Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Aplicaciones Web

... y un vistazo a Ruby on Rails...

Raúl Montes T.

Para compartir documentos científicos...

Necesitamos

- representar el documento ➜ HTML
  - referencias a otros documentos ➜ Hyperlinks
- representar la referencia a un documento ➜ URL(I)
- servir el documento ➜ web server
- obtener el documento ➜ web browser
- que client y server puedan entenderse ➜ HTTP

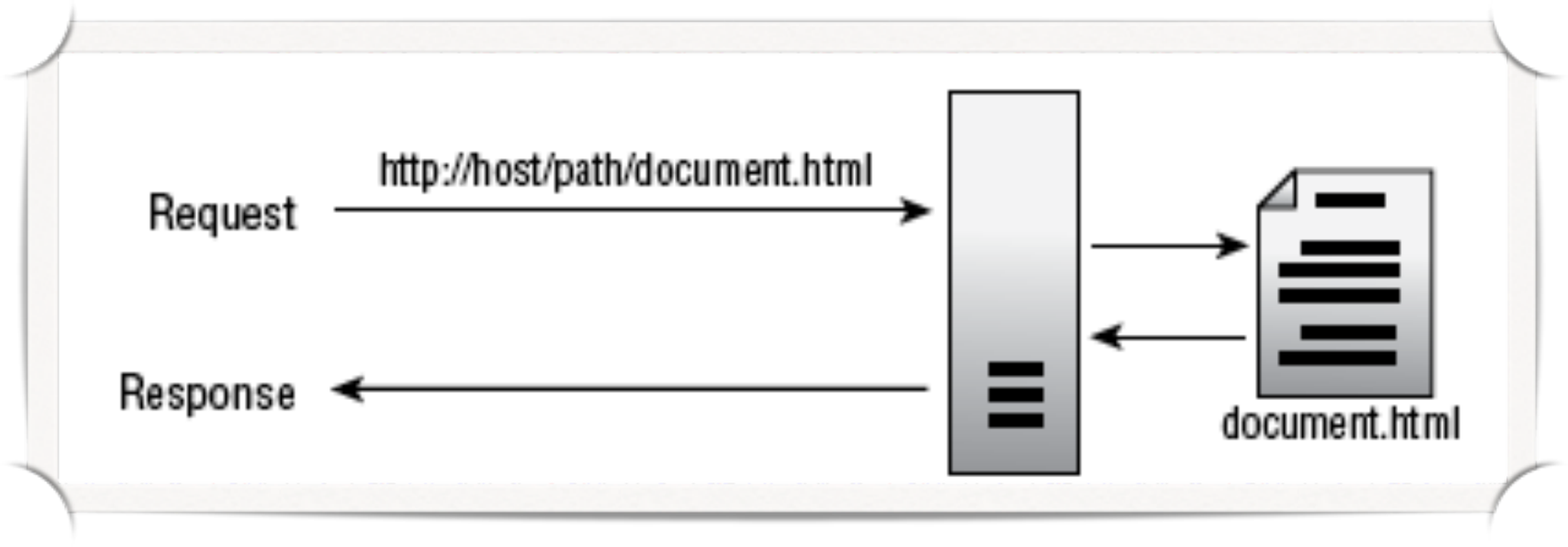
Los inicios de la Web

HTTP

http://hostname/doc.html

~1990-93

El Rol de HTTP y HTML



Principales Requests en Hypertext Transfer Protocol

GET - Busca y entrega el recurso asociado al URI

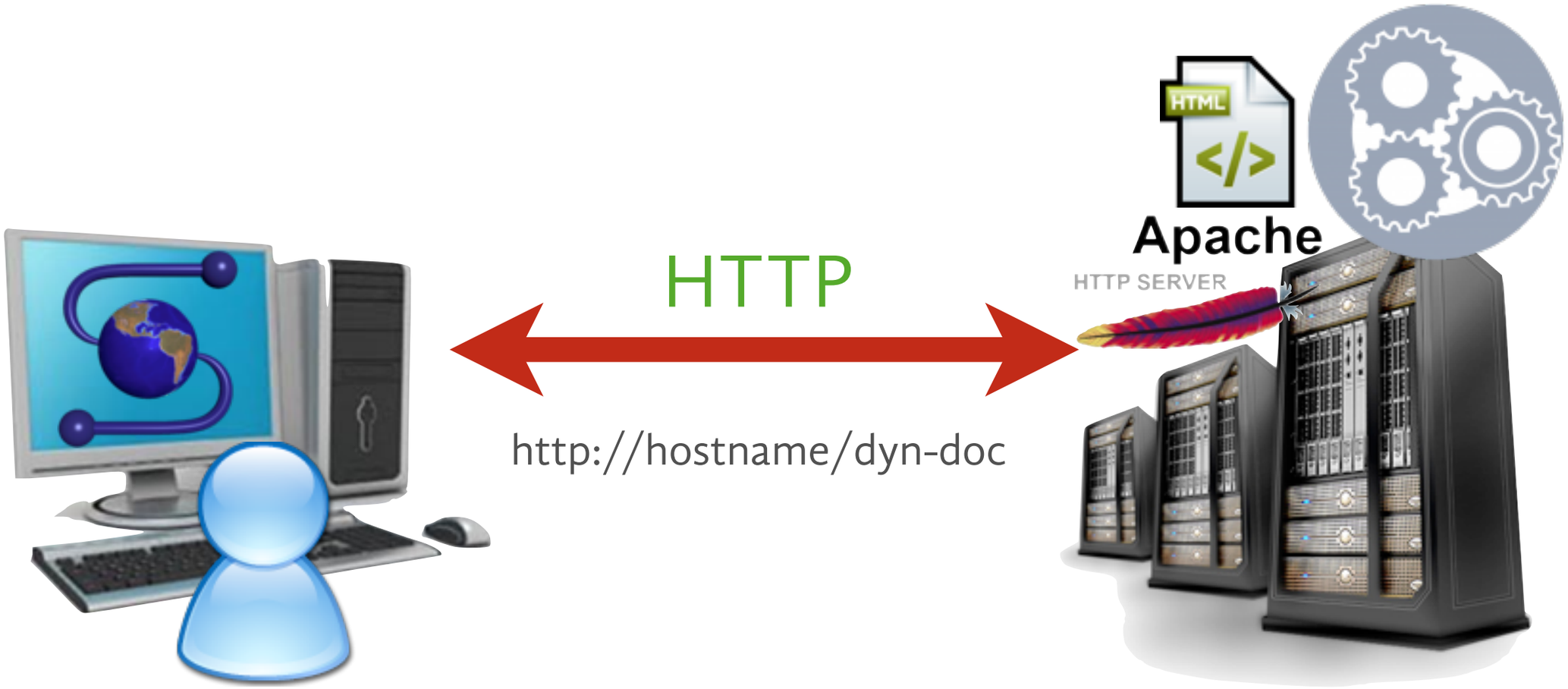
POST - Acepta la entidad adjunta en el URI indicado

PUT - Acepta la entidad adjunta y cambia el recurso

DELETE - Elimina el recurso asociado al URI

Ooooohhhmmm... be the browseeeeeeer...

