Oi, pessoal! Esse é o curso de Lógica de Programação e hoje nós vamos falar sobre "funções". Observem um exemplo que eu trouxe aqui pra vocês, esse exemplo deve ser algo facílimo pra quem chegou até aqui, certo? No caso, a gente criou uma variável "v1" e "v2" que recebe 2 e 3, respectivamente e inicializou uma variável chamada "resultado". "resultado" vai receber a soma de "v1" com "v2" e nós vamos imprimir esse resultado. Simples, nós já fizemos coisas bem mais complexas do que isso. Porém, eu posso querer utilizar a funcionalidade, ou digamos, a operação "soma" mais de uma vez no meu programa e as linguagens de programação me permitem que eu crie as minhas próprias funções.  
  
Mas o que são as funções? Funções são pequenos módulos de programas que podem ser reaproveitados. Se vocês observarem, nós temos funções desde o início do nosso curso. Observem que "escreva" é uma função, o que "escreva" faz? Imprime na tela. Nós temos a função "início", que é a nossa função principal, é aquela que é executada, quando nosso programa inicia. O que a gente vai ver agora é como construir uma função nossa. E nós podemos fazer isso de algumas formas. Primeiro passo, eu vou construir uma função exemplo na qual nós vamos somar dois valores e imprimir o resultado desses valores, fazer exatamente aquilo que a gente tem aqui já pronto. Primeira coisa, eu preciso dizer que vou criar uma função, eu preciso dizer qual é o nome dela função "soma". O que soma vai fazer? "soma" vai somar dois valores e imprimir o resultado dessa soma. Então, eu preciso dizer quais são os parâmetros que a minha função precisa.  
  
Com o que ela vai trabalhar? Se eu vou somar dois valores, eu preciso passar dois valores pra que ela possa fazer a soma. Eu vou lá, vou até mudar o nome pra que vocês não confundam um ponto importante aqui. Minha função soma vai receber "inteiro a" e o "inteiro b" e vai imprimir "escreva", ou melhor, vamos fazer o seguinte, vamos criar uma outra variável aqui vai se chamar "result", que vai receber "a+b" e vamos imprimir lá "O resultado é:" "result". Algo muito parecido com o que nós tínhamos abaixo. Se nós executarmos nosso programa, observem que o que está sendo impresso é "O resultado é: 5". Nesse caso, qual é o trecho que está sendo executado? A linha 14. Por que nós não executamos a função "soma" antes da "inicio"? Por que a função "soma" só vai ser executada quando ela for utilizada dentro da função "início", quando a gente chamar ela. E a gente chama ela como? Pelo nome que nós demos aqui. Lembrando que assim como as variáveis, a função "soma" tem esse nome porque eu dei esse nome a ela. Vocês podem dar qualquer nome a essa função. Vou fazer uma alteração no nosso programa de forma que ele fique mais útil para a nossa função. Observem que eu vou tirar toda essa parte do resultado, do processamento e da impressão e vou, simplesmente, chamar a função "soma" e passar os meus valores "v1" e "v2". Observem o seguinte aqui, pessoal, vou executar e vocês verão que deu certo, o resultado é o mesmo, 2 e 3, o resultado é 5. Nós criamos uma função, chamamos ela de "soma" e dissemos que ela recebe por parâmetro "a" e "b" que são também inteiros. Só que quando a gente chamou a função "soma" no nosso programa início, nós passamos "v1" e "v2", por quê? Porque esses parâmetros não tem relação nenhuma com o programa principal, nós sabemos que nossa função tem uma assinatura, ou seja, ela precisa receber um inteiro e um inteiro, ou seja, dois inteiros, tanto faz o nome deles. Observem que eu mandei "v1" e "v2" e mesmo assim, sem essas variáveis de parâmetro chamarem-se "v1" e "v2" o meu programa funcionou. O que a gente pode perceber disso? Que "v1" vai ser atribuído a variável "a" e "v2" a variável "b". Dentro da função, vai ser trabalhado com as variáveis que foram passadas por parâmetro. Vou trabalhar com "a" e "b" e não mais com "v1" e "v2". Eu posso passar valores inteiros, valores diretos, sem solicitar, ou criar, ou passando uma variável. Nesse caso, o resultado é 12, por quê? O primeiro parâmetro é "v1" que é "2", o segundo parâmetro é "10" eu não passei "v2". Então "b" recebe 10, eu vou somar "2" com "10", o resultado é "12". OK? Posso criar funções para qualquer situação no meu programa, eu posso ter muitas funções, mas cuidado que eu só posso ter funções com um nome, eu não posso ter mais de uma função com o mesmo nome nesse nosso programa. Existe uma outra forma de criar a função que é torná-la ainda mais útil. O que acontece? Digamos que esse nosso programa, nós não quiséssemos imprimir "O resultado é:", nós queremos usar ele em diferentes contextos, primeiro pra imprimir o resultado, depois pra saber quanto é, e verificar se é par ou ímpar. Se nós deixarmos com essa impressão dentro da função, nós não teríamos como usar o resultado dela. Pra tornar ela mais útil, existe um comando chamado "retorne". Observem que criamos uma função que imprime. Mas nós também podemos criar uma função que retorna um processamento. Vou fazer uma pequena alteração nela aqui. Se ela vai retornar, eu preciso dizer qual é o tipo de retorno que ela vai ter. Ela vai continuar com a mesma funcionalidade, ela vai somar dois valores, só que ao invés de imprimir, ela vai retornar o valor somado.  
  
Posso tirar tudo o que eu tenho aqui dentro e simplesmente dizer o seguinte, "retorne a+b". Quando eu chamar "soma" ela volta pro programa principal, retornando para uma variável "a+b", nesse caso, eu vou fazer novamente uma alteração. Vou criar uma variável "resultado", a mesma que nós tínhamos antes, e vou usar aqui "resultado" recebe o retorno da função "soma" e agora eu posso imprimir então "resultado". Vamos executar. 12, certo? Por que nós temos o retorno e a função sem retorno. Uma função sem retorno ela pode ser utilizada pra inicializar uma variável, para processar, inicializar um vetor, uma matriz, eu posso utilizar ela pra imprimir algo, uma mensagem personalizada. A função com retorno ela se torna mais útil, porque ela pode ser usada em diferentes contextos. Observem aqui, que nesse caso, quem vai se preocupar em imprimir o valor pro usuário é o programa principal, tornando ele mais reutilizável. Retomando, toda função precisa ter um nome e ela pode ou não receber parâmetros, nós podemos criar uma função que só imprima "Olá", "Bom dia". Nesse caso, nós temos parâmetros que são utilizados dentro dela. Essas funções elas podem ou não ter retorno. Se ela não tem retorno o nome dela é só função e o nome. Se ela tem retorno, a assinatura dela é função, o tipo de retorno e o nome dela. Lembrando que as funções com retorno acabam tornando isso mais úteis, porque delegam para o programa principal a impressão e o uso do resultado. OK? Pessoal, conhecer funções é essencial para progredir nos conteúdos de programação. Eu sugiro que vocês façam muitos exercícios que vocês tentem refazer aqueles que foram propostos na forma de funções e, caso vocês tenham dúvidas, voltem a assistir o nosso vídeo. Até lá!