



Curso de Front-End

Clase 15



Agenda de la clase



Agenda

- Repaso.
- Vue.js.
- Ejercicios.



Repaso



¿Qué es un objeto? (En JavaScript)

Un **objeto** es una **entidad** que contiene una **colección de propiedades**.

Una propiedad es una asociación de clave-valor.

Las propiedades son similares a las variables comunes de JavaScript, la diferencia es que las propiedades son variables que están unidas a un determinado objeto.

El valor de una propiedad puede ser una función y en ese caso a la propiedad se le llama **método**.

```
var nombreDelObjeto = {  
  clave1: valor1,  
  clave2: valor2,  
  clave3: valor3  
};
```

Propiedades. (Cada propiedad se separa con una coma ",").

Las claves y valores se separan con dos puntos ":".



Ejemplo de un objeto

```
var persona1 = {  
  nombre: "María",  
  apellido: "Rodríguez",  
  edad: 36,  
  email: "maria.rod@gmail.com",  
  nacionalidad: ["Uruguay", "España"],  
  nombreCompleto: function () {  
    return this.nombre + " " + this.apellido;  
  },  
};
```

La palabra `this` es una palabra reservada del lenguaje. En JavaScript tiene un comportamiento muy especial, y por lo tanto se recomienda leer estos [docs de MDN](#).

```
console.log(persona1.apellido); // Retorna "Rodríguez".  
persona1["apellido"] = "Pérez"; // Asigna "Pérez" como apellido, usando "Bracket notation".  
persona1.apellido = "López";    // Asigna "López" como apellido, usando "Dot notation".  
persona1.nombreCompleto();      // Retorna "María López" (notar que el apellido no es "Rodríguez" ni "Pérez").
```



Ejemplo de un objeto

```
var personal = {  
  nombre: "María",  
  apellido: "Rodríguez",  
  edad: 36,  
  email: "maria.rod@gmail.com",  
  nacionalidad: ["Uruguay", "España"],  
  nombreCompleto: function () {  
    return this.nombre + " " + this.apellido;  
  },  
};
```



Para pensar... ¿Está bueno guardar el dato de la edad de la persona? ¿Qué otro dato más útil se podría haber guardado en su lugar? En dicho caso, ¿cómo se haría para saber la edad de la persona? ¿Se podría crear un método para ello?



A veces, editar el DOM de forma
“directa” no es la mejor opción...



¿Editar el DOM de forma “directa”? (1/3)

Con el paso del tiempo, las **necesidades** de desarrollo Front-End **han cambiado**. Antiguamente los sitios web solían ser simples páginas que mostraban información de una empresa.

Actualmente, **muchos sitios web se parecen más a una aplicación de escritorio** que a una página informativa. Ej: Gmail, WhatsApp Web, Google Slides, Figma, etc.

Estos **sitios interactivos** requieren de una gran cantidad de modificaciones del DOM. En estos casos, manipular el DOM de forma directa deja de ser útil y empieza a ser difícil de usar. Se podría decir que es útil siempre y cuando las modificaciones al DOM sean pequeñas y pocas.

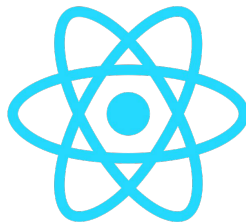


¿Editar el DOM de forma “directa”? (2/3)

A la hora de desarrollar **web apps** del lado del **cliente** (a las cuales también se les puede llamar *Single Page Applications*), existen librerías y/o frameworks muy poderosos y que simplifican mucho la tarea.

Algunos ejemplos son:

- [React](#) de Facebook.
- [Angular](#) de Google.
- [Vue.js](#) de Evan You.



👉 Estas librerías/frameworks se basan en un **paradigma** de programación llamado “**declarativo**”, que se contrapone al **paradigma “imperativo”** que veníamos utilizando en las últimas clases (cuando manipulábamos el DOM de forma “directa”).



¿Editar el DOM de forma “directa”? (3/3)

En este curso de Front-End les daremos un **pantallazo inicial** sobre este tipo de librerías/frameworks basados en el **paradigma declarativo**.

Elegimos **Vue.js (versión 2)** simplemente por su facilidad de uso para principiantes. En este momento, sería casi imposible darles una introducción a React o Angular ya que tienen una curva de aprendizaje mucho más empinada.

En cursos más avanzados de Hack Academy se utilizará React.





