Shared function

SetPedAnimation

Sets the current animation of a player or ped. Not specifying the type of animation will automatically cancel the current one.

> **Warning:** It is possible that an animation will be cancelled if you use setElementFrozen on the ped, but this does not happen all the time.

Syntax

```
bool setPedAnimation ( ped thePed [, string block = nil, string anim = nil, int time = -1, bool loop = true, bool updatePosit
ion = true,
                       bool interruptable = true, bool freezeLastFrame = true, int blendTime = 250, bool retainPedState = fal
se ] )
```

OOP Syntax Help! I don't understand this!

Method: *ped:setAnimation(...)*
Counterpart: *getPedAnimation*

Required Arguments

- **thePed:** the player or ped you want to apply an animation to.

Optional Arguments

NOTE: When using optional arguments, you might need to supply all arguments before the one you wish to use. For more information on optional arguments, see optional arguments.

- **block:** the animation block's name.
- **anim:** the name of the animation within the block.
- **time:** how long the animation will run for in milliseconds.
- **loop:** indicates whether or not the animation will loop.
- **updatePosition:** will change the actual coordinates of the ped according to the animation. Use this for e.g. walking animations.
- **interruptable:** if set to *false* other tasks wont be able to interupt the animation. Setting this to 'false' also gives this function more power to override other animations that are running. For example, squatting after a jump can be terminated.
- **freezeLastFrame:** if set to *true* after animation the last frame will be frozen, otherwise the animation will end and controls will return.
- **blendTime:** how long the animation will mixed with the previous one in milliseconds.

- **retainPedState:** will restore the task which was playing before calling this function. Useful for restoring the crouch task after animation ends. This may be extended in the future to support other states/tasks.

Returns

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