

The logo for Oracle Academy is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is framed by two horizontal dark gray bars, one at the top and one at the bottom.

# ORACLE

## Academy

# Java Fundamentals

2-3

## Procedimentos e Argumentos

**ORACLE**  
Academy



Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

# Objetivos

- Esta lição abrange os seguintes objetivos:
  - Alternar entre o Scene editor e o Code editor e descrever as diferenças visuais existentes entre eles
  - Localizar e descrever a finalidade do painel Methods e da guia Procedures
  - Usar procedimentos para mover objetos
  - Adicionar procedimentos de programação ao Code editor
  - Demonstrar como é possível alterar os valores dos procedimentos
  - Criar comentários de programação
  - Reordenar, editar, excluir, copiar e desativar instruções de programação
  - Testar e depurar uma animação



# Exibir o Code Editor

- Clique no Edit Code (no scene editor) para exibir o Code editor
- É no Code editor que você programa sua animação



**ORACLE**  
Academy

JF 2-3  
Procedimentos e Argumentos

Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

4

## Selecionar Instância

- Primeiro, selecione a instância que deseja programar
- Isso garante que você está criando uma instrução de programação para a instância correta
- Selecione uma instância clicando nela na janela de cena pequena ou usando o menu pull-down da instância abaixo da janela de cena pequena

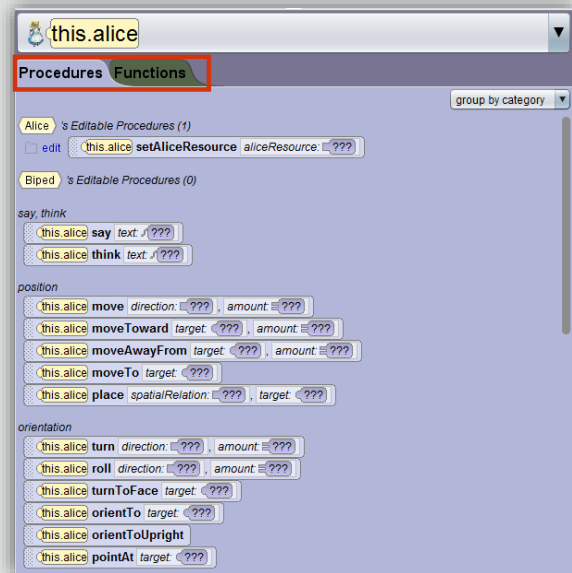


Quando você clica no objeto na janela da cena, sempre certifique-se de que o objeto, e não o mundo propriamente dito, tenha sido selecionado. Você pode verificar isso por meio do nome que é exibido no menu drop-down.

# Painel Methods

O Painel Methods contém duas guias:

- Procedures:
  - todos os procedimentos predefinidos de um objeto
- Functions:
  - todas as funções predefinidas de um objeto

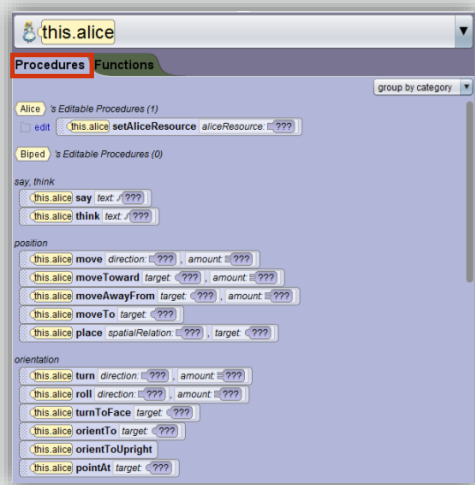


Todas as instruções de programação no Alice serão provenientes dessas guias. Procedimentos das ações e funções para informação.

# Guia Procedures

- A guia Procedures exhibe procedimentos predefinidos da instância selecionada, bem como procedimentos que você mesmo define

Um procedimento é uma parte de código de programa que define como o objeto deve executar uma tarefa. O Alice 3 tem um conjunto de procedimentos para cada classe. No entanto, os usuários podem criar ou "declarar" novos procedimentos.

