

A continuació trobaràs enllaços a recursos i tutorials, que t'ajudaran a resoldre l'activitat d'aquest apartat:

Tasca ->[POO 1](#)



Ambats aquí tenim consciència del que són els tipus de dades bàsics en PHP, com operar amb ells, encapsular-los en funcions, agrupar-los en arrays... PERÒ, amb només això ens arriba per representar qualsevol element de la nostra realitat en un programa informàtic? En realitat sí! Però existeix una manera més intuïtiva i pràctica de fer-ho. Amb objectes o, en altres paraules, creant els nostres propis tipus de dades. Dedica rem dos temes a això, en aquest repassarem temes bàsics que ja vas treballar a fonaments però aplicats a PHP.

SOM-HI!

Quan els programes creixen...

Lògicament, no sempre estarem desenvolupant programes en un únic arxiu, per això, ara que els programes es comencen a fer una miqueta més grans, estaria bé saber conèixer mecanismes bàsics d'importació de fitxers en PHP.

Sentències include i require

->[Sentències include y require de PHP](#)

Include, include_once, require, require_once

->[¿Cuál es la diferencia entre require, require_once, include, include_once en PHP?](#)

Què és una classe? I un objecte?

Definim el concepte clau d'aquest tema i les seves parts.

Class vs Object

->[What Is The Difference Between Class And Object?](#)

Diferències entre Class i Object

->[Difference between object and class](#)

Atributs i mètodes d'una classe

->[101: Introduction to Class, Attributes, Objects & Methods](#)

Classes i objectes en PHP

->[Objetos](#)

Modificadors d'accés i Herència

Una de les gràcies de programar en objectes és que ens permet encapsular informació de diferents maneres. Decidir com encapsulem la informació defineix críticament el nostre disseny. A més, podem definir jerarquies de classes (Herència) mecanisme

que ens permetrà modularitzar i no repetir codi font entre altres avantatges.

Modificadors i herència de classes en PHP

->[Modificadores y herencia de clases PHP](#)

Herència en PHP

->[Modificadores y herencia de clases en PHP](#)

La paraula clau "static"

A vegades tenim la necessitat de definir propietats o mètodes que són en valor o comportament a totes les instàncies d'una mateixa classe. Tot i que podem suplir aquest comportament amb un millor disseny de classes, convé saber que existeix aquesta manera de fer les coses.

Late static binding a PHP

->[Late Static Binding en PHP](#)

📌 Per saber més

Tenim un resum de tot plegat en aquest llistat de vídeos (fins al capítol 11):

Programación Orientada a Objetos con PHP | Presentación del curso. Cap 0



✉ [Contacteu amb el suport del lloc](#)

Heu iniciat sessió com a [Mario Gomez](#) (Surt)

[Política de Cookies](#) - [Política de Privacitat](#) - [Condicions Generals d'Us](#)

Funciona amb [Moodle](#)



Unió Europea
Fons Europeu
de Desenvolupament Regional

[Privacy Policy](#)