

# Intro 2 JS

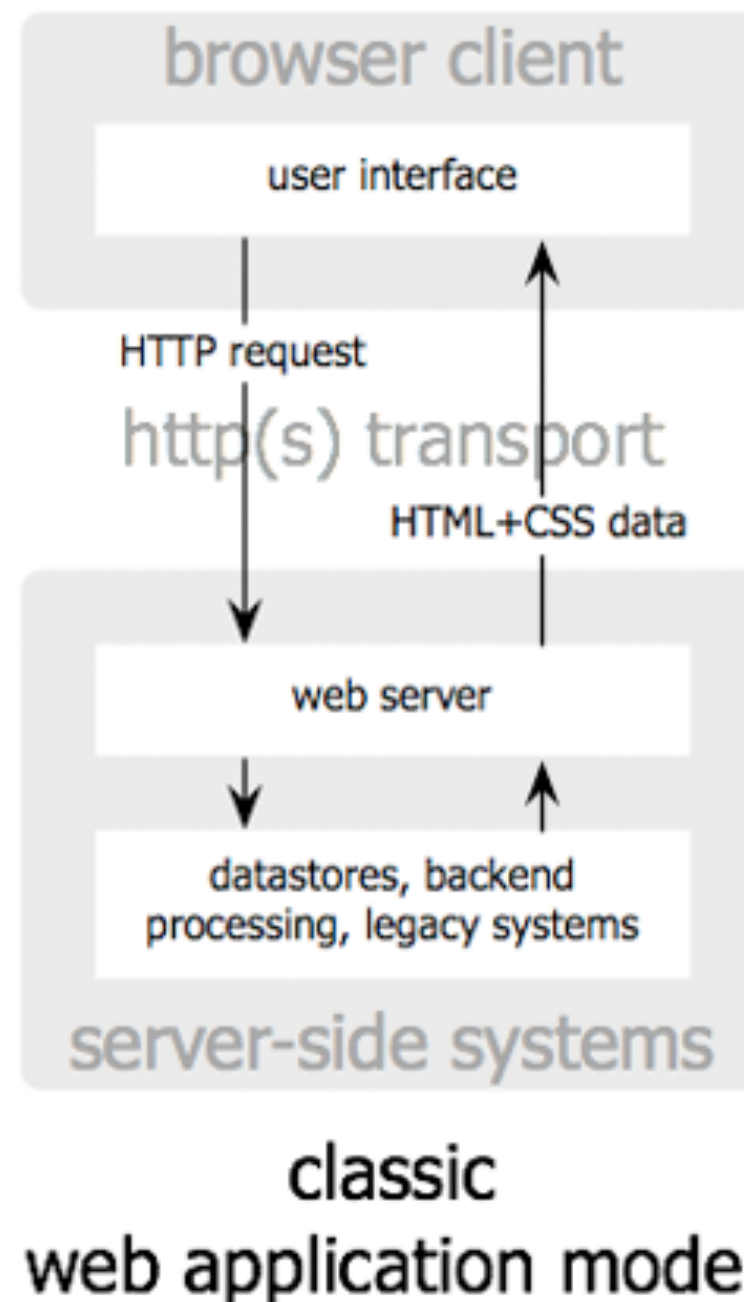
The collection of technologies used by Google was christened AJAX, in a seminal essay by Jesse James Garrett of web design firm Adaptive Path.

"**Ajax** isn't a technology. It's really several technologies, each flourishing in its own right, coming together in powerful new ways.

