

Layout

Rápido y fácilmente crea un layout básico con 24 columnas.

Layout básico

Crea un layout básico usando columnas.

:::demo Con `row` y `col` , puede fácilmente manipular el layout usando el atributo `span` .

```
<el-row>
  <el-col :span="24"><div class="grid-content bg-purple-dark"></div></el-col>
</el-row>
<el-row>
  <el-col :span="12"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="12"><div class="grid-content bg-purple-light"></div></el-col>
</el-row>
<el-row>
  <el-col :span="8"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="8"><div class="grid-content bg-purple-light"></div></el-col>
  <el-col :span="8"><div class="grid-content bg-purple"></div></el-col>
</el-row>
<el-row>
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple-light"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple-light"></div></el-col>
</el-row>
<el-row>
  <el-col :span="4"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="4"><div class="grid-content bg-purple-light"></div></el-col>
  <el-col :span="4"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="4"><div class="grid-content bg-purple-light"></div></el-col>
  <el-col :span="4"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="4"><div class="grid-content bg-purple-light"></div></el-col>
</el-row>

<style>
  .el-row {
    margin-bottom: 20px;
    &:last-child {
      margin-bottom: 0;
    }
  }
  .el-col {
    border-radius: 4px;
  }
  .bg-purple-dark {
    background: #99a9bf;
  }
  .bg-purple {
    background: #d3dce6;
```

```

    }
    .bg-purple-light {
      background: #e5e9f2;
    }
    .grid-content {
      border-radius: 4px;
      min-height: 36px;
    }
    .row-bg {
      padding: 10px 0;
      background-color: #f9fafc;
    }
  }
</style>

```

⋮

Espaciado de columnas

El espaciado de columnas está soportado.

⋮demo Row provee el atributo `gutter` para especificar el espacio entre columnas y su valor por defecto es 0.

```

<el-row :gutter="20">
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
</el-row>

<style>
  .el-row {
    margin-bottom: 20px;
    &:last-child {
      margin-bottom: 0;
    }
  }
  .el-col {
    border-radius: 4px;
  }
  .bg-purple-dark {
    background: #99a9bf;
  }
  .bg-purple {
    background: #d3dce6;
  }
  .bg-purple-light {
    background: #e5e9f2;
  }
  .grid-content {
    border-radius: 4px;
    min-height: 36px;
  }

```

```
.row-bg {
  padding: 10px 0;
  background-color: #f9fafc;
}
</style>
```

...

Layout híbrido

Crea un complejo layout híbrido combinando el básico de 1/24 columnas.

:::demo

```
<el-row :gutter="20">
  <el-col :span="16"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="8"><div class="grid-content bg-purple"></div></el-col>
</el-row>
<el-row :gutter="20">
  <el-col :span="8"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="8"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="4"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="4"><div class="grid-content bg-purple"></div></el-col>
</el-row>
<el-row :gutter="20">
  <el-col :span="4"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="16"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="4"><div class="grid-content bg-purple"></div></el-col>
</el-row>

<style>
.el-row {
  margin-bottom: 20px;
  &:last-child {
    margin-bottom: 0;
  }
}
.el-col {
  border-radius: 4px;
}
.bg-purple-dark {
  background: #99a9bf;
}
.bg-purple {
  background: #d3dce6;
}
.bg-purple-light {
  background: #e5e9f2;
}
.grid-content {
  border-radius: 4px;
  min-height: 36px;
}
```

```

}
.row-bg {
  padding: 10px 0;
  background-color: #f9fafc;
}
</style>

```

...

Offset de columnas

Puedes especificar offsets para las columnas.

demo Puedes especificar el offset para una columna mediante el atributo `offset` de Col.

```

<el-row :gutter="20">
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="6" :offset="6"><div class="grid-content bg-purple"></div></el-col>
</el-row>
<el-row :gutter="20">
  <el-col :span="6" :offset="6"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="6" :offset="6"><div class="grid-content bg-purple"></div></el-col>
</el-row>
<el-row :gutter="20">
  <el-col :span="12" :offset="6"><div class="grid-content bg-purple"></div></el-col>
</el-row>

<style>
.el-row {
  margin-bottom: 20px;
  &:last-child {
    margin-bottom: 0;
  }
}
.el-col {
  border-radius: 4px;
}
.bg-purple-dark {
  background: #99a9bf;
}
.bg-purple {
  background: #d3dce6;
}
.bg-purple-light {
  background: #e5e9f2;
}
.grid-content {
  border-radius: 4px;
  min-height: 36px;
}
.row-bg {
  padding: 10px 0;
}

```

```
background-color: #f9fafc;
}
</style>
```

...

Alineación

Usa flex layout para un alineamiento flexible de columnas.

