Código 10: Loops

Usar loops é uma forma de ganhar mais pontos na linha "Complexidade de codificação" na seção Técnica da **rubrica de julgamento**. Também pode ajudar com "Função de aplicativo" e "Experiência e design do usuário".

Nesta lição, você vai ...

- Saiber mais sobre loops
- Criar um aplicativo que você pode levar para festas de aniversário

Termos chave

- **Loops -** uma maneira de dizer a um computador para fazer algo (um bloco de código) muitas vezes seguidas
- **For Loop -** repete um bloco de código um determinado número de vezes
- **For Each Loop -** repete uma vez para cada item em uma lista
- **While Loop -** repete um bloco de código até que uma condição não seja mais verdadeira

Aprender

rotações

Atividade

Quantos anos você tem?

Reflexão

Loopty Loop!

Recursos adicionais

Tutorial do Hangman

Inspiração - Loops

Nesta lição, você aprenderá sobre loops. Os **loops** são uma forma de dizer ao computador para fazer algo várias vezes seguidas. Os computadores são realmente bons em fazer as coisas repetidamente e rapidamente.

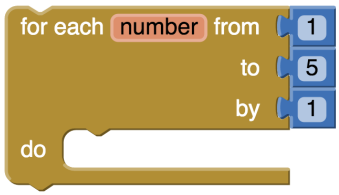
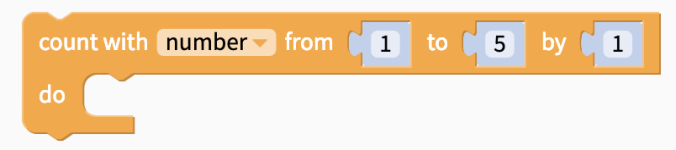
Imagine que você foi solicitado a escrever seu nome 100 vezes consecutivas. Isso pode levar muito tempo e você pode cometer alguns erros ao longo do caminho. Esta é uma tarefa perfeita para um computador, que o conseguiria fazer muito rápido e sem cometer erros. Você pode tirar vantagem disso usando loops. Um loop é um bloco de código que se repetirá continuamente.

Existem dois tipos de loops, “while loops” e “for loops”. Os **loops while** serão repetidos até que uma condição não seja mais verdadeira e os **loops for** se repetirão um certo número de vezes. Você também aprenderá sobre **loops for each**, que são um tipo de **loop for** que se repete uma vez para cada item em uma lista. Vamos examinar cada um deles com mais detalhes.

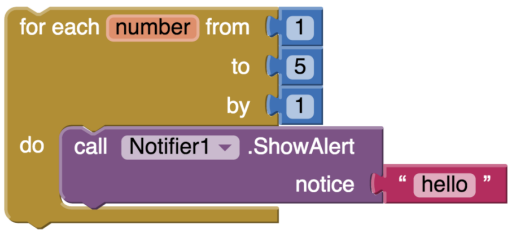
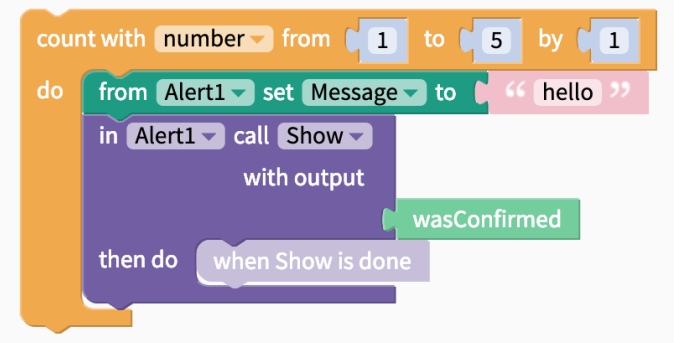
For loops

Os **loops For** repetirão um bloco de código um determinado número de vezes. A razão que eles são chamados **for loops** é que você pode dizer ao seu aplicativo quantas vezes você quer que ele repita para o código **for** . Você pode pensar em loops for como dizer ao seu aplicativo, “repita isso, por 17 vezes” ou “repita isso, por 5 vezes”.