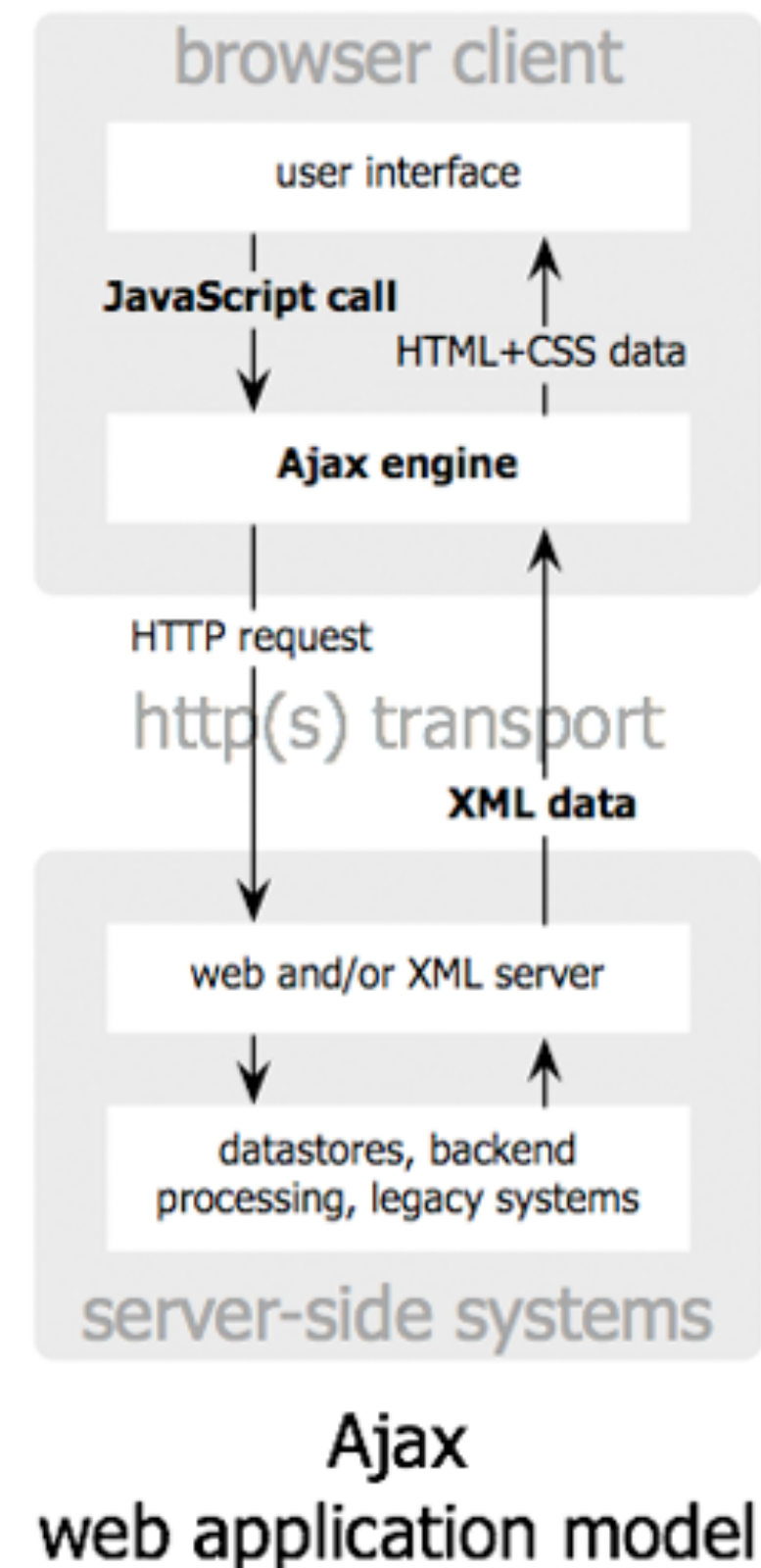
Ajax incorporates:

- standards-based presentation using **XHTML** and **CSS**;
- dynamic display and interaction using the Document Object Model (DOM);
- data interchange and manipulation using XML and XSLT;
- asynchronous data retrieval using XMLHttpRequest;
- + **JavaScript** binding everything together."





Jesse James Garrett / adaptivepath.com



Jesse James Garrett, 2005

(To view this link you'll need the Wayback Machine)

## Server Side Programming

Until now, we have been learning about \*client side\* web development. The client side, in this case, is the web browser and the technologies that go into it (HTML, CSS, JavaScript).

