

1.platform application architecture

client | server

2.network communication

- ① layer: interface network transport application
- ② protocol: mac ip tcp/udp http
- ③ packet: mac frame(mac) ip packet(ip) tcp/udp packet(port) http packet

3.source code/program structure

- ① API/driver/runtime: underlying function/class
- ② library/package/module/framework/engine: base function/class
- ③ declaration&implementation: specific function/class
- ④ **flow**: entry function/class

4.platform language

c/c++ java c# **perl python**

5.web application architecture

browser(html/css/javascript) | server(script)

6.http communication

- ① http url
- ② http request packet

method request-uri http-version | request-header:value | request-data

- ③ http response packet

http-version status-code reason-phrase | response-header:value | response-data

method: get post put delete

7.web language

presentation layer: html css

business layer: javascript flex/acrionscript **php** java-web **python** ruby

data layer: sql

8.LIB

shell: source /xx/xx . /xx/xx

perl: require */path/xx* use <module> *@INC* perl -V export *PERLLIB/PERL5LIB="/xx/xx"*

python: import /xx/xx *sys.path* export *PYTHONPATH*="/xx/xx"

9.program/process

Program: *data structure, logical, function, class/object*

Process: self- *data handling*; system call – *file system, database, network, web, GUI, process/thread*