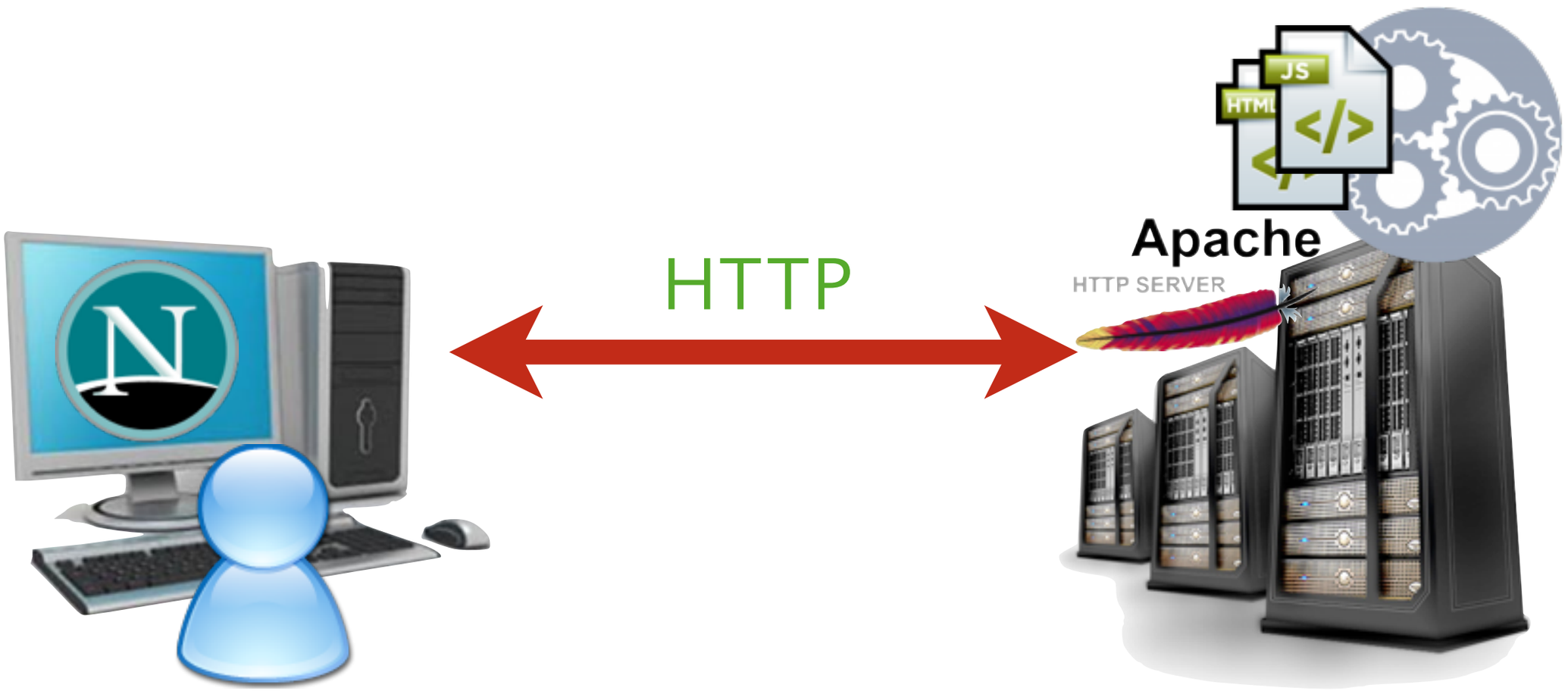
Los inicios de la Web



Necesitamos contenido dinámico

~1993

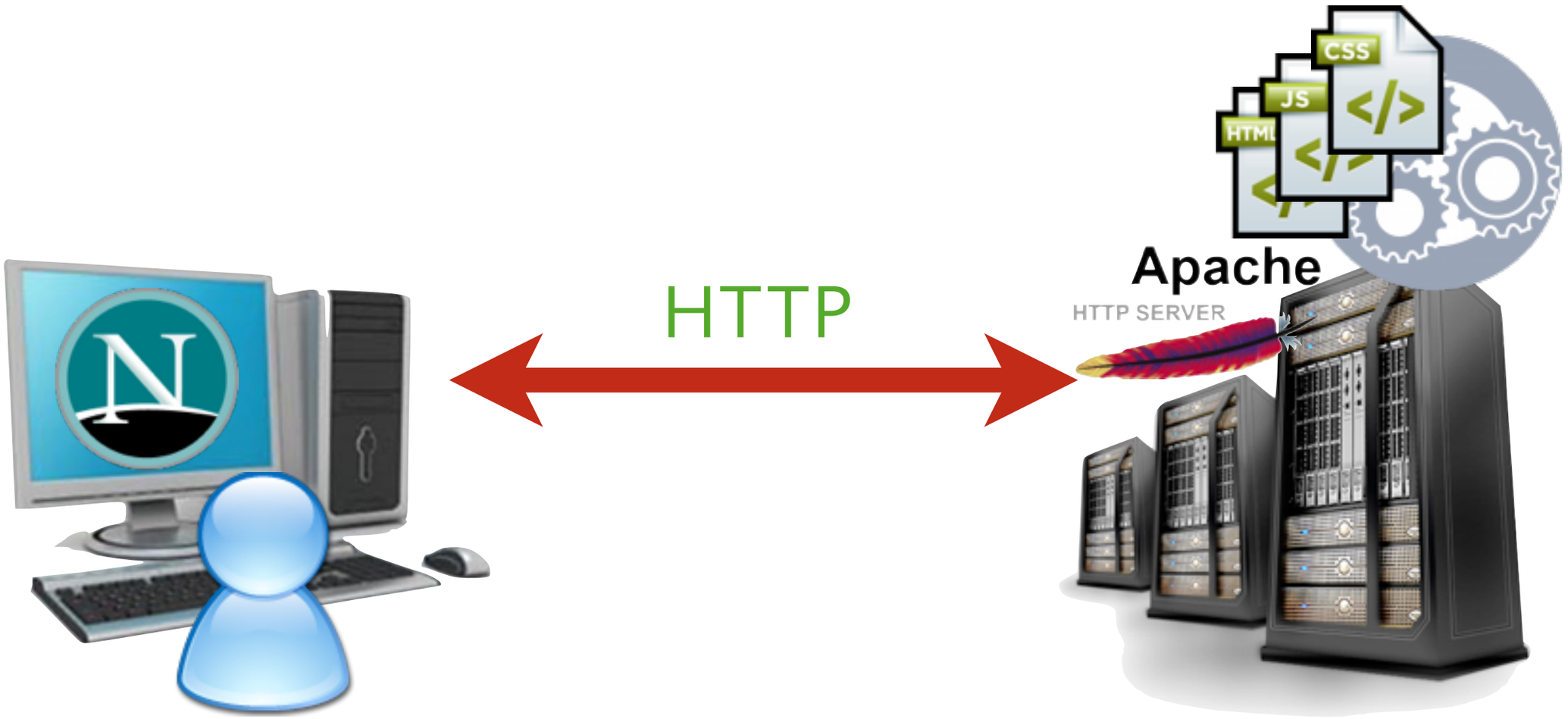
Los inicios de la Web



También en el lado del cliente

~1995

Los inicios de la Web



Necesitamos representar estilo más complejo

~1994-96

Aplicaciones Web