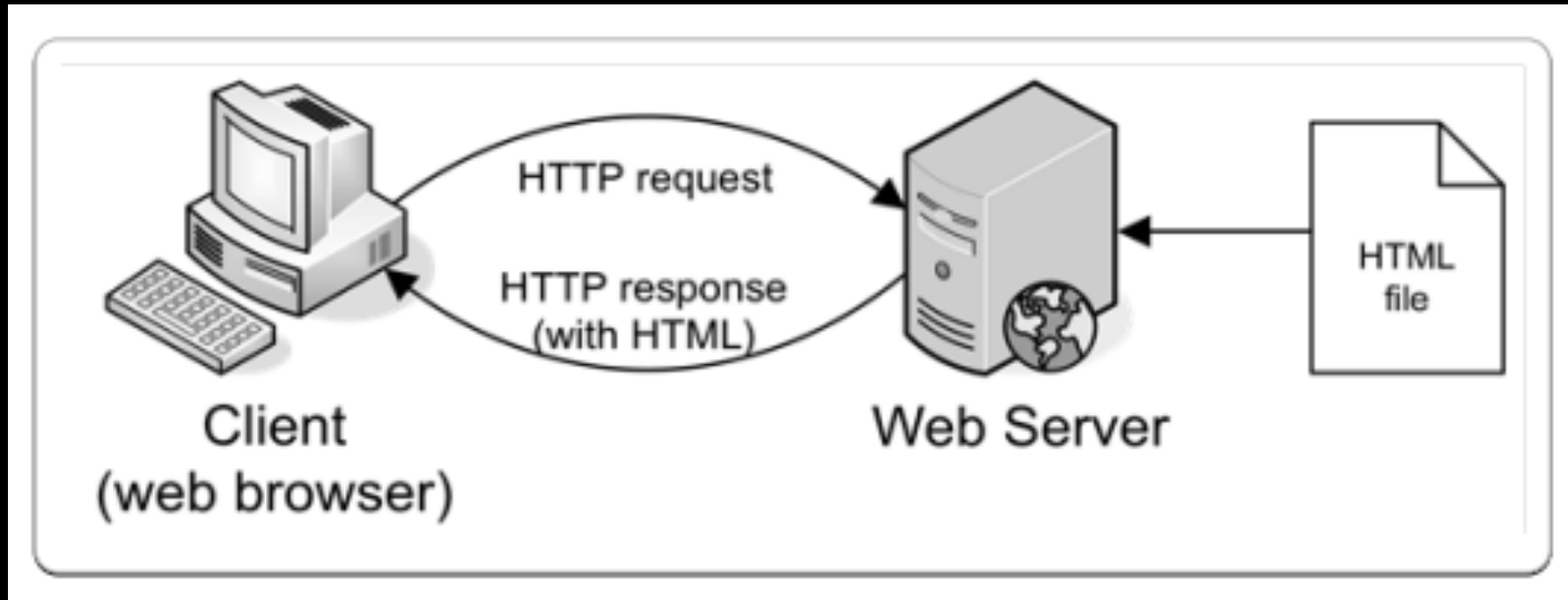
## Static vs. Dynamic Web Pages

Server side programming languages (e.x. JavaScript, PHP, .NET, Python, Perl) make creating **dynamic web pages** possible. Using HTML and CSS we have been creating **static web pages** in this class. Static web pages will always contain **the same content or information** in them no matter what, until you change the source code of the page. Dynamic web pages are capable of producing **different content** for a web page or web site using the same source code, based on the application logic written by the developer. (**RSS feeds** were one of the first instances of this w/ **Web 2.0**).

Dynamic web pages will often be powered by a server side programming language and a database (e.x. JavaScript, MySQL, SQL Server, Oracle). Together, these technologies can determine what information should be returned to the browser and construct dynamic content for the client side to present to the user.

