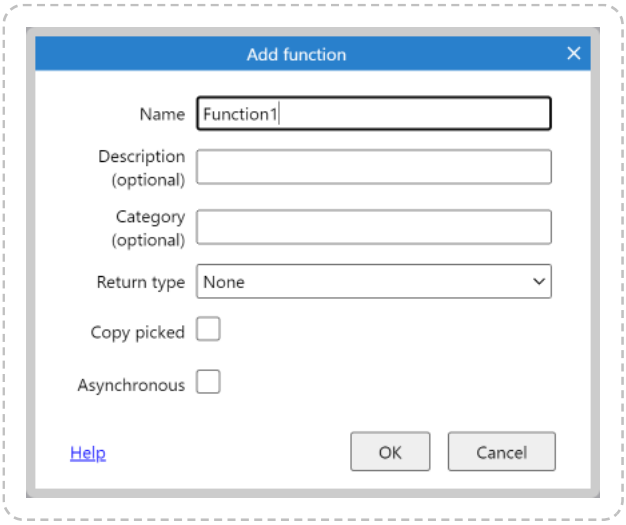
ADD/EDIT FUNCTION DIALOG

View online: https://www.construct.net/en/make-games/manuals/construct-3/interface/dialogs/function

The add/edit function dialog appears when adding or editing a function.



This dialog has the following fields.

Name

The name of the function. Note that if the function is used as an expression (its *Return type* is not *None*), the name must be a valid expression, so cannot contain special characters or whitespace. Functions used as actions (with a *Return type* of *None*) can use any name.

Description Optional

An optional description of the function, for your organisational purposes. This is displayed in the Add action dialog or Expressions dictionary depending on the return type of the function, and can be a helpful reminder of what the function does.

Category Optional

An optional category for the function, for your organisational purposes. Functions with the same category are grouped together in the *Add action dialog* and *Expressions dictionary*, providing a way to arrange related functions together. This field autocompletes with existing category names used in the project, making it easier to use the same category names.

Return type

The return type of the function. This also determines whether the function is used as an action or an expression. Functions with a return type of *None* are used as actions; otherwise they are used as expressions. A return type of *Any* means the function can return either a

number or a string. Functions with a return type must have a name which is a valid expression, so cannot contain special characters or whitespace.

> *Once a function is used in your project, the return type cannot be changed.*

Copy picked

Normally calling a function will run its actions with all objects reset to picked. For example calling a function that modifies a Sprite will modify all instances of that Sprite regardless of whether any conditions picked specific instances before calling the function. Enabling *Copy picked* means the function will run with the same picked instances as the event that calls it, so actions still run on the same instances picked by any previous conditions.

Asynchronous

Tick to mark the function as *asynchronous* (or *async* for short). This means calls to this function can be used with the System action *Wait for previous actions to complete* if the function does any of its own waiting. Note this has a small performance overhead, so for best performance leave this disabled if you don't need it.