Vue.js



¿Qué es Vue.js? (1/2)

- Es un **framework JavaScript** para **Front-End**.
- Fue escrito por una sola persona: [Evan You](#) (aunque actualmente existe una comunidad de colaboradores).
- No es tan popular como otros frameworks como Angular y React, pero es más **simple** y por lo tanto más **fácil de aprender**, sobre todo en su **versión 2**. De todas maneras, Vue.js está [creciendo muy rápidamente](#).
- Se dice que Vue.js es “**progresivo**” dado que permite empezar a trabajar con él de una forma muy simple e ir agregándole funcionalidades de a poco, a medida que se necesite.
- Se dice que sigue el paradigma de programación **declarativo**, en lugar del imperativo (que veníamos usando hasta ahora para manipular el DOM).



¿Qué es Vue.js? (2/2)

- Es open source. Pero cuidado, aún la comunidad no es tan grande como la de Angular o React.
- La [versión 2.6](#) tiene aprox. 11.900 líneas de código. La versión `min` pesa 87kB.
- No soporta Internet Explorer 8 (y anteriores).
- Documentación: <https://v2.vuejs.org/v2/guide/>.



Instalación de Vue.js



Instalación de Vue.js

La instalación de Vue.js es muy sencilla. Similar a como se instaló Bootstrap, se puede copiar y pegar el link del CDN.

```
<body>
  ...
  ...
  ...
  <script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
</body>
```

Nota: En caso de necesitarlo, se podría instalar Vue.js y Bootstrap en la misma página.



Utilización de Vue.js



Utilización de Vue.js (1/4)

1^{er} paso: Crear una **instancia (un objeto) de Vue.js** haciendo uso del [operador new](#). Además es necesario definir sobre qué elemento HTML tendrá “control” dicho objeto.

HTML:

```
<div id="app">  
  
</div>
```

JS:

```
var miVueApp = new Vue({  
  el: "#app",  
  data: {  
  
  },  
});
```

En este ejemplo, se creó un objeto llamado **miVueApp**.

Además, con el atributo **el** se especificó que **miVueApp** tendrá “control” sobre el elemento con **id** “**app**”.

Tanto el nombre **miVueApp** como **app** son arbitrarios.



Utilización de Vue.js (2/4)

2^{do} paso: Definir los **datos** que deben ser “gestionados” por Vue.js, dentro del objeto **data**. En general suelen ser los datos **que se quieren mostrar** en la página web.

HTML:

```
<div id="app">
  <p>{{ mensaje }}</p>
</div>
```

JS:

```
var miVueApp = new Vue({
  el: "#app",
  data: {
    mensaje: "Hola"
  }
});
```

En este ejemplo, se creó un atributo `mensaje` (nombre arbitrario) al cual se le asignó el string `"Hola"`. Para mostrar dicho dato en el HTML (más exactamente, en el DOM), se utilizan las **dobles llaves**: `{{ }}`.

Automáticamente, Vue.js se encargará de tomar el contenido de `mensaje` y mostrarlo en el lugar definido en el HTML (en el `<p>`).

Notar que en ningún lugar del archivo JS se hace referencia al lugar dónde aparecerán los datos. **Es mágico** 🧙.



Utilización de Vue.js (3/4)

3er paso: En caso de necesitarlo, también es posible **enviar datos desde el HTML al JS**, es decir, el camino inverso al anterior. El ciclo completo se conoce como: **Two-way-Binding**.

HTML:

```
<div id="app">
  <p>{{ mensaje }}</p>
  <input type="text" v-model="mensaje">
</div>
```

JS:

```
var miVueApp = new Vue({
  el: "#app",
  data: {
    mensaje: "Hola"
  }
});
```

En este ejemplo, se creó un elemento `<input>` y se le agregó el atributo `v-model` con el valor "mensaje" (que coincide con el nombre de atributo `mensaje` dentro de `data`).

De este modo, lo que el usuario escriba dentro del `<input>`, automáticamente se guardará dentro de `mensaje`, sobrescribiendo el valor inicial "Hola". Y no sólo eso, también se actualizará el contenido del `<p>`. **Esto es muy mágico** 🤖🤖🤖🤖🤖🤖.



Utilización de Vue.js (4/4)

Notar que ahora, gracias a Vue.js, en el archivo JavaScript no es necesario escribir código para manipular el DOM de forma directa. Por eso se dice que es **declarativo** (en lugar de imperativo).

Es decir que, gracias a un framework como Vue.js, es posible **separar mejor la “lógica de negocio” de la manipulación del DOM**.

👉 No olvidar leer la documentación: <https://v2.vuejs.org/v2/guide/> (**versión 2**).



