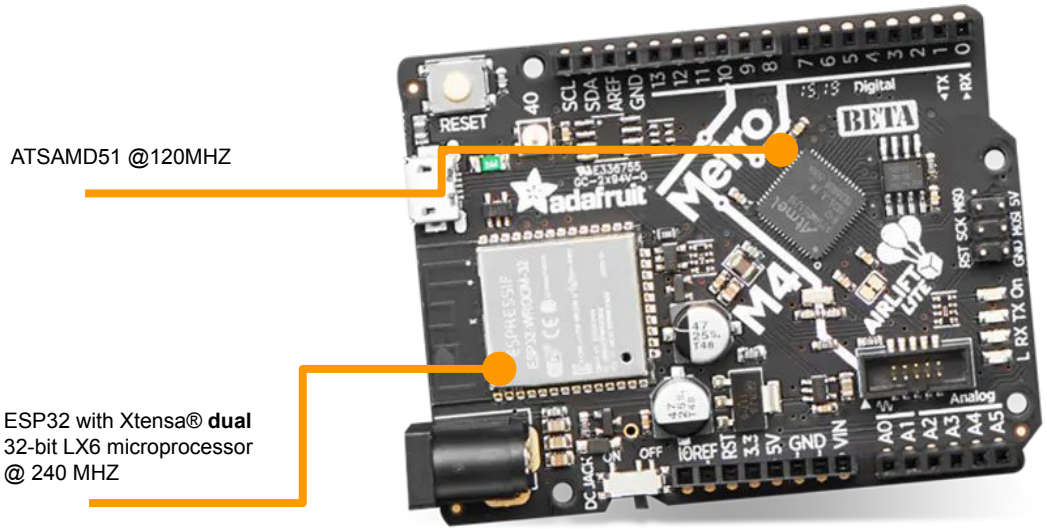
bits and Bytes

Our **Metro M4 Airlift Lite** has **three 32 bit processors!**

This is important because it means we store information in 32 bit (4-byte) blocks.

When we come back… C++



Arduino and C++

Arduino is our chosen programming language and it is based on C++

The Arduino code reference will tell you all about the language.
It is like a dictionary for computer code with three main sections:

Variables:

https://www.arduino.cc/reference/en/#variables

Functions:

https://www.arduino.cc/reference/en/#functions

Structure:

https://www.arduino.cc/reference/en/#structure

W3Schools.com has a well organized and easy to follow C++ reference as well!

https://www.w3schools.com/cpp/default.asp

We can use any part of the Arduino core language and almost all of the C++ language it is based on.

Arduino and C++

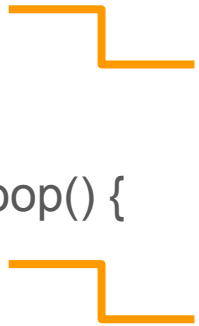
Basic structure and bareMinimum

```
void setup(){

}

void loop() {

}
```



This is **whitespace** or empty space.

It can be made of **spaces** typed on the space bar line enters or **tabs** with the tab key.

C++ is not affected by whitespace. Your code can have as much or as little as you want.

C++ uses several types of brackets to group code in different ways

( ) - **Parenthesis** - used to group things like in math, usually follow conditionals like if() and while() to contain the conditions and also to group the arguments passed to a function.

[ ] - **Brackets** - used to contain members of an array and to declare an array

{ } - **Curly Braces** - used to group the commands that follow if() and while() conditionals and those that make up a function and also to group the variables in a struct() data type

Arduino and C++

Basic structure and bareMinimum

```
void setup(){

}

void loop() {

}
```

This is **whitespace** or empty space.

It can be made of **spaces** typed on the space bar line enters or **tabs** with the tab key.

C++ is not affected by whitespace. Your code can have as much or as little as you want.

Since C++ does not pay attention to whitespace it is important to use punctuation so the compiler can read what you are writing.

, commas are used to separate lists of things

; semicolons are used to **terminate** a line of code

You will spend a lot of time paying attention to brackets, commas, and semicolons while learning C++

Arduino and C++

Variables:

https://www.arduino.cc/reference/en/#variables

We can store different **types** of data.
We will know these as **data types.**
These are common data types:

int	char
long	void
float	String()
bool	array

There are **community standards** for writing **variable names** and **definitions** in the Arduino language

variables are written in all lower case

camelCase (like a camel's hump) is used when we write variable names with more than one word in them like this: **arnieMartin**

ALL_CAPS_WITH_UNDERSCORES is used to give names to #define declarations like this: #define LED_PIN 13

The standards help avoid overlap between names of known variables and functions and make it easier for people to read and understand each other's code.

Arduino and C++

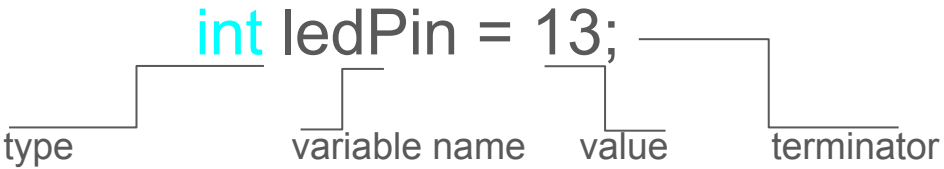
Variables:

https://www.arduino.cc/reference/en/#variables

We can **declare** a **variable** by using its **type** and give it a **name** and **value** like this:

Declare only

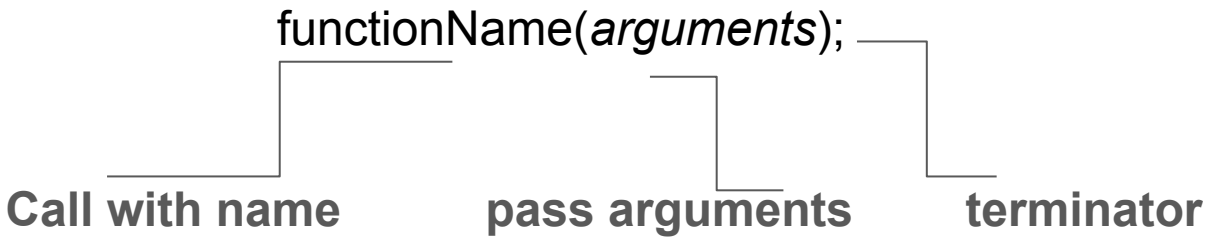
Declare with value



functions();

Functions in C++ are like pre-packaged code that does specific... well functions!
https://www.arduino.cc/reference/en/#functions

When we wrote code to blink an LED we used a function called pinMode() and another called digialWrite(). To **call** a function we simply type its name followed by a pair of parentheses and a **terminator**



Structure

Structure:
https://www.arduino.cc/reference/en/#structure

Structure is how we control the flow of code, what code executes and when.

C++ uses **conditional statements** to control the flow of code.

Some common conditionals are:

if(*something is true or false*){

}

while(*something is true or false*){

}