

* Giao thức HTTP (4 bước):
  + Tạo yêu cầu
* Server response thuần html cho client thì kết thúc session
* OSI : 7 tầng
* TCP/IP : 4 tầng
  + Ứng dụng
  + Giao vận
  + Internet \*
  + Tiếp cận mạng
* Lập trình viên (3 tầng) -> 3 lớp:
  + giao diện (View) -> ở client
  + modul xử lý (Controller) -> ở client và server
  + quản lý dữ liệu (Model: DAO, BO, BEAN) -> ở server

Form :

* + - name = ?
    - action = ? //chỉ định 1 file khác để xử lý form
    - method = “post / get”
    - Trường Input :
      * type = ?
      * name = ?
      * value = ?
    - 2 trường ko dung input -> select button & textarea

<Frame> : có phép load nhiều trang cùng 1 thời điểm (\* HTML5 -> iframe)

* + - 1 trang : frame
    - 2 trang trở lên : frameset
      * Dòng : Rows
      * Cột : Cols

Đối tượng:

* Thuộc tính
* Hành động –(gọi)-> phương thức đáp ứng
* Đối tượng.phuongthuc (đáp ứng hành động của đối tượng)
* Đối tượng.thuoctinh (tính chất để mô tả rõ cho đối tượng)

Document chứa:

* Form
* Links
* Có thuộc tính “target” rất qtr -> định vùng (frame)

Có 4 cách để nhúng php vào html:

* <? …. ?>
* <?php …. ?>
* <script language=”php”></script>
* <% … %>

\*\* rất mạnh về kiểu string và array

* + - * Chuỗi: ‘ ‘ ; “ “; ‘ “ “ ‘ ; “ ‘ ‘ “ ; ‘ \’ \’ ‘;” \” \” “.
      * Trước mỗi biến dùng $
      * Kiểu Array:
        + Khai báo:

<?php $a = Array(1, 2, 3) ?>

<?php $a = Array(

“key1” => “giá trị 1”,

“key2” => “giá trị 2”,

“key3” => “giá trị 3”);

Echo $a[“key1”] #in ra “giá trị 1” ?>

* + - * Khai báo biến : KO bắt đầu chữ số, \_
      * Toán tử:
        + + , - , \* , / , % , ++ , --
        + <> , != , === , == , …
        + & , | , ^ , ~ , << , >>
        + && , and , || , or , !
      * Dấu . để nối 2 chuỗi
      * 2 dạng foreach:
        + Foreach($array as $value) câu lệnh;
        + Foreach($array as $key => $value) câu lệnh;
* Nhúng file

require: nếu dùng -> lỗi -> tbao + ngắt luôn  
include -> lỗi -> đưa ra thông báo + tiếp tục

Kết nối csdl từ DAO -> ko truyền liền về Controller

DAO chỉ trả về cho BO

Chỉ BO mới gọi được DAO

Client -> Controller -> Model -> Controller -> View -> Client

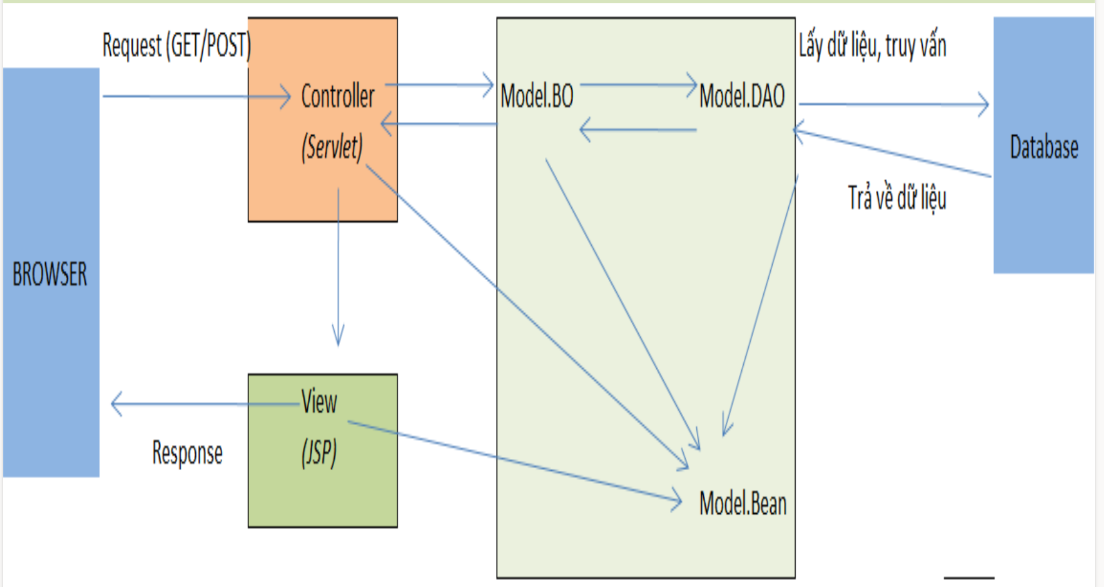
Model : chứa dữ liệu + xử lý -> View+Controller phụ thuộc vào model

Controller : quyết định đối tượng được View

* Controller:
  + B1: Gọi model.BO
  + B2: Nhận dữ liệu từ Model.BO
  + B3: Gọi view tương ứng
* Model:
  + Truy vấn csdl -> Mỗi bản ghi trả về 1 đối tượng -> thực thể entity (class) mô tả cho 1 bản ghi – 1 đối tượng

(BO-DAO -> M\_Student ; BEAN -> E\_Student)

* BEAN: 1 class mô tả cho 1 thực thể thực tế (tất cả các properties là private)
  + If return Boolean => thay get thành is



* BEAN: Chứa thực thể
  + Có các thuộc tính (VD: name, address, alive… của 1 đối tượng)
* BO là Service