trabajo coordinado entre cliente y servidor

no pueden correr por si
solas

HTTP

poco acceso a recursos
del server

no son simplemente programas sino
verdaderos ecosistemas

~1999-2005

Aplicaciones Web

JavaScript

CSS

HTML

Front-end

HTTP

Ruby

BD

Back-end

Componentes

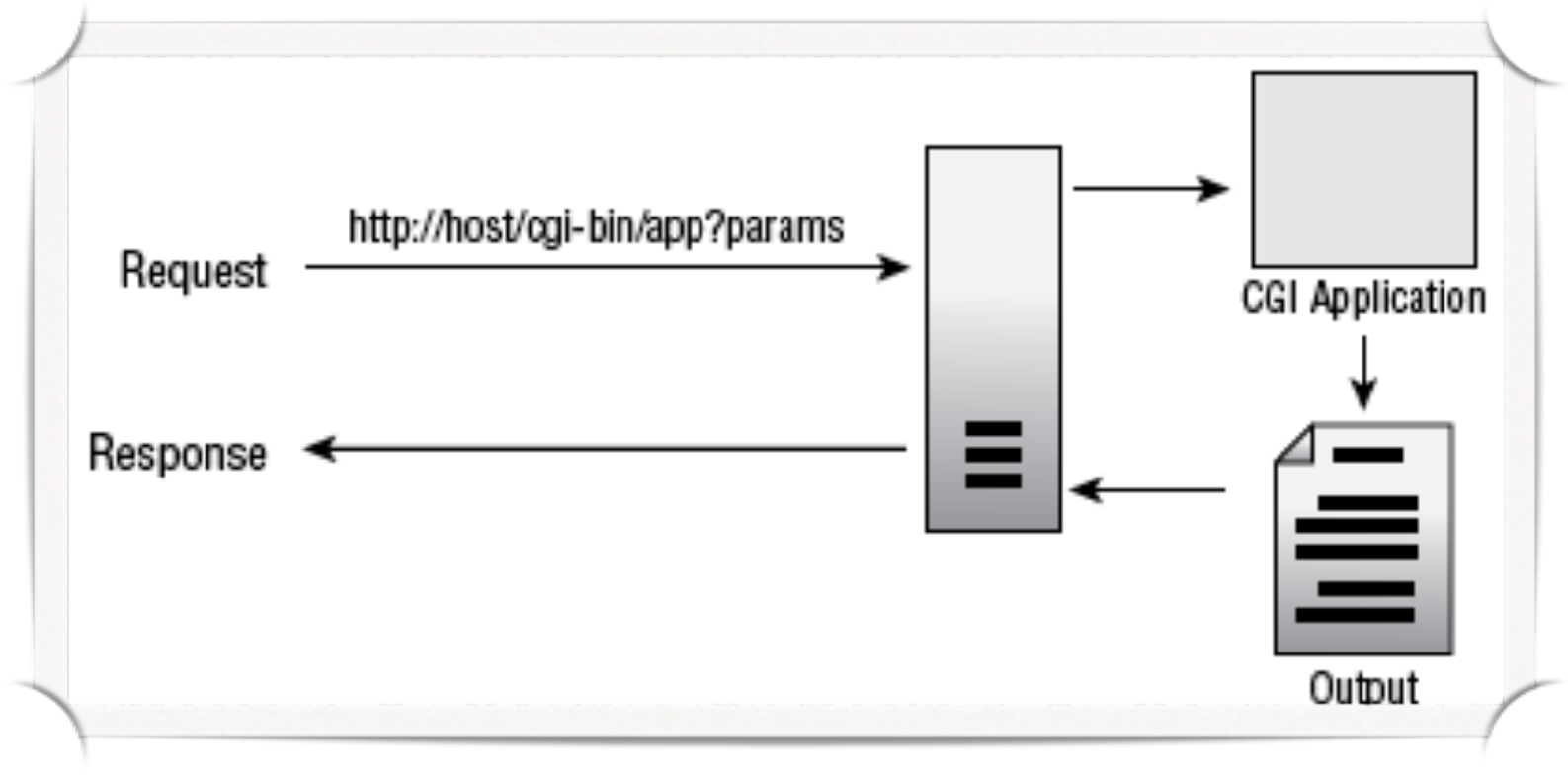
- **HTML** para estructura de la UI
- **CSS** para controlar el aspecto de la UI
- **JavaScript** para comportamiento de la UI
- código **Ruby** (u otro) para lógica de la aplicación
- motor de **BD** para persistencia de datos
- **REST API** para interactuar con otras aplicaciones

