Structured Programming

101 Level

Presented by Ezra Christensen of Tarrant Makers
09/14/2013

Programming 101

- Introduction
- Programming a Halloween Robot
  - Syntax (aka structure)
  - Variables
  - Functions
  - If / then … else statements
  - For Loops, While loops
  - Showing information
- Review
- Appendix

Introduction

Processing Environment

https://processing.org/download/

Why Processing?

- The *Arduino* development environment is based on Processing.
- The programming concepts and principles used in Processing Environment are similar to many other programming languages.

Our Halloween Robot

Our Halloween Robot

- Our skeleton is missing its brain.
- We need to program it in order to enjoy the holiday.
- For this course, we'll assume that our robot has some working sensor inputs and controllable outputs.

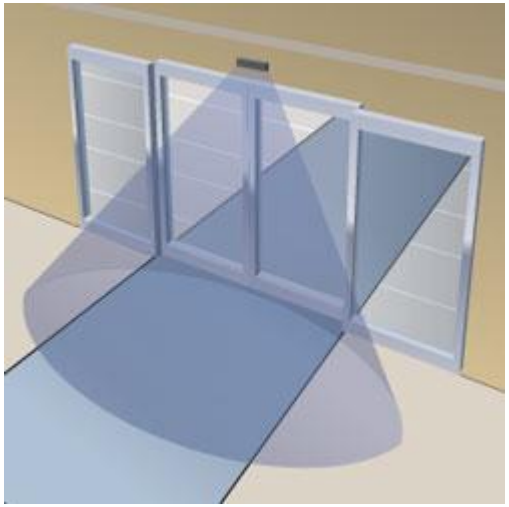
Making Things Happen

Here is what we want the robot to do:

If someone steps onto our porch,

it Activates the Skeleton!

What do we need to know first?



Motion Sensor - like an automatic door uses.

Making Things Happen

Variables and Functions

```
int today = day();
```

Variables

- store information that can be referenced later
- can be of different *Data Types*

Functions

- call entire blocks of code
- built-in functions, libraries of function, or we can build them
- functions may or may not return data

Main Data Types

Name	Code	Description and examples
Integer	int	Whole numbers (no decimals), positive or negative. E.g., 4, -20, 128
Floating number	float	Decimal numbers, positive or negative. E.g., 5.730, -2.5
Text	String	Text of words, sentences, paragraphs. Must be enclosed in quotes. E.g., “Hello”, “What is your name?”
True / False	boolean	Only has two states: true or false.

Making Things Happen

Seeing the otherwise invisible

```
int today = day();
print(today);
```

The program won't tell us what it's doing, unless we ask it to, or do something else to make it show us.

- print() function

<u>Parameters</u>

- give functions a value(s) for them to use

Making Decisions

We know what day it is, what do we do now? (in English)

October 2013

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31 		

Making Things Happen

If / Then statement

```
int today = day();
if (today == 31) {
  print("Here is some candy!");
}
```

- If statements will only execute code only IF certain condition(s) are met.
- == logical comparison for "is equal to"
- { } used to begin and end of a group of code.

Making Things Happen

If / Then … Else statement

```
int today = day();
if (today == 31) {
  print(“Here is some candy!”);
}
else {
  print(“Boo!”);
}
```

- The ELSE block will run IF the statement(s) is False

When will things happen?

We need to control when our code will run.

If we leave it out in the open, it will run away immediately.

When do we want our code to run?

Making Things Happen

Functions of our own

```
void personDetected() {
    if (day() == 31) {
      print(“Here is some candy!”);
    }
    else {
      print(“Boo!”);
    }
}
```

Making Things Happen

Firing our function

```
void setup() {
  //the setup() function runs automatically
  //when you start your program
  personDetected();
}
```

- We call the function we defined by using its name, the same way we called the built-in day() function.

Making Things Happen

Doing things repeatedly

```
void giveCandy() {
  println("Candy #1 given.");
  println("Candy #2 given.");
  println("Candy #3 given.");
  println("Candy #4 given.");
}
```

- This might work sometimes but gives us little control and wouldn't work well if we didn't know the exact number when we coded.

