Shared function

GetWeaponProperty

This function gets a weapon property of the specified custom weapon (clientside only) or specified player-held weapon (both client and server).

Syntax

```
int getWeaponProperty ( int weaponID/string weaponName, string weaponSkill, string property )
```

OOP Syntax Help! I don't understand this!

> **Note**: *This OOP syntax is for custom weapons only.*
> **Method**: *weapon:getProperty(...)*
> **Counterpart**: *setWeaponProperty*

Required Arguments

- **weaponID or weaponName:** The ID or name of the weapon you want to get info of. Names can be:

  - grenade
  - teargas
  - molotov
  - colt 45
  - silenced
  - deagle
  - shotgun
  - sawed-off
  - combat shotgun
  - uzi
  - mp5
  - ak-47
  - m4
  - tec-9
  - rifle
  - sniper
  - rocket launcher
  - rocket launcher hs
  - flamethrower
  - minigun
  - satchel
  - bomb
  - spraycan
  - fire extinguisher
  - camera

- **weaponSkill:** Either: "pro", "std" or "poor"
- **property:** The property you want to get the value of:

  - "weapon_range" - *float*
  - "target_range" - *float* - **Max targeting range**
  - "accuracy" - *float*
  - "damage" - *int* - **Note: Changing the standard M4 stat will change how much damage vehicle guns (e.g: Rustler) do.**
  - "maximum_clip_ammo" - *int*
  - "move_speed" - *float* - **How fast player can move with weapon**
  - "flags" - *int* - **(specify a flag to toggle it on/off) See Weapon Flags**

    - "flag_aim_no_auto" - *bool* - **Disable auto up/down for non-aimed firing**
    - "flag_aim_arm" - *bool* - **Uses other arm for aiming**
    - "flag_aim_1st_person" - *bool* - **Uses 1st person aim**
    - "flag_aim_free" - *bool* - **Can only use free aiming**
    - "flag_move_and_aim" - *bool* - **Can move and aim at same time**
    - "flag_move_and_shoot" - *bool* - **Can move and fire at same time**
    - "flag_type_throw" - *bool* - **Is a throwing weapon**
    - "flag_type_heavy" - *bool* - **Can't jump**
    - "flag_type_constant" - *bool* - **Fires every frame within loop (ie paint spray)**

    - "flag_type_dual" - *bool* - **Can use 2x guns at same time**
    - "flag_anim_reload" - *bool* - **Weapon has reload anims**
    - "flag_anim_crouch" - *bool* - **Has crouching anims**
    - "flag_anim_reload_loop" - *bool* - **Loop from end of reload to fire loop start**
    - "flag_anim_reload_long" - *bool* - **Force a longer reload time**
    - "flag_shot_slows" - *bool* - **Slows down (area effect)**
    - "flag_shot_rand_speed" - *bool* - **Random speed (area effect)**
    - "flag_shot_anim_abrupt" - *bool* - **Force the anim to finish player after aim/fire rather than blending out (area effect)**
    - "flag_shot_expands" - *bool* - **Expands (area effect)**

- "anim_loop_start" - *float* - **Start of aimed firing animation loop**
- "anim_loop_stop" - *float* - **End of aimed firing animation loop (Reduce to increase firing rate)**
- "anim_loop_bullet_fire" - *float* - **Time in aimed firing animation when weapon should be fired (Must be between Start and End)**
- "anim2_loop_start" - *float* - **Start of non-aimed firing animation2 loop**
- "anim2_loop_stop" - *float* - **End of non-aimed firing animation2 loop (Reduce to increase crouch firing rate)**
- "anim2_loop_bullet_fire" - *float* - **Time in non-aimed firing animation2 when weapon should be fired (Must be between Start and End)**
- "anim_breakout_time" - *float* - **Time after which player can break out of attack and run off**

The following properties are get only:

- "fire_type" - *int* - **Type - instant hit (e.g. pistol), projectile (e.g. rocket launcher), area effect (e.g. flame thrower)**
- "model" - *int*
- "model2" - *int*
- "weapon_slot" - *int*
- "anim_group" - *int*
- "skill_level" - *int*
- "required_skill_level" - *int*
- "firing_speed" - *float* - **Projectile/area-effect (e.g. flame thrower) only**
- "radius" - *float* - **Area effect (e.g. flame thrower) only**
- "life_span" - *float* - **Time taken for shot to dissipate**
- "spread" - *float* - **Angle inside which shots are created**
- "fire_offset" - *vector* - **Offset from weapon origin to projectile starting point**
- "aim_offset" - *int* - **Index into (mystery) array of aiming offsets**
- "default_combo" - *int* - **Base combo for this melee weapon**
- "combos_available" - *int* - **How many further combos are available**

Returns

On success:

int: The weapon property

On failure:

bool: False if the passed arguments were invalid