## Request-Response Procedure

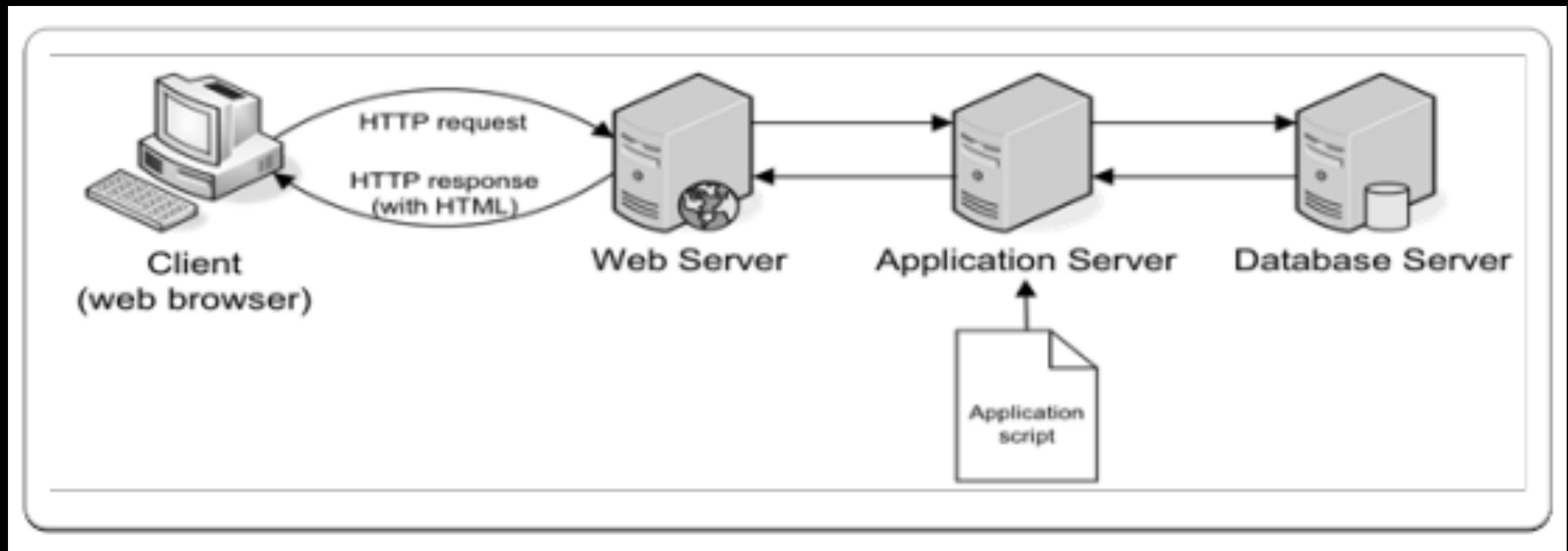
The HTTP [request-response](#) procedure explains how you are able to access web sites over the internet. When working with regular client side technologies, the request-response cycle is pretty simple.



1. You enter a website address <http://example.com> into the browser's address bar.
2. The browser looks up the address for <http://example.com> and requests the web page associated with it.
3. The request traverses the internet for the server that lives at <http://example.com>
4. The server at <http://example.com> receives the request and sends the (HTML) web page for <http://example.com>.
5. The browser receives the web page sent from the server and shows it to the user.

## Server Side

1. You enter a website address <http://example.com> into the browser's address bar.
2. The browser looks up the address for <http://example.com> and requests the web page associated with it.
3. The request traverses the internet for the server that lives at <http://example.com>.
4. The server at <http://example.com> receives the request and interprets it as a PHP file.
5. The PHP interpreter executes the PHP code and passes any MySQL statements to the database to be processed.
6. The PHP interpreter and MySQL return the resulting data to the web server as HTML.
7. The web server sends the generated page to the browser, which displays it to the user.



# JS Syntax

## Variables

Variables are containers that you can store values in. You start by declaring a variable with the var keyword, followed by any name you want to call it

```
var theVariable;
```

```
var theVariable = 'Bob';
```

——> String Data

```
var theVariable = 10;
```

——> Numeric Data

```
var theVariable = true;
```

——> Boolean

```
var theVariable = [536, 3, 3354684, 325]
```

——> Array

## JS Syntax

### let

allows you to declare variables that are limited in scope to the block, statement, or expression on which it is used. This is unlike the **var** keyword, which defines a variable globally, or locally to an entire function regardless of block scope.

At the top level of programs and functions, let, unlike var, does not create a property on the global object.

**\*\* Note:** the browser is not able to identify local variables.