

EVENT VARIABLES

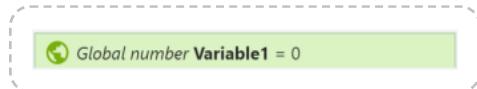
View online: <https://www.construct.net/en/make-games/manuals/construct-3/project-primitives/events/variables>

Event variables are number or text values which are either **global** to the whole project or **local** to a range of events. They are modified using the **Event Variable dialog**. To add an event variable, **right-click** on an event, another variable, or an empty space in the event sheet, and select Add global variable or Add local variable, or press the **V** keyboard shortcut. Variables at the root level of the event sheet (not indented beneath anything else) become global variables, whereas variables in groups or sub-events become local variables.

Event variables are modified with the system actions in the *Global & local variables* category. They can be retrieved by simply using their name in **expressions**.

Global variables

Global variables show a globe icon. They are always at the top level of an event sheet - they are not **sub-events** or inside any **groups**.



Global variables store their values between layouts. Events in any layout can access any global variable, even if it was created in a different event sheet that is not **included**.

Global variables can be moved to another event sheet by cutting and pasting them. After being cut, references to the global variable will disappear because it has been removed; this is normal and nothing to worry about. When you paste the global variable, the references that disappeared will reappear again. Alternatively you can **right-click** the global variable and select Move to event sheet....

Local variables

Local variables are variables placed nested under other events, or inside a group. They also show with a different icon to global variables.



The main difference between global and local variables is **local variables can only be accessed in their scope**. A local variable's scope is its level of sub-events. All other events at the same level of indentation, or lower levels, can access the local variable. Events above it (less indented) cannot access the local variable.

For example, if an event variable is in a group of events, it becomes a local variable. Then, it will only appear as an option for a variable in events inside that group. In other groups or in other event sheets it does not appear at all and cannot be accessed. This makes the variable *local* to the scope in which it is placed.

Local variables convenient for temporarily holding variables over a short range of events, such as to calculate an average value (where a temporary *sum* variable may be necessary). It also helps keep the project simple, since it prevents the need to create more global variables, which appear everywhere in the project even if they are not needed everywhere.

The scope of local variables is designed to mimic how the scope of variables works in real programming languages.

Function parameters

Function parameters are a special kind of local variable, scoped to a function event. For more information see the section on [Functions](#).

Static and constant variables

By default, local variables reset to their initial value whenever entering their scope (usually every tick), like local variables in programming languages. If the variable is marked *static* in the Event Variable dialog it will persist its value permanently, like a global variable.

Both global and local variables can be marked *constant*. This makes them read-only: they can be retrieved and compared, but not changed.

Finding references to variables

Paid plans only You can quickly see a list of all references to a global or local variable by **right-clicking** it and selecting Find all references.... This will open the [Find Results bar](#) with a list of all places in the project the variable is used. This is also helpful for identifying if there are no references so the variable can be safely deleted.