

INSTANCE VARIABLES

View online: <https://www.construct.net/en/make-games/manuals/construct-3/project-primitives/objects/instance-variables>

Instance Variables are added to **object types** but store numbers, text or booleans (on/off flags) individually for each **instance**. This makes them ideal for things like health counters in a game, since each instance tracks its own value. Instance variables are added to object types with the **Object Instance Variables dialog**, and the initial values for each instance can be set from the **Properties Bar**.

[Click here to open an example of instance variables.](#)

Instance variables can also be used to help control instances independently of each other. For example, a Boolean instance variable could be used to determine if an enemy is hunting down the player (*true*) or running away (*false*). If instances all have different values, the condition *Is boolean instance variable set* can be used to apply actions to enemies hunting down the player. Inverting the condition (picking instances with the value being *false*) can then be used to apply actions to enemies running away. The end result is a number of instances of the same object type acting independently: some chasing and others running away. This is a simple example - much more complex methods can be made using multiple instance variables. In other words, an instance's *state* can be controlled using instance variables.

Instance variables can also be added to **Families** Paid plans only. All the object types in the family then *inherit* the instance variable.

Autocompleting string instance variables

When using string instance variables, Construct will offer to autocomplete the instance variable with other strings it is referenced with in both event sheets and properties. The autocomplete options will appear in both the **Parameters Dialog** (after typing the first `"` character) and the **Properties Bar**.

This is useful for string instance variables that represent a fixed set of states, such as "idle", "searching" and "attacking". If your event sheets or properties reference a set of strings like this, then they will be offered for autocomplete in properties and parameters, helping show the list of available strings and avoiding typos from re-entering the values.