:::demo Puede habilitar flex layout asignando el atributo `type` a 'flex', y definir el layout de elementos hijos asignando el atributo `justify` con los valores start, center, end, space-between o space-around.

```
<el-row type="flex" class="row-bg">
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple-light"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
</el-row>
<el-row type="flex" class="row-bg" justify="center">
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple-light"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
</el-row>
<el-row type="flex" class="row-bg" justify="end">
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple-light"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
</el-row>
<el-row type="flex" class="row-bg" justify="space-between">
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple-light"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
</el-row>
<el-row type="flex" class="row-bg" justify="space-around">
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple-light"></div></el-col>
  <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
</el-row>

<style>
  .el-row {
    margin-bottom: 20px;
    &:last-child {
      margin-bottom: 0;
    }
  }
  .el-col {
    border-radius: 4px;
  }
  .bg-purple-dark {
    background: #99a9bf;
  }
}
```

```

.bg-purple {
  background: #d3dce6;
}
.bg-purple-light {
  background: #e5e9f2;
}
.grid-content {
  border-radius: 4px;
  min-height: 36px;
}
.row-bg {
  padding: 10px 0;
  background-color: #f9fafc;
}
</style>

```

...

Responsive Layout

Tomando el ejemplo de Bootstrap responsive design, existen 5 breakpoints: xs, sm, md, lg y xl.

:::demo

```

<el-row :gutter="10">
  <el-col :xs="8" :sm="6" :md="4" :lg="3" :xl="1"><div class="grid-content bg-
purple"></div></el-col>
  <el-col :xs="4" :sm="6" :md="8" :lg="9" :xl="11"><div class="grid-content bg-
purple-light"></div></el-col>
  <el-col :xs="4" :sm="6" :md="8" :lg="9" :xl="11"><div class="grid-content bg-
purple"></div></el-col>
  <el-col :xs="8" :sm="6" :md="4" :lg="3" :xl="1"><div class="grid-content bg-
purple-light"></div></el-col>
</el-row>

<style>
.el-col {
  border-radius: 4px;
}
.bg-purple-dark {
  background: #99a9bf;
}
.bg-purple {
  background: #d3dce6;
}
.bg-purple-light {
  background: #e5e9f2;
}
.grid-content {
  border-radius: 4px;
  min-height: 36px;
}

```

```
}  
</style>
```

...

Clases útiles para ocultar elementos

Adicionalmente, Element provee una serie de clases para ocultar elementos dadas ciertas condiciones. Estas clases pueden se agregadas a cualquier elemento del DOM o un elemento propio. Necesita importar el siguiente archivo CSS para usar estas clases:

```
import 'element-ui/lib/theme-chalk/display.css';
```

Las clases son:

- `hidden-xs-only` - oculto en viewports extra pequeños solamente
- `hidden-sm-only` - oculto en viewports pequeños solamente
- `hidden-sm-and-down` - oculto en viewports pequeños y menores
- `hidden-sm-and-up` - oculto en viewports pequeños y superiores
- `hidden-md-only` - oculto en viewports medios solamente
- `hidden-md-and-down` - oculto en viewports medios y menores
- `hidden-md-and-up` - oculto en viewports medios y mayores
- `hidden-lg-only` - ocultos en viewports grandes solamente
- `hidden-lg-and-down` - ocultos en viewports grandes y menores
- `hidden-lg-and-up` - ocultos en viewports grandes y superiores
- `hidden-xl-only` - oculto en viewports extra grandes solamente

Atributos Row

Atributos	Descripción	Tipo	Valores aceptados	Valor por defecto
gutter	espaciado de la grilla	number	—	0
type	modo del layout , puedes usar flex, funciona en navegadores modernos	string	—	—
justify	alineación horizontal del layout flex	string	start/end/center/space-around/space-between	start
align	alineación vertical del layout flex	string	top/middle/bottom	—
tag	tag de elemento propio	string	*	div

Atributos Col

Atributos	Descripción	Tipo	Valores aceptados	Valor por defecto
span	número de columnas que abarca la cuadrícula	number	—	24

offset	especifica el espacio en el lado izquierdo del grill	number	—	0
push	número de columnas que la grilla se mueve hacia la derecha	number	—	0
pull	número de columnas que la grilla se mueve hacia la izquierda	number	—	0
xs	<768px Columnas responsive u objeto con propiedades de la columna	number/object (e.g. {span: 4, offset: 4})	—	—
sm	≥768px Columnas responsive u objeto con propiedades de la columna	number/object (e.g. {span: 4, offset: 4})	—	—
md	≥992px Columnas responsive u objeto con propiedades de la columna	number/object (e.g. {span: 4, offset: 4})	—	—
lg	≥1200px Columnas responsive u objeto con propiedades de la columna	number/object (e.g. {span: 4, offset: 4})	—	—
xl	≥1920px Columnas responsive u objeto con propiedades de la columna	number/object (e.g. {span: 4, offset: 4})	—	—
tag	tag de elemento propio	string	*	div