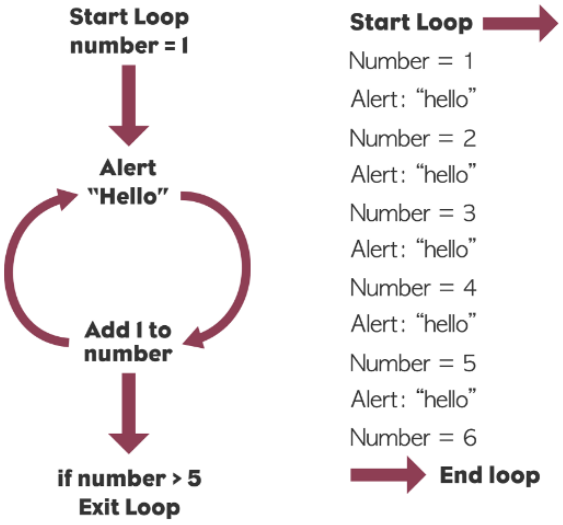
Os **For loops** usam uma variável para contar quantas vezes o código foi repetido, chamado de **count**. Você controla quantas vezes o loop se repete definindo onde o contador começa e termina. Você também define o quanto o contador sobe cada vez que o código se repete. Na maioria dos cenários, você deseja que o contador aumente em 1 cada vez que o loop se repetir.

App Inventor	Thunkable
	

A parte que diz o *número* é o contador. Por enquanto, o contador se chama *número* , mas você pode mudar isso. *O número* começará em 1 e parará quando for igual a 5. Cada vez que o código dentro do loop se repete, o *número* aumenta em 1. Portanto, este loop repetirá o código dentro dele 5 vezes. Agora, este circuito não faz nada, já que o *faz parte* está vazio. Vamos examinar um exemplo.

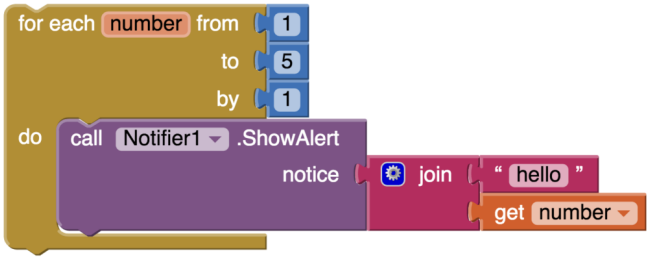
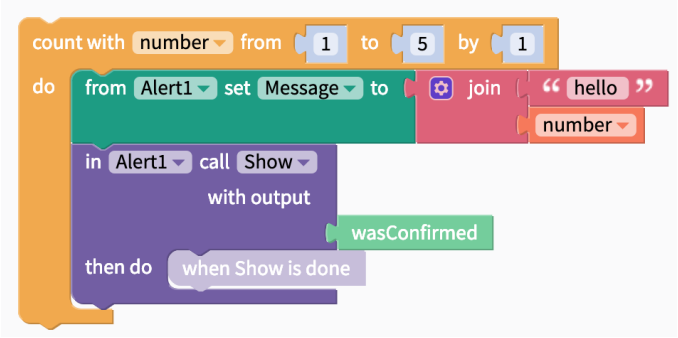
App Inventor	Thunkable
	

Nós não mudamos nada sobre o *número* variável, mas nós adicionamos um código para o *fazer* parte do loop. Cada vez que este loop é executado, o aplicativo irá alertar seu usuário “hello”, então o usuário será alertado 5 vezes. Veja como o aplicativo será executado neste ciclo:

