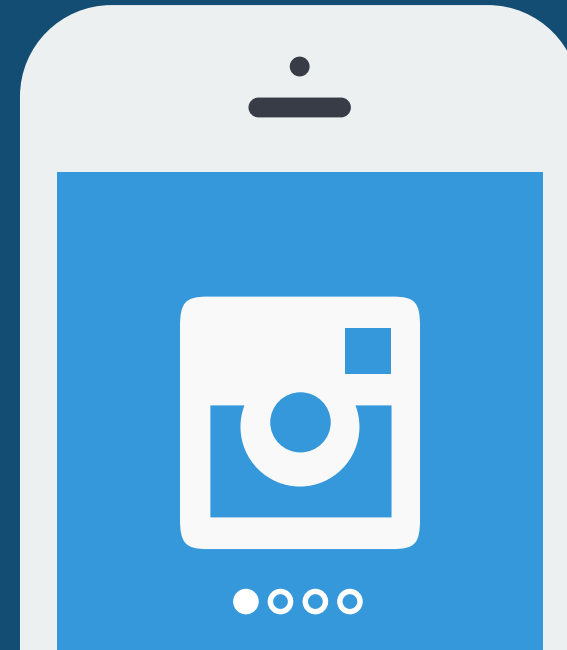
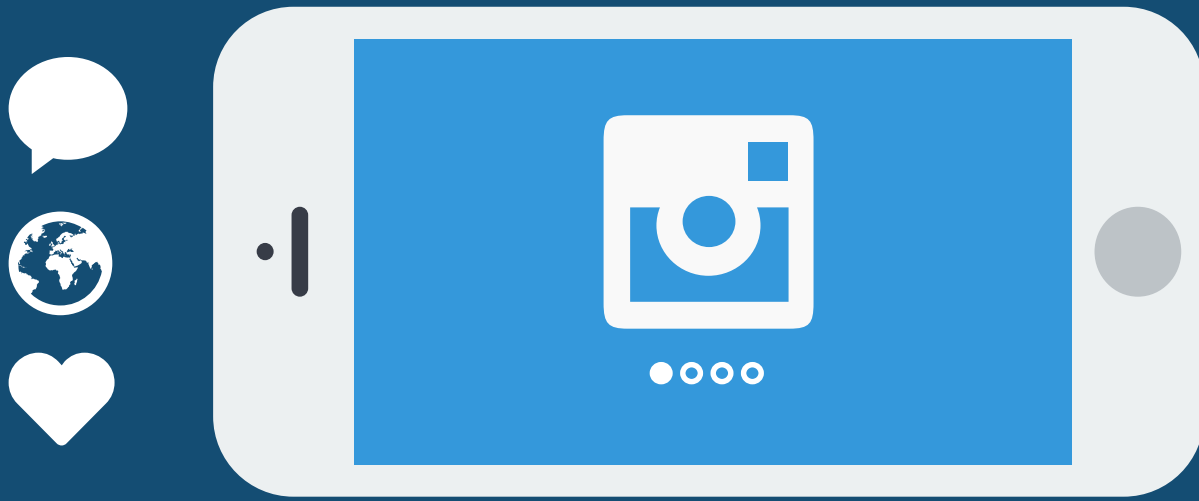


# targettrust

treinamento e tecnologia



# JSON AJAX JSONP



# JSON

- JSON (**Notação de Objetos JavaScript**) é um formato de dados permutável. Ele assemelha - se próximo a um sub tipo da sintaxe do **JavaScript**, mas ele não é um tipo de script. Isto é útil quando escrevemos qualquer tipo de aplicação baseada em **JavaScript**, incluindo web sites e extensões de navegadores.



# Usando JSON

```
var foo = {};  
foo.bar = "new property";  
foo.baz = 3;  
var JSONfoo = JSON.stringify(foo);  
//JSONfoo now holds {"bar":"new property","baz":3}.
```

Para fazer JSONfoo voltar a ser um objeto JavaScript faça:

```
var backToJS = JSON.parse(JSONfoo);
```



# Então JSON é JavaScript?



**Não exatamente. Embora o JSON se pareça com o JavaScript, tem as seguintes regras restritivas:**

- O JSON representa seis tipos de valores: objetos, arrays, números, strings, booleans e o null literal.
- Datas não são reconhecidas como um tipo de valor único.
- O conceito de um identificador JavaScript não é compreendido pelo JSON. Todos os nomes-chave têm que ser strings JSON.
- Strings JSON têm que estar entre aspas.
- Números JSON não podem ter zeros à esquerda.



