

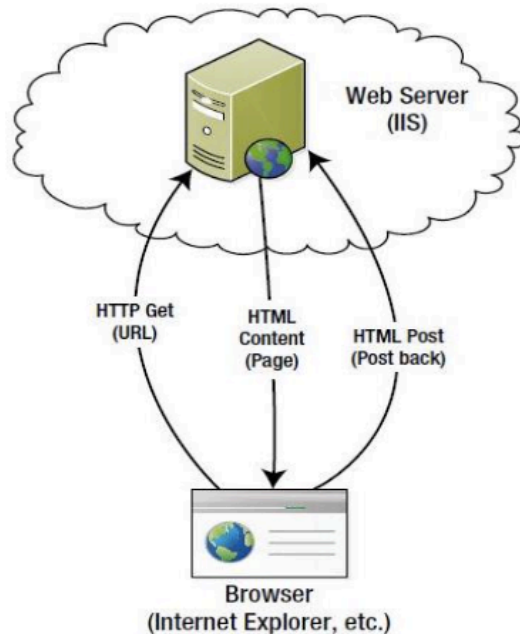
# Web Page

---

## HTTP

No início a web, era bastante simples. O modelo incluía um **servidor web**, que é responsável por armazenar e permitir o acesso ao conteúdo da rede, e o navegador (**cliente**) que renderiza as páginas de volta para o cliente.

# Web Page



Se a página web inclui um formulário com campos (*fields*), o navegador pode prover a entrada desses dados. quando a página for submetida, estes dados são enviados para o servidor web através de um **HTTP POST**.

Aspectos importantes desse modelo:

- O modelo é muito centrado na página.
- Desenvolvimento básico com um servidor web e um cliente.

**Figure 1-1.** A simple page-centric web model

# Web page

---

## Web centrada na página

Como já mencionado, os sites web são predominantemente focados em páginas web. Uma página é requisitada, retornada e renderizada. Dados nas páginas são postados para o servidor, processados, e uma página atualizada é retornada e renderizada. Como o servidor web é **stateless**, ele não sabe nada sobre a página anterior que foi retornada.

# Web page

## **Modelo Client/Server**

Nesse modelo, quando construímos uma aplicação web, temos de considerar dois componentes um servidor e um cliente. O servidor, por exemplo, o IIS, responde pelas requisições **HTTP**. Para conteúdo estático, os arquivos HTML podem ser simplesmente armazenados em uma pasta virtual dentro do **IIS** e nenhuma programação é requerida. Para conteúdo dinâmico, é necessário uma aplicação web para gerar **HTML (ASP.NET)**.

**ASP.NET** permite que você escreva código dinamicamente para gerar HTML. Por exemplo, a página pode consultar um banco de dados e popular um grid usando os dados retornados do banco de dados. Como também, os dados apresentados em um HTTP POST, podem ser escritos no banco de dados.

# Web page

## **Melhorar a experiência web**

Em uma abordagem centrada na página, um grande obstáculo é a iteratividade com o usuário. Para resolver esses problemas, duas principais melhorias foram introduzidas:

- Script do lado do cliente
- AJAX

# Web page

## **Script do lado do cliente**

Todo navegador agora prover a habilidade para rodar script do lado cliente, os quais são predominantemente escritos em JavaScript, embora possam ser escritos em outras linguagens tais como C#, VBScript, etc. a habilidade para rodar script no cliente é uma grande melhoria. Por exemplo, um script pode esconder/exibir ou modificar um conteúdo, com base na entrada do usuário.

**Atenção!** JavaScript pode ser desabilitado no cliente e você deve considerar, e testar, como sua página irá funcionar com o script desabilitado.