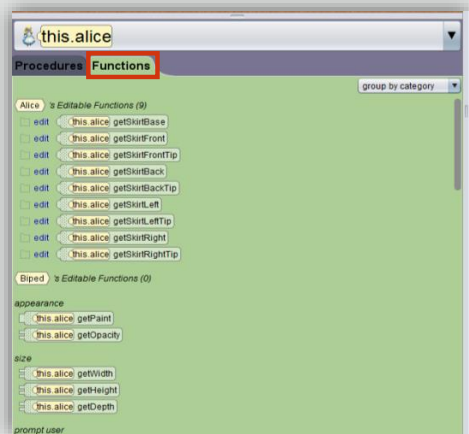


Você analisará a criação de seus próprios procedimentos mais adiante no curso. O Alice 3 tem vários procedimentos embutidos que permitirão a você manipular seus objetos.

# Guia Functions

- A guia Functions exibe funções predefinidas da instância selecionada, bem como funções que você mesmo define

Uma Função calcula e responde uma pergunta sobre um objeto: "Qual é largura ou altura de um objeto?" ou "Qual é a distância desse objeto em relação a outro?" O Alice 3 tem um conjunto de funções para cada classe. No entanto, os usuários podem declarar novas funções.



Uma função é usada para retornar informações sobre seu objeto e deve ser usada em conjunto com procedimentos para criar instruções de programação.



## Guias do Code Editor

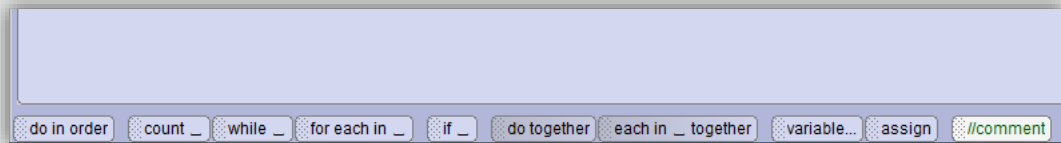
- O Menu Class é exibido à esquerda da guia Scene
- Você cria suas instruções de programação na guia myFirstMethod
- Por padrão, o Alice cria uma instrução de controle Do In Order no procedimento myFirstMethod
- A área identificada como "drop statement here" corresponde ao local em que você inserirá as instruções de programação



O Alice executa o código em uma ordem de procedimentos. O código que está no início será executado primeiro e, depois, as linhas seguintes de código, em ordem, até atingir o fim do código.

# Instruções de Controle

- Na parte inferior da guia myFirstMethod, você encontrará instruções de controle do Alice 3



Essas instruções de controle podem ser inseridas no code editor clicando nelas e arrastando-as para qualquer posição que você deseje.

# Movimento do Objeto

- O movimento do objeto é egocêntrico:
  - Os objetos se movem com base na direção para onde estão direcionados
- Um objeto pode se mover em seis direções:
  - Para cima
  - Para baixo
  - Para frente
  - Para trás
  - Para a direita
  - Para a esquerda

Um objeto, assim como suas subpartes, pode mover-se nessas direções.

# Exemplos de Procedimentos de Movimento

Procedimento	Descrição
<b>Move</b>	Move o objeto em qualquer uma das seis direções
<b>Move Toward</b>	Move o objeto em direção a outro objeto
<b>Move Away From</b>	Afasta o objeto de outro objeto
<b>Move To</b>	Move o objeto de sua posição atual até o ponto central do objeto de destino
<b>Move and Orient To</b>	Move o objeto de sua posição atual até o ponto central do objeto de destino e ajusta a orientação do objeto para que corresponda à orientação do objeto de destino
<b>Delay</b>	Encerra o movimento de um objeto durante determinado número de segundos O procedimento Delay pode ser usado para desacelerar a execução de uma animação
<b>Say</b>	Cria uma bolha de diálogo com texto que faz com que o objeto pareça estar falando

# Exemplos de Procedimentos de Rotação

Procedimento	Descrição
<b>Turn</b>	Gira um objeto para a esquerda, para a direita, para frente ou para trás em seu ponto central Giro para a esquerda ou para a direita no eixo vertical do objeto; giro para frente ou para trás no eixo horizontal do objeto
<b>Roll</b>	Rola um objeto para esquerda ou para a direita em seu ponto central usando o eixo horizontal do objeto

Turn e Roll são os dois procedimentos que costumam ser combinados com mais frequência quando se inicia o Alice. A escolha do procedimento errado geralmente cria algumas animações "interessantes".

