

Código 9: Lógica Avançada e Condicionais

Usar condicionais é uma forma de ganhar mais pontos na seção "complexidade do código" da [rubrica de julgamento](#) . Também pode ajudar com "funcionamento do aplicativo" e "experiência e design do usuário".

Nesta lição, você vai ...

- Aprender a escrever declarações condicionais que usam lógica
- Melhorar seu aplicativo Magic 8 Ball para que uma pessoa não possa fazer uma pergunta que não tenha um ponto de interrogação.

Termos Chave e Conceitos

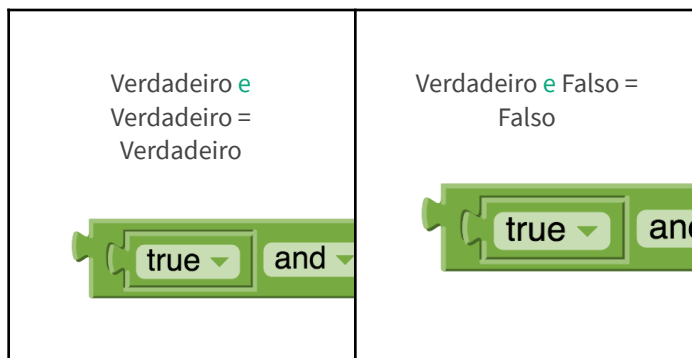
- **Operadores lógicos** - permitem que os computadores tomem decisões com base em várias condições
- **E (and)** - um operador de programação que produzirá verdadeiro quando todas as condições de entrada forem verdadeiras
- **Ou (or)** - um operador de programação que produzirá verdadeiro quando pelo menos uma das condições de entrada for verdadeira
- **Não (not)** - produz o oposto da condição

Inspiração - Condições

Na última lição de código, você aprendeu como fazer seu aplicativo fazer coisas diferentes usando instruções condicionais. Nesta lição, você aprenderá sobre operadores lógicos. Os **operadores lógicos** permitem que os computadores tomem decisões com base em várias condições. No App Inventor, esses são blocos que precisam de duas entradas. Existem três operadores lógicos principais que você aprenderá nesta seção: **e (and)** , **ou (or)** , e **não (not)** .

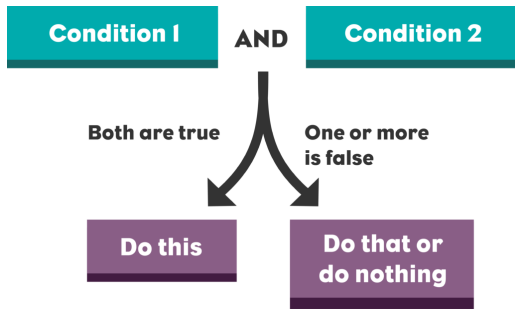
Operador E (and)

O operador **E (and)** produzirá true (verdadeiro) quando todas as condições de entrada forem verdadeiras. Se alguma das condições de entrada for falsa, a saída será falsa. Aqui estão todos os resultados possíveis ao usar o operador **E (and)** .



<p>Falso e Verdadeiro = Falso</p> 	<p>Falso e Falso = Falso</p> 
---	--

Veja como você pode pensar em usar **E (and)** em uma instrução condicional:




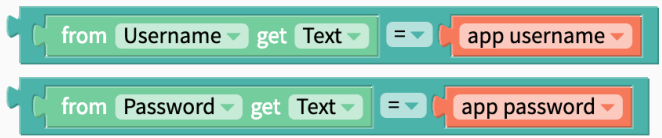
*Nota: Usar o operador **E (and)** é diferente de usar uma instrução **else if** porque ambas as condições são avaliadas ao mesmo tempo, em vez de uma após a outra.*

Exemplos

Você deve usar **E (and)** quando precisar que duas condições sejam verdadeiras para que algo aconteça. Aqui estão alguns exemplos de quando você pode querer usar **E (and)** em seu aplicativo. Imagine que seu aplicativo permite que as pessoas façam login. Você só deseja que seu aplicativo faça login se o nome de usuário e a senha estiverem corretos. Aqui está o que você deseja que seu aplicativo faça:

