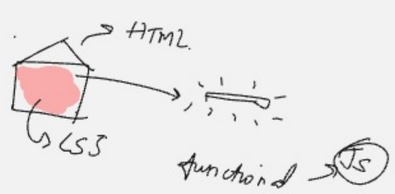
* Website → set of code → HTML, CSS, JS

House



History of Web

Web 1.0 → Web 2.0 → Web 3.0

Static websites ⇒ Readonly

(HTML) → data → images / text / links

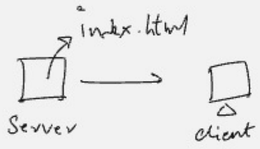
Pros
* Simple, easy to create
* low bandwidth → Share it
* minimal security risk

1980

Cons
Not interactive
Limited fn.

XSS
Cross Site Scripting

< > tags



Web2.0

Type of data/content shared

Interactive part of web. (2 way comm)

→ [HTML, CSS, Javascript, AJAX, Python, PHP, Ruby]

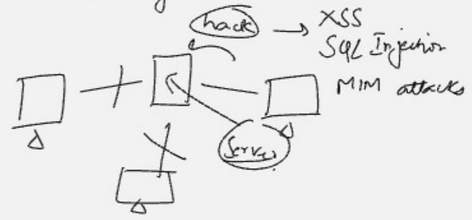
→ <forms> ⇒ Input through HTML

(1990 – 2010/2020)

Pros
→ Rich & better UX
→ Real time communication

Cons
→ Slow in nature
→ More complex architecture.
→ Security threat.

Cloud



Web 3.0

park → personalised & secured.

→ Shared space in Internet

failure & restored to previous version

data

* decentralised ⇒ control of web.

AI, Blockchain, AR/VR

Smart contract, token/gas

bitcoin / ethereum

* Pros → Ownership personalised space.
* Cons → complex.

URL → Uniform Resource Locator.

https://www.google.com/index.html

protocol | domain | file/resource

192.168.0.0 → IP

{ www.google.com: }

→ Browser
- Browser cache
- OS cache
- Router cache
- ISP DNS → Interprovin
- Root DNS → hostinger / godaddy

→ IP

CSS → Inline, Internal, External

→ Server (Backend logic / Server files/data)

IP ⇒ TCP / TLS Handshake → request

← response/data.

Requests → GET, POST, UPDATE, DELETE, PUT, OPTIONS → APIs → Response.

Browser → Renders on UI

combining all [HTML, CSS, Js files.]