# Criar uma Instrução de Programação

- Na guia Procedures, clique e arraste o procedimento desejado até myFirstMethod no Code editor



## Selecionar e Definir Valores de Argumentos

- Depois que a instrução de programação for criada, use os menus drop-down para definir os valores de cada argumento
- Para acessar o menu drop-down do argumento, clique no triângulo que apontada para baixo à direita do valor do argumento



Os valores de toda instrução que foi adicionada ao code editor podem ser alterados a qualquer momento.

## Executar o Programa

- Clique no botão Run para executar as instruções de programação
- Execute o programa com frequência para testar os resultados desejados e, se necessário, altere os valores dos argumentos

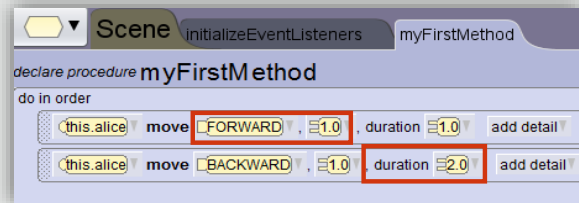


Como acontece em todos os ambientes de programação, nunca crie uma quantidade de código muito extensa antes de testar se o código funciona. Codifique pequenas animações e teste-as regularmente.



# Argumentos

- Os argumentos são selecionados depois que o procedimento é lançado no Code editor
- Os tipos de argumentos podem incluir o seguinte:
  - Objeto
  - Direção
  - Valor da direção
  - Duração de tempo
  - Texto

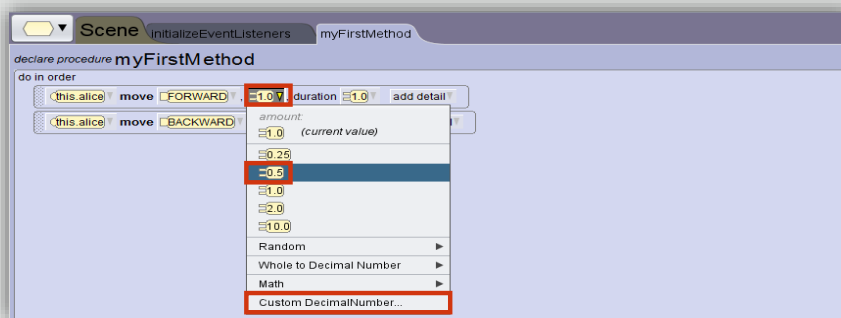


Um argumento é um valor que o procedimento usa para completar sua tarefa. Um programa de computador usa argumentos para informar como implementar o procedimento.

Os argumentos no Alice ensinam sobre o uso de variáveis dentro de outros ambientes de programação. E os valores deles podem ser alterados pelo código a qualquer momento.

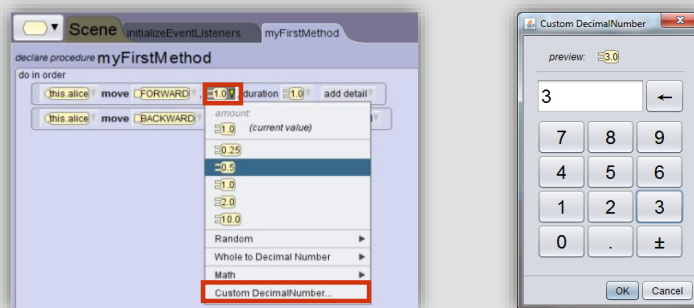
# Menu Argument

- O menu Argument oferece valores de argumentos padrão que podem ser selecionados
- Se nenhum dos padrões for adequado, selecione a opção Custom DecimalNumber... para especificar um valor de argumento mais preciso



## Etapas para Editar Argumentos

- Ao lado do valor do argumento, clique na seta para exibir o menu de valores
- Selecione um novo valor
- Use a opção Custom DecimalNumber... para especificar um valor que difere da lista padrão de valores



Usando esses métodos, você conseguirá atribuir aos argumentos qualquer valor que esteja dentro do escopo do tipo do argumento.