De páginas a Aplicaciones Web

- Arquitectura original de la WWW consideraba:
  - documentos codificados en HTML
  - interacción cliente servidor mediante HTTP
  - cliente capaz de desplegar contenido HTML entregado por el servidor

Aplicaciones web, los inicios

Common Gateway Interface (CGI)

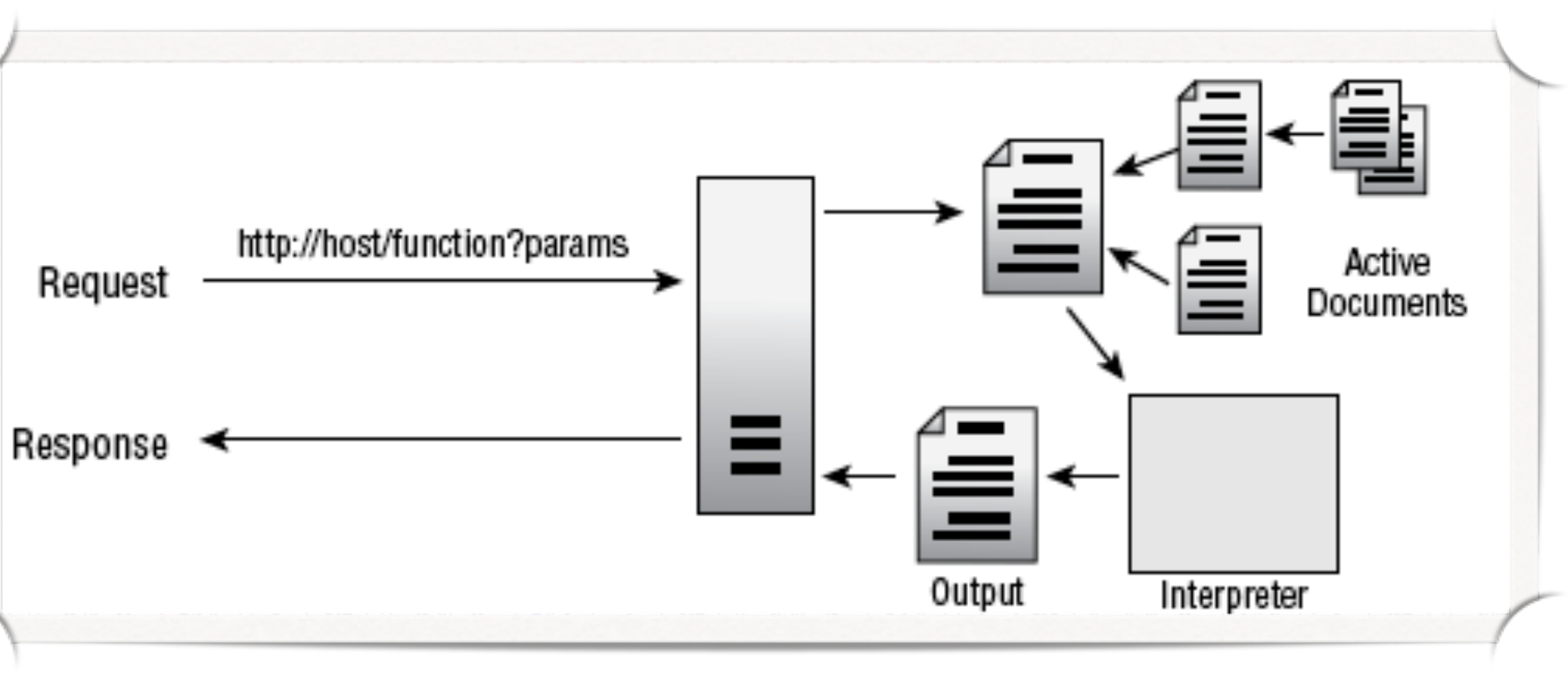


Segunda Etapa: el documento activo

PHP

Ejemplos: SSI, ASP y...

documento contiene código que debe ser interpretado
antes de devolverlo al cliente