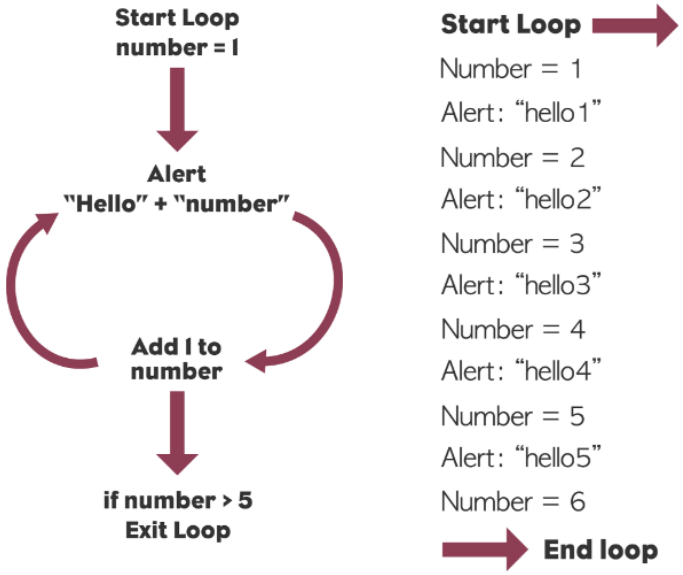


Este loop seria útil se você quisesse alertar o usuário "alô" 5 vezes. Isso pode não parecer muito útil, pois não é tão difícil colocar 5 blocos notificadores em uma linha dizendo "olá". Mas, e se você quisesse alertar o usuário "alô" 100 vezes? Isso seria muito mais fácil de fazer com um loop do que colocar 100 blocos em uma linha.

Outra maneira que os loops for podem ser úteis é usando a variável counter em seu código. Cada vez que o loop for executado, a variável do contador terá um valor diferente e isso pode ser muito útil. Aqui está um exemplo.

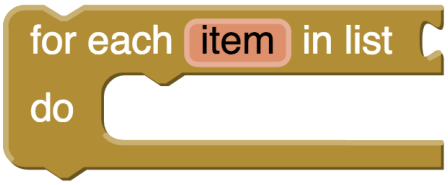
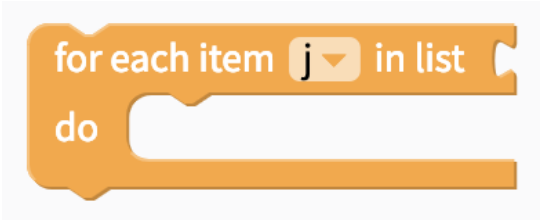
App Inventor	Thunkable
	

Neste **loop for** , estamos usando o número da variável no código, anexando-o à palavra "hello". O número aumenta em 1 a cada vez, então o aplicativo continuará imprimindo coisas diferentes cada vez que for executado. Aqui está como o aplicativo executará este loop agora:



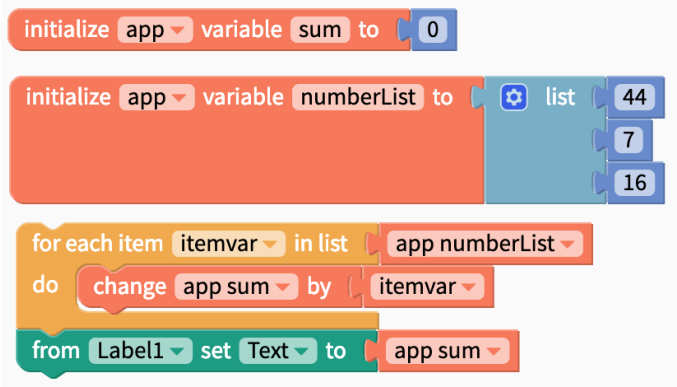
For Each Loops

Outro tipo útil de **loop** for que você pode usar no App Inventor e Thunkable é este:

App Inventor	Thunkable
	

Aqui, a variável do contador é chamada de item e este loop já está configurado para se repetir para o número de itens em uma lista. Esses loops são muito úteis sempre que você precisa fazer algo com uma lista. Digamos que você tenha uma lista de números e queira somar todos os números da lista e armazená-los em uma variável chamada soma. Veja como você faria isso com um para cada loop.

