

```
{  
  "notas": [  
    {  
      "para": "Jorge",  
      "de": "Juan",  
      "cuerpo": "¡No olvides la tarea!"  
    },  
    {  
      "para": "Carmen",  
      "de": "Jose Luis",  
      "cuerpo": "¡Llego mas tarde!"  
    },  
    {  
      "para": "Hilda",  
      "de": "Gerardo",  
      "cuerpo": "Te veo mañana a las 6:00pm"  
    }  
  ]  
}
```

# Introducción a JSON y AJAX

Arquitectura de la Información  
MA. Judith Miguel Hernández

# Objetos en Javascript

Objetos son variables, pero pueden contener diferentes valores.

```
var persona= {  
    nombre: "Juan",  
    edad: 30,  
    email: "juan@gmail.com"  
}
```

Para acceder a los valores de un objeto

```
var nom= persona.nombre  
var anios= persona.edad  
var correo= persona.email
```

Ó

```
var nom= persona["nombre"]  
var anios= persona["edad"]  
var correo= persona ["email"]
```

```
var elementoHTML= document.getElementById("elementoParrafo")  
elementoHTML.innerHTML= "¡Hola mundo!"
```

# INTRODUCCIÓN A JSON

- JSON significa JavaScript Object Notation (Notación de Objetos en Javascript) y es un estándar para almacenar y transportar datos.
- Se utiliza principalmente para intercambiar datos entre sistemas desarrollados en diferentes lenguajes.
- JSON separa los datos de la representación (pantalla o vista)
- JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores.
- JSON es texto, pero se puede convertir en un objeto Javascript fácilmente, se consideran tipos de datos semiestructurados

# SINTAXIS JSON

- La sintaxis JSON es un subconjunto de la sintaxis de JavaScript, pero JSON no es Javascript, sus reglas son:

- ☐ Los datos inician con llaves
- ☐ Los datos están en pares de nombre : valor
- ☐ Los datos están separados por comas.
- ☐ Las llaves apoyan objetos
- ☐ Los corchetes sostienen arreglos o matrices

```
{  
  "notas": [  
    {  
      "para": "Jorge",  
      "de": "Juan",  
      "cuerpo": "¡No olvides la tarea!"  
    },  
    {  
      "para": "Carmen",  
      "de": "Jose Luis",  
      "cuerpo": "¡Llego mas tarde!"  
    },  
    {  
      "para": "Hilda",  
      "de": "Gerardo",  
      "cuerpo": "Te veo mañana a las 6:00pm"  
    }  
  ]  
}
```

# TIPOS DE DATOS JSON

- Cadenas (String)

```
{ "nombre": "Juan" }
```

- Números

```
{ "edad": 30 }
```

- Objetos

```
{  
  "empleado": { "nombre": "Juan", "edad": 30,  
                "city": "New York" }  
}
```

- Arreglos

```
{  
  "empleados": [ "Juan", "Ana",  
                  "Pedro" ]  
}
```

- Booleanos

```
{ "venta": true }
```

- Nulos

```
{ "apellido": null }
```

# Ejemplo de representación de datos JSON

```
{
  "arrayColores":[{
    "nombreColor":"rojo",
    "valorHexadec":"#f00"
  },
  {
    "nombreColor":"verde",
    "valorHexadec":"#0f0"
  },
  {
    "nombreColor":"azul",
    "valorHexadec":"#00f"
  },
  {
    "nombreColor":"cyan",
    "valorHexadec":"#0ff"
  },
  {
    "nombreColor":"magenta",
    "valorHexadec":"#f0f"
  },
  {
    "nombreColor":"amarillo",
    "valorHexadec":"#ff0"
  },
  {
    "nombreColor":"negro",
    "valorHexadec":"#000"
  }
]
}
```

Nombre: arrayColores

Valor: un arreglo [] de objetos JSON.

```
{
  "arrayColores":[{
    "rojo":"#f00",
    "verde":"#0f0",
    "azul":"#00f",
    "cyan":"#0ff",
    "magenta":"#f0f",
    "amarillo":"#ff0",
    "negro":"#000"
  }
]
}
```

Nombre: arrayColores

Valor: un arreglo [] de un solo objeto JSON.

```
{
  "rojo":"#f00",
  "verde":"#0f0",
  "azul":"#00f",
  "cyan":"#0ff",
  "magenta":"#f0f",
  "amarillo":"#ff0",
  "negro":"#000"
}
```

Valor: un solo objeto con siete pares nombre:valor.

# EJEMPLO ARCHIVO JSON QUE REPRESENTA UNA TABLA

```
{  
  "notas": [  
    {  
      "para": "Jorge",  
      "de": "Juan",  
      "cuerpo": "¡No olvides la tarea!"  
    },  
    {  
      "para": "Carmen",  
      "de": "Jose Luis",  
      "cuerpo": "¡Llego mas tarde!"  
    },  
    {  
      "para": "Hilda",  
      "de": "Gerardo",  
      "cuerpo": "Te veo mañana a las 6:00pm"  
    }  
  ]  
}
```

NOTAS		
PARA	DE	CUERPO
Jorge	Juan	¡No olvides la tarea!
Carmen	Jose Luis	¡Llego más tarde!
Hilda	Gerardo	Te veo mañana

# CONVIRTIENDO JSON A JAVASCRIPT

- Imagina que recibimos este código JSON de un servidor web:

```
{  
  nombre: "Juan",  
  edad: 30,  
  email: "juan@gmail.com"  
}
```

