Password Encryption

Why encrypt your password ?

It is necessary to hash passwords before storing them in DB.

So, if the DB were to be hacked, we could not recover passwords in clear format. Many users, contrary to the recommendations, use exactly the same password for a big number of sites. If this password is saved in clear, and the database compromised, this would be catastrophic for them.

Today, we don't use SHA1 or MD5 to hash passwords, the latter being now vulnerable because of machines with high computing power.

We must add a 'salt': an additional string of characters, which aims to make the password more complex before hashing.

Some developers use a weak salt and weak algorithm for generating a hash, for example:

```
$hash = md5($password . $salt); // works, but dangerous
```

OR

```
$salt = 'myS@lT89e0[0j';
$hashedPass = sha1($password . $salt);
```

But the password_hash() function can simplify our lives and our code can be secure. When you need to hash a password, just feed it to the function and it will return the hash which you can store in your database.

Encrypt with password_hash

This is the basic syntax to use password_hash function :

$hash = password_hash($password, PASSWORD_DEFAULT);

That's it! The first parameter is the password string that needs to be hashed and the second parameter specifies the algorithm that should be used for generating the hash.

The default algorithm is currently 'bcrypt', but a stronger algorithm may be added as the default later at some point in the future and may generate a larger string. If you are using PASSWORD_DEFAULT in your projects, be sure to store the hash in a column that's capacity is beyond 60 characters

You can provide your own salt using a third argument.
Check to official documentation for more advanced informations.

Using password_verify

The *password_verify*( ) function is used to check if a password match a hash.
It takes a plain password and the hashed string as two arguments.
It returns true if the hash matches the plain password.

For example :

```
if (password_verify($pass, $hashPass)) {
    echo 'Password ok';
} else {
    echo 'Connection refused';
}
```