

The background of the slide is a light gray gradient, decorated with numerous realistic water droplets of various sizes. Some droplets are in the top left corner, others are scattered along the bottom edge, and a few are on the right side. The droplets have highlights and shadows, giving them a three-dimensional appearance.

# AULA 2 – ESTRUTURAS DE CONDIÇÃO

PROFESSOR LUIZ RICARDO LIMA

ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

# SINTAXE DO PROGRAMA

Todo programa arduino inicia sua execução na função **setup()** e depois faz um laço na função **loop()**. No início das funções existe um { e no final um }.

Cada instrução encerra com ; (ponto e vírgula) que faz parte do comando.

Comentários são informações que não serão executadas.

Elas devem ser colocados das seguintes formas:

**`/* comentário */`** ou

**`// comentário`**

# OPERADORES LÓGICOS

OU:    |    ou    ||

E:    &    ou    &&

# OPERADORES RELACIONAIS

Maior que: >

Maior ou igual: >=

Menor que: <

Menor ou igual: <=

Igual a: ==

Diferente: !=

Os resultados desse operadores sempre são Verdadeiro (número diferente de zero) ou Falso (número igual a zero).

# ESTRUTURA if()

```
if(condição){
```

Comandos

```
}
```

- Início e fim do que está “dentro” do **if** é marcado pelas chaves **{ }**.

# Estrutura if- else

```
if(condição){
```

Comandos

```
}else{
```

comandos do caso contrário

```
}
```

- Else não é obrigatório.
- Início e fim também são marcados pelas chaves.

# INDENTAÇÃO

Derivado do Inglês indentation.

É o espaçamento dado entre algumas linhas de programa para melhor organizá-lo.

Seu uso não é obrigatório para que o programa funcione, mas é uma boa prática e considerada obrigatória entre programadores.

Há um compromisso entre tamanho do código e legibilidade.

Há vários padrões de indentação. Veremos 2 aqui:

# WHITESMITHS

```
main()  
{  
    int a; // declara "a" co  
    a = 3 + 2; // soma 3 com  
    printf("Soma = %i\n", a);  
}
```



# WHITESMITHS

```
if(temperatura < 20)
{
    Aquecedor = 1;
    Arcondicionado = 0;
}
else
{
    Aquecedor = 0;
    if(temperatura > 25)
    {
        ArCondicionado = 1;
    }
    else
    {
        Arcondicionado = 0;
    }
}
```

# EXERCÍCIOS

- 1) Baixar o programa “ifsemin.txt” e organizar a indentação deste programa segundo o padrão de indentação aprendido em sala de aula. Testar o programa e explicar nos comentários como ele funciona.
- 2) Crie um programa que teste se uma variável do tipo float é maior que 6. Caso verdadeiro a mensagem “aluno aprovado” deve aparecer na tela.
- 3) Continuando a partir do exercício anterior, se a nota do aluno for menor que 6 faça com que a mensagem “Aluno em exame” apareça na tela.
- 4) No mesmo programa do exercício 2, caso a nota seja menor que 2,5, a mensagem “Aluno reprovado” deve aparecer na tela. Caso a nota esteja entre 2,5 e 6 a mensagem “aluno em exame” deve aparecer. 0 e 10 são válidos também.

# EXERCÍCIOS

- 5) A partir do exercício 4, como ele seria modificado para considerar o zero como valor não válido mudando apenas um operador?
- 6) Refaça o exercício 5 de forma a utilizar a não utilizar nenhum else.
- 7) Refaça o exercício 5 de forma que, se o número não estiver dentro do intervalo correto, uma mensagem de erro apareça.
- 8) Como ficaria para que a mensagem de erro para valores menores que zero fosse diferente de valores maiores que 10?

# Exercícios envolvendo leitura de números

- 1 - Elaborar um programa onde 2 números sejam lidos pela serial e salvo em variáveis adequadas. Imprima na tela do computador, sem usar nenhuma variável a mais, o resultado:
  - a) da soma dos dois
  - b) da multiplicação dos dois
  - c) da subtração dos dois.
  - d) da divisão do primeiro pelo segundo e do segundo pelo primeiro.
- 2 – Refaça o exercício anterior, salvando os resultados em outras variáveis do tipo inteiro e mostrando todos os resultados ao mesmo tempo.
- 3 – Refaça o exercício 1 fazendo com que o programa peça um terceiro número pela serial. Esse número deve escolher a operação que será realizada entre os dois números, soma ou subtração.
- 4 – A partir do exercício 3, acrescente as opções de multiplicação e divisão.

