

# SetPedControlState

This function makes a ped or player press or release a certain control.

**Note:** You can't use `enter_exit` or `enter_passenger` on a ped. Please use `setPedEnterVehicle` and `setPedExitVehicle`.

**Note:** To make a ped crouch, set the control to *true* and reset to *false* after one frame. Use `setTimer` with 0ms. Do the same to make the ped stand again.

## Syntax

```
bool setPedControlState ( ped thePed, string control, bool state )
```

**OOP Syntax** Help! I don't understand this!

**Method:** *Ped:setControlState(...)*

## Required Arguments

- **thePed:** the ped you want to press or release a control.
- **control:** the name of the control of which to change the state. See control names for a list of valid names.
- **state:** the new control state. *true* means pressed, *false* is released.

## Returns

Returns *true* if successful, *false* if otherwise.

## Remarks

If you set the control state on the `localPlayer` element, then the control state is synchronized to every other client as if the input was made through the game input. But if you set any control states to server-side peds then the control state is not automatically synchronized. This behaviour is unaffected by being the syncer of the server-side ped or not. Let's assume that the syncer client calls `setPedControlState` on the ped. The other game clients - the ones where `setPedControlState` was not manually called by scripts - will see the ped teleport around. It will clearly look as if synchronization were broken.