

Paginación

Si tiene que mostrar muchos datos en una página, utilice la paginación.

Uso básico

::demo Asigne en el atributo `layout` los diferentes elementos que quiere utilizar separados por coma. Los elementos de paginación son: `prev` (un botón para navegar a la página anterior), `next` (un botón para navegar a la siguiente página), `pager` (lista de página), `jumper` (un `input` para saltar a una página determinada), `total` (total de elementos), `size` (un `select` para seleccionar el tamaño de la página) y `->` (todo elemento situado luego de este símbolo será movido a la derecha).

```
<div class="block">
  <span class="demonstration">When you have few pages</span>
  <el-pagination
    layout="prev, pager, next"
    :total="50">
  </el-pagination>
</div>
<div class="block">
  <span class="demonstration">When you have more than 7 pages</span>
  <el-pagination
    layout="prev, pager, next"
    :total="1000">
  </el-pagination>
</div>
```

...

Números de paginas

::demo De forma predeterminada, Pagination colapsa los botones del paginador adicionales cuando tiene más de 7 páginas. Esto se puede configurar con el atributo `pager-count`.

```
<el-pagination
  :page-size="20"
  :pager-count="11"
  layout="prev, pager, next"
  :total="1000">
</el-pagination>
```

...

Paginación pequeña

Usa una paginación pequeña en caso de espacio limitado.

::demo Solo ponga el atributo `small` como `true` y la Paginación se volverá pequeña.

```
<el-pagination
  small
```

```
    layout="prev, pager, next"
    :total="50">
</el-pagination>
```

...

Más elementos

Agrega más módulos basados en su escenario.

...demo Este ejemplo es un completo caso de uso. Utiliza los eventos `size-change` y `current-change` para manejar el tamaño de página y el cambio de página. El atributo `page-sizes` acepta un arreglo de enteros, cada uno representa un diferente valor del atributo `sizes` que es un `select`, ejemplo `[100, 200, 300, 400]` indicará que el `select` tendrá las opciones: 100, 200, 300 o 400 elementos por página.

```
<template>
  <div class="block">
    <span class="demonstration">Total item count</span>
    <el-pagination
      @size-change="handleSizeChange"
      @current-change="handleCurrentChange"
      :current-page.sync="currentPage1"
      :page-size="100"
      layout="total, prev, pager, next"
      :total="1000">
    </el-pagination>
  </div>
  <div class="block">
    <span class="demonstration">Change page size</span>
    <el-pagination
      @size-change="handleSizeChange"
      @current-change="handleCurrentChange"
      :current-page.sync="currentPage2"
      :page-sizes="[100, 200, 300, 400]"
      :page-size="100"
      layout="sizes, prev, pager, next"
      :total="1000">
    </el-pagination>
  </div>
  <div class="block">
    <span class="demonstration">Jump to</span>
    <el-pagination
      @size-change="handleSizeChange"
      @current-change="handleCurrentChange"
      :current-page.sync="currentPage3"
      :page-size="100"
      layout="prev, pager, next, jumper"
      :total="1000">
    </el-pagination>
  </div>
  <div class="block">
    <span class="demonstration">All combined</span>
```

```

    <el-pagination
      @size-change="handleSizeChange"
      @current-change="handleCurrentChange"
      :current-page.sync="currentPage4"
      :page-sizes="[100, 200, 300, 400]"
      :page-size="100"
      layout="total, sizes, prev, pager, next, jumper"
      :total="400">
    </el-pagination>
  </div>
</template>
<script>
  export default {
    methods: {
      handleSizeChange(val) {
        console.log(`${val} items per page`);
      },
      handleCurrentChange(val) {
        console.log(`current page: ${val}`);
      }
    },
    data() {
      return {
        currentPage1: 5,
        currentPage2: 5,
        currentPage3: 5,
        currentPage4: 4
      };
    }
  }
}
</script>

```

...

Oculte la paginación cuando el resultado es solo una pagina simple

Cuando sólo hay una página, oculte la paginación configurando el atributo `hide-on-single-page`.

...demo

```

<div>
  <el-switch v-model="value">
  </el-switch>
  <el-pagination
    :hide-on-single-page="value"
    :total="5"
    layout="prev, pager, next">
  </el-pagination>
</div>

<script>
  export default {

```

```

data() {
  return {
    value: false
  }
}
}
</script>

```

...

Atributos

Atributo	Descripción	Tipo	Valores aceptados	Por defecto
small	usar paginación pequeña	boolean	—	false
page-size	cantidad de elementos por página, soporta el modificador .sync	number	—	10
total	total de elementos	number	—	—
page-count	total de páginas. Asigna <code>total</code> o <code>page-count</code> y las páginas serán mostradas; si necesitas <code>page-sizes</code> , <code>total</code> es requerido	number	—	—
pager-count	número de paginadores. La paginación colapsa cuando el número total de páginas excede este valor.	number	odd number between 5 and 21	7
current-page	número actual de la página, soporta el modificador .sync	number	—	1
layout	layout de la paginación, elementos separados por coma	string	sizes, prev, pager, next, jumper, ->, total, slot	'prev, pager, next, jumper, ->, total'
page-sizes	opciones para la cantidad de elementos por página	number[]	—	[10, 20, 30, 40, 50, 100]
popper-class	clase propia para el dropdown del select del número de páginas	string	—	—
prev-text	texto para el botón <code>prev</code>	string	—	—
next-text	texto para el botón <code>next</code>	string	—	—
disabled	si Pagination esta disabled	boolean	—	false
hide-	si ocultar cuando sólo hay una página	boolean	—	-

on-single-page				
----------------	--	--	--	--

Eventos

Nombre del evento	Descripción	Parámetros
size-change	se dispara cuando <code>page-size</code> cambia	nuevo valor de <code>page-size</code>
current-change	se dispara cuando <code>current-page</code> cambia	nuevo valor de <code>current-page</code>
prev-click	Se dispara cuando el botón <code>prev</code> recibe el clic y la pagina actual cambia	la nueva pagina actual
next-click	Se dispara cuando el botón <code>next</code> recibe el clic y la pagina actual cambia	la nueva pagina actual

Slot

Nombre	Descripción
—	Elemento propio. Para utilizar esto necesitas declarar <code>slot</code> en el <code>layout</code>