Parameters in Alice 3 Assessment

By Yossra Hamid
Under the supervision of Professor Susan Rodger
Duke University
November 2015

This is a multiple-choice assessment, designed for beginners in Alice. It should take approximately 10 minutes.

For preparation, complete the “Bunny Visits - Parameters” Tutorial.
1. Which correctly describes what a **PARAMETER** is?

   a. Piece of information that a method/function needs to execute correctly *(Example: Giving Alice a distance when using the “move” method)*
   
   b. Three-dimensional shape such as a person, animal, piece of furniture, building or anything else put in an Alice world
   
   c. Determines whether or not the object is visible.
   
   d. Something that tells the world or objects when to carry out methods

2. For which scenario would you need a parameter?

   a. You want one of your characters to perform a different action for every character they walk up to in your world.
   
   b. You want to create a list of friends in the world
   
   c. You want to store the color of an object to use later in your world
   
   d. You want to find out the height of one of the objects in your world

3. Which of these is a **STRING** parameter?

   a. ![Parameter 1](image1)
   
   b. ![Parameter 2](image2)
   
   c. ![Parameter 3](image3)
   
   d. ![Parameter 4](image4)
4. Which event correctly uses the parameter in the `hippo.visit` method to have the bunny greet the object that is clicked on?

a. 

```
declare procedure mouseClicked (event getScreenDistanceFromLeft) do in order
this.hippo visit friend: this.chicken
```

b. 

```
declare procedure mouseClicked (event getModelAtMouseLocation) do in order
this.hippo visit friend: this.tortoise
```

c. 

```
declare procedure mouseClicked (event drop statement here) do in order
this.hippo greet friend: this.chicken
drop statement here
```

d. 

```
declare procedure mouseClicked (add detail) do in order
this.hippo add detail
```

5. Which of these has a section that can be replaced by the parameter, “distance”:

a. 

```
declare procedure mouseClicked (event drop statement here) do in order
this.hippo move up, 1.0
```

b. 

```
declare procedure mouseClicked (add detail) do in order
this.hippo say "hello"
```

c. 

```
declare procedure mouseClicked (add detail) do in order
this.hippo turnToFace this.camera
```

d. 

```
declare procedure mouseClicked (add detail) do in order
while true isFace this.camera
```

```
```
```
```
```
```