App Inventor	Thunkable



Cada vez que o loop for executado, a variável sum obterá um item de numberList adicionado a ela. O loop irá parar automaticamente depois que todos os números da lista forem adicionados. Aqui está o que o loop faz:

- Há um item na lista, iniciar loop.
  - Soma Global = 0 + 44
- Os itens ainda estão em loop, repita.
  - Soma Global = 44 + 7
- Os itens ainda estão em loop, repita
  - Soma Global = 51 + 16
- Não há itens restantes, saia do loop.
  - Defina o texto Label1 para 67

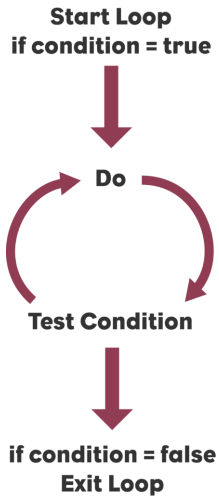
Nota: Você deve ter notado que as variáveis de contador nesta seção são muito parecidas com as variáveis locais que você aprendeu em **Codificação 5: Variáveis**. *Assim como as variáveis locais, você só pode usá-las dentro do loop.*

While Loops

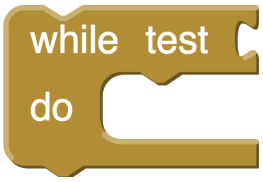
Loops while são loops que continuarão até que uma condição não seja mais verdadeira. O motivo pelo qual eles são chamados de **loops while** é porque o código se repetirá enquanto uma condição ainda for verdadeira. Você pode pensar nos **loops while** dizendo ao seu aplicativo " *enquanto* isso acontece, repita isso" ou " *enquanto* isso não mudou, repita isso".

Você está dando uma festa e quer que a música continue tocando até que todos os convidados saiam. Você pode descrever sua festa como este ciclo: • Enquanto (convidados na festa> 0) • fazer: continue tocando música	E se você também quiser que sua música pare de tocar quando for depois da meia-noite? Você pode programar loops while para encerrar o loop com base em várias condições usando a lógica. Agora você pode descrever sua festa como este loop. Enquanto (convidados na festa> 0) e (hora <meia-noite) fazer: continue tocando música Nesse caso, a música parava assim que todos saíssem da festa ou se já passasse da meia-noite.

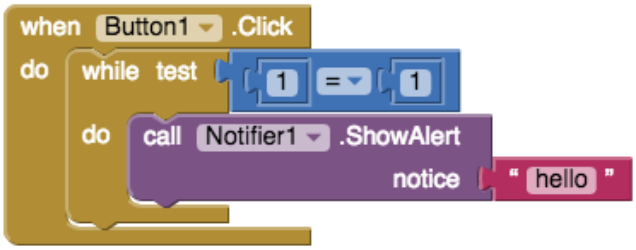
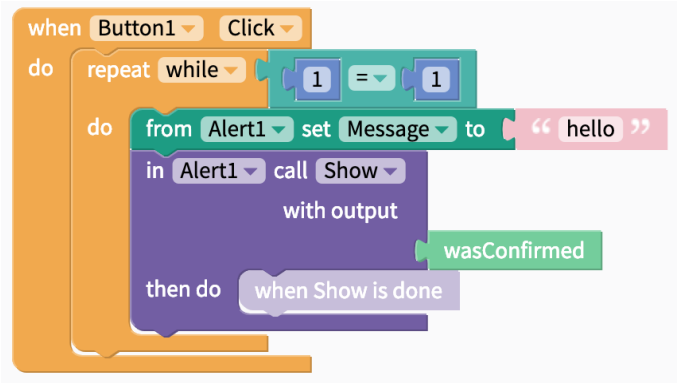
Para usar um **loop while**, você precisa configurar uma condição que comece como verdadeira. Se sua condição começar como falsa, seu loop nunca será executado. O loop verificará a condição todas as vezes antes de se repetir para garantir que a condição ainda seja verdadeira.