## Argumentos como Espaços Reservados

- Quando um procedimento é lançado no Code editor, todos os valores dos argumentos devem ser especificados
- Haverá vezes em que você seleciona o valor de um argumento como um espaço reservado temporário que será substituído mais tarde

À medida que aprender a construir estruturas de codificação mais complexas, você usará cada vez mais valores de espaços reservados no Alice 3.

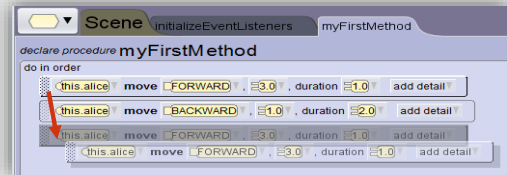
# Argumentos como Espaços Reservados

- Por exemplo
  - Talvez você queira que um objeto mova-se para frente, mas não sabe ao certo a distância desse deslocamento
  - Selecione um valor de espaço reservado igual a 2 metros, execute a animação e verifique se é preciso selecionar um valor diferente. Caso seja, edite o valor
  - Você também pode especificar um valor de espaço reservado que será substituído mais tarde por uma função ou uma variável

À medida que aprender a construir estruturas de codificação mais complexas, você usará cada vez mais valores de espaços reservados no Alice 3.

## Etapas para Reordenar Instruções de Programação

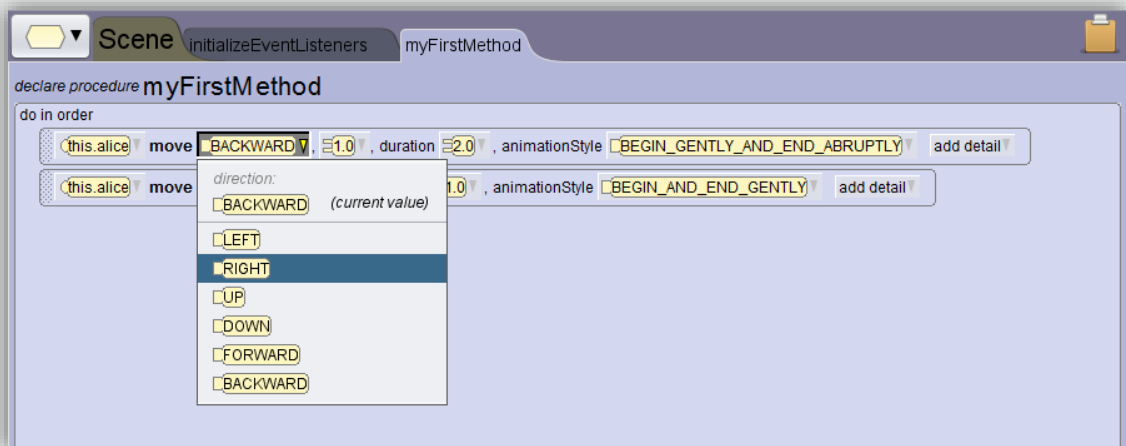
- Usar "**Arrastar e Soltar**":
  - selecione a alça da instrução de programação
- Arraste a instrução de programação até a nova posição
- Solte a instrução de programação em sua nova posição cancelando a seleção da alça\*



- **Observação:** uma linha de indicador de posição verde aparecerá para ajudar você a alinhar a instrução de programação na posição desejada

