IOBJECTTYPE SCRIPT INTERFACE

View online: https://www.construct.net/en/make-games/manuals/construct-3/scripting/scripting-reference/object-interfaces/iobjecttype

---

The `IObjectType` script interface represents an object type in the project, e.g. a Sprite object type. An object type can have multiple instances created, which are represented by the IWorldInstance script interface if it appears in a layout, otherwise the IInstance interface.

`IObjectType` derives from IObjectClass which is a base class shared with families (IFamily).

Getting an IObjectType

References to the project's object types are typically accessed through the IRuntime interface `objects` property. For example `runtime.objects.Sprite` would refer to the `IObjectType` interface for the *Sprite* object type, assuming one was added to the project.

> *Try not to confuse object classes with object instances. A common mistake is to try to use something like `runtime.objects.Sprite.x` to get the X co-ordinate of a Sprite instance. However `runtime.objects.Sprite` is an IObjectType, which does not have a position. First add another call to get an instance before trying to read instance properties, for example `runtime.objects.Sprite.getFirstInstance().x`.*

Examples

The following examples demonstrate using some features of `IObjectType`:

- Instance destroy event demonstrates the use of the `"instancedestroy"` event
- Iterating instances demonstrates the use of the `instances()` iterator to modify all instances

Object type APIs

---

setInstanceClass(Class)

Set a custom class to be used to represent instances of this object type. The class must derive from the default class. This can only be called in `runOnStartup`, before any instances have been created. For more information see the guide on subclassing instances.

---

createInstance(layerNameOrIndex, x, y, createHierarchy, template)

Create a new instance of the object type at a position. The layer to create on is specified either by a case-insensitive string of the layer name or its zero-based index. The position is

given in layout co-ordinates. If `createHierarchy` is true, all children of the created instance in the scene-graph hierarchy will also be created automatically with their connections in place. If `template` is a valid template name then the new instance will be based on the template rather than an arbitrary instance. The return value is the script interface for the created instance.

> *See Setting up a hierarchy in the Layout View manual entry for more information about hierarchies.*

> *See the Templates manual entry for more information on what templates are and how to start using them.*

---

getAllFamilies()

Return an array of IFamily listing all the families this object type belongs to.

***families()**

Iterates over all the families this object type belongs to (as `IFamily`).

---

isInFamily(family)

Returns a boolean indicating if this object type belongs to the specified `IFamily`.