8.1. Inclusiones y auto loading

1º DAW – Programación
David Soler Talens

Inclusiones (I)

- **Consiste en tomar todo el código existente en el archivo especificado y copiarlo en el archivo que utiliza la declaración de inclusión.**
- **Ventajas**
  - Nos permite *organizar el código* en diferentes ficheros
  - Nos permite *re-usar código* de forma más cómoda.
- **Dos opciones:**
  - *require()* → cuando el archivo es requerido por la aplicación.
  - *include()* → cuando el archivo no es crítico y la aplicación puede continuar cuando el archivo no se encuentra.

Inclusiones (II)

- **Include**

  include 'fichero_a_incluir';

  include('fichero_a_incluir'); ← sintaxis alternativa

  - Si el fichero no se encuentra, se emite un *warning*
  - el código que contiene el fichero incluido, hereda el ámbito de las variables de la línea en la cual ocurre la inclusión

- **include_once**

  include_once 'fichero_a_incluir';

  include_once('fichero_a_incluir'); ← sintaxis alternativa

  - Previene que se pueda cargar más de una vez un fichero de inclusión
  - Evita sobre-escribir código por error.

Inclusiones (III)

- **Require**

  require 'fichero_a_incluir';

  require('fichero_a_incluir'); ← sintaxis alternativa

  - Si el fichero no se encuentra, se emite un *error y se detiene el script*
  - el código que contiene el fichero incluido, hereda el ámbito de las variables de la línea en la cual ocurre la inclusión

- **Require_once**

  require_once 'fichero_a_incluir';

  require_once('fichero_a_incluir'); ← sintaxis alternativa

  - Previene que se pueda cargar más de una vez un fichero de inclusión
  - Evita sobre-escribir código por error.

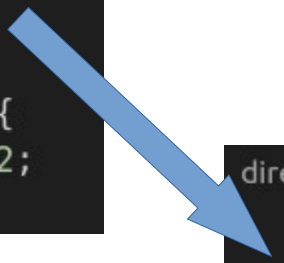
Inclusiones (IV)

directorio > funciones.php > ...

```php
<?php

function uno(){
    echo "uno";
}

function dos($valor){
    return $valor * 2;
}
```



directorio > usoFunciones.php

```php
<?php
require_once("funciones.php");

echo dos(5);
echo "<br>";
uno();
```

Auto-loading (I)

- **Como norma general**
  - es una buena idea guardar las clases PHP en archivos separados
  - una clase por cada archivo -> necesitamos requiere_once
- **PROBLEMA -> *se solían generar largas listas con require_once***
- **SOLUCIÓN -> *autoloading***

1) el código intenta crear una instancia que PHP desconoce todavía
2) PHP llama automáticamente a la función __autoload()
3) La función localiza e incluye el archivo de la clase
4) PHP carga la clase y crea el objeto

Auto-loading (II)

el método "mágico" es auto-invocado cuando se desconoce la clase, pasando la referencia desconocida a __autoload()

El fichero debe existir y coincidir con el nombre de clase

Incluimos los ficheros de clase

Se construye la ruta a los ficheros de clase

```php
<?php

function __autoload($nombreClase) {
    $nombreClase = strtolower($nombreClase);
    $directorio = "classes/{$nombreClase}.php";
    if(file_exists($directorio)) {
        require_once($directorio);
    } else {
        die("El archivo {$nombreClase}.php
            no se ha podido encontrar.");
    }
}
```

Auto-loading (III)

- **Existe una alternativa mejor y más flexible**

  spl_autoload_register("funcion_de_carga");

- **Nos permite tener mas de una función cargadora de clases**
- **spl_autoload_register es la función PREFERIDA**
- **__autoload ha sido declarada OBSOLETA a partir de PHP 7.2.0**

Auto-loading (IV)