

Lembre-se ! Veja a documentação de referência da linguagem em:
<https://www.arduino.cc/reference/pt/>

Objetivo – ambientar-se com o monitor serial e com o plotter serial e interagir com a placa/ programa por meio de interações no teclado.

1. Compile o seguinte programa:

```
void setup() {  
    Serial.begin(9600);  
}  
int x = 0;  
void loop() {  
    Serial.print("Numero: ");  
    Serial.println(x);  
    delay(300);  
    x++;  
}
```

Ligue o **Monitor Serial** (Ferramentas/ Monitor Serial) e veja o resultado.

2. Compile o seguinte programa:

```
void setup() {  
    Serial.begin(9600);  
}  
int x = 0;  
void loop() {  
    Serial.print("Numero: ");  
    Serial.println(x);  
    Serial.print(" DEC: ");  
    Serial.print(x);  
    Serial.print(" HEX: ");  
    Serial.print(x, HEX);  
    Serial.print(" BIN: ");  
    Serial.println(x, BIN);  
    delay(300);  
    x++;  
}
```

Ligue o **Monitor Serial** (Ferramentas/ Monitor Serial) e veja o resultado. Note que utilizamos um novo elemento (DEC, HEX, BIN) junto à informação de impressão do valor (print), de forma a obtermos, respectivamente, a visualização nos sistemas Decimal, Hexadecimal e Binário (também foi usado o ‘print’, ao invés de ‘println’, no qual o segundo comando acrescenta uma quebra de linha – ‘\n’ – após a impressão).

3. Compile o seguinte programa:

```
long aleatorio;  
  
void setup() {  
    Serial.begin(9600);  
}
```

```

void loop() {
  Serial.print("\nEntre 0 e 99:");
  aleatorio = random(100);
  Serial.println(aleatorio);
  Serial.print("\nEntre 5 e 14: ");
  aleatorio = random(5, 15);
  Serial.println(aleatorio);
  delay(300);
}

```

Ligue o **Monitor Serial** (Ferramentas/ Monitor Serial) e veja o resultado. O comando ‘\n’ inserido dentro da instrução print faz com que seja pulada uma linha (‘\n = new line’). A impressão será algo como: Entre 0 e 99:7; Entre 5 e 14: 14; Entre 0 e 99:73 ; ... Se você compilar o programa novamente e executá-lo novamente os valores sorteados deverão ser os mesmos. Isto porque o gerador de números aleatórios não foi inicializado. Uma forma de inicializá-lo é por meio do uso da função `randomSeed()`¹ (semente aleatória). Veja:

```

long aleatorio;

void setup() {
  Serial.begin(9600);

  // inicializa o gerador de números aleatórios.
  // um pino analógico desconectado irá retornar um
  // valor aleatório de tensão em analogRead()
  randomSeed(analogRead(0));
}

void loop() {
  Serial.print("\nEntre 0 e 99:");
  aleatorio = random(100);
  Serial.println(aleatorio);
  Serial.print("\nEntre 5 e 14: ");
  aleatorio = random(5, 15);
  Serial.println(aleatorio);
  delay(300);
}

```



Ligue o **Monitor Serial** (Ferramentas/ Monitor Serial) e veja o resultado. Se você compilar o programa novamente e executá-lo novamente os valores sorteados deverão ser diferentes a cada momento. Compare a execução do programa usando / não usando o comando `randomSeed()`.

Na sequência, veremos o uso do plotter, que é outro programa monitor da interface serial, mas capaz de gerar interfaces gráficas.

1 <https://www.arduino.cc/reference/pt/language/functions/random-numbers/randomseed/>

4. Compile o seguinte programa:

```
long aleatorio;
void setup() {
  Serial.begin(9600);
  randomSeed(analogRead(0));
}
void loop() {
  aleatorio = random(50);
  Serial.println(aleatorio);
}
```

Ligue o **Monitor Serial** (Ferramentas/ Monitor Serial) e veja o resultado. Depois, feche a tela. Ligue o **Plotter Serial** (Ferramentas/ Plotter Serial) e veja o resultado.

Vamos criar mais uma variável de exibição. Compile o seguinte programa:

```
long aleatorio1, aleatorio2;
void setup() {
  Serial.begin(9600);
  randomSeed(analogRead(0));
}
void loop() {
  aleatorio1 = random(50);
  Serial.println(aleatorio1);
  aleatorio2 = random(50);
  Serial.println(aleatorio2);
}
```

Ligue o **plotter serial**. Viu alguma diferença? Não são duas variáveis? O plotter simplesmente continuou a mostrar os valores continuamente.

Agora, teste assim: compile o programa que segue.

```
long aleatorio1, aleatorio2;
void setup() {
  Serial.begin(9600);
  randomSeed(analogRead(0));
}
void loop() {
  aleatorio1 = random(50);
  Serial.print(aleatorio1);
  Serial.print(", ");
  aleatorio2 = random(50);
  Serial.print(aleatorio2);
  Serial.print("\n");
}
```

Ligue o **plotter serial** e veja a diferença. O uso do separador (',') entre os valores passados ao comando print (e não é o println), faz com que o plotter interprete a mudança de variável. Para uma nova série de valores, usamos o terminador de nova linha ('\n') – poderia ter sido utilizado Serial.println(aleatorio2) ao invés de primeiro mostrar o valor e depois pular uma linha, o resultado seria a mesma interpretação por parte do plotter serial.

Mas a legenda ficou confusa, não é mesmo? Vamos acertá-la? Compile o programa que segue:

```
*/  
long aleatorio1, aleatorio2;  
void setup() {  
    Serial.begin(9600);  
    randomSeed(analogRead(0));  
}  
void loop() {  
    aleatorio1 = random(50);  
    Serial.print("Aleatorio1: ");  
    Serial.print(aleatorio1);  
    Serial.print(",");  
    Serial.print("Aleatorio2: ");  
    aleatorio2 = random(50);  
    Serial.print(aleatorio2);  
    Serial.print("\n");  
}
```

Ligue o **plotter serial**. Agora a legenda deverá estar ok.

Desafios:

1. Pesquise como ler um valor digitado no teclado pelo monitor serial. Depois, faça:
 - a) mostre o valor digitado utilizando Serial.println;
 - b) use um valor digitado para controlar um led (por exemplo, 1 para o led ligado na saída 8 e 2 para ligar o led na saída 9 – ou use as saídas/ interfaces que quiser)