Esta é a aparência dos **loops while** no App Inventor e Thunkable:

App Inventor	Thunkable
	

Com **loops while** é possível cometer erros! Se você escolher uma condição que nunca será falsa, seu loop nunca terminará. Isso é chamado de loop infinito. Aqui está um exemplo:

App Inventor	Thunkable
	

Visto que 1 sempre será igual a 1, essa condição nunca pode ser falsa. Quando executamos este código no App Inventor, nosso telefone travou e não pudemos fazer nada. Você também pode receber uma mensagem dizendo que o App Inventor Companion App parou de funcionar. Quando você está programando, você quebrará coisas o tempo todo! Reinicie seu aplicativo complementar e tente novamente.

Atividade: Quantos anos você tem?

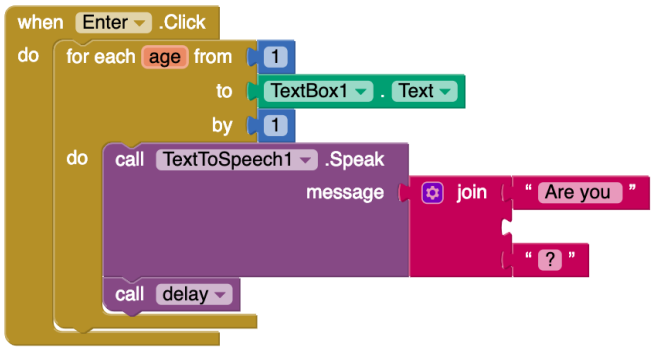
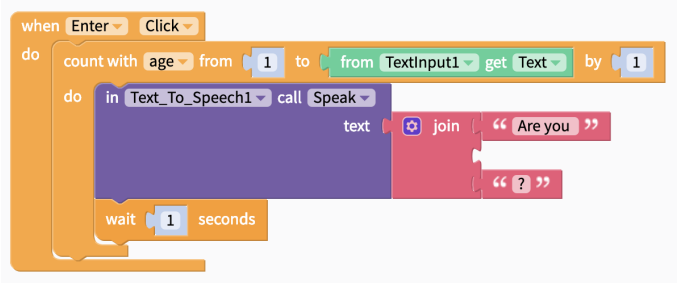
Quantas vezes você já foi a uma festa de aniversário e ouviu aquela música que diz… você tem 1, tem 2, tem 3….? Se você ainda não ouviu, dê uma olhada naquele vídeo!

https://www.youtube.com/watch?v=EPi9DLfIK4M

Essa música leva muito tempo para ser cantada, especialmente quando a pessoa tem 98 anos, como a avó do vídeo. Poderíamos fazer isso muito mais rápido usando loops. Imagine que pudéssemos criar um aplicativo onde a pessoa simplesmente coloca sua idade e o aplicativo canta a música para ela!

Nesta atividade, você vai fazer um aplicativo que pode levar para festas de aniversário.

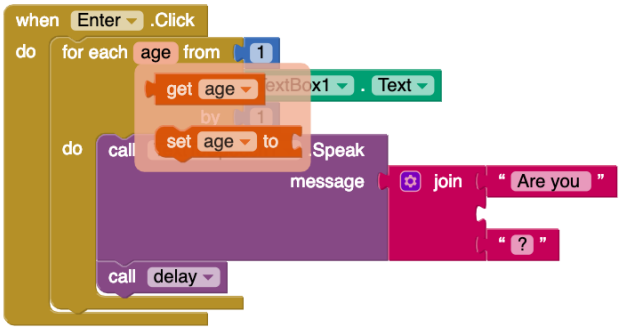
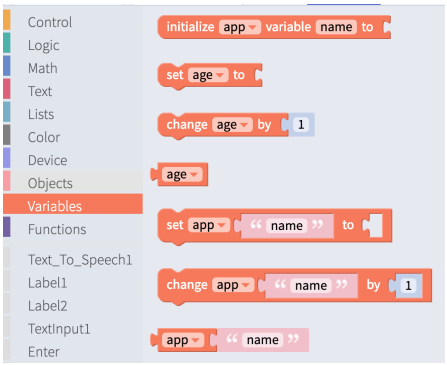
Você precisará usar um **loop for** :

App Inventor	Thunkable
	

Este **loop for** conta de 1 até o número que o usuário colocou na caixa de texto. Cada vez que o loop é executado, o contador aumenta em 1. Este código está *quase* completo, mas não diz a idade cada vez que conta. Você precisa descobrir como fazer esse bloco dizer a idade cada vez que for importante.

