Functions

Alice
Definition of *Function*

An instruction that returns a result.
Built-in Functions

We have been using built-in functions.
How a function works

A **function** receives value(s), performs some computation on the value(s), and returns (sends back) a value.
Result of Functions

A value
- The current value of a property
- The result of a comparison of objects (using properties)
- The result of an expression that uses parameters and/or properties

The *return type* of a function describes the kind of value it returns, (the operation being performed)
- a number
- a specific object
- a color
- a Boolean (true or false)
- other property values…
Rolling the ball

Suppose we want to roll a ball along the ground.
We want a realistic motion rather than a slide.
The ball must simultaneously move and roll.

Do together
move ball forward 1 meter
turn ball forward 1 revolution

realisticRoll
Demo

**realisticRoll - first try**

◮ Our design assumed that the ball's motion is relative to the ground. But the ball turns relative to its own sense of direction.

**realisticRoll - second try**

◮ AsSeenBy ground can be used to make the ball turn relative to the ground.
Revising the approach

The ball is made to roll 1 revolution. Suppose we want the ball to roll a certain distance (say 3 or 4 or even 10 meters) along the ground.

How many times should the ball turn 1 complete revolution?
Number of revolutions

The number of revolutions depends on the size of the ball

The number of revolutions is \( \frac{\text{distance}}{\pi \times \text{diameter}} \)

But there is no built-in function to return the number of revolutions

We will write our own!
Parameters

We want to return the value computed as:

\[
\text{distance/}(\ PI \ast \ diam)\]

Obviously, what is needed is:

- the ball’s diameter
- the ball object has a built-in \textit{width} function
- the distance the ball is to travel
- can be sent as a parameter to the function
Demo

**realisticRoll - third attempt**

Concepts illustrated in this example

- A function must have a *return* statement to send back a computed value.
- In this example, a **math expression** is created to compute a number value.
- The order of evaluation is indicated by the use of parentheses, as in traditional math expressions.
Create a function

1. Function tab
2. Click on button
3. Name the function
4. Select the return type
5. Click Okay
Function Editor

Functions return values
Write the function

```
( distance \rightarrow \left( 3.14 \times \frac{\text{subject} = \text{toyBall1}'s \text{width}}{\text{distance}} \right) \right)
```
Calling the function

We should run the animation with several test values to be sure it works as expected.

What happens if you use a negative value?
Levels of Functions

As with methods, functions can be either class-level or world-level. (The function just presented was class-level.)

The guidelines for class-level methods also apply to class-level functions:

- No references to other objects.
- No references to world-level functions you have written (built-in world-level functions are fine to use).
Execution Control with If/Else and Boolean Functions

Example: Single Condition

Alice
Thinking About More Advanced Worlds

No doubt you have started to think about building animations like simulations and video games…

To build more advanced worlds, you will need to write code that involves decisions
Examples of Decisions

In a car-race simulation, the driver steers the car around curves and past mile-markers.

- If the car stays on the road, the score increases.
- If the car goes off the road into the stands, the car crashes.
- If the driver gets the car over the finish-line, the time is posted and the driver wins!
Logical Expressions

A decision is made based on current conditions.

A condition is checked in a logical expression that evaluates to true or false (Boolean) value.

- Car on road \(\rightarrow\) true
- Car over finish line \(\rightarrow\) false
If/Else

In Alice, a logical expression is used as the condition in an If/Else control structure.

Decisions (using If/Else) are used in
- functions
- methods
Example: Boolean Function

Suppose you are building a simulation system used to train air traffic controllers.

One of the tasks of an traffic controller is to be alert for possible collisions in the flight space.
One factor in determining whether two aircraft are in danger of collision is the vertical distance (difference in altitudes) between them.

We can write a function that checks the vertical distance against a minimum difference in altitudes.

The function returns \textit{true} if they are too close, otherwise \textit{false}.

\begin{verbatim}
isTooClose

Parameters: \textit{aircraftOne, aircraftTwo, minimumDistance}

If the vertical distance between \textit{aircraftOne} and \textit{aircraftTwo} is less than \textit{minimumDistance}
    return \textit{true}
Else
    return \textit{false}
\end{verbatim}
Demo

FlightCollision

Concepts illustrated in this example

A world-level relational operator is used to create a Boolean expression as the condition.

The absolute value function is used to make sure the computed difference in altitude is not a negative number.
To avoid a collision, the aircraft that is above the other should move up and the lower aircraft should move down.

```plaintext
avoidCollision

Parameters: aircraftOne, aircraftTwo

If aircraftOne is above aircraftTwo
   Do together
      aircraftOne move up
      aircraftTwo move down
Else
   Do together
      aircraftOne move down
      aircraftTwo move up
```
Demo

Concepts illustrated in this example

- Decisions were made to
  - control whether a method is called
  - determine which set of instructions are immediately executed