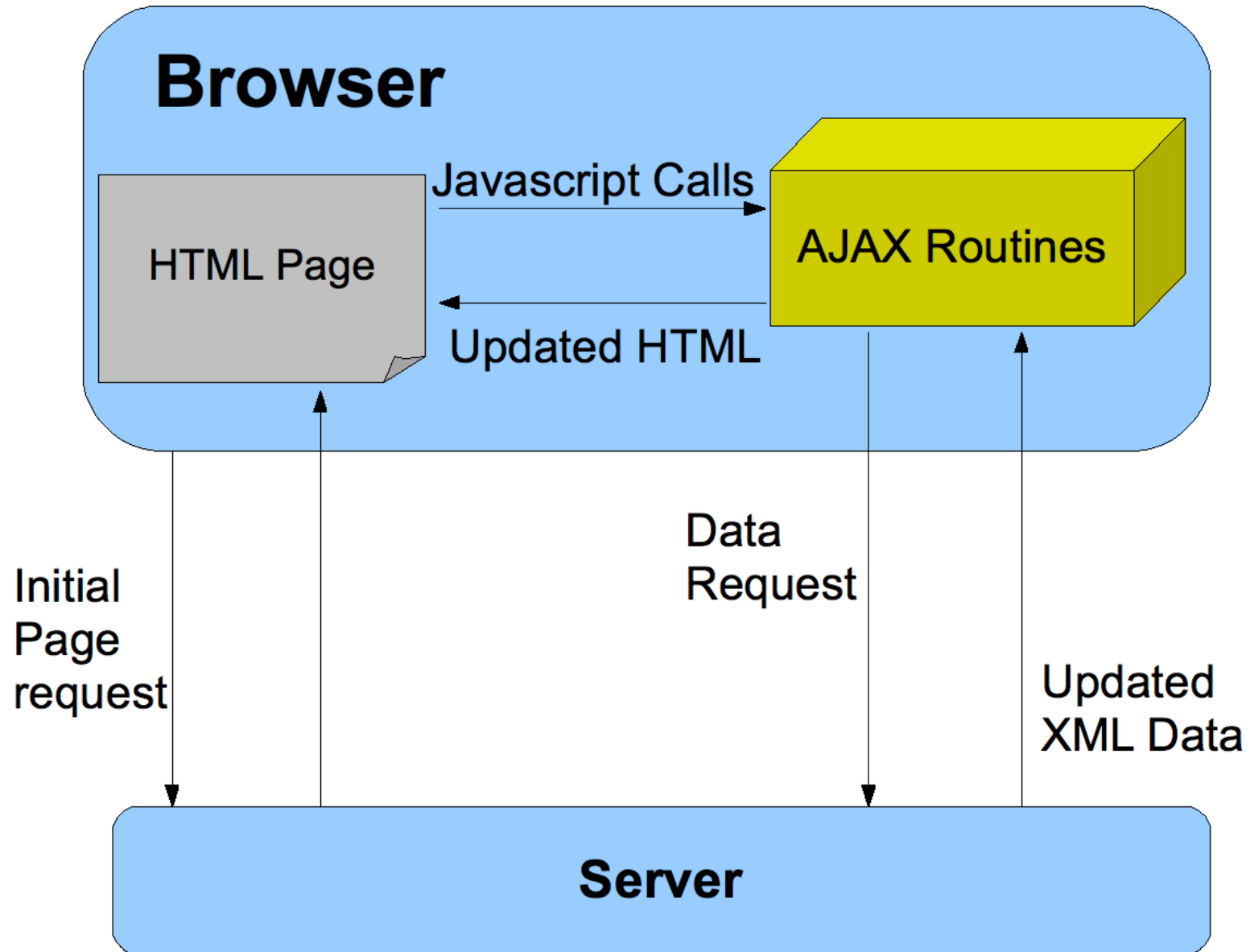


# Ajax



- Es el acrónimo de Asynchronous Javascript And XML.
- Nos permite intercambiar información con el servidor sin necesidad de recargar la página mediante peticiones asíncronas.
- A pesar de que su nombre lo dice, XML no es requerido para usar AJAX, de hecho, se utiliza JSON.

# Ajax



- Con jQuery, hacer uso de AJAX es muy sencillo.
- Utilizaremos la web UAPI <https://dev.datos.ua.es/uapi> , a la cual haremos solicitudes utilizando AJAX y jQuery.
- jQuery provee varias funciones para trabajar con AJAX.
  - La más comúnmente usada \$.ajax()
  - Mas info sobre JQuery+Ajax:
    - <http://api.jquery.com/jquery.ajax/>

# Ajax - JQuery

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-2.1.0.min.js"></script>
  <meta charset="utf-8">
  <title>jQuery Ajax</title>

  <script>

    function testAjax() {

      $.ajax({

        url: 'https://dev.datos.ua.es/uapi/72CB7RhNf9RKS4PTW5wm/datasets/6/data ',
        type: 'GET',
        dataType:"jsonp",
        success: function(data) {
          $('body').append('<pre>' + JSON.stringify(data, null, 2) + '</pre>');
        },
        error: function(jqXHR, textStatus, error) {

          alert( "error: " + jqXHR.responseText);
        }
      });
    }
  </script>
</head>
<body>
  <button onclick="testAjax();">click para probar</button>
</body>
</html>
```

El ejemplo hace uso de jQuery AJAX para obtener un objeto json de la UAPI.

Los parametros pasados a la función AJAX:

**url:** La dirección a donde enviar la solicitud.

**type:** Tipo de request (solicitud).  
Ejemplo: GET, POST, PUT, DELETE, etc. En caso de utilizar POST o PUT, por ejemplo, se puede enviar un objeto en otro parámetro a la misma función llamado data. Ej: data: {'clave': 'valor'},.

**datatype:** El tipo de respuesta que se espera, en este caso es json.

Las funciones success y error son llamadas cuando el request fue exitoso o falló respectivamente.