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

Client-side JavaScript

Frameworks en el browser

Raúl Montes T.

UI de primeras aplicaciones Web

- Carga de un documento HTML completo
- Interacción de usuario para enviar u obtener datos implicaba request HTTP y carga de nuevo documento HTML
- Lento, interacción interrumpida por página en blanco mientras se carga siguiente documento

Luego vino el DOM y Dynamic HTML

- 1995-1998
- Mutación del documento en el cliente por la detección de interacción del usuario
- Permitió cargar un documento inicial – más completo – para crear una interacción más fluida con JavaScript en el cliente (tabs, steps form, etc.)
- Una mejora, pero de necesitar enviar datos u obtener nuevos datos, requests por un documento completo son inevitables -> página blanca, interacción lenta e interrumpida, ...

XMLHTTPRequest

- 1999 en Internet Explorer como control ActiveX
- 2000-2005 en otros browsers, en 2006 se convirtió en estándar
- Permitió enviar y recibir datos asíncronamente, manteniendo documento e interacción de usuario
- Uso de DOM para actualizar documento con llegada de datos

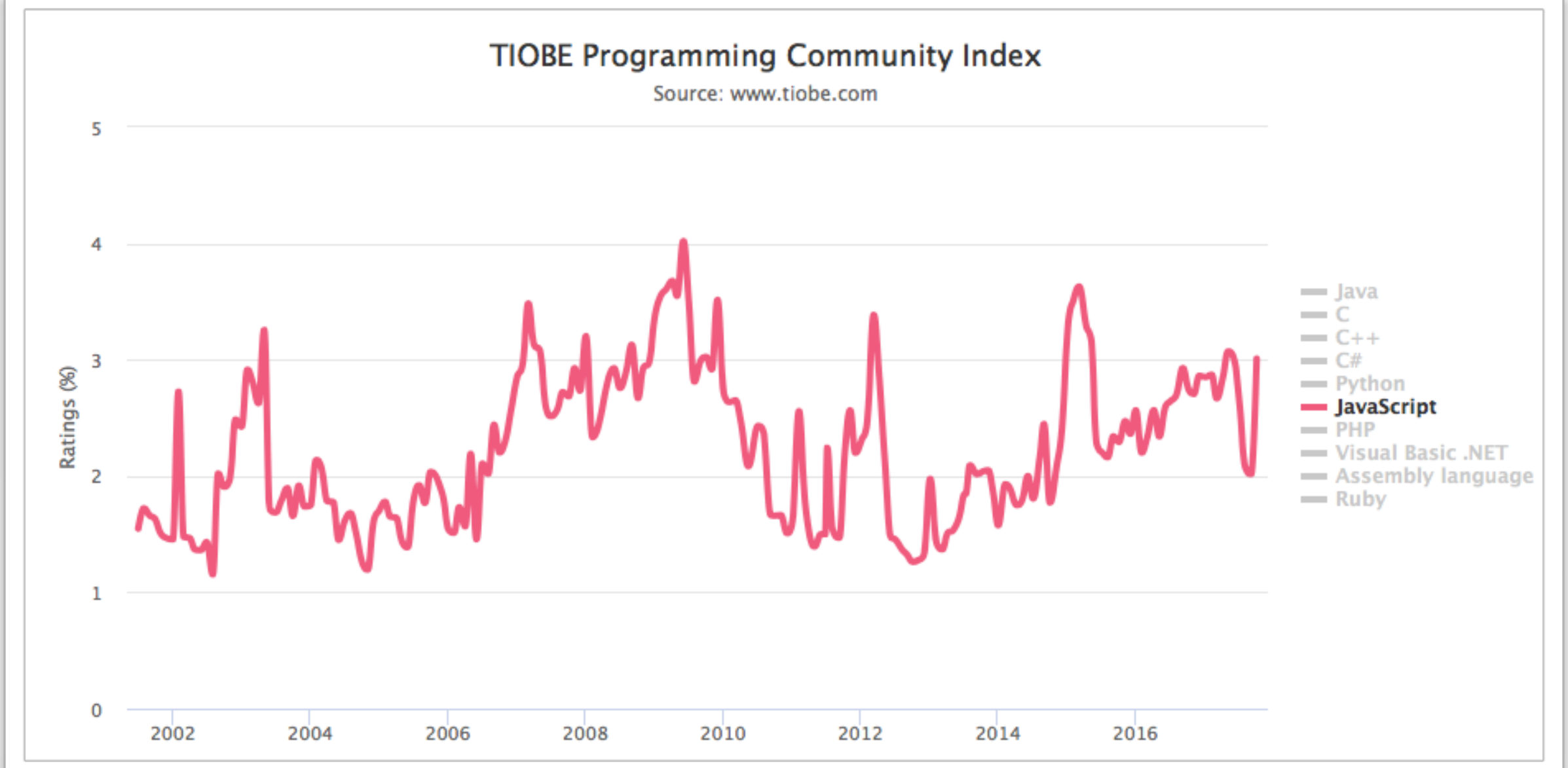
Ajax

Garret creó el término para referirse al conjunto de tecnologías que permite esta interacción mucho más fluida