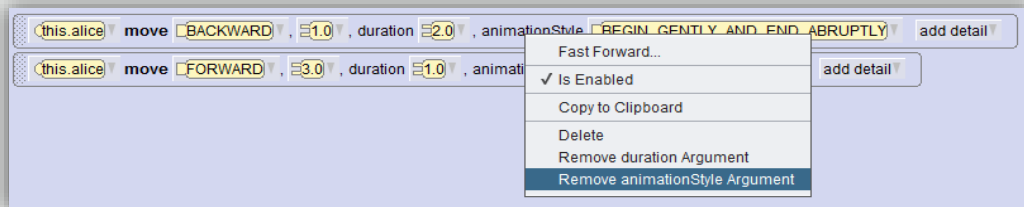
# Editar Instruções de Programação

- Use as listas drop-down para editar os valores de uma instrução de programação

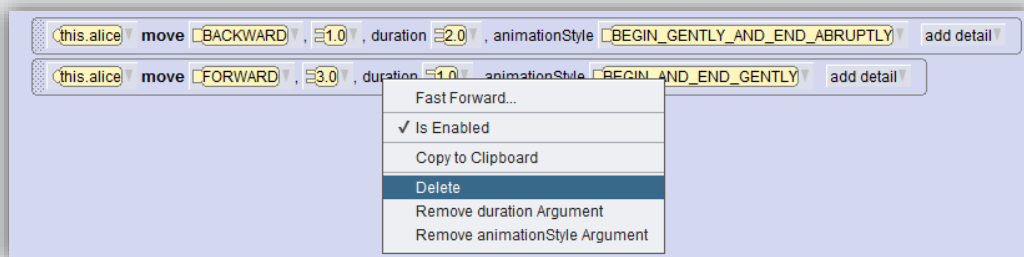


## Excluir Instruções de Programação

- Clique com o botão direito do mouse nas instruções de programação para remover partes da instrução



- Ou exclua a instrução inteira



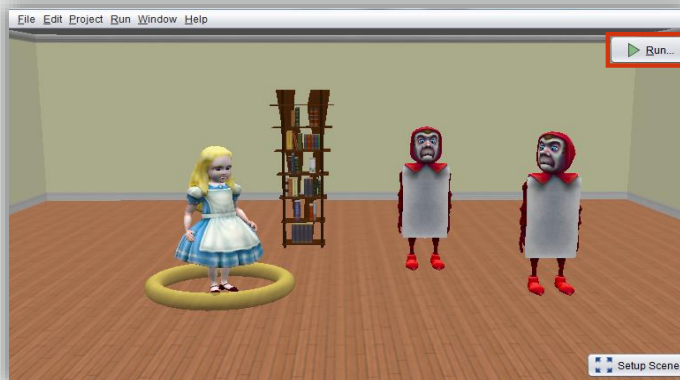
A exclusão de partes da instrução evita que você precise excluir e recriar instruções de código desde o início. Esta é uma ferramenta muito útil dentro do Alice que economiza bastante tempo.

Se você excluir uma instrução e, em seguida, se arrepender, poderá sempre usar a opção desfazer (CTRL + Z) para cancelar a exclusão.



## Editar e Testar o Programa

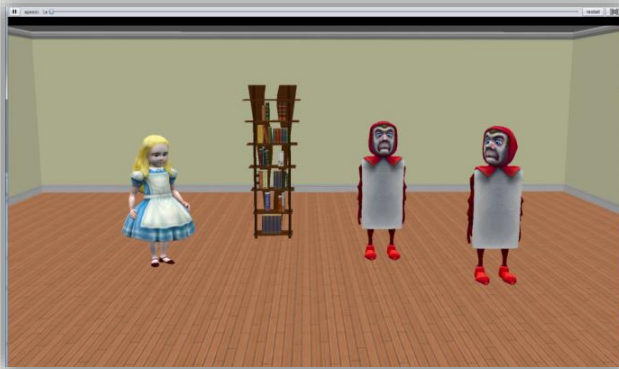
- Execute a animação para testá-la e, se necessário, edite o código
- Talvez sejam necessários vários ciclos de teste e edição para fazer com que a animação seja executada como desejado



Após cada alteração, é necessário testar novamente a animação. Pode ser que algo que estava funcionando perfeitamente antes não esteja mais funcionando como deveria devido a uma alteração que foi feita em alguma parte.

