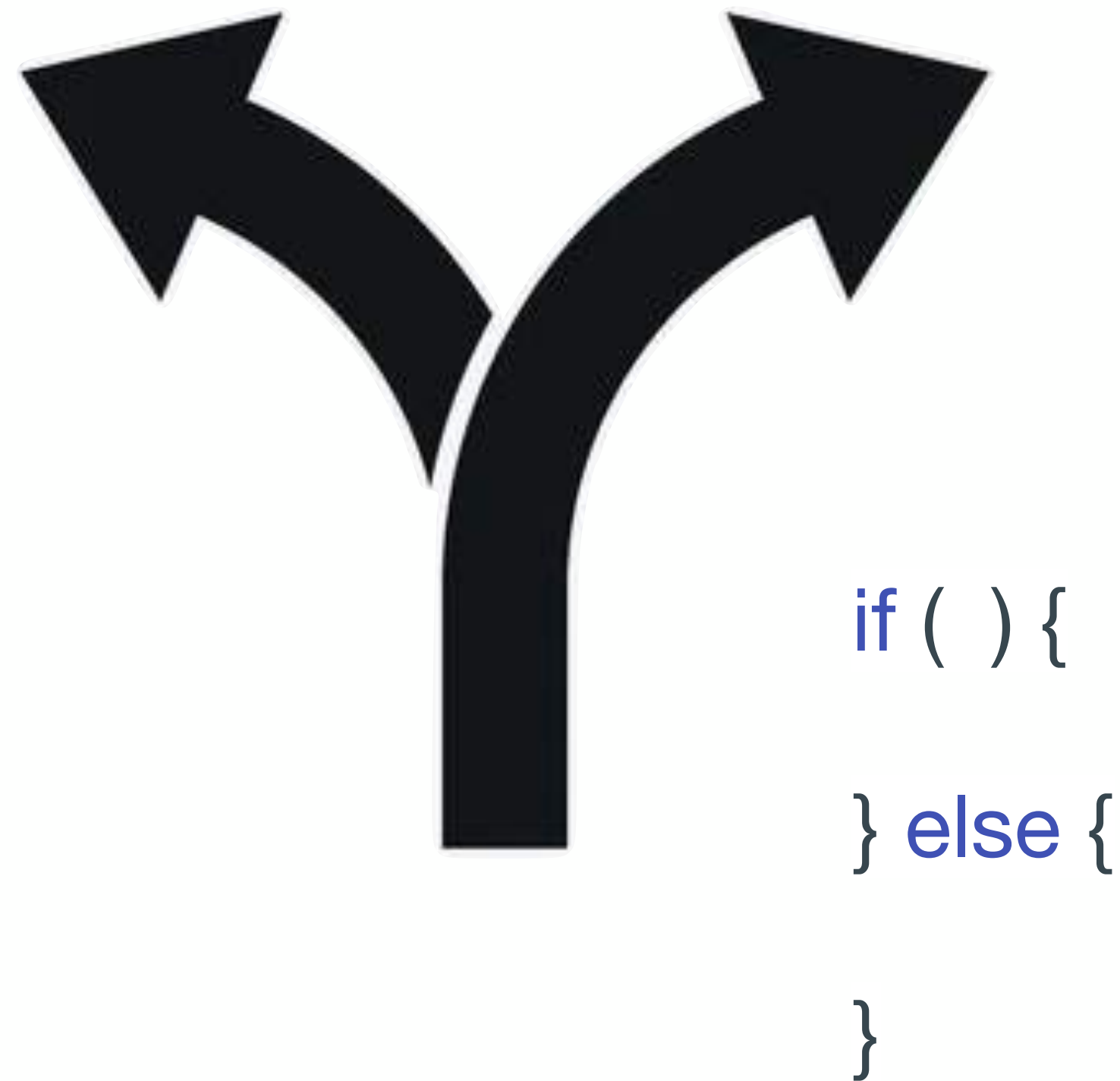


# Electronics Engineering Eca

Block C Lesson 6

# Conditional logic



- Make decisions
- respond to different situations
- behave differently based on what's happening
- Without it, the program would just follow the same instructions all the time, no matter what.

# Conditional logic

```
int ledPin = 9;    // LED connected to pin 9
int buttonPin = 7; // Button connected to pin 7

void setup() {
  pinMode(ledPin, OUTPUT); // Set LED pin as output
  pinMode(buttonPin, INPUT); // Set button pin as input
}

void loop() {
  int buttonState = digitalRead(buttonPin); // Read the button's state

  if (buttonState == 1) { // If buttonState equals 1 (HIGH)
    digitalWrite(ledPin, HIGH); // Turn on the LED
  } else { // Otherwise (buttonState is 0 or LOW)
    digitalWrite(ledPin, LOW); // Turn off the LED
  }
}
```

# Conditional logic

```
int buttonState = digitalRead(buttonPin);
```

- digitalRead is a function that returns a integer variable based on the physical pin
- This reads whether the button is pressed (HIGH, represented as 1) or not pressed (LOW, represented as 0).
- The value is stored in the buttonState variable.