- HTML y CSS para la vista
- DOM para interactuar con la vista
- XML, JSON, HTML o JS para intercambio de info
- El XMLHTTPRequest para requests asíncronos
- JavaScript para unirlos a todos... ~~y atarlos en las tinieblas~~

Primeras aplicaciones basadas en Ajax

- Outlook Web App (2000), Oddpost (2002)
  - aunque muy poco conocidas
- Gmail (2004), Kayak.com (2004) Google Maps (2005)
  - se comenzó a masificar el uso de Ajax


TIOBE Programming Community Index
Source: www.tiobe.com
Ratings (%)
5
4
3
2
1
0
2002
2004
2006
2008
2010
2012
2014
2016
Java
C
C++
C#
Python
JavaScript
PHP
Visual Basic .NET
Assembly language
Ruby

Ajax

- El uso de JavaScript explotó
- Nació la idea de Single Page Applications (SPA)
- Complejidad de aplicaciones en el lado del cliente se incrementó enormemente
- Cumplimiento de estándares de browsers más usados y compatibilidad entre ellos seguía siendo un gran impedimento
- Surgimiento de librerías para simplificar API JavaScript y lidiar con diferentes browsers

2005

2005

2006

2006

2007

Estas librerías proveen muchas facilidades para recorrer, examinar y manipular el DOM y utilizar Ajax

Pero complejidad de aplicaciones cada vez mayor llevó al surgimiento de frameworks de más alto nivel