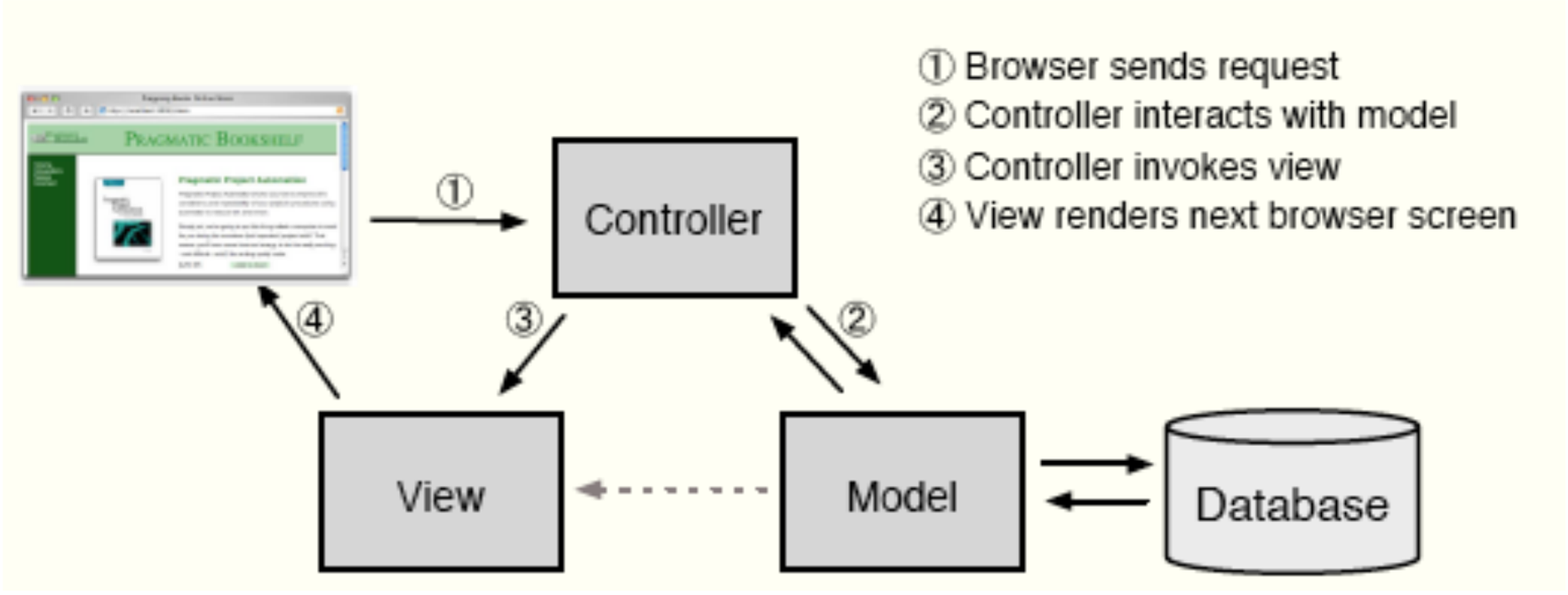
Surgimiento de Arquitectura MVC

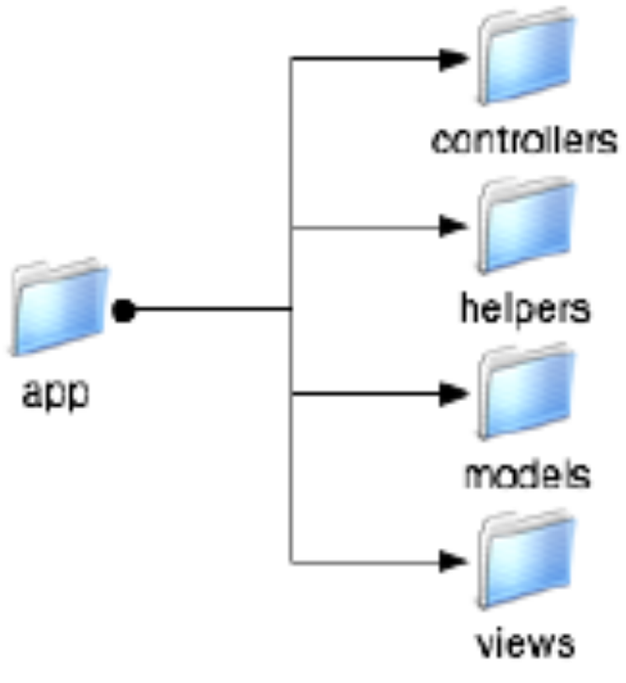
- **Vista** - código que aplicación usa para interactuar con el usuario (HTML, CSS, JavaScript)

- **Modelo** - objetos y métodos con la lógica de la aplicación (Ruby, por ejemplo)

- **Controlador** - código que sirve de nexo entre vista y modelo (Ruby, por ejemplo)