# Types of variables

```
int ledPin = 9; // Pin connected to LED
```

```
float temperature = 23.56;
```

```
String message = "Arduino is running!";
```

```
bool ledState = false;
```

Data Type	Size	Range	Description	Example
void	N/A	N/A	Indicates no return value (used in functions).	void setup() {}
bool	1 byte	true or false	Stores a true/false value.	bool isOn = true;
char	1 byte	-128 to 127	Stores a single character (ASCII).	char letter = 'A';
unsigned char	1 byte	0 to 255	Stores positive-only single-byte values.	unsigned char value = 255;
byte	1 byte	0 to 255	Same as unsigned char, often used for pins.	byte ledPin = 13;
int	2 bytes	-32,768 to 32,767	Stores whole numbers.	int count = 100;
unsigned int	2 bytes	0 to 65,535	Stores positive-only whole numbers.	unsigned int count = 200;
word	2 bytes	0 to 65,535	Alias for unsigned int, often for time values.	word timer = 1000;
long	4 bytes	-2,147,483,648 to 2,147,483,647	Stores large whole numbers.	long bigValue = 1000000;
unsigned long	4 bytes	0 to 4,294,967,295	Stores large positive-only whole numbers.	unsigned long time = 5000;

# Types of variables

```
int ledPin = 9; // Pin connected to LED
```

```
float temperature = 23.56;
```

```
String message = "Arduino is running!";
```

```
bool ledState = false;
```

<b>int</b>	2 bytes	-32,768 to 32,767	Stores whole numbers.	int count = 100;
<b>unsigned int</b>	2 bytes	0 to 65,535	Stores positive-only whole numbers.	unsigned int count = 200;
<b>word</b>	2 bytes	0 to 65,535	Alias for unsigned int, often for time values.	word timer = 1000;
<b>long</b>	4 bytes	-2,147,483,648 to 2,147,483,647	Stores large whole numbers.	long bigValue = 1000000;
<b>unsigned long</b>	4 bytes	0 to 4,294,967,295	Stores large positive-only whole numbers.	unsigned long time = 5000;
<b>short</b>	2 bytes	-32,768 to 32,767	A smaller version of int.	short value = 123;
<b>unsigned short</b>	2 bytes	0 to 65,535	Stores positive-only small numbers.	unsigned short num = 50;
<b>float</b>	4 bytes	±3.4028235E+38 (6-7 decimal places)	Stores decimal values.	float pi = 3.14;
<b>double</b>	4 bytes (8 on some platforms)	Same as float (on most boards).	Stores higher-precision decimal values.	double voltage = 5.25;
<b>size_t</b>	2 or 4 bytes	Platform-dependent	Used for sizes or counts, e.g., array lengths.	size_t length = 50;
<b>String</b>	Variable	Depends on content	Stores sequences of characters (text).	String name = "Hello";

# Operators

Operator	Meaning	Example	Explanation
==	Equal to	if (x == 10)	Checks if x is equal to 10.
!=	Not equal to	if (x != 10)	Checks if x is not equal to 10.
>	Greater than	if (x > 10)	Checks if x is greater than 10.
<	Less than	if (x < 10)	Checks if x is less than 10.
>=	Greater than or equal to	if (x >= 10)	Checks if x is greater than or equal to 10.
<=	Less than or equal to	if (x <= 10)	Checks if x is less than or equal to 10.
&&	Logical AND (both conditions must be true)	if (x > 5 && x < 15)	Checks if x is greater than 5 <b>and</b> less than 15.
	Logical OR (at least one condition must be true)	if (x > 5    x < 15)	Logical OR (at least one condition true)

# Functions

Type of return value

Function name

Input parameters

```
type functionName(type parameter1, type parameter2) {  
    // Code using parameters  
}
```





# Functions

Example:

```
int addNumbers(int a, int b) {  
    return a + b;  
}
```

Implementation:

```
int result = addNumbers(5, 3); // Call the function with 5 and 3 as arguments  
  
Serial.print("The sum is: ");  
Serial.println(result); // Display the result on the Serial Monitor
```

# Functions

Bool return type:

```
bool functionName() {  
    // Code to evaluate a condition  
    return condition; // return either true or false  
}
```

Example:

```
bool isButtonPressed() {  
    if (digitalRead(buttonPin) == HIGH) {  
        return true; // Button is pressed  
    } else {  
        return false; // Button is not pressed  
    }  
}
```

Implementation:

```
if (isButtonPressed()) {  
    Serial.println("Button is pressed.");  
} else {  
    Serial.println("Button is not pressed.");  
}
```