

Layout

Quickly and easily create layouts with the basic 24-column.

Basic layout

Create basic grid layout using columns.

demo With `row` and `col`, we can easily manipulate the layout using the `span` attribute.

```
<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>

```

...

Column spacing

Column spacing is supported.

demo Row provides `gutter` attribute to specify spacings between columns, and its default value is 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>
```

...

Hybrid layout

Form a more complex hybrid layout by combining the basic 1/24 columns.

:::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>

```

...

Column offset

You can specify column offsets.

demo You can specify the number of column offset by setting the value of `offset` attribute of 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>

```

...

Alignment

Use the flex layout to make flexible alignment of columns.

demo You can enable flex layout by setting `type` attribute to 'flex', and define the layout of child elements by setting `justify` attribute with start, center, end, space-between or 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

Taking example by Bootstrap's responsive design, five breakpoints are preset: xs, sm, md, lg and 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>
```

...

Utility classes for hiding elements

Additionally, Element provides a series of classes for hiding elements under certain conditions. These classes can be added to any DOM elements or custom components. You need to import the following CSS file to use these classes:

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

The classes are:

- `hidden-xs-only` - hide when on extra small viewports only
- `hidden-sm-only` - hide when on small viewports and down
- `hidden-sm-and-down` - hide when on small viewports and down
- `hidden-sm-and-up` - hide when on small viewports and up
- `hidden-md-only` - hide when on medium viewports only
- `hidden-md-and-down` - hide when on medium viewports and down
- `hidden-md-and-up` - hide when on medium viewports and up
- `hidden-lg-only` - hide when on large viewports only
- `hidden-lg-and-down` - hide when on large viewports and down
- `hidden-lg-and-up` - hide when on large viewports and up
- `hidden-xl-only` - hide when on extra large viewports only

Row Attributes

Attribute	Description	Type	Accepted Values	Default
gutter	grid spacing	number	—	0
type	layout mode, you can use flex, works in modern browsers	string	—	—
justify	horizontal alignment of flex layout	string	start/end/center/space-around/space-between	start
align	vertical alignment of flex layout	string	top/middle/bottom	—
tag	custom element tag	string	*	div

Col Attributes

Attribute	Description	Type	Accepted Values	Default
span	number of column the grid spans	number	—	24
offset	number of spacing on the left side of the grid	number	—	0
push	number of columns that grid moves	number	—	0

	to the right			
pull	number of columns that grid moves to the left	number	—	0
xs	<768px Responsive columns or column props object	number/object (e.g. {span: 4, offset: 4})	—	—
sm	≥768px Responsive columns or column props object	number/object (e.g. {span: 4, offset: 4})	—	—
md	≥992px Responsive columns or column props object	number/object (e.g. {span: 4, offset: 4})	—	—
lg	≥1200px Responsive columns or column props object	number/object (e.g. {span: 4, offset: 4})	—	—
xl	≥1920px Responsive columns or column props object	number/object (e.g. {span: 4, offset: 4})	—	—
tag	custom element tag	string	*	div