# JSONP

- Vimos que podemos usar JSON para transportar dados entre servidor e cliente, e que isso pode ser feito com relativa segurança. Mas e para buscar dados em outros domínios?
- Sei que o **Twitter** tem uma **API** poderosa para obter dados históricos de **tweets**, mas sou limitado pela política de mesma origem.
- Ou seja, a menos que meu cliente esteja no domínio twitter.com, o uso de um **XHR get** comum me retornará nada mais do que um erro **HTTP**.



# JSON Padding

- Enquanto o padding (prefixo) é tipicamente o nome da função callback que é definida dentro do contexto de execução do navegador, ela pode ser uma atribuição de uma variável, trecho de if , ou qualquer outro trecho de código Javascript.
- O uso típico do JSONP provê acesso cross-domain a uma API JSON existente, encapsulando o conteúdo JSON em uma chamada de função.



# AJAX

- Asynchrhonous Javascript and XML;
- Encapsula técnicas para serem utilizadas em páginas dinâmicas;
- Outras linguagens, além de XML, também são aceitas;
- Ao invés de fazer o usuário enviar uma requisição junto com a página toda, manda só a parte que interessa;





# AJAX

- Denominado JavaScript assíncrono + XML enquanto não era uma tecnologia em si, mas um termo empregado em 2005 por Jesse James Garrett descreve uma "nova" aproximação ao uso de um número de tecnologias recentes juntas, incluindo: HTML ou XHTML, Cascading Style Sheets, JavaScript, The Document Object Model, XML, XSLT, e o mais importante: objeto XMLHttpRequest.



# Como fazer uma requisição HTTP

```
var httpRequest;  
if (window.XMLHttpRequest) { // Mozilla, Safari, ...  
    httpRequest = new XMLHttpRequest();  
} else if (window.ActiveXObject) { // IE 8 and older  
    httpRequest = new ActiveXObject("Microsoft.XMLHTTP");  
}
```



# AJAX

- Primeiro decida com qual script/página você quer se conectar;
- Depois, decida qual área da sua página você quer inserir o resultado;
- PS: vamos para um servidor HTTP;
- Queremos evitar quaisquer problemas com “Cross origin requests”;



```
function makerequest(serverPage, objID) {  
  
    var obj = document.getElementById(objID);  
  
    xmlhttp.open("GET", serverPage);  
  
    xmlhttp.onreadystatechange = function() {  
  
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {  
  
            obj.innerHTML = xmlhttp.responseText;  
  
        } }  
  
    xmlhttp.send(null); }
```



# Manuseio da Resposta do Servidor

**A lista completa dos readyState valores é a seguinte:**

- 0 (não inicializado)
- 1 (carregamento)
- 2 (carregado)
- 3 (interativa)
- 4 (completo)



# Exercício 1



- Crie um HTML e faça uma chamada à função **makeRequest()** no evento “**onLoad**” do `<body>`.
- Passe como parâmetro um arquivo texto (txt) com qualquer conteúdo e o ID de uma div (use aspas simples).
- Mude o txt para html e teste novamente.
- Usando este código, vamos para o próximo exercício.



# Exercício 2

Crie um menu usando o seguinte padrão:

```
<a href="pagina1.html"
onclick="makerequest('pagina1.html','res');
return false;"> Página 1</a>
```

O menu deve apontar para quatro páginas.

```
<div style="width: 770px; text-align:
left;">

    <h1>Page 2</h1><p>TEXTTO...</p>

</div>
```



# Exercício 3

- Construir um formulário HTML e incluir novos campos conforme o usuário interage com o formulário. Exemplo: Seleção de Estado em um Combo, carregar as cidades correspondentes.

**Dica: usar PHP**

