



FACULTAD
DE INGENIERIA

Universidad de Buenos Aires

Desarrollo de Aplicaciones I

Ramos Matias

Clase 6

- JSON
- Materialize
- Ajax - POST



JSON (JavaScript Object Notation)

- JSON es un formato de intercambio de datos ligero. Es fácil para leerlo y escribirlo.
- JSON es un formato de datos basado en un subconjunto de la sintaxis de JavaScript:
 - literales de arrays y literales de objetos.
- JSON es completamente independiente del lenguaje



Literales de Objetos

- Un objeto es un conjunto desordenado de pares nombre/valor.
- Un objeto comienza con { y termina con }. Cada nombre es seguido por : y los pares nombre/valor están separados por , (coma).

```
var persona = { "nombre" : "Juan", "edad" : 35 };  
alert(persona.nombre); // Juan  
alert(persona.edad); // 35  
  
alert(persona["nombre"]); // Juan
```

Literales de Arrays

- Un array es una colección de valores.
- Un array comienza con [y termina con]. Los valores están separados por , (coma).

```
var nombres = [ "Juan", "Pablo", "Ramiro" ];  
alert(nombres[0]); // Juan  
alert(nombres[2]); // Ramiro  
  
var personas = [{ "nombre" : "Juan", "edad" : 35 },  
                 { "nombre" : "Anibal", "edad" : 26 }];  
alert(personas[0].nombre); // Juan  
alert(personas[1].nombre); // Anibal
```

JSON.parse

- El método `JSON.parse()` analiza una cadena de texto como JSON, transformando opcionalmente el valor producido por el análisis.

```
var str = '{ "nombre" : "Juan", "edad" : 35 } ';  
var obj = JSON.parse(str);  
  
alert(obj.nombre); //Juan
```

JSON.stringify

- JSON.stringify() convierte un valor a notación JSON representándolo:
 - Las propiedades de objetos no-array no se garantiza que se realicen en un orden particular.
 - Los objetos Boolean, Number, y String se convierten a sus valores primitivos.

```
var obj = { "nombre" : "Juan", "edad" : 35 };  
  
var str = JSON.stringify(obj);  
// { "nombre" : "Juan", "edad" : 35 }
```

Material Design

- Creado y diseñado por Google
- El objetivo de Google es desarrollar un sistema de diseño que permita una experiencia de usuario unificada en todos sus productos en cualquier



Container

- La clase *container* permite centrar el contenido de tu página.
- Se establece en ~ 70% del ancho de la ventana
- Usamos el contenedor para contener nuestro contenido dentro del body

```
<body>
  <div class="container">
    <!-- Page Content goes here -->
  </div>
</body>
```

El sistema de cuadrículas

- La cuadrícula estándar tiene 12 columnas.
- No importa el tamaño del navegador, cada una de estas columnas siempre tendrá el mismo ancho.
- s1 significa small-1, que en inglés simple significa "1 columna en pantallas pequeñas".

```
<div class="row">  
  <div class="col s1">1</div>  
  <div class="col s1">2</div>  
  <div class="col s1">3</div>  
  <div class="col s1">4</div>  
  <div class="col s1">5</div>  
  <div class="col s1">6</div>  
  <div class="col s1">7</div>  
  <div class="col s1">8</div>  
  <div class="col s1">9</div>  
  <div class="col s1">10</div>  
  <div class="col s1">11</div>  
  <div class="col s1">12</div>  
</div>
```

Las columnas están en las filas

Cuando creamos nuestro diseño todas las columnas deben estar contenidas dentro de una fila y que debe agregar la clase col a sus divs internos para convertirlos en columnas

```
<div class="row">
  <div class="col s1">1</div>
  <div class="col s1">2</div>
  <div class="col s1">3</div>
  <div class="col s1">4</div>
  <div class="col s1">5</div>
  <div class="col s1">6</div>
  <div class="col s1">7</div>
  <div class="col s1">8</div>
  <div class="col s1">9</div>
  <div class="col s1">10</div>
  <div class="col s1">11</div>
  <div class="col s1">12</div>
</div>
```

