU DON'T HAVE TO

- What if you wanted to run the same code some specific number of times OR until some argument is true/false?

```
for (int i = 0; i < count; i++)  
{  
    Code to run  
}
```

Runs “Code to run” ‘count’ number of times

```
while (argument1)  
{  
    Code to run  
}
```

Runs “Code to run” as long as argument1 is TRUE

```
do  
{  
    Code to run  
} while (argument1)
```

Runs “Code to run” at least once, as long as argument1 is TRUE (runs code then checks argument)