MVC, MVP, MVVM, ... MV*

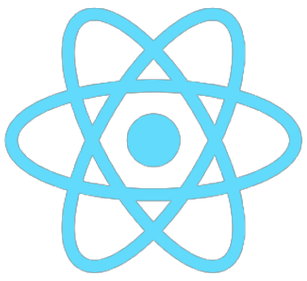
Sproutcore, 2010

BACKBONE.JS 2010

Knockout. 2010

ANGULARJS by Google 2010

ember 2011

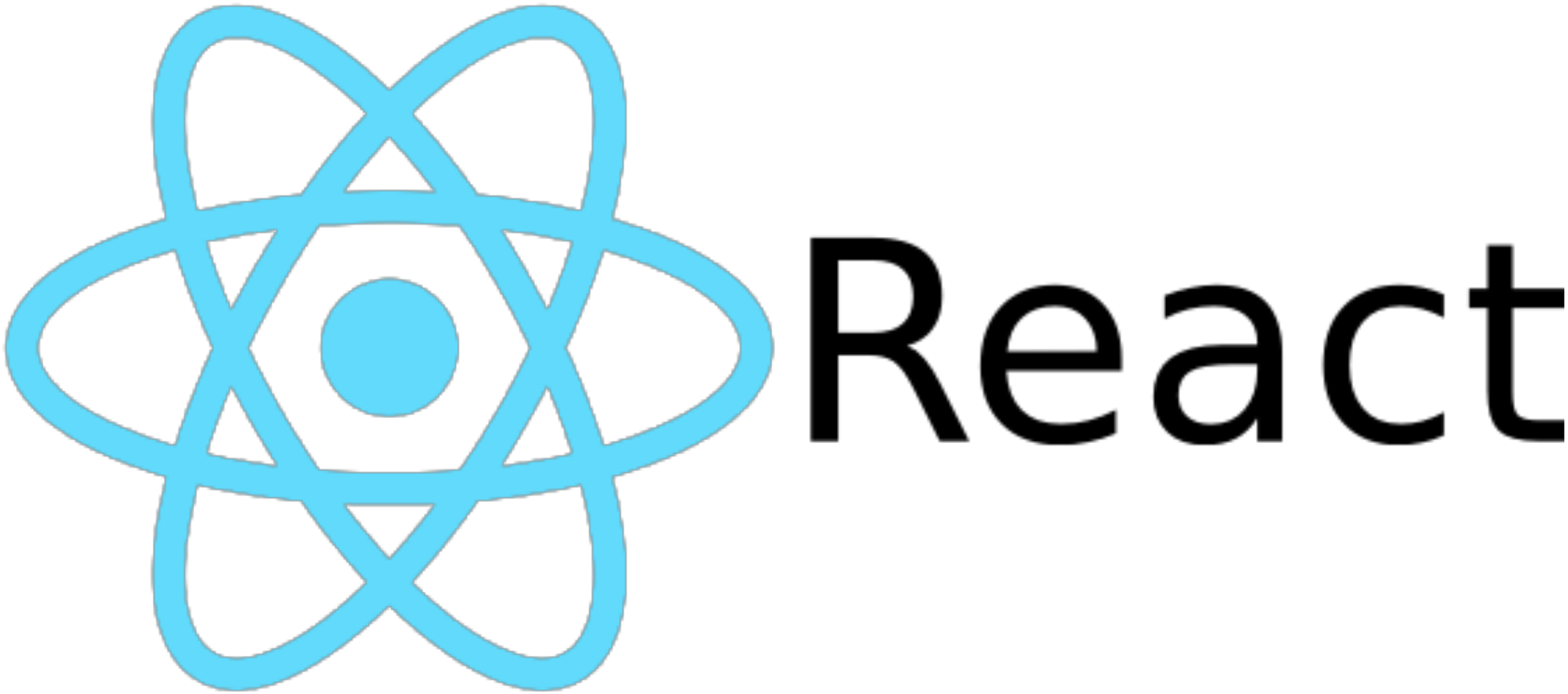


React 2013

Vue, 2014

RIOT.js by MUUT 2014

aurelia 2016


React

React

- Inicialmente la V de MVC
- Derivó a Component based architecture (CBA)
  - Componentes: View, UI state, Event handling
  - Flux: Application state, Business Logic

React: características importantes

- Flux (one-way data flow): properties flow down, actions flow up

- Virtual DOM: cambios se calculan en representación virtual para ejecutar mínimas operaciones en DOM

- JSX: extensión de sintaxis de JavaScript para representar vistas en “HTMLish”

State-less/ful components

- Components pueden (*stateful*) o no (*stateless*) mantener estado
- Componentes *stateless* generalmente usan la notación de función (que recibe props y retorna JSX)
- Estado (`this.state`) se inicializa como objeto en constructor y se muta sólo mediante llamados a `setState`
  - `this.setState({ propToChange: value });`

Patrón relevante: dumb/smart components

- Dumb components sólo reciben props y retornan JSX
  - generalmente *stateless* y con notación de función
  - suelen ser componentes “puros” (concepto de *pure function*)
- Smart components generalmente necesitan estado y manejar el ciclo de vida del componente
  - Se encargan de data fetching/updating
  - Entregan props a sus *children*

Docs

- Docs oficiales: https://reactjs.org/docs/hello-world.html
- Manejo de forms: https://reactjs.org/docs/forms.html
- Tutorial oficial: https://reactjs.org/tutorial/tutorial.html
- Otro tutorial que comienza más simple: http://buildwithreact.com/tutorial