El sistema de cuadrículas

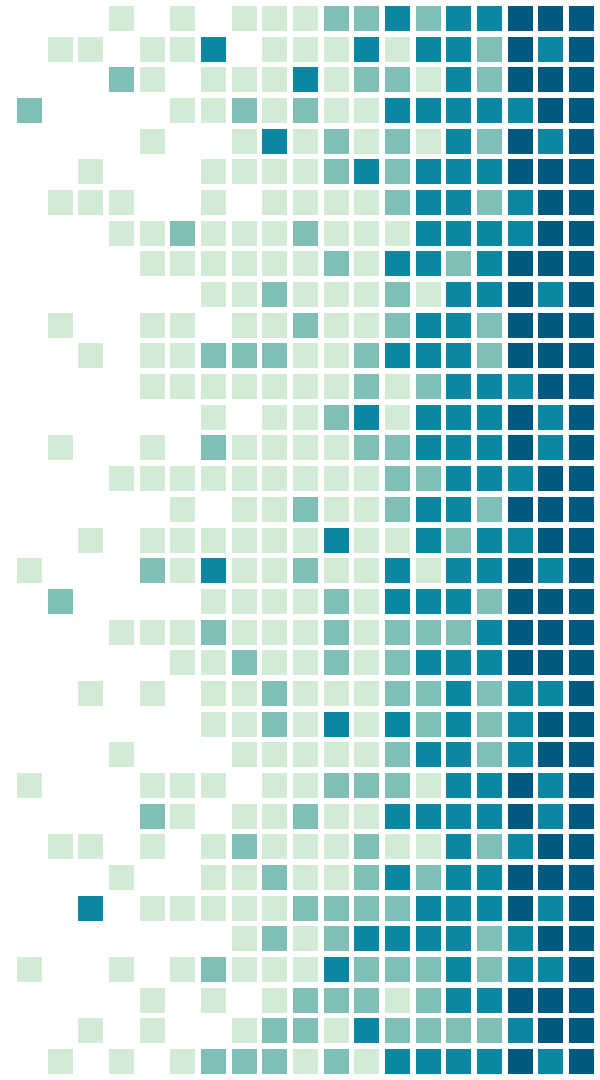
col s1	col s1	col s1	col s1	col s1	col s1	col s1	col s1	col s1	col s1	col s1	col s1
col s4				col s4				col s4			
col s4				col s8							
col s6						col s6					
col s12											



innerHTML y value

Al recuperar un elemento del DOM, vamos a tener acceso a dos atributos.

- Si es un input a través del `.value` accedemos al valor cargado
- Si es un elemento donde el contenido esta presente entre la etiqueta de apertura y cierre `.innerHTML`



Listar elementos

Los navegadores evalúan en tiempo de ejecución el texto plano que cargamos

- Si detecta una etiqueta HTML valida, dibuja su interpretación grafica
- Si no detecta una etiqueta, escribe el texto literal

```
let listaUl = document.getElementById("device")
for (let di of devices) {
  listaUl.innerHTML +=
    `<p>${di.description}</p>`;
}
```

Listar elementos

Una vez ya graficados los elementos en pantalla, ya podemos recuperarlos del DOM, para acceder a sus datos o configurarle los listeners.



Ajax POST

La diferencia entre la ejecución de un método GET o un método POST, es donde viaja la información.

Por lo cual hay que avisarle al servidor que tipo de información va a viajar.

- La info viaja en el cuerpo de la petición
- La configuración de que tipo de info va a viajar se hace en la cabecera de la petición

```
let xhr = new XMLHttpRequest();
xhr.onreadystatechange = ()=>{
  |  ~~~
}
xhr.open("POST", url, true);
xhr.setRequestHeader("Content-Type",
  | "application/json");
xhr.send(JSON.stringify(datos));
```