
JsRender

Sintaxi bàsica de JsRender

JsRender és una biblioteca **JavaScript** que permet generar **HTML** fàcilment utilitzant **plantilles**. És part de la família d'eines **JsViews**, que també inclou **JsViews** per a la vinculació de dades.

La sintaxi bàsica de les plantilles **JsRender** ens permet aconseguir un control més gran sobre com generar contingut dinàmicament:

1. Interpolació de dades:

La interpolació de dades és el procés de **substitució de les** variables o expressions amb els seus valors corresponents a la plantilla. Per aconseguir-ho utilitzem la sintaxi:

```
{{:variable}}
```

Per exemple:

```
<script id="myTemplate" type="text/x-jsrender">
  <div>
    <p>El meu nom és {{:nom}} i tinc {{:edat}} anys.</p>
  </div>
</script>
```

En aquest exemple, **{{:nom}}** i **{{:edat}}** es substituiran pels valors de les propietats **nom** i **edat** dels objectes de dades quan es renderitzi la plantilla.

2. Sentències condicionals:

Podem utilitzar sentències condicionals per controlar quin contingut es renderitza a la plantilla.

Utilitzarem la sintaxi:

```
{{if condició}} ... {{else}} ... {{/if}}.
```

Per exemple:

```
<script id="myTemplate" type="text/x-jsrender">
  {{if edat >= 18}}
    <p>Major d'edat</p>
  {{else}}
    <p>Menor d'edat</p>
  {{/if}}
</script>
```

En aquest cas, si la propietat **edat** és **major o igual a 18**, es renderitzarà "**Major d'edat**", sinó es renderitzarà "**Menor d'edat**".

3. Iteració sobre arrays:

Per renderitzar una llista de dades emmagatzemades en un array, utilitzarem la sintaxi:

```
{{for array}} ... {{/for}}.
```

Per exemple:

```
<script id="myTemplate" type="text/x-jsrender">
  <ul>
    {{for persones}}
      <li>{{:nom}} - {{:edat}}</li>
    {{/for}}
  </ul>
</script>
```

Esta plantilla itera sobre l'array **persones** i renderitza cada element com un element de llista ****.

4. Iteració sobre les propietats d'un objecte:

Per renderitzar les propietats d'un objecte, utilitzem la sintaxi :

```
{{props object}} ... {{/props}}.
```

Per exemple:

```
<script id="myTemplate" type="text/x-jsrender">
  <ul>
    {{props persona}}
      <li>{{:prop}}: {{:~val}}</li>
    {{/props}}
  </ul>
</script>
```

En aquest cas, cada propietat de l'objecte **persona** es renderitzarà com un element de llista **** amb el nom de la propietat i el seu valor.

Estes són algunes de les característiques bàsiques de la sintaxi de les plantilles JsRender. Es poden combinar i utilitzar en conjunt per aconseguir la lògica i la flexibilitat necessàries per generar el contingut HTML dinàmic.

Utilització de JsRender

- **Pas 1: Configuració.** Primer, necessitem incloure JsRender en el nostre projecte. Descarregarem l'arxiu des del lloc web oficial de JsRender o utilitzant un CDN. Per exemple:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jsrender/1.0.13/jsrender.min.js"></script>
```

- **Pas 2: Crear una plantilla JsRender** utilitzant la sintaxi de plantilles per generar HTML. Podem definir una plantilla dins d'una etiqueta `<script>` amb un tipus específic:

```
<script id="plantilla-persona" type="text/x-jsrender">
  <div>
    <p>Nom: {{:nom}}</p>
    <p>Edat: {{:edat}}</p>
  </div>
</script>
```

- **Pas 3: Renderitzar la plantilla.** Una volta tenim creada la plantilla, podem usar JsRender per renderitzar-la amb dades específiques. Per exemple:

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

<script>
  // Dades per a la plantilla
  var dades = {
    nom: "Jaume Aragó",
    edat: 30
  };

  // Renderitzar la plantilla amb les dades
  var plantilla = $.templates("#plantilla-persona");
  var htmlSortida = plantilla.render(dades);

  // Afegir l'HTML generat al DOM
  $("#sortida").html(htmlSortida);
</script>
```

- **Pas 4: Utilitzar bucles i condicionals.** JsRender també ens permet utilitzar bucles i condicionals a les nostres plantilles per manejar dades de manera més dinàmica. Per exemple:

```
<script id="plantilla-persones" type="text/x-jsrender">
  {{for persones}}
    <div>
      <p>Nom: {{:nom}}</p>
      <p>Edat: {{:edat}}</p>
    </div>
  {{/for}}
</script>

<div id="sortida"></div>

<script>
  // Dades per a la plantilla
  var dades = {
    persones: [
      { nom: "Jaume Aragó", edat: 30 },
      { nom: "Verònica Mascarós", edat: 25 },
      { nom: "Andrei Micleusanu", edat: 40 }
    ]
  };

  // Renderitzar la plantilla amb les dades
  var plantilla = $.templates("#plantilla-persones");
  var htmlSortida = plantilla.render(dades);

  // Afegir l'HTML generat al DOM
  $("#sortida").html(htmlSortida);
</script>
```

JsRender és una eina potent per generar HTML dinàmicament en aplicacions web. Amb la seua sintaxi senzilla i la seua integració fluida amb JavaScript, pot facilitar la creació d'interfícies d'usuari interactives i responsives.

JsRender amb JSON extern

- JSON (dades.json)

```
{
  "persona": {
    "nom": "Isabel",
    "edat": 28,
    "casada": true
  },
  "persones": [
    {"nom": "Joan", "edat": 35},
    {"nom": "Carla", "edat": 40},
    {"nom": "Pep", "edat": 30}
  ]
}
```

- HTML:

```
<!DOCTYPE html>
<html lang="ca">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemple JsRender</title>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jsrender/1.0.13/jsrender.min.js"></script>
</head>

<body>

  <div id="persona"></div>
  <hr>
  <div id="persones"></div>

  <script id="plantilla-persona" type="text/x-jsrender">

    <div>
      <p>Nom: {{:nom}}</p>
      <p>Edat: {{:edat}}</p>
      {{if casada}}
        <p>Està casada</p>
      {{else}}
        <p>No està casada</p>
      {{/if}}
    </div>

  </script>

  <script id="plantilla-persones" type="text/x-jsrender">

    {{for persones}}
      <div>
        <p>Nom: {{:nom}}</p>
        <p>Edat: {{:edat}}</p>
      </div>
    {{/for}}

  </script>
```

```
<script>

$(function() {
  // Carregar les dades JSON externes

  $.getJSON("dades.json", function(dades) {
    // Renderitzar la plantilla per a una sola persona
    var plantillaPersona = $.templates("#plantilla-persona");
    var htmlPersona = plantillaPersona.render(dades.persona);

    $("#persona").html(htmlPersona);

    // Renderitzar la plantilla per a l'array de persones

    var plantillaPersones = $.templates("#plantilla-persones");
    var htmlPersones = plantillaPersones.render(dades);

    $("#persones").html(htmlPersones);
  });
});

</script>

</body>
</html>
```

En aquest exemple, es carreguen les dades des d'un fitxer JSON extern (**dades.json**). Després, s'utilitzen dos plantilles diferents (**plantilla-persona** i **plantilla-persones**) per renderitzar les dades.

La primera plantilla mostra les dades d'una sola persona, mentre que la segona itera sobre un array de persones i les mostra una per una.