App Inventor Code Thunkable Code

Presa? Aqui esta uma dica!

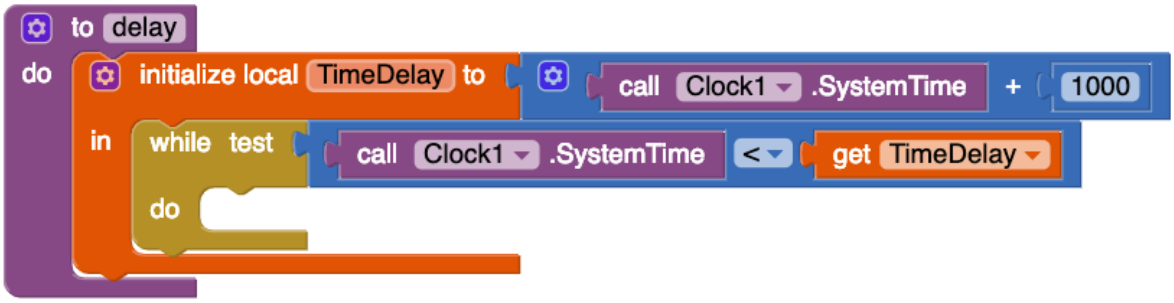
App Inventor	Thunkable
A idade é uma variável e você pode pegar blocos passando o mouse sobre eles 	*A idade é uma variável e você pode obtê-la na gaveta de variáveis.* 

O bloco "delay" do App Inventor explicado!

Este aplicativo possui um procedimento denominado atraso.

Este bloco dá ao aplicativo tempo para terminar de dizer uma coisa antes de passar para a próxima. Tente removê-lo de seu loop for, você notará que o aplicativo não funciona mais. Isso ocorre porque o aplicativo se move pelo loop muito rápido e não tem tempo para dizer todas as idades.

Então, o que o bloco de atraso está realmente fazendo?



Este bloco chama um procedimento denominado *delay* . Você viu os procedimentos no aplicativo de contagem pela primeira vez em **Codificação 5: Variáveis**. Um **procedimento** é um bloco de código que possui um nome que permite reutilizá-lo em diferentes lugares.

Delay primeiro cria uma variável chamada *TimeDelay* . *Clock1.SystemTime* fornece a hora atual em segundos. Em seguida, adicionamos um segundo a isso, dando-nos um tempo que é 1 segundo no futuro. Isso torna a variável *TimeDelay* igual a um tempo que está 1 segundo no futuro.

Em seguida, usamos um **loop while** . A condição no **loop while** diz ao aplicativo para não fazer nada enquanto a hora atual estiver a menos de um segundo no futuro. Isso permanece verdadeiro por um segundo, depois disso o clock se recupera e o loop for continua a contar. Este loop faz o aplicativo esperar 1 segundo.

Reflexão

Nesta lição, você aprendeu sobre os loops for , while e for each . Você também criou um aplicativo que usava dois tipos diferentes de loops!

- Como você acha que pode usar loops em seu aplicativo final?
- Isso lhe dá alguma ideia de como você pode construir alguns de seus recursos?

Recursos adicionais: Tutorial do Hangman

Inventor de aplicativos: tutorial do Hangman (avançado)

Você quer um pouco mais de prática usando loops? Você pode tentar este tutorial da ex-aluno do Technovation, Jennifer John.

https://youtu.be/GNRJP_Adteo