

Agrupar código y/o classes en PHP

Namespace & Use

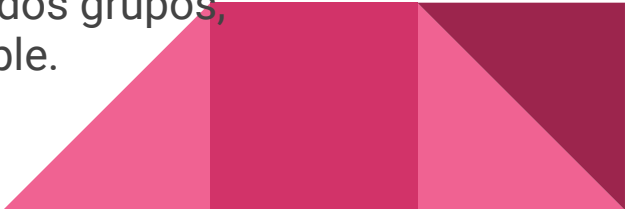
Uso de Namespace

Namespace sirve para definir un nombre a un “espacio”. Son calificadores que resuelven dos problemas diferentes:

- Permiten una mejor organización al **agrupar** clases que trabajan juntas para realizar una tarea.
- Permiten que el **mismo nombre se use para más de una clase**.

Por *ejemplo*, puedes tener un conjunto de clases que describen una tabla de datos, como Table, Row y Cell, y otro conjunto de clases que describen muebles, como Table, Chair y Bed.

Los namespaces pueden usarse para organizar las clases en dos grupos, evitando además que se confundan las dos clases Table y Table.



Alcance de un namespace

Abarca todo el código que sigue después de su declaración y hasta el final del archivo o hasta otra declaración de namespace.

En otras palabras, el namespace agrupa/define a **todas las clases, funciones y constantes que se declaran en el archivo**. O bien hasta la siguiente declaración de un namespace (poco común).



Declaración de un Namespace

Se debe realizar siempre al principio del código. *Ejemplo:*

```
namespace Html;
```

Para una mayor organización, es posible tener namespaces **anidados**:

```
namespace Code\Html;
```

Permite establecer jerarquias.



Uso de un namespace declarado (Use)

Para utilizar una clase que está en un namespace, puedes hacer referencia a ella:

1. Usando el **nombre completo** del namespace o
2. Utilizar la palabra clave **use** para simplificar el acceso mediante un *Alias*.



Ejemplo de uso

// Archivo **UsuarioApp.php**

```
<?php
```

```
namespace App;
```

```
class Usuario {
```

```
    public function __construct() {
```

```
        echo "Usuario de la aplicación";
```

```
    }
```

```
}
```

// Archivo **UsuarioAdmin.php**

```
<?php
```

```
namespace Admin;
```

```
class Usuario {
```

```
    public function __construct() {
```

```
        echo "Usuario administrativo";
```

```
    }
```

```
}
```

Ejemplo de uso - mediante nombre completo

```
<?php
```


```
require 'UsuarioApp.php';
```

```
require 'UsuarioAdmin.php';
```

```
// Llamando directamente a la clase con el nombre completo del namespace
```


```
$appUser = new \App\Usuario();           // Salida: "Usuario de la aplicación"
```

```
$adminUser = new \Admin\Usuario();       // Salida: "Usuario administrativo"
```



Ejemplo de uso - mediante Use (como Alias)

```
<?php  
require 'UsuarioApp.php';  
require 'UsuarioAdmin.php';  
  
// Usando la palabra clave 'use' para simplificar  
use App\Usuario as AppUsuario;  
use Admin\Usuario as AdminUsuario;  
  
$appUser = new AppUsuario();  
$adminUser = new AdminUsuario();
```



Propósito de Use

1. Importar clases o namespaces, especialmente cuando el nombre completo de la clase es largo.
2. Evitar conflictos de nombres al crear un alias para una clase o función importada.



Ejemplo de Use para importar una clase

```
// Archivo1.php
```

```
<?php
```

```
namespace Proyecto\Modulos;
```

```
class ClaseEjemplo {
```

```
    public function mensaje() {
```

```
        return "Mensaje desde ClaseEjemplo en Proyecto\\Modulos.";
```

```
    }
```

```
}
```

```
?>
```



Ejemplo de Use para importar una clase

```
// Archivo2.php
```

```
<?php
```

```
require 'Archivo1.php';
```

```
use Proyecto\Modulos\ClaseEjemplo; // Importa la clase “ClaseEjemplo”
```

```
$ejemplo = new ClaseEjemplo();
```

```
echo $ejemplo->mensaje();
```

```
// Salida: Mensaje desde ClaseEjemplo en Proyecto\Modulos.
```

