

Card

Conteneur intégrant des informations.

Usage

Le composant Card comprend un titre, un contenu et des opérations.

:::demo Card est composé d'un `header` et d'un `body`. `header` est optionnel et son contenu nécessite l'utilisation d'un slot.

```
<el-card class="box-card">
  <div slot="header" class="clearfix">
    <span>Card name</span>
    <el-button style="float: right; padding: 3px 0" type="text">Bouton</el-button>
  </div>
  <div v-for="o in 4" :key="o" class="text item">
    {{ 'List item ' + o }}
  </div>
</el-card>

<style>
  .text {
    font-size: 14px;
  }

  .item {
    margin-bottom: 18px;
  }

  .clearfix:before,
  .clearfix:after {
    display: table;
    content: "";
  }
  .clearfix:after {
    clear: both
  }

  .box-card {
    width: 480px;
  }
</style>
```

:::

Card simple

Le header peut être omis.

:::demo

```

<el-card class="box-card">
  <div v-for="o in 4" :key="o" class="text item">
    {{ 'List item ' + o }}
  </div>
</el-card>

<style>
  .text {
    font-size: 14px;
  }

  .item {
    padding: 18px 0;
  }

  .box-card {
    width: 480px;
  }
</style>

```

...

Images

Affichez un contenu plus riche grâce à la configuration.

:::demo L'attribut `body-style` définit le style CSS du `body`. Cet exemple utilise aussi `el-col` pour la mise en page.

```

<el-row>
  <el-col :span="8" v-for="(o, index) in 2" :key="o" :offset="index > 0 ? 2 : 0">
    <el-card :body-style="{ padding: '0px' }">
      
      <div style="padding: 14px;">
        <span>Yummy hamburger</span>
        <div class="bottom clearfix">
          <time class="time">{{ currentDate }}</time>
          <el-button type="text" class="button">Operating</el-button>
        </div>
      </div>
    </el-card>
  </el-col>
</el-row>

<style>
  .time {
    font-size: 13px;
    color: #999;
  }

```

```

.bottom {
  margin-top: 13px;
  line-height: 12px;
}

.button {
  padding: 0;
  float: right;
}

.image {
  width: 100%;
  display: block;
}

.clearfix:before,
.clearfix:after {
  display: table;
  content: "";
}

.clearfix:after {
  clear: both
}
</style>

<script>
export default {
  data() {
    return {
      currentDate: new Date()
    };
  }
}
</script>

```

...

Ombres

Vous pouvez définir quand l'ombre des Cards doivent apparaître.

::demo L'attribut `shadow` détermine quand l'ombre doit apparaître. Les valeurs possibles sont `always`, `hover` ou `never`.

```

<el-row :gutter="12">
  <el-col :span="8">
    <el-card shadow="always">
      Always
    </el-card>
  </el-col>
  <el-col :span="8">

```

```
<el-card shadow="hover">
  Hover
</el-card>
</el-col>
<el-col :span="8">
  <el-card shadow="never">
    Never
  </el-card>
</el-col>
</el-row>
```

...

Attributs

Attribut	Description	Type	Valeurs acceptées	Défaut
header	Titre de la Card. Accepte aussi un template DOM passé via <code>slot#header</code> .	string	—	—
body-style	Style CSS du body.	object	—	{ padding: '20px' }
shadow	Quand l'ombre doit apparaître	string	always / hover / never	always