Making Things Happen

Repeating actions: For loop

```
void giveCandy(int num) {
  for (int i = 1; i <= num; i = i+1) {
    println("Candy #" + i + " given.");
  }
}
```

- Let's examine the parts.
  - What we can recognize
  - New parts

Making Things Happen

Repeating actions: For loop

```
void giveCandy(int num) {
  for (int i = 1; i <= num; i = i+1) {
    println("Candy #" + i + " given.");
  }
}
```

- Function declaration
- { } Start and End blocks of code

Making Things Happen

Repeating actions: For loop

```
void giveCandy(int num) {
  for (int i = 1; i <= num; i = i+1) {
    println("Candy #" + i + " given.");
  }
}
```

- Variable declarations

Making Things Happen

Repeating actions: For loop

```
void giveCandy(int num) {
  for (int i = 1; i <= num; i = i+1) {
    println("Candy #" + i + " given.");
  }
}
```

- IF test
  - IF still true, the code block runs again, otherwise it stops

Making Things Happen

Repeating actions: For loop

```
void giveCandy(int num) {
  for (int i = 1; i <= num; i = i+1) {
    println("Candy #" + i + " given.");
  }
}
```

- Afterthought - runs once each time the FOR code is executed.
- Goal is to make the IF eventually false

Making Things Happen

Repeating actions: For loop

```
void giveCandy(int num) {
  for (int i = 1; i <= num; i++) {
    println("Candy #" + i + " given.");
  }
}
```

- Adding strings together.
  - We can use the i variable within the loop

Making Things Happen

Repeating actions: While loop

```
void entertainPerson() {
  while (personStillHere() ) {
    //tellJoke();
    println("Person is still here");
    delay(9000);
  }
}
```

- The code block will continue to loop WHILE the condition is true.

Making Things Happen

Repeating actions: While loop

```
void entertainPerson() {
  while (personStillHere() ) {
    //tellJoke();
    println("Person is still here");
    delay(9000);
  }
}
```

- We want to DELAY between loops, to give people time to leave.

Review

Variables

String a = “Pi”;

float b = 3.14159;

int c = 3;

Review

Functions

```
void personDetected() {
    //lines of code, end each line in a ;
}

void giveCandy(int num) {
    println(“Here are “ + num + “ candy.”);
}
```

Review

If / Then statements

```
if (a > b) {
  //Will run if a is greater than b;
}
else {
  //Will run if a is not greater than b
}
```

Review

For Loops

```
for (int i = 0; i < num; i = i+1) {
  println(“Practice attempt #” + i);
}
```

Review

While Iteration

```
while (peopleClapping()) {
    println("Thank you, and you're welcome.");
    bow();
    delay(5000);
}
```

Appendix

Extra information on

- Functions
- If / then… else statements
- Reference materials

How old are you?

Even on Halloween, should we give EVERYONE the same amount of candy?

Parents and teens?

Little kids?

Making Things Happen

Functions that tell you something

```
float detectHeight() {
  float hght = random(20,96);
  return hght;
}
float personHeight = detectHeight();
```

- We can tell a function to RETURN a value
- We can use our function to assign values
- We can pass more than one parameter into a function, separate with commas: see

Making Things Happen

Multiple IFs possible

```
void evaluateFate() {
  float personHeight = detectHeight();
  if (personHeight < 36) {
    giveCandy(4);
  }
  else if (personHeight > 66) {
    print("Aren't you a little old to be trick or treating?");
  }
  else {
     giveCandy(2);
}    }
```

Main Data Types

Name	Code	Description and examples
Integer	int	Whole numbers (no decimals), positive or negative. E.g., 4, -20, 128
Floating number	float	Decimal numbers, positive or negative. E.g., 5.730, -2.5
Text	String	Text of words, sentences, paragraphs. Must be enclosed in quotes. E.g., “Hello”, “What is your name?”
Boolean	boolean	Only has two states: true or false.

Appendix

For more information, you can visit the official Processing reference.

http://www.processing.org/reference/