## Depurando e Testando

- Depuração e teste consistem no processo de executar a animação várias vezes e ajustar as instruções de controle, os procedimentos e os argumentos depois de cada execução
- Durante a depuração, salve o programa com frequência



## Inserir Instruções de Programação Temporárias para Ajudar na Depuração

- Você pode inserir instruções de programação temporárias no código para ajudar na depuração
- Por exemplo:
  - imagine que você tenha um objeto que não está se movendo para frente
- Insira temporariamente uma instrução de programação Say para anunciar que o objeto está prestes a se mover para frente

A utilização de instruções de teste pode ser muito útil porque ela permite a você obter informações sobre o fluxo do código. E, por meio da exibição desse fluxo na tela, você pode avaliar o resultado.

## Inserir Instruções de Programação Temporárias para Ajudar na Depuração

- Teste o programa para ver se a instrução de programação Say é executada ou não
- Se ela for executada, mas o objeto não se mover, isso indica um tipo de problema
- Se nem a instrução Say nem a instrução Move forem executadas, isso pode indicar outro tipo de problema

A utilização de instruções de teste pode ser muito útil porque ela permite a você obter informações sobre o fluxo do código. E, por meio da exibição desse fluxo na tela, você pode avaliar o resultado.

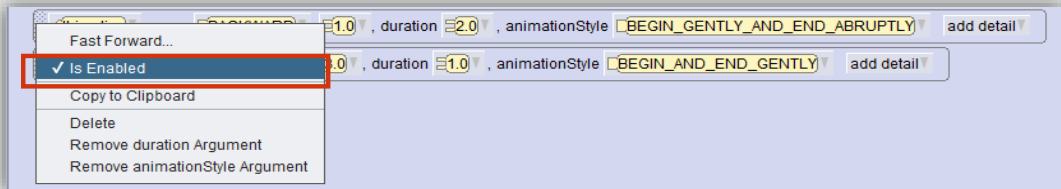
## Desativando Instruções de Programação

- As instruções de programação podem ser desativadas no Code editor
- Desative as instruções de programação para:
  - Ajudar a isolar partes do código durante o teste
  - Ajudar a focar na programação, no teste e na depuração de uma instrução específica

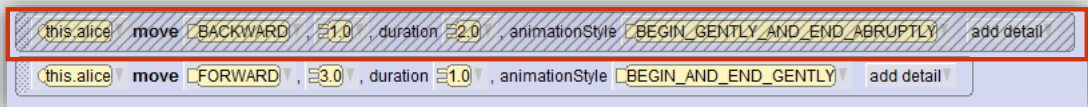
A desativação de blocos de código pode economizar muito tempo, evitando que você precise ficar constantemente testando um código que já foi totalmente testado.

## Etapas para Desativar uma Instrução de Programação

- Clique com o botão direito do mouse na instrução de programação
- Desmarque a seleção de "Is Enabled" na lista drop-down



- A cor da instrução de programação mudará para marcas tracejadas cinzas para indicar que a instrução está desativada



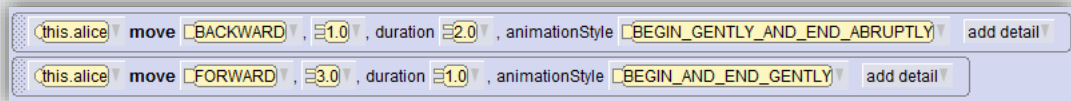
Sempre ative novamente todo o código antes de teste final.

