

- * limbaj de scripting, nu e compilat, este interpretat
- * AMP → Apache, Mysql, PHP (LAMP = se adaugă Linux)
stack
- * toate tehnologiile de backend au nevoie de ceva să le ruleze
→ un modul din Apache (de exemplu) face asta (le execută)

! Întrebări la examen din notițe!

- * e greșit să încercăm fișiere php cu: `file:///...`, trebuie prin protocolul http, altfel browserul e doar viewer

`<?php ... ?>` → exclusiv pe backend, clientul nu-l știe interpretat

`<? ?>` → delimitator de script
→ poate lipsi dacă este la finalul fișierului

- * parametrii se transmit prin valoare implicit

- * `&` → pt referință (ca în C)

- * global `$pi`; → face ca variabila să fie globală

- * `$_GET`, `$_POST` → tablouri asociative care memorează și încapsulează toți parametrii veniți prin request

- Se recomandă să se deșpartă ~~view~~ view-ul de point-ul care solosească datele (acelasi script este folosit și ca view și ca console.)

- am scris bine prima dată :(
- la început, fiecare DB ~~are~~ propriile librării ce trebuiau incluse
 - mysqli - (mysql "improved") → varianta nouă
 - * toate acestea depind de un anumit SGBD specific
 - PDO (PHP Data Objects) → asigură independența față de serverul de date folosit (SGBD)
- ⇒ nu mai trebuie schimbat codul dacă schimb baza de date;
se modifică doar string-ul de conectare
- * se pot include alte fișiere PHP într-un fișier PHP → *require e mai bun!*
 - * dacă se fac 2 include-uri → risc de erori ⇒ folosim include-once
(include fișierul doar dacă nu a mai fost adăugat)