

## 1- Què és un namespace?. Per què cal treballar amb namespaces?

### a) Per què els espais namespace (o espai de noms)

Quan un projecte creix en complexitat i es treballa en equip, probablement s'haurà d'integrar el codi de diversos desenvolupadors. Tard o d'hora, 2 o més desenvolupadors hauran definit classes amb el mateix nom. Aquest problema es coneix com a **col·lisió de noms**.

Per resoldre'l, podeu utilitzar espais de noms. PHP admet treballar amb **namespace** (espais de noms) des de la versió 5.3.

### b) Què és un namespace (o espai de noms)

És més fàcil entendre els espais de noms per analogia amb l'estructura de directoris d'un sistema de fitxers.

Un directori emmagatzema fitxers relacionats, cosa que és similar a un espai de noms que agrupa classes relacionades.

Un directori no ens permet tenir dos fitxers amb el mateix nom. Tanmateix, podem tenir fitxers amb els mateixos noms en directoris diferents. De la mateixa manera, els espais de noms imiten el mateix principi.

Per definició, els espais de noms ens proporcionen una manera d'agrupar classes relacionades i ens ajuden a evitar possibles col·lisions de noms.

Els espais de noms no es limiten a les classes de grup. Poden agrupar altres identificadors, incloses funcions, constants, variables, etc.

## 2- Operador use. Treballant amb un àlies d'un namespace

a) Pot ser útil donar un **àlies** a un **namespace** o a una classe per facilitar-ne l'utilització dins d'un codi. Això es fa amb la paraula clau o **use**.

b) L'utilització de l'operador **use** ens permet:

- Haver d'utilitzar continuament el nom complet d'un namespace dins del codi d'un projecte.
- Importar una classe des d'un namespace en comptes d'haver d'importar el namespace.

## 3- Sentències include i require de PHP

a) Les sentències **include** i **require** afegixen tot el codi PHP dins d'un fitxer i l'insereixen dins d'un altre fitxer PHP que utilitza la sentència **include/require**.

b) Incloure fitxers és molt útil per reutilitzar codi.

c) Les sentències include i require són idèntiques, excepte en cas d'error:

- **include** produirà un avís (E\_WARNING) però l'script continuarà
- **require** produirà un error fatal (E\_ERROR) i l'script s'aturarà

d) Si volem que l'execució continuï i mostri als usuaris una sortida, fins i tot si falta el fitxer d'inclusió, utilitzarem la sentència **include**. Això podria ser aconsellable per a fitxers com ara capçaleres, peus de pàgina o menús de navegació.

e) Si volem que l'execució s'aturi en cas d'error, utilitzarem la sentència **require**. Això és aconsellable per a fitxers de configuració o connexions a bases de dades.