Creative Code Lessons For Ikamva Youth

By Marion Walton & Lyndon Daniels

Based on Marion Walton's Training Resources at
https://ikamvacodes.wordpress.com/creative-code/
CREATIVE CODE is a Dr Marion Walton initiative

Your First Steps in Programming

When we talk to other people we need to communicate with them in a language that they can understand, as well as ourselves. Computers are no different in this way, because we communicate with computers in languages that are understood both by computers and ourselves.

Computers also have many different languages that we can use to communicate with them such as HTML, JavaScript and Processing to name a few. This is very similar to the way in which many people we know also use different languages to communicate with each other such as Xhosa, English, Zulu and Afrikaans.

Processing is one of the many different languages we use to communicate with a computer, and is also the language we will be learning in this course to create code. You don't need to learn every different type of computer language, just the one's that will help you complete your tasks. These tasks will always have a context such as making a Web Site has the context of Web Development or making a Mobile App has the context of Mobile App Development. As the context changes for development so too will the **Computer Programming Languages** used.

Processing and HTML5 are both Computer Programming Languages, but because they are used in different contexts the languages look and act very differently.

```
void setup() {
  size(640, 360);
}

void draw(){
  text("Hello World", 10, 10, 40, 60);
}
```

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>Hello World</title>
    </head>
    <body>
        <h1>Hello World</h1>
    </body>
</html>
```

The program on the left is coded in Processing and the program on the right is coded in HTML5. Although both programs do the same thing when they are run, which is, print the words "Hello World" to the screen, the way in which the same result is achieved differentiates substantially from one computer programming language to another because of the language's context.

The NameTag App

The task of creating code is known as **programming** and it involves creating a list of instructions that tell a computer what we would like it to do. These instructions are called a Computer Program or just simply a **Program** and will eventually become an application more often referred to as an **app**, just like the apps you have installed on your phone such as games, web browsers and text editors to name a few of the many different types of apps.

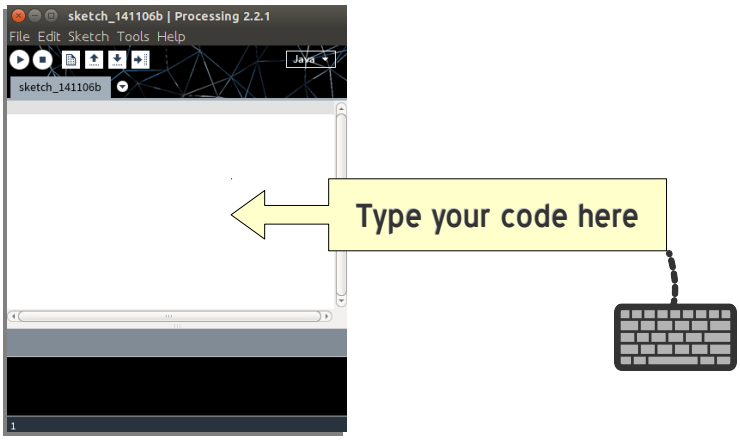
We've all used apps before either on our phones or on a computer, so lets have a look at how easy it is to create our own simple app in Processing that will allow us to make a nametag.

Although we will be making an app we will still need software and a computer or a phone to help us do this. Processing is a special type of application that helps us make our own apps.

Processing should already be installed on your computers, if not go to processing.org to download and install it. Ask your teacher for help if you're not sure how to do this.

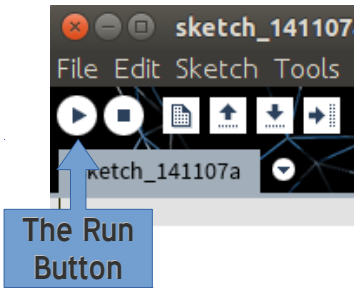
Lets Start Processing

Launch the Processing application and you'll notice that it immediately allows you to start typing in the white text area. This is the area that you will add your code to.



Exercise 02

Create Your Own Name Tag Art with the NameTag App



Below you will find the code for the NameTag app. All you need to do is copy and paste the code into Processing, click the run button and start creating your custom nametag.
Try pressing the b, r, w and s keys for variations. Then have look at the code and try changing the numbers where you see the fill() command. What happens when you run the program with the changes?

```
void setup() {

	size(480, 120);
	smooth();
	background(232,28,28);
}
void draw() {
	if (mousePressed) {
		fill(7,124,20);
	}else{
	fill(255);
	}

	if ((key == 'b') || (key == 'B')) {
		fill(0);
	}
	if ((key == 'r') || (key == 'R')) {
		fill(255,0,0);
	}
	if ((key == 'w') || (key == 'W')) {
		fill(255);
	}

	ellipse(mouseX, mouseY, 20, 20);
}

void keyPressed() {
	if ((key == 's') || (key == 'S')) {
		selectOutput("Select a file to write to:", "fileSelected");

	}
}

void fileSelected(File selection) {
	if (selection == null) {
		println("Window was closed or the user hit cancel.");
	}else{
		println("User selected " + selection.getAbsolutePath());
		save(selection.getAbsolutePath());
	}
}
```