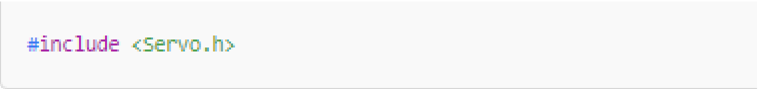
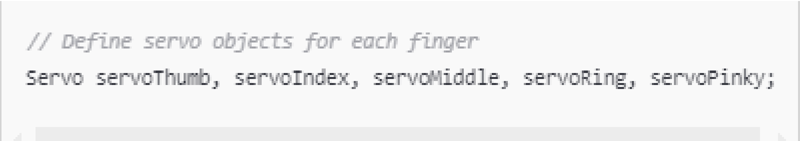
## Header and Servo Object Definition



```
#include <Servo.h>
```

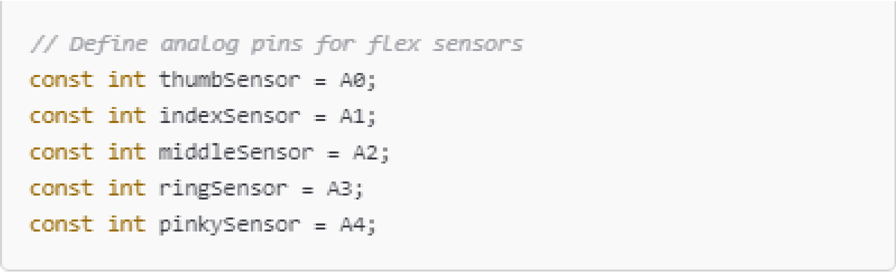
- **Meaning**: This includes the Servo library, allowing our code to control servo motors.
- **Use**: Without this library, we can't use `Servo` class functions like `.attach()` or `.write()` to control the motors.



```
// Define servo objects for each finger
Servo servoThumb, servoIndex, servoMiddle, servoRing, servoPinky;
```

- **Meaning**: Creates five servo objects, one for each finger of the robotic hand.
- **Use**: Each object controls an individual servo motor corresponding to a finger.

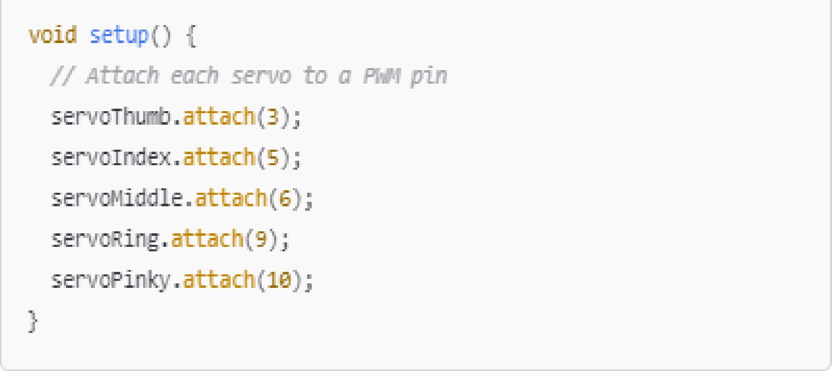
## Analog Pin Assignments for Flex Sensors



```
// Define analog pins for flex sensors
const int thumbSensor = A0;
const int indexSensor = A1;
const int middleSensor = A2;
const int ringSensor = A3;
const int pinkySensor = A4;
```

- **Meaning**: Assigns specific analog pins (`A0` to `A4`) to the flex sensors for each finger.
- **Use**: These pins will read the analog signals from the flex sensors, which change based on how much the corresponding finger is bent.