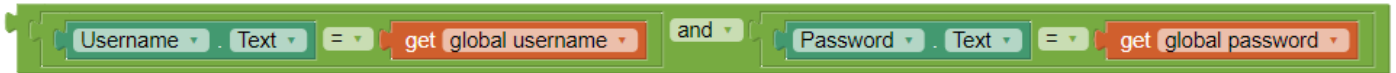
- Se (o nome de usuário está correto) e (a senha está correta) → então permita que o usuário faça o login
- Se um deles estiver incorreto → então não permita que o usuário faça login

Estas são as suas duas condições no App Inventor e Thinkable.

App Inventor	Thinkable
	

Aqui estão essas duas condições em um bloco **E (and)**.

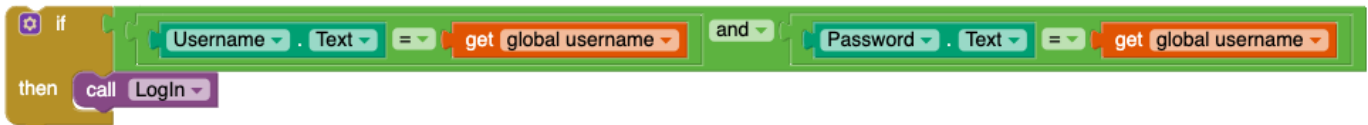
App Inventor



Thunkable



Agora você pode colocá-los em uma instrução if (se). Esta instrução if só fará o login da pessoa se ambas as condições forem verdadeiras.

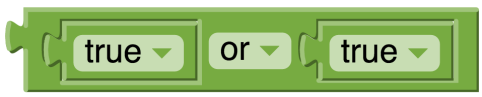


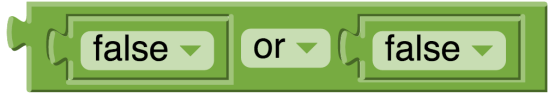


Aqui estão mais alguns exemplos de quando você pode usar **E (and)**.

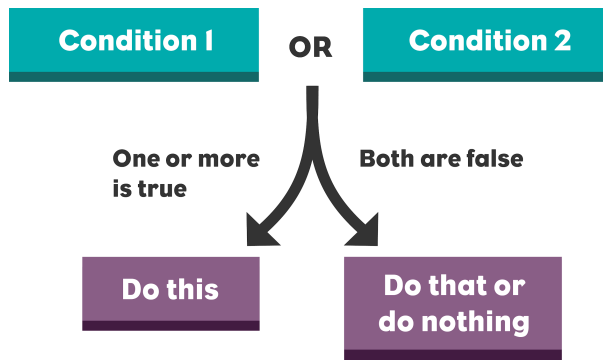
- Ganhando um jogo
 - Se (o jogador termina o nível) e (o tempo não acabou) → então o usuário ganha
- Encontrar uma mercearia nas proximidades:
 - Se (a loja está aberta) e (a loja está perto do usuário) → então aparecem nos resultados da pesquisa
- Postagem de fotos com legendas:
 - Se (o usuário selecionou uma imagem) e (o usuário inseriu uma legenda) → então permita que o usuário poste a imagem
- Você pode pensar em mais exemplos?

Operador Ou (or)

Para que o operador **Ou (or)** produza true (verdadeiro), apenas uma das entradas precisa ser true. Aqui estão todos os resultados possíveis ao usar o operador **Ou (or)**.

