|  |
| --- |
| Circle Language Spec: Objects |

## Attributes

An attribute is value stored inside an object. An attribute is actually a sub-object with special properties.

An attribute is often an object of a simple nature, for instance a number, a boolean, a date or a simple piece of text. But an attribute can really be any type of object.

The main aspect that makes a sub-object an attribute, is the fact that it has a fixed logical residence inside its parent. The parent is the sole container of it. It is not an object that can be shifted around like other objects. It is fixed inside the parent. An object usually doesn’t get a fixed logical residence, so this is what’s special about an attribute.

Another aspect, that makes a sub-object an attribute, is that it is always created, never destroyed, never recreated again, and a different object is never assigned to it. Another object’s *state* can be assigned to it, but that’s all that can be changed about it.

A third aspect of an attribute, is that is usually has a small piece of binary data stored directly inside it. For instance a number is stored as a piece of binary. Direct storage of a piece of binary data is not required for an object to be considered an attribute, though.

A logical aspect of an attribute, is that it is more part of what the object *is*. The other sub-objects are often just considered references to other objects, not as much part of the object itself.

You can point to an attribute, but an attribute can *not* be a pointer to something else.

If an attribute either becomes a pointer, gets destroyed or recreated it would loose its status as attribute, and will just be a related object again.

Any object reference could be made an attribute. The attributes are scattered around an object’s list of related items. The attributes could be returned by the object as a separate list, instead of looking them up in an object’s collection of related objects.

Here follows a summary of the aspects of an attribute:

- Requirements:

- Fixed logical residence inside an object

- Always created, never destroyed, never recreated, never a pointer

- Logically:

- More part of what the object *is*

- Usually:

- Has a binary value stored directly inside it

- Is an object of a simple nature

- And an extra:

- An object can return a list of just its attributes, instead of just all related objects.

An attribute is a value stored inside an object. An attribute is actually a sub-object with special properties, as explained in the article *Attributes*. Because an attribute is an object, it is represented by a circle drawn with a solid line.



An attribute is often an object of a simple nature, for instance a number, a date or a simple piece of text. The value is textually expressed. An object’s literal value is shown inside the object, usually in the center of the symbol:



Or closer to the top of the symbol if any other symbols are shown inside the attribute.

