# Serial.print()

Prints data to the serial port as human-readable ASCII text. This command can take many forms. Numbers are printed using an ASCII character for each digit. Floats are similarly printed as ASCII digits, defaulting to two decimal places. Bytes are sent as a single character. Characters and strings are sent as is. For example-

* *Serial*.print(78) gives "78"
* *Serial*.print(1.23456) gives "1.23"
* *Serial*.print('N') gives "N"
* *Serial*.print("Hello world.") gives "Hello world."

An optional second parameter specifies the base (format) to use; permitted values are BIN(binary, or base 2), OCT(octal, or base 8), DEC(decimal, or base 10), HEX(hexadecimal, or base 16). For floating point numbers, this parameter specifies the number of decimal places to use. For example-

* *Serial*.print(78, BIN) gives "1001110"
* *Serial*.print(78, OCT) gives "116"
* *Serial*.print(78, DEC) gives "78"
* *Serial*.print(78, HEX) gives "4E"
* *Serial*.print(1.23456, 0) gives "1"

### Syntax

*Serial*.print(val);  
*Serial*.print(val, format);

*Serial*: serial port object. See the list of available serial ports for each board on the [Serial main page](https://www.arduino.cc/reference/en/language/functions/communication/serial).

val: the value to print. Allowed data types: any data type.

# Serial.println()

Prints data to the serial port as human-readable ASCII text followed by a carriage return character (ASCII 13, or '\r') and a newline character (ASCII 10, or '\n'). This command takes the same forms as [Serial.print()](https://www.arduino.cc/reference/en/language/functions/communication/serial/print).

### Example Code

int analogValue = 0; // variable to hold the analog value

void setup() {

// open the serial port at 9600 bps:

Serial.begin(9600);

}

void loop() {

// read the analog input on pin 0:

analogValue = analogRead(0);

Serial.print(analogValue); // print as an ASCII-encoded decimal

Serial.print(analogValue, DEC); // print as an ASCII-encoded decimal

Serial.println(analogValue, HEX); // print as an ASCII-encoded hexadecimal

Serial.println(analogValue, OCT); // print as an ASCII-encoded octal

Serial.println(analogValue, BIN); // print as an ASCII-encoded binary

// delay 10 milliseconds before the next reading:

delay(10);

}