<p>Verdadeiro ou Verdadeiro = Verdadeiro</p> 	<p>Verdadeiro ou Falso = Verdadeiro</p> 
<p>Falso ou Verdadeiro = Verdadeiro</p> 	<p>Falso ou Falso = Falso</p> 

Veja como você pode pensar em usar **Ou (or)** em uma instrução condicional:

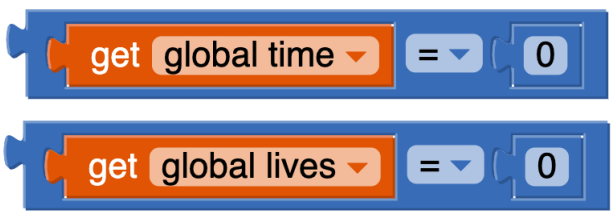
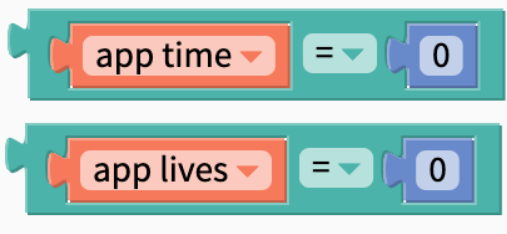


*Nota: O operador **Ou (or)** também pode parecer semelhante a **else if** para você. O operador **Ou (or)** é diferente porque as duas condições são avaliadas ao mesmo tempo, em vez de uma após a outra. O operador **Ou (or)** é melhor usar quando você tem duas condições que deveriam ter o mesmo resultado se fossem verdadeiras.*

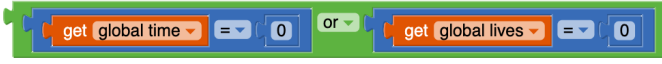
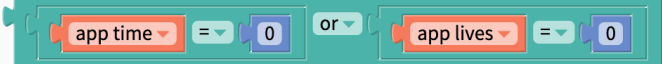
Exemplos

Aqui estão alguns exemplos de quando você pode querer usar **Ou (or)** em seu aplicativo. Imagine que você está criando um jogo. Você quer que o jogo acabe quando o tempo acabar ou quando o jogador perder todas as suas vidas.

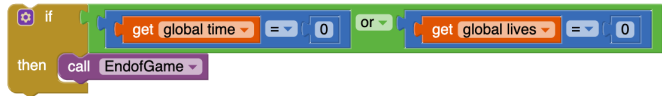
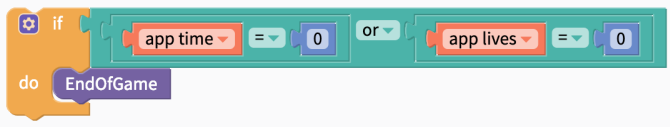
Aqui estão suas duas condições no App Inventor e Thinkable.

App Inventor	Thunkable
	

Aqui estão essas duas condições em um bloco **Ou(or)**:

App Inventor	Thunkable
	

Agora você pode colocá-los em uma instrução if. Esta declaração if encerrará o jogo se pelo menos uma das condições for verdadeira.

App Inventor	Thunkable
	

Aqui estão mais alguns exemplos de quando você pode usar **Ou (or)**.

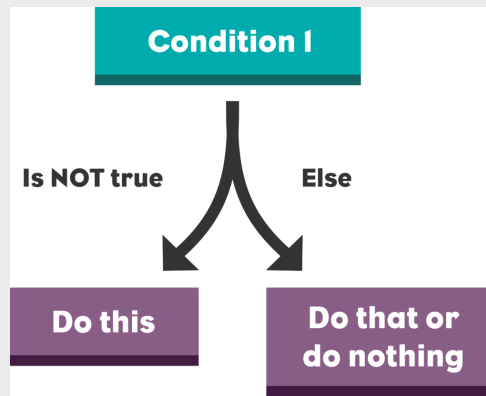
- Mostrando resultados de pesquisa
 - Se (o título corresponder) ou (a descrição corresponder) → mostre-o nos resultados da pesquisa
- Recomendar vídeos a um usuário
 - Se (os amigos do usuário gostarem) ou (é semelhante a algo que o usuário gostou) → então recomende ao usuário

Operador Não (not)

O operador **Não (not)** muda o valor de uma condição de entrada para ser o oposto do que é.



