

SetCameraMatrix

This function sets the camera's position and direction. The first three arguments are the point at which the camera lies, the last three are the point the camera faces (or the point it "looks at").

Note: Calling this function takes the camera's focus away from the player and sets the camera in a fixed position and rotation. The camera's focus can be brought back to the player using the `setCameraTarget` function.

Syntax

Server

```
bool setCameraMatrix ( player thePlayer, float positionX, float positionY, float positionZ [, float lookAtX, float lookAtY, float lookAtZ, float roll = 0, float fov = 70 ] )
```

OOP Syntax [Help! I don't understand this!](#)

Method: *player:setCameraMatrix(...)*

Variable: *.cameraMatrix*

Counterpart: *getCameraMatrix*

Required Arguments

- **thePlayer:** The player whose camera is to be changed.
- **positionX:** The x coordinate of the camera's position.
- **positionY:** The y coordinate of the camera's position.
- **positionZ:** The z coordinate of the camera's position.
- **Instead of six coordinates, or two vectors, a Matrix can be supplied.**

Optional Arguments

- **lookAtX:** The x coordinate of the point the camera faces.
- **lookAtY:** The y coordinate of the point the camera faces.
- **lookAtZ:** The z coordinate of the point the camera faces.
- **roll:** The camera roll angle, -180 to 180. A value of 0 means the camera sits straight, positive values will turn it counter-clockwise and negative values will turn it clockwise. -180 or 180 means the camera is upside down.
- **fov:** the field of view angle, 0.01 to 180. The higher this value is, the more you will be able to see what is to your sides.

Client

```
bool setCameraMatrix ( float positionX, float positionY, float positionZ [, float lookAtX, float lookAtY, float lookAtZ, float roll = 0, float fov = 70 ] )
```

OOP Syntax [Help! I don't understand this!](#)

Note: *This is under the static class **Camera***

Method: *Camera.setMatrix(...)*

Variable: *.matrix*

Counterpart: *getCameraMatrix*

Required Arguments

- **positionX:** The x coordinate of the camera's position.
- **positionY:** The y coordinate of the camera's position.
- **positionZ:** The z coordinate of the camera's position.
- **Instead of six coordinates, or two vectors, a Matrix can be supplied.**

Optional Arguments

- **lookAtX:** The x coordinate of the point the camera faces.
- **lookAtY:** The y coordinate of the point the camera faces.
- **lookAtZ:** The z coordinate of the point the camera faces.
- **roll:** The camera roll angle, -180 to 180. A value of 0 means the camera sits straight, positive values will turn it

counter-clockwise and negative values will turn it clockwise. -180 or 180 means the camera is upside down.

- **fov:** the field of view angle, 0.01 to 180. The higher this value is, the more you will be able to see what is to your sides.

Returns

Returns *true* if the arguments are valid, *false* otherwise.