## Setup Function

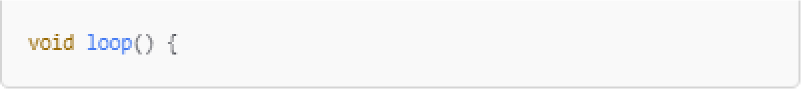


```
void setup() {
  // Attach each servo to a PWM pin
  servoThumb.attach(3);
  servoIndex.attach(5);
  servoMiddle.attach(6);
  servoRing.attach(9);
  servoPinky.attach(10);
}
```

- **Meaning**: The `setup()` function initializes the servo motors by attaching them to specific PWM pins (Pulse Width Modulation pins).
- **Use**:

  - `servoThumb.attach(3)` links the thumb servo motor to pin 3 of the Arduino.
  - Similarly, other fingers are linked to PWM pins 5, 6, 9, and 10.
  - Servos require PWM signals to adjust their position accurately.

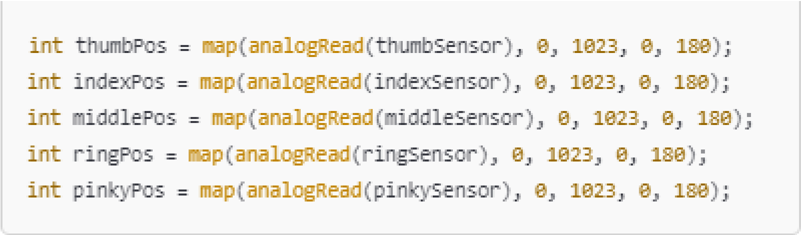
## Loop Function



```
void loop() {
```

- **Meaning**: The `loop()` function runs continuously, reading sensor data and moving the servos based on that data.
- **Use**: This is where the robotic hand continuously updates the finger positions.

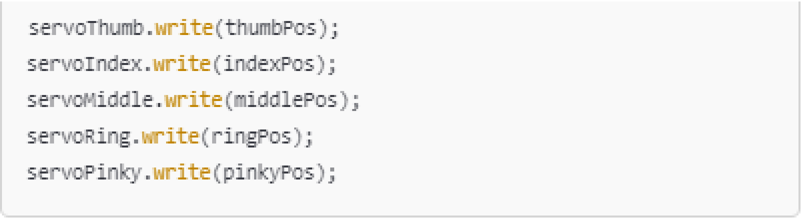
## Reading Flex Sensor Values



```
int thumbPos = map(analogRead(thumbSensor), 0, 1023, 0, 180);
int indexPos = map(analogRead(indexSensor), 0, 1023, 0, 180);
int middlePos = map(analogRead(middleSensor), 0, 1023, 0, 180);
int ringPos = map(analogRead(ringSensor), 0, 1023, 0, 180);
int pinkyPos = map(analogRead(pinkySensor), 0, 1023, 0, 180);
```

- **Meaning**: Reads the analog values from the flex sensors and maps them to servo angles (0° to 180°).
  - `analogRead(thumbSensor)` reads the bending level of the thumb's flex sensor as a value between 0 and 1023.
  - `map(x, 0, 1023, 0, 180)` converts this value to a servo angle between 0° (completely open) and 180° (completely closed).
- **Use**: This mapping ensures that the servo position corresponds to the degree of finger bending detected by the flex sensors.

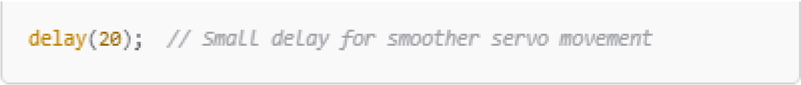
## Controlling Servo Motors



```
servoThumb.write(thumbPos);
servoIndex.write(indexPos);
servoMiddle.write(middlePos);
servoRing.write(ringPos);
servoPinky.write(pinkyPos);
```

- **Meaning**: Sends the mapped angle values to each servo motor to adjust their positions.
- **Use**:
  - `servoThumb.write(thumbPos)` moves the thumb servo to the position dictated by the flex sensor's reading.
  - Other fingers are similarly controlled.

## Adding a Delay



```
delay(20);  // Small delay for smoother servo movement
```

- **Meaning**: Introduces a delay of 20 milliseconds in each loop iteration.
- **Use**:
  - Prevents abrupt movements by allowing the servos to transition smoothly between positions.
  - Reduces excessive CPU usage by creating a pause before the next loop.