

PasswordHash

Note: Using **passwordHash** is the recommended way of storing passwords.

This function creates a new password hash using a specified hashing algorithm.

Warning: It is strongly recommended to use the async version of the function (i.e. provide a callback function). Otherwise, you will experience short freezes due to the slow nature of the bcrypt algorithm

Syntax

```
string passwordHash ( string password, string algorithm, table options [, function callback ] )
```

Required Arguments

- **password:** The password to hash.
- **algorithm:** The algorithm to use:
 - *bcrypt*: use the bcrypt hashing algorithm. Hash length: 60 characters. **Note that only the prefix `$2y$` is supported (older prefixes can cause security issues).**
- **options:** table with options for the hashing algorithm, as detailed below.

Optional Arguments

- **callback:** providing a callback will run this function asynchronously, the arguments to the callback are the same as the returned values below.

Options for each hashing algorithm

- *bcrypt*:
 - *cost* (int), default: 10. Visit [this link](#) to determine the number of rounds appropriate for your server.
 - *salt* (string), default: empty string
 - an empty string will automatically generate a salt with the *cost* provided
 - **deprecated:** if a string is provided, it will be used as a salt (**do not do this!**)
 - the provided string should be longer than or equal to 22 characters

Returns

Returns the hash as a string if hashing was successful, *false* otherwise. If a callback was provided, the aforementioned values are arguments to the callback, and this function will always return *true*.