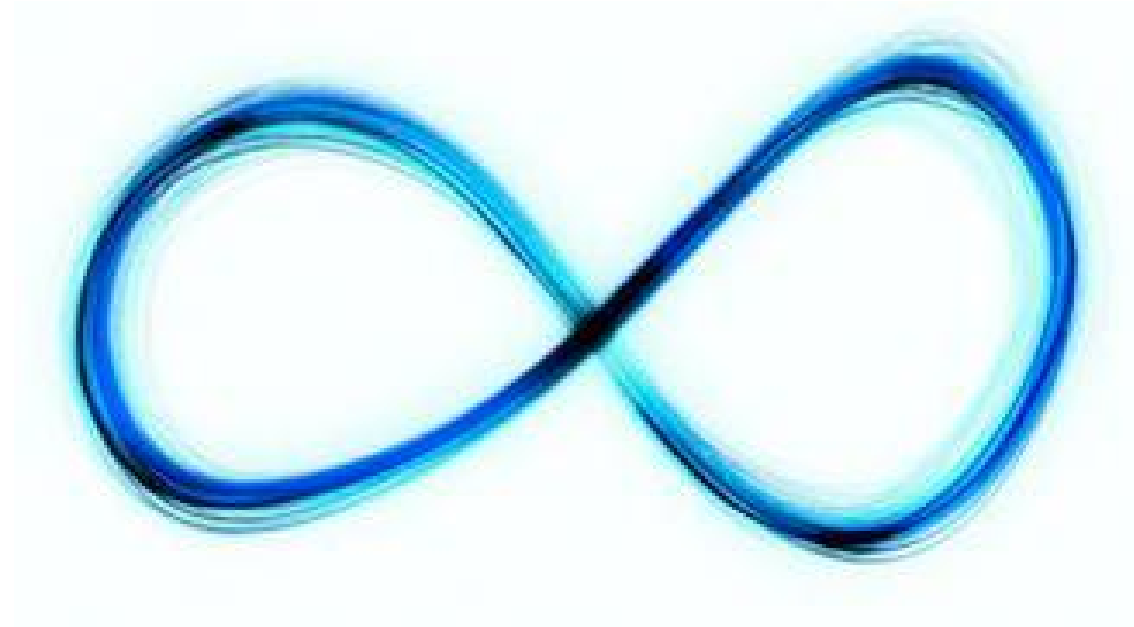


Estruturas de repetição



Escola CDI

Professor: Paulo Trentin

O que são?

Quando queremos executar vários comandos semelhantes, por exemplo ligar e desligar 5 leds, normalmente fazemos um código linha a linha:

```
digitalWrite(2, HIGH); delay(100);  
digitalWrite(2, LOW); delay(100);  
digitalWrite(4, HIGH); delay(100);  
digitalWrite(4, LOW); delay(100);  
digitalWrite(5, HIGH); delay(100);  
digitalWrite(5, LOW); delay(100);  
digitalWrite(6, HIGH); delay(100);  
digitalWrite(6, LOW); delay(100);  
digitalWrite(7, HIGH); delay(100);  
digitalWrite(7, LOW); delay(100);
```

O que são?

E se trocássemos todas as linhas de código anteriores por:

```
int contador = 2;
```

```
while (contador < 8){  
    digitalWrite(contador, HIGH); delay(100);  
    digitalWrite(contador, LOW); delay(100);  
    contador++;  
}
```

Testando...

Crie uma variável inteira chamada contador e inicie ela com o valor 0.

Dentro da função **setup** insira:

```
pinMode(13, OUTPUT);  
int contador = 0;  
while (contador < 20){  
    digitalWrite(13, HIGH); delay(500);  
    digitalWrite(13, LOW); delay(500);  
    contador++;  
}
```

Obs : A função **loop** não terá nada, mas lembre-se: ela é obrigatória

Entendendo o **while**

while (execucao < 20)

Usamos um repetidor chamado while. Lemos o comando acima desta maneira: "**Enquanto** o valor da variável execucao for menor que 20, faça"

Seguimos aqui o mesmo princípio do contador, afinal dentro do while, temos: execucao++; que incrementa o contador com + 1;

O while faz o papel da função **loop**?

Implemente o seguinte código:

```
void setup(){\n  pinMode(13, OUTPUT);\n  while (1){\n    digitalWrite(13, HIGH);\n    delay(1000);\n    digitalWrite(13, LOW);\n    delay(1000);\n  }\n}
```

```
void loop(){}
```

Com qual programa de exemplo ele se parece?

O repetidor **for**

O repetidor **while** é útil para fazermos tarefas que comecem com o "enquanto". Porém muitas tarefas na programação exigem mais detalhes.

Por exemplo, usando o **while** precisamos incrementar o contador dentro do código.

Outro detalhe é que precisamos iniciar o contador antes de usar o repetidor **while**.

Com o repetidor **for**, tudo é feito em uma linha!

O repetidor **for**

Veja o exemplo:

```
for (int contador = 0; contador < 20; contador++){  
    // Aqui o código para repetir 20 vezes  
}
```

O valor da variável contador, pode ser usado dentro do for.

O repetidor **for** - exemplo de uso

Implemente o seguinte código fonte:

```
void setup() {  
    pinMode(13, OUTPUT);  
  
    for (int contador = 0; contador < 20; contador++) {  
        digitalWrite(13, HIGH); delay(500);  
        digitalWrite(13, LOW); delay(500);  
    }  
}  
  
void loop() { }
```

Entendendo o repetidor **for**

Ele é dividido em três sessões:

1. **Início da variável contadora**: Aqui um contador é criado para ser usado dentro do **laço** de repetição **for** (repetidor for).
2. **Condição de teste**: Nesta sessão ele testa se ainda pode executar seu código ou se é hora de parar
3. **Rotina pós execução**: Aqui normalmente é feito um incremento no contador.

Entendendo o repetidor **for**

```
for ( sessão 1 ; sessão 2 ; sessão 3 ){  
  
}
```

O código dentro de cada sessão é facultativo, ou seja, poderíamos ter um for implementado desta maneira:

```
for ( ; ; ){  
  
}
```

Faca o teste e veja o que acontece!

Onde uso o repetidor **for**?

Normalmente quando você deseja percorrer um vetor, ou matriz, você irá usá-lo.

Ele faz o trabalho sujo de repetir uma sequência de códigos de forma finita (ou não, mas aí você terá um looping infinito!).

O que é um looping infinito?

Explicação:

<http://www.tugatronica.com/loop-infinito-para-leigos/>

<http://www.tocadoelfo.com.br/2011/07/looping-infinito-uma-definicao.html>

`for (; ;);` ou `while (1);`

Quando não há uma condição de parada ou ela nunca é alcançada, configura-se o looping infinito.

Exercício

Desenvolva um programa com um laço **for** que faça com que os LEDs 2 à 13 pisquem por 20 milissegundos cada.

Essa mesma rotina deve ser executada 10 vezes.

Lembre-se: a função **pinMode** precisa ser executada apenas uma vez!

Exercício - resposta

```
void setup(){  
    // Define os pinos 2 à 13 como saída  
    for (int referencia = 2; referencia < 14; referencia++){  
        pinMode(referencia, OUTPUT);  
    }  
    // Executa por 10 vezes o código de dentro deste for  
    for (int cont = 0; cont < 10; cont++){  
        // Liga e desliga os LEDs 2 à 13 por 20 ms  
        for (int led = 2; led < 14; led++){  
            digitalWrite(led, HIGH); delay(20);  
            digitalWrite(led, LOW); delay(20);  
        }  
    }  
}
```

```
void loop(){ }
```