# SWITCH CASE

```
int cor = 1;
Serial.println("Por favor escolha uma cor");
Serial.println("1: vermelho\n2: ver\n3: Azul");
cor = LeInteiro();
switch(cor)
{
    case 1:
        Serial.println("Você escolheu vermelho");
        break;
    case 2:
        Serial.println("Você escolheu verde");
        break;
    case 3:
        Serial.println("Você escolheu azul");
        break;
    default:
        Serial.println("Você nao escolheu nenhuma cor");
        break;
}
```

# SWITCH CASE

Variável de controle, de modo geral deve ser um int ou um char

```
switch (cor) {  
  case 1:   
    Serial.print("Voce escolheu vermelho\n");  
    break;
```

Diagram illustrating the components of a switch statement:

- Variável de controle**: Points to the variable `cor` in the `switch` statement.
- Valor do caso**: Points to the case label `case 1:`.
- Fim do caso**: Points to the `break;` statement.

# DICIONÁRIO/GLOSSÁRIO

- SWITCH – CHAVE E EQUIVALENTE AO ESCOLHA NO PORTUGOL
- CASE – CASO
- BREAK – QUEBRA E MARCA O FIM DE CADA CASO
- DEFAULT – PADRÃO E É EQUIVALENTE AO OUTROCASO NO PORTUGOL
- FALLTHROUGH – CAIR ATRAVÉS

# FALLTHROUGH

```
switch(nota)
{
    case 4:
        Serial.print("Faltou pouco para um C.");
    case 0: case 1: case 2: case 3:
        Serial.print("O conceito eh D");
    break;
}
```

- É possível executar algumas ações para uma única condição no switch e outras ações para esta mesma condição e para outras sem ter que escrevê-las duas vezes.
- Essa estrutura dentro do switch case se chama fallthrough e pode ser vista acima.



# FALLTHROUGH

```
switch(nota)
{
    case 4:
        Serial.print("Faltou pouco para um C.");
        //fallthrough
    case 0: case 1: case 2: case 3:
        Serial.print("O conceito eh D");
        break;
}
```

- Ao esquecer um break, isso pode se parecer com um fallthrough. Para evitar esse tipo de erro e facilitar a vida de quem for ler o código posteriormente, uma prática comum é evidenciar os fallthrough com um comentário como mostrado acima.

# Exercícios envolvendo leitura de números

- 1 - Elaborar um programa onde 2 números sejam lidos pela serial e salvo em variáveis adequadas. Imprima na tela do computador, sem usar nenhuma variável a mais, o resultado:
  - a) da soma dos dois
  - b) da multiplicação dos dois
  - c) da subtração dos dois.
  - d) da divisão do primeiro pelo segundo e do segundo pelo primeiro.
- 2 – Refaça o exercício anterior, salvando os resultados em outras variáveis do tipo inteiro e mostrando todos os resultados ao mesmo tempo.
- 3 – Refaça o exercício 1 fazendo com que o programa peça um terceiro número pela serial. Esse número deve escolher a operação que será realizada entre os dois números, soma ou subtração.
- 4 – A partir do exercício 3, acrescente as opções de multiplicação e divisão.
- 5 – Refaça o exercício 4 usando switch/case ao invés de if/else.
- 6 - Crie um programa que teste se uma variável do tipo int é maior que 6. Caso verdadeiro a mensagem “aluno aprovado” deve aparecer na tela. Se a nota do aluno for menor que 6 faça com que a mensagem “Aluno em exame” apareça na tela.  
Caso a nota seja menor que 2,5, a mensagem “Aluno reprovado” deve aparecer na tela.  
Caso a nota esteja entre 2,5 e 6 a mensagem “aluno em exame” deve aparecer. 0 e 10 são válidos também.