<p>não Verdadeiro = Falso</p> 	<p>não Falso = Verdadeiro</p> 
---	--

Veja como você pode pensar nisso em uma declaração condicional.


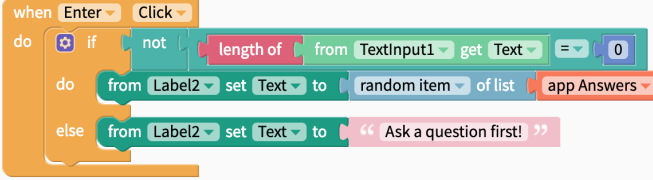


Exemples

Em seu aplicativo Magic 8 Ball, você queria verificar se o usuário havia inserido uma pergunta na caixa de texto. Você pode usar o operador **Não (not)** para se certificar de que a caixa de texto não está vazia. Esta é a aparência dessa condição no App Inventor e Thinkable:

App Inventor	Thinkable
	

Aqui está como você pode colocar essa condição em uma instrução if.

App Inventor	Thunkable
	

Esta instrução if só dará ao usuário uma resposta se a caixa de texto não estiver vazia.

Aqui estão alguns outros exemplos em que você pode usar o operador **Não (not)**.

- Você deseja excluir os resultados da pesquisa sobre cachorros, então você verifica a palavra “cachorrinho” usando o operador **Não (not)**.
 - Se não (o artigo contém a palavra “cachorrinho”) → mostre os resultados
- Você pode combinar operadores lógicos para fazer mais coisas. Veja como você poderia mostrar resultados de pesquisa que têm a palavra “gatinho” e não têm a palavra “cachorrinho”.
 - Se não (o artigo contém a palavra “cachorrinho”) e (o artigo contém a palavra “gatinho”) → mostre os resultados

Atividade - Melhorias no Magic 8 Ball - Operadores Lógicos

Nesta atividade, você está indo para fazer mais uma melhoria para o seu aplicativo Magic 8 Ball, utilizando operadores lógicos. Seu aplicativo dirá aos usuários para “fazer uma pergunta primeiro” sempre que

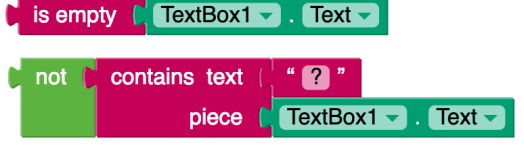
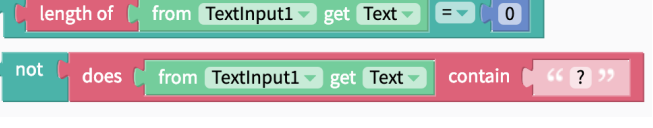
1. O usuário faz uma pergunta que não contém um ponto de interrogação
2. A caixa de texto está vazia

Pergunta 1: Primeiro, quais são as duas condições que você precisa verificar. Como você poderia escrever essas condições no App Inventor?

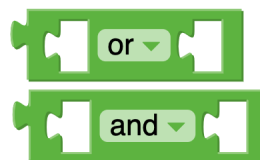
Resposta - Tente você mesma antes de checar!

Condição 1: a caixa de texto está vazia



Condição 2: o texto não tem ponto de interrogação

App Inventor	Thunkable
	

Pergunta 2: Você acha que precisa usar o operador **E (and)** ou o operador **Ou (or)** para isso? Onde você acha que isso deve entrar em seu código?



Resposta - Tente você mesma antes de checar!

App Inventor	Thunkable
	

Use o código do Magic 8 Ball das duas últimas lições. Se você precisa do código inicial, aqui está novamente.

[Código Inicial App Inventor](#) [Código Inicial Thunkable](#)

Reflexão

Nesta lição, você aprendeu como usar operadores lógicos para fazer declarações condicionais mais avançadas.

- Quais são algumas outras condições que podem ser úteis para adicionar ao aplicativo Magic 8 Ball?
- Você consegue pensar em alguns lugares em seu aplicativo em que gostaria de usá-lo?