## Etapas para Ativar de Novo uma Instrução de Programação Desativada

- Clique com o botão direito do mouse na instrução de programação que foi desativada
- Selecione "Is Enabled" na lista drop-down



- Isso ativa novamente o código e remove as linhas cinzas



## Comentários de Programação

- A inclusão de comentários de programação em uma animação ajuda os usuários a compreenderem o fluxo da programação
- Comentários:
  - Descrevem a intenção das instruções de programação
  - Não afetam a funcionalidade do programa porque são ignorados durante a execução
  - Em geral, são colocados acima do bloco de instruções de programação que descrevem
  - Nos programas grandes, geralmente são escritos primeiro, como um resumo das instruções de programação

Os comentários são cruciais para entender como o código funciona. Vale a pena adquirir o hábito de adicionar comentários durante a codificação. Nunca deixe para inseri-los no fim; é bem mais fácil adicioná-los à medida que você cria o código.

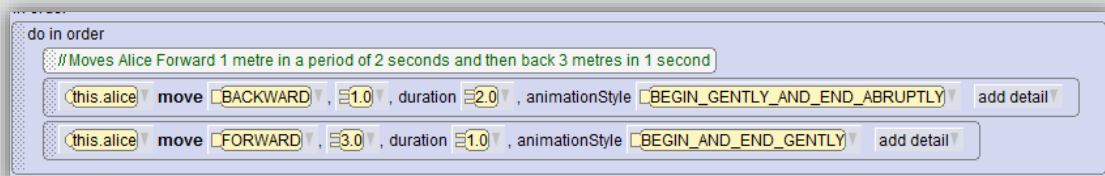


## Etapas para Inserir Comentários em um Programa

- Arraste e solte a caixa de comentários acima de um segmento de código
- A caixa de comentários está localizada na parte inferior de myFirstMethod

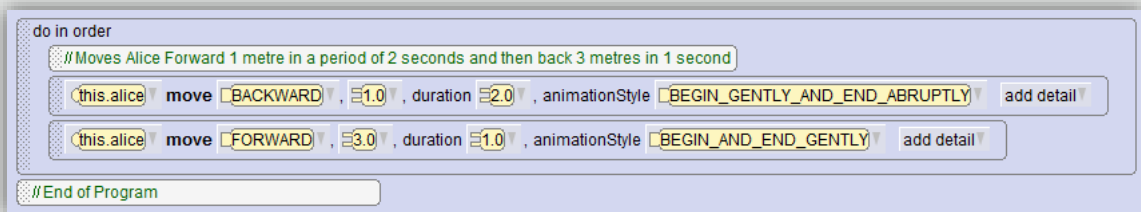


- Digite comentários que descrevam a sequência de ações no segmento de código



# Usando Comentários para Organizar o Programa

- Os comentários podem ser uma ótima ferramenta para organizar o desenvolvimento de um programa
- No caso de programas maiores, crie um comentário que indique o fim do programa
- Um comentário "fim do programa" reduz a rolagem quando se adiciona instruções de programação a um procedimento myFirstMethod muito longo



Os comentários não são compilados como parte do código. Portanto, não afetam o modo como o programa é executado. Você pode adicionar quantos comentários desejar. Eles podem ser usados para explicar a função e o layout do código.

# Terminologia

- Estes são os principais termos usados nesta lição:
  - Argumento
  - Code editor
  - Comentários
  - Funções
  - Painel Methods
  - Orientação
  - Procedimento

# Resumo

- Nesta lição, você deverá ter aprendido a:
  - Alternar entre o Scene editor e o Code editor e descrever as diferenças visuais existentes entre eles
  - Localizar e descrever a finalidade do painel Methods e da guia Procedures
  - Usar procedimentos para mover objetos
  - Adicionar procedimentos de programação ao Code editor
  - Demonstrar como é possível alterar os valores dos procedimentos
  - Criar comentários de programação
  - Reordenar, editar, excluir, copiar e desativar instruções de programação
  - Testar e depurar uma animação



The Oracle Academy logo is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font, with the letters "R" and "A" slightly overlapping. Below "ORACLE" is the word "Academy" in a dark gray, sans-serif font. The entire logo is framed by a thin black border, with dark gray horizontal bars at the top and bottom.

# ORACLE

## Academy