Vue.js y Arrays



Vue.js y Arrays

En caso de que se necesite **mostrar el contenido de un array**, es necesario utilizar un atributo especial llamado `v-for`. Es lo que se conoce como [List Rendering](#).

HTML:

```
<ul id="app">
  <li v-for="fruta in frutas">
    {{ fruta.nombre }}
  </li>
</ul>
```

JS:

```
var app = new Vue({
  el: "#app",
  data: {
    frutas: [
      { nombre: "Manzana", color: "Rojo" },
      { nombre: "Banana", color: "Amarillo" },
      { nombre: "Pera", color: "Verde" }
    ]
  }
});
```

En este ejemplo, el `v-for` se aplicó a un ``, pero se podría haber aplicado a cualquier elemento HTML que se quisiese repetir tantas veces como ítems hubiesen dentro del array.



Vue.js y Atributos HTML



Vue.js – Atributos HTML

En caso de que se necesite colocar un dato (de `data`) dentro de un atributo HTML (ej: `href` en un `<a>`), no será posible usar la notación doble llave `{{ }}`. Será necesario utilizar la notación `v-bind:href`.

HTML:

```
<div id="app">
  <a v-bind:href="direccion">
    {{ titulo }}
  </a>
</div>
```

JS:

```
var app = new Vue({
  el: "#app",
  data: {
    direccion: "https://ha.dev",
    titulo: "Hack Academy"
  }
});
```

Ver [documentación](#).



Ejercicio 1

Blog



Ejercicio 1 – Blog

Supongamos que los desarrolladores de Back-End nos enviaron un array que contiene un montón de objetos JavaScript y cada objeto del array es un artículo de un blog.

```
var articles = [  
  {  
    id      : 1,  
    title   : "Parturient montes nascetur ridiculus mus",  
    content : "<p>Donec quam felis...</p>",  
    author  : "Bill Gates",  
    url     : "http://google.com",  
    image   : "http://images.com/photo.jpg",  
  },  
  {  
    ...  
  }  
];
```



Ejercicio 1 – Blog (cont)

Se pide:

- Mostrar en una página web el listado de artículos respetando el diseño de la próxima slide.
- Descargar el archivo `articles.js` desde [aquí](#) y moverlo a la carpeta `/js` del proyecto.
- Partir del proyecto base (`.zip`) que ya fue compartido hace algunas clases.



Blog de Hack Academy

Sed nisl nunc rhoncus



Nulla consequat massa quis enim. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu.

In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend

tellus.

Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum.

Autor: **Eric Garrett**

[Leer artículo](#)

Id turpis integer aliquet



Integer tincidunt. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum.

In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus.

Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum.

Autor: **Shawn Romero**

[Leer artículo](#)

Sidebar

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5
- Item 6

[Iniciar sesión](#)

Blog de Hack Academy

`.container`

`.row`

`.col-md-8`

Sidebar

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5
- Item 6

Iniciar sesión

`.col-md-4`

Blog de Hack Academy

Título <h2> con link <a>.

Sed nisl nunc rhoncus



Nulla consequat massa quis enim. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu.

In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus.

Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum.

Autor: **Eric Garrett**

Leer artículo

Imágenes de ancho 80px.

Botones con colores customizados.

Id turpis integer aliquet



Integer tincidunt. Aenean felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu.

In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus.

Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum.

Autor: **Shawn Romero**

Leer artículo

Sidebar

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5
- Item 6

Iniciar sesión



Ejercicio 1 – Blog (cont)

A nivel de JavaScript, sólo deberían escribir el archivo `app.js`:

En archivo `articles.js`:

```
var articles = [
  {
    id      : 1,
    title   : "...",
    content : "...",
    author  : "...",
    url     : "https://...",
    image   : "https://...",
  },
  {
    ...
  }
];
```

Este archivo viene dado. No modificarlo.

En archivo `app.js`:

```
var app = new Vue({
  el: "#posts",
  data: {
    articles: articles
  }
});
```

El primer `articles` es un nombre arbitrario para la propiedad (dentro de `data`) que contendrá la lista de artículos.

El segundo `articles` hace referencia a la variable declarada en el archivo `articles.js`

Por un tema de consistencia, se utilizó el mismo nombre.



Ejercicio 2

Editor



Ejercicio 2

Rehacer ejercicio de la Clase 11, usando **Vue.js**. Notar que no lleva botón “Guardar”.

Ejemplo de Vue.js

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Menú de Edición

Título

Párrafo

Tamaño del título (en rem)

Color del título

- ☐ Rojo
☐ Verde
☒ Azul

Para la parte del tamaño y color del título, [usar esto](#).