Oi, pessoal! Esse é o curso de Lógica de programação e hoje nós vamos falar sobre "desvios condicionais". Como a gente viu no nosso primeiro vídeo, o nosso programa, o nosso algoritmo, ele executado sempre de cima para baixo e lido da esquerda para a direita pelo computador, ou seja, tudo aquilo que nós escrevemos lá na função "início", é executado de cima para baixo até o seu final, até o final da última chave, como eu estou indicando aqui. Com as estruturas de "desvios condicionais" nós podemos alterar o fluxo do nosso programa conforme verificações e testes que nós vamos fazer em cima das variáveis. Então, pra isso, nós vamos conhecer hoje três novos comandos, que é o "se", o "se não" e o "se não se". Pra esse primeiro conceito agora, eu gostaria de trabalhar esse exemplo com vocês. Nós fizemos já um exercício antes, que verificava, qual dos times tinha vencido. Só que nós não podíamos fazer nenhuma ação sobre o resultado das comparações.  
  
A partir das estruturas de desvios condicionais, nós podemos, sim, imprimir algo conforme o resultado de uma comparação. Nesse caso aqui, nós vamos trabalhar da seguinte forma: Nós temos duas variáveis a "golsTime1" e a "golsTime2", ou seja, é o número de gols marcados em cada uma das equipes numa partida de futebol. Para que eu consiga fazer essas verificações, eu vou utilizar uma palavra que a gente utiliza no cotidiano que é a palavra "se". O "se" é utilizado para criar uma condição. Nesse caso, a gente já estudou lá nos operadores relacionais, que eu posso verificar a relação de grandeza entre as variáveis. Nesse caso, pra verificar qual time venceu, nós vamos verificar qual deles é maior. Eu vou fazer o seguinte: "se" "golsTime1" for maior que "golsTime2", eu vou imprimir que quem venceu foi "venceu o time 1". Afinal, "golsTime1" é maior que time 2, significa que eu vou imprimir isso. Observem que, nós temos chaves aqui, igual ao que tínhamos nas nossas funções, lá no programa e na função início. Nesse caso, se essa condição for verdadeira, ou seja, se ela retornar verdadeira, eu vou executar o que está aqui dentro. Somente se isso for verdadeiro. Então, se a gente executasse nesse momento, não teríamos nenhum resultado, afinal "golsTime1" não é maior que "golsTime2", então ele não entrou dentro dessa verificação.  
  
Com isso, nós também podemos adicionar uma nova palavra, uma nova palavra reservada chamada "se não". Ou seja, se isso não for verdade, faça o que está aqui dentro. Nesse caso, eu poderia escrever aqui "venceu o time 2". Certo? Então o que nós teríamos aqui? Venceu o time 2. Então ele verificou que "golsTime1" é maior que "golsTime2"? Não, não é. Então faça isso, certo? Então, esse é o contrário disso, ok? Agora, se nós alterarmos nosso programa para 2 gols no time 1 e 2 gols no time 2. Venceu time 2, porquê? "golsTime1" é maior que "golsTime2"? Não eles são iguais, então ele entra aqui. Nesse caso, o nosso programa está errado, porque a gente não pode afirmar que o time 2 ganhou, se os dois fizeram o mesmo número de gols. Pra esse tipo de situação, eu posso adicionar um ou mais comandos chamados "se não se". Que permitem que, se a primeira condição não for verdadeira, permite que eu verifique se existem outras condições que são verdadeiras. Nesse caso, eu quero que vocês vejam aqui comigo. Se "golsTime1" for maior que "golsTime2", venceu o time 1. "Senão se" "golsTime2" for maior que "golsTime1" eu posso dizer então que venceu o time 2, caso contrário, "senão" eu vou escrever que "A partida empatou". Certo? Observem a sequência comigo: Se isso for verdade, ele executa isso. Se não, ou seja, se isso não for verdade, verifica-se se isso aqui é verdade, "golsTime2" maior que "golsTime1". Se isso não for verdade, ou seja, se nenhuma dessas for verdadeira, a partida empatou.  
  
Chamo a atenção de vocês ao seguinte: "se" pode aparecer uma única vez nessa sequência, "se não se" pode aparecer tantas quanto forem necessários e "se não" também uma única vez. Certo? Na sequência, não quer dizer que eu não possa ter mais de um "se" dentro de um mesmo programa. Eu posso ter tantos quantos forem necessários, mas nesse caso aqui, só um "se", vários "se não se" e somente um "se não". Ao executar esse programa, agora ele está funcionando. A partida empatou. Primeiro exemplo pronto! Vamos pro exemplo 2. No caso do exemplo 2, eu quero que vocês observem que nós vamos trabalhar com os operadores lógicos. Nesse nosso programa aqui, o exemplo 2, eu criei uma variável chamada "valor" e atribui o valor 4 a ela. Estou fazendo um teste básico. Se "valor" dividido por 2, restar zero e "valor" menor que 10 eu vou entrar num teste "valor é par e menor que 10". Ou seja, o nosso valor teria satisfeito essa condição e essa. Agora, se isso não for verdade, ele vai imprimir "não entrou no teste". Ou seja, ele passou no teste de cima. Com o valor 4, ao ser executado, nós temos o retorno do programa, "valor é par e menor que 10". Ou seja, 4 é par e 4 é menor que 10. Agora, se eu alterar o valor para 5, qual será o resultado? "Não entrou no teste" Mas 5 não é menor que 10? É menor que 10, mas 5 não é par. Lembrem-se que o "e" faz com que só seja verdadeira a condição se ambas as comparações forem verdadeiras. Nesse caso, essa era falsa e essa era verdadeira. Então, ele não entrou no valor do "se", mas caiu na condição "senão".  
  
Como é que nós faríamos pra esse nosso funcionar, sem fazer grandes alterações? Poderíamos alterar essa condição "e" por "ou" e aí, na hora de imprimir, nós teríamos lá "valor é par ou menor que 10". Por que? Quando eu utilizo a comparação "ou", ou o operador lógico "ou", tanto faz se esta, ou essa, ou e ambas forem verdadeiras, ele vai entrar no laço. Nesse caso aqui, valor é igual a 5. O resto da divisão por 2 de 5 é zero? Não porque ele é um valor ímpar. Ou o valor é menor que 10? Sim, 5 é menor que 10. Então, como tanto faz se essa ou essa forem verdadeiras, ele entra, ele vai imprimir essa mensagem aqui. Valor é par ou menor que 10. Por que eu alterei o "ou", tirei o "e" e coloquei o "ou"? Porque eu não posso garantir que é par e menor que 10, afinal, 5 não passou nesse teste, tá? Isso tem a ver com o teste que a gente fez aqui, teste lógico. Certo? Pessoal, o uso das estruturas de desvios condicionais é imprescindível dentro da programação. Dessa forma, eu sugiro que vocês pratiquem fazendo os exercícios disponíveis no curso e, caso tenham dúvidas, voltem a assistir o vídeo. Até mais.