Para manipularlo en el Javascript hacemos

```
var txt='{nombre: "Juan",  
edad: 30, email:juan@gmail.com"}'  
var obj= JSON.parse (txt)  
var name= obj.nombre  
var age= obj.edad  
var mail= obj.email
```



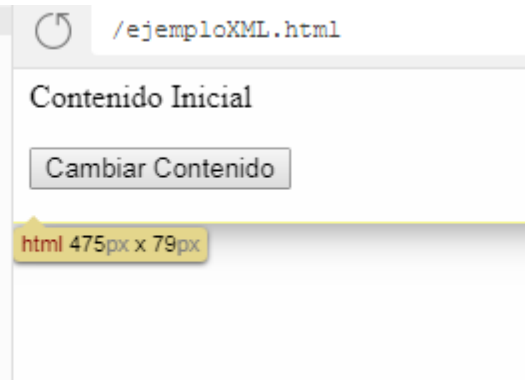
# Introducción a AJAX (JavaScript Asíncrono + XML )

- AJAX no es una tecnología por sí misma, es un término que utiliza conjuntamente varias tecnologías existentes (HTML, CSS, JS, JSON, XML)
- Su funcionamiento se basa en el objeto **XMLHttpRequest** para hacer peticiones al servidor
- Sirve para:
  1. Actualizar una página web sin volver a cargar la página.
  2. Solicitar datos de un servidor - después de que la página se haya cargado
  3. Recibir datos de un servidor - después de que la página se haya cargado
  4. Enviar datos a un servidor de forma no visible (background)

# Petición de XMLHttpRequest

Más info: [https://www.w3schools.com/xml/dom\\_httprequest.asp](https://www.w3schools.com/xml/dom_httprequest.asp)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo XML </title>
    <script type="text/javascript" src="javascript.js"></script>
  </head>
  <body>
    <div id="demo">Contenido Inicial</div>
    <p>
      <input type="button" value="Cambiar Contenido" id="Cambiar">
    </p>
  </body>
</html>
```



```
window.onload=function(){

  var btn_Cambiar = document.getElementById("Cambiar");

  btn_Cambiar.addEventListener("click", cambiarTexto);

  var div_Demo= document.getElementById("demo");

  function cambiarTexto()
  {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
      if (this.readyState == 4 && this.status == 200) {
        div_Demo.innerHTML= xhttp.responseText;
      }
    };

    xhttp.open("GET", "contenido.txt", true);
    xhttp.send();
  }
}
```

Crea el objeto

**onreadystatechange** especifica una función que se ejecutará cada vez que cambie el estado del objeto

Cuando las propiedades **readyState** es 4 y **status** es 200 la respuesta está lista

**responseText** devuelve la respuesta del servidor como una cadena de texto, puede ser un JSON

Especifica el tipo y fuente de solicitud

Envía la solicitud

# Ejemplo solicitando un archivo JSON

archivo: json.txt

```
{  
  "notas": [  
    {  
      "para": "Jorge",  
      "de": "Juan",  
      "cuerpo": "¡No olvides la tarea!"  
    },  
    {  
      "para": "Carmen",  
      "de": "Jose Luis",  
      "cuerpo": "¡Llego mas tarde!"  
    },  
    {  
      "para": "Hilda",  
      "de": "Gerardo",  
      "cuerpo": "Te veo mañana a las 6:00pm"  
    }  
  ]  
}
```

código javascript asociado, se solicita al servidor el archivo json.txt

```
// JavaScript File  
window.onload=function(){  
  
    var btn_Cambiar = document.getElementById("Cambiar")  
  
    btn_Cambiar.addEventListener("click", cambiarTexto)  
  
    var div_Demo= document.getElementById("demo")  
  
    function cambiarTexto()  
    {  
        var xhttp = new XMLHttpRequest();  
        xhttp.onreadystatechange = function() {  
            if (this.readyState == 4 && this.status == 200) {  
                consumirJSON(this)  
            }  
        }  
    };  
  
    xhttp.open("GET", "json.txt", true)  
    xhttp.send()  
}
```

# Se procesa la respuesta del servidor

```
function consumirJSON(respuestaTXT)
{
    var objJson = JSON.parse(respuestaTXT.responseText);

    //var nodo = xmlDoc.getElementsByTagName("nota")
    var estructuraHTML;
    var i
    var fondo= ["#ff6666", "#85e085", "#4d94ff", "#ff6666", "#85e085", "#4d94ff", "#ff6666", "#85e085", "#4d94ff"];
    div_Demo.innerHTML="";
    //for ( i = 0; i < objJson.length; i++)
    for (i in objJson.notas)
    {
        estructuraHTML= ""
        estructuraHTML+= "Nota " + i + ": <br>"
        estructuraHTML += "Para: " + objJson.notas[i].para+ "<br>"
        estructuraHTML += "De: " + objJson.notas[i].de + "<br>"
        estructuraHTML += "Mensaje: " + objJson.notas[i].cuerpo + "<br><br><br>"
        var nuevoDiv = document.createElement("div")
        nuevoDiv.innerHTML= estructuraHTML;
        nuevoDiv.style.backgroundColor=fondo[i];
        nuevoDiv.style.fontFamily="Arial";
        nuevoDiv.style.width="500px";
        div_Demo.appendChild(nuevoDiv);
    }
}
```