MVC


① Browser sends request
② Controller interacts with model
③ Controller invokes view
④ View renders next browser screen
Controller
View
Model
Database
①
②
③
④

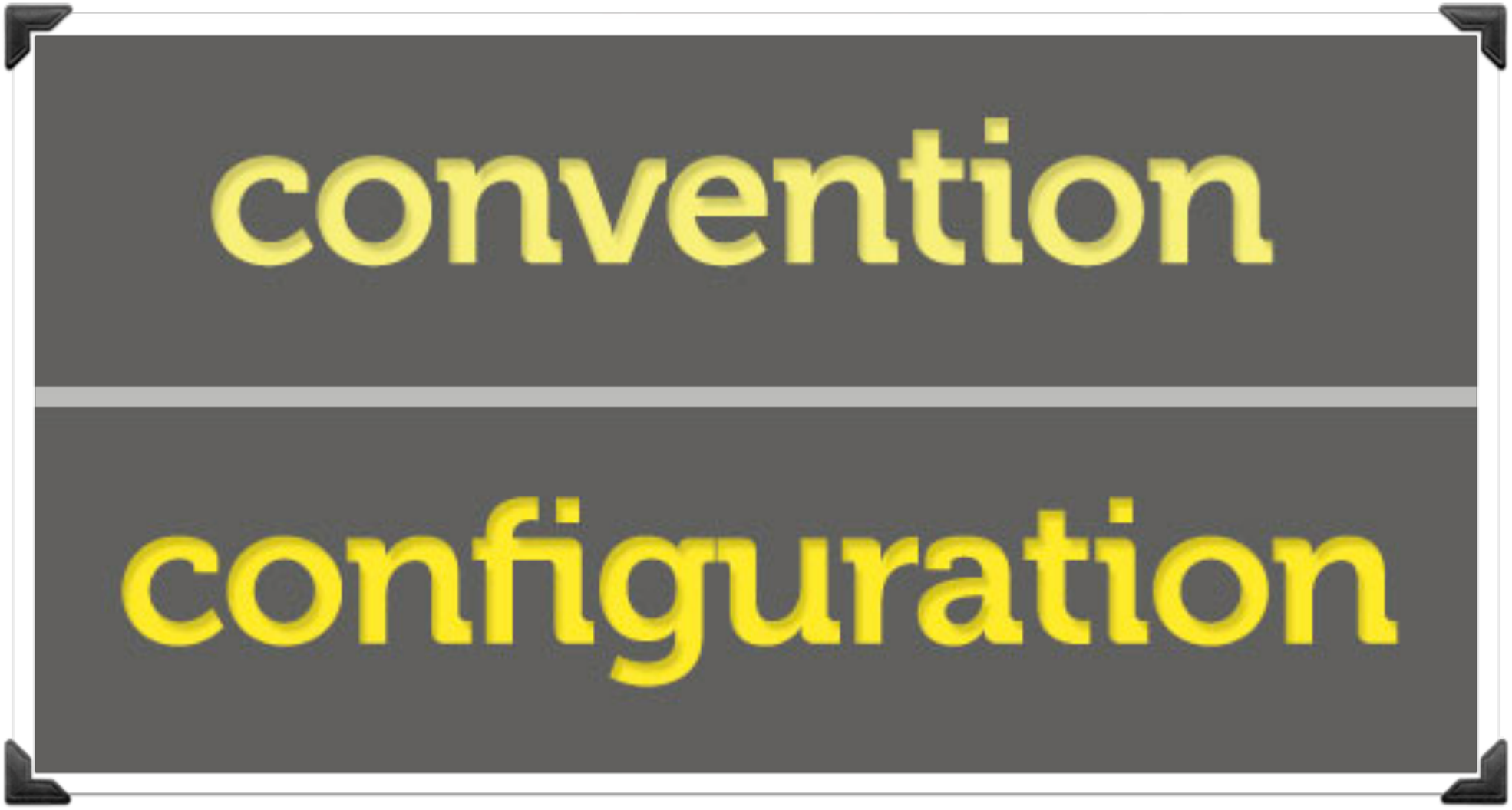

app
controllers
helpers
models
views

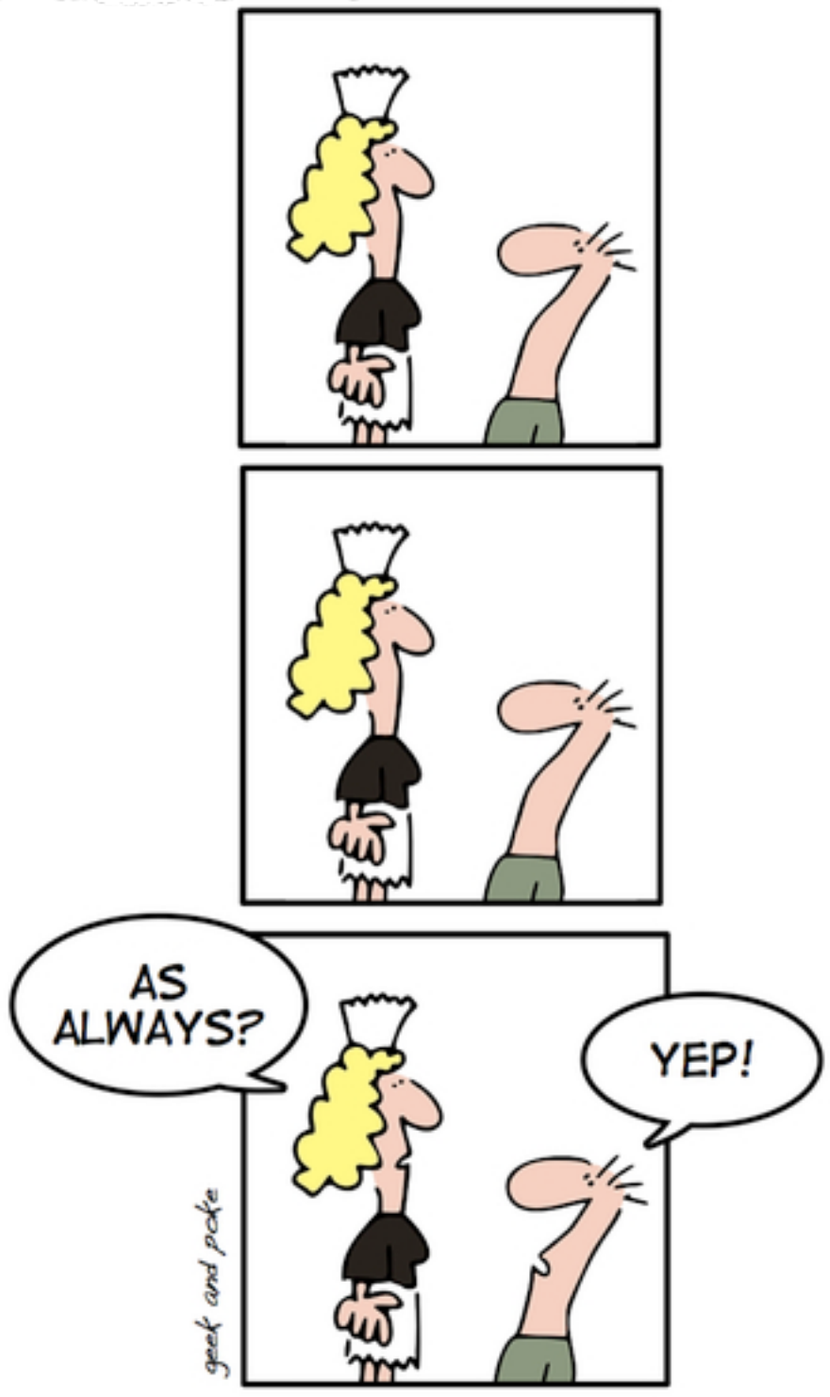
RoR es un framework MVC

- programador construye modelos, vistas, controladores

- framework proporciona los soportes para unir todas estas partes

- utilización de “defaults” o convenciones facilita el proceso de tejido


convention
configuration



SIMPLY EXPLAINED – PART 18:
CONVENTION OVER CONFIGURATION

Soporte del Modelo: Active Record Module

- modelo por lo general tiene una contrapartida en una BD relacional
- filas de la tabla son objetos del modelo
- atributos de objetos son columnas de la tabla
- métodos de clase para facilitar las búsquedas, ordenamientos, etc

Ejemplo

```
require 'active_record'

class Order < ActiveRecord::Base
end
```

```
order = Order.find(1)
puts "Customer #{order.customer_id},
amount=$#{order.amount}"
order.pay_type = "Purchase order"
order.save
```

Soporte de la Vista

- La mayor parte de la vista es HTML
- Hay partes dinámicas generadas por el método/acción del controlador
- Contenido dinámico se maneja con plantillas (templates)
  - ERb
  - XML Builder
  - RJS

y el Controlador

- el gran orquestador
- responsable del enrutamiento a la acción correspondiente
- maneja el cache, la sesión y módulos auxiliares (helpers)

Hello World

- `$ rails new helloworld`
  - crea una nueva aplicación (directorio completo)
- `$ rails server`
  - inicia un servidor en el puerto 3000
- `$ rails generate controller Say hello`
  - crea controlador Say y acción hello
  - crea automáticamente vista asociada