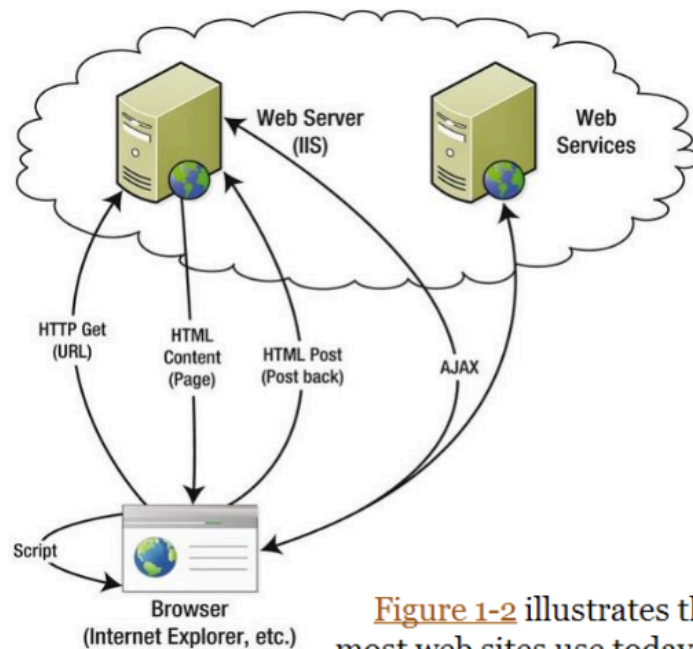
# Web page

## **AJAX**

**AJAX** é um acrônimo para assíncrono JavaScript e XML. **AJAX** é uma coleção de tecnologias que habilita script do lado cliente, comunicar-se com o servidor web, fora do típico cenário reflexo de página. Você pode usar o AJAX para requisitar dados do servidor web, atualizar o conteúdo da página com esses dados, usando o *Document Object Model* (**DOM**). Isto permite que porção da página web seja atualizada sem refrescar a página.

# Web page

## AJAX



Com a inclusão do script do lado cliente e requisições **AJAX**, podemos agora criar soluções web mais iterativas. Claro, isso requer aplicações mais complexas e um conjunto de tecnologias extras para se trabalhar, tanto no cliente quanto no servidor.

**Figure 1-2** illustrates the more robust model that most web sites use today.



# Web page

## **Reverendo as tecnologias Web**

Vamos, rapidamente rever, as várias tecnologias, que você necessitaria usar, para construir aplicações Web interativa e ótimo visual.

- HTML – HyperText Markup Language
- CSS – Cascading Style Sheets
- DOM – Document Object Model
- ECMAScript – JavaScript é um dialeto do ECMAScript Standard.

# Web page

## HTML5?

**HTML5** é um conjunto de especificações relacionadas a padronização dos navegadores Web, muitos dos quais não tem nada a ver com a HTML. Existem algumas coisas que você deve ter em mente:

- Muitas das especificações ainda não foram finalizadas.
- Nem todo navegador dão suporte as essas especificações
- Cada vendedor de navegador decide como implementar essas características.

# Web page

## **HTML5 – novos elementos de marcação**

O elemento <div> ainda é suportado, mas novos elementos foram fornecidos. As novas tag de conteúdo são:

- <article>
- <aside>
- <footer>
- <header>
- <hgroup>
- <nav>
- <section>

# Web page

## **HTML5 – novos elementos de marcação**

Os novos tipos são:

- color
- datetime (bem como datetime-local, date, time, month, e week)
- email
- number
- range
- search
- tel
- url

# Web page

## HTML5 – novos elementos de marcação

- <audio>
- <figcaption>
- <figure>
- <meter>
- <output>
- <progress>
- <vídeo>

E também temos o novo **<canvas>**, elemento que fornece capacidade gráfica.

# Web page

## HTML5 – outras funcionalidades

As novas funcionalidades são:

- Offline Cache – você pode especificar as páginas que são requeridas off-line, incluindo arquivos CSS e JavaScript. A especificação, também inclui uma API e manipuladores de eventos, que permite você monitorar e controlar a cache local.
- Drag and Drop – fornece a habilidade para selecionar um item e arrastá-lo para a página web.
- Web workers – permite executar um script em uma thread separada.
- Web storage – inclui **sessionStorage** e **localStorage** (armazena dados no cliente).
- Geolocation – define uma API que pode ser chamada do JavaScript para determinar a localização geográfica corrente.
- Web sockets – fornece comunicação assíncrona entre páginas web (navegador) e servidor.

