

GetPedTask

This function is used to get any simple or complex task of a certain type for a ped.

It can provide feedback on all tasks relating to a ped. For example, while jumping, getPedSimplestTask will return TASK_SIMPLE_IN_AIR. If you wanted to know specifically if the player has jumped, you would use this function. If you did you will discover that while jumping Primary task 3 is TASK_COMPLEX_JUMP.

Syntax

```
string, string, string, string getPedTask ( ped thePed, string priority, int taskType )
```

OOP Syntax Help! I don't understand this!

Method: *ped:getTask(...)*

Required Arguments

- **thePed:** The ped whose task you want to retrieve.
- **priority:** A string determining which set of tasks you want to retrieve it from. This must be either "primary" or "secondary".
- **taskType:** An integer value representing the task type (or slot) you want to get the task from. Types can be:
 - **PRIMARY TASKS**
 - **0:** TASK_PHYSICAL_RESPONSE
 - **1:** TASK_EVENT_RESPONSE_TEMP
 - **2:** TASK_EVENT_RESPONSE_NONTEMP
 - **3:** TASK_PRIMARY
 - **4:** TASK_DEFAULT
 - **SECONDARY TASKS**
 - **0:** TASK_SECONDARY_ATTACK
 - **1:** TASK_SECONDARY_DUCK
 - **2:** TASK_SECONDARY_SAY
 - **3:** TASK_SECONDARY_FACIAL_COMPLEX
 - **4:** TASK_SECONDARY_PARTIAL_ANIM
 - **5:** TASK_SECONDARY_IK

Returns

Returns the name of the most complex task. See list of player tasks for valid strings. Returns *false* if invalid arguments are specified or if there is no task of the type specified.

Returns between 1 and 4 strings. The first string contains the name of the most complex task, with simpler sub-tasks being named in the following strings. See list of player tasks for valid strings. Returns *false* if invalid arguments are specified or if there is no task of the type specified.