

伪类和伪元素

The next set of selectors we will look at are referred to as **pseudo-classes** and **pseudo-elements**. There are a large number of these, and they often serve quite specific purposes. Once you know how to use them, you can look at the list to see if there is something which works for the task you are trying to achieve. Once again the relevant MDN page for each selector is helpful in explaining browser support.

Prerequisites:	Basic computer literacy, basic software installed , basic knowledge of working with files , HTML basics (study Introduction to HTML), and an idea of how CSS works (study CSS first steps).
Objective:	To learn about the pseudo-class and pseudo-element selectors.

What is a pseudo-class?

A pseudo-class is a selector that selects elements that are in a specific state, e.g. they are the first element of their type, or they are being hovered over by the mouse pointer. They tend to act as if you had applied a class to some part of your document, often helping you cut down on excess classes in your markup, and giving you more flexible, maintainable code.

Pseudo-classes are keywords that start with a colon. For example, `:hover` is a pseudo-class.

Simple pseudo-class example

Let's look at a simple example. If we wanted to make the first paragraph in an article larger and bold, we could add a class to that

paragraph and then add CSS to that class, as shown in the first example below:

**Veggies es bonus vobis,
proinde vos postulo essum
magis kohlrabi welsh onion
daikon amaranth tatsoi
tomatillo melon azuki bean
garlic.**

Gumbo beet greens corn soko endive
gumbo gourd. Parsley shallot courgette
tatsoi pea sprouts fava bean collard
greens dandelion okra wakame tomato.
Dandelion cucumber earthnut pea
peanut soko zucchini.

Interactive editor

```
.first {  
  font-size: 120%;  
  font-weight: bold;  
}
```

```
<article>  
  <p class="first">Veggies es  
    bonus vobis, proinde vos postulo  
    essum magis kohlrabi welsh onion  
    daikon amaranth tatsoi tomatillo  
      melon azuki bean garlic.  
</p>  
  
  <p>Gumbo beet greens corn soko
```

However, this could be annoying to maintain — what if a new paragraph got added to the top of the document? We'd need to move the class over to the new paragraph. Instead of adding the class, we

could use the `:first-child` pseudo-class selector — this will *always* target the first child element in the article, and we will no longer need to edit the HTML (this may not always be possible anyway, maybe due to it being generated by a CMS).

**Veggies es bonus vobis,
proinde vos postulo essum
magis kohlrabi welsh onion
daikon amaranth tatsoi
tomatillo melon azuki bean
garlic.**

Gumbo beet greens corn soko endive
gumbo gourd. Parsley shallot courgette
tatsoi pea sprouts fava bean collard
greens dandelion okra wakame tomato.
Dandelion cucumber earthnut pea
peanut soko zucchini.

Interactive editor

```
article p:first-child {  
  font-size: 120%;  
  font-weight: bold;  
}
```

```
<article>  
  <p>Veggies es bonus vobis,  
  proinde vos postulo essum magis
```



All pseudo-classes behave in this same kind of way. They target some bit of your document that is in a certain state, behaving as if you had added a class into your HTML. Take a look at some other examples on MDN:

- [:last-child](#)
- [:only-child](#)
- [:invalid](#)

注意：编写伪类和元素之前没有任何元素选择器是有效的。在上面的示例中，您可以编写 `:first-child` 规则并将应用于作为元素的第一个子元素的任何 `<article>` 元素，而不仅仅是段落的第一个子元素 — `:first-child` 等同于 `*:first-child`。然而，通常你想要更多的控制，所以你需要更具体。



利用 Equinix Metal
的自助服务特性和
Equinix Fabric 中内
置的连接

Mozilla 广告

不想看广告？

一些伪类仅在用户以某种方式与文档交互时适用。这些**用户操作伪类**，有时称为**动态伪类**，当用户与之交互时，就好像一个类已添加到元素中一样。例子包括：

- [:hover](#) - 上文提到的; 这仅适用于用户将指针移到元素（通常是链接）上的情况。
- [:focus](#) — 仅当用户通过单击或使用键盘控件聚焦元素时才适用。

Hover over me

Interactive editor

```
a:link,
a:visited {
  color: rebeccapurple;
  font-weight: bold;
}
```

```
a:hover {
  color: hotpink;
}
```

```
<p><a href="">Hover over me</a></p>
```

Reset

什么是伪元素？

伪元素的行为方式类似。但是，它们的作用就好像您在标记中添加了一个全新的 HTML 元素，而不是将类应用于现有元素。

伪元素以双冒号开头 `::`。`::before` 是伪元素的一个例子。

注意：一些早期的伪元素使用单冒号语法，因此您有时可能会在代码或示例中看到这种情况。现代浏览器支持使用单冒号或双冒号语法的早期伪元素，以实现向后兼容。

例如，如果您想选择段落的第一行，您可以将其包装在一个 `` 元素中并使用元素选择器；但是，如果您包裹的字数比父元素的宽度长或短，那将失败。由于我们往往不知道一行中可以容纳多少个单词——因为如果屏幕宽

度或字体大小发生变化，它就会发生变化——不可能通过添加 HTML 来稳健地做到这一点。

伪元素选择器 `::first-line` 会可靠地为你做这件事——如果单词的数量增加或减少，它仍然只会选择第一行。

Veggies es bonus vobis, proinde vos

postulo essum magis kohlrabi welsh onion daikon
amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko endive

gumbo gourd. Parsley shallot courgette tatsoi pea
sprouts fava bean collard greens dandelion okra
wakame tomato. Dandelion cucumber earthnut pea
peanut soko zucchini.

Interactive editor

```
article p::first-line {  
  font-size: 120%;  
  font-weight: bold;  
}
```

```
<article>  
  <p>Veggies es bonus vobis, proinde vos  
postulo essum magis kohlrabi welsh onion daikon  
amaranth tatsoi tomatillo  
melon azuki bean garlic.</p>  
  
  <p>Gumbo beet greens corn soko endive gumbo  
gourd. Parsley shallot courgette tatsoi pea  
sprouts fava bean collard  
greens dandelion okra wakame tomato.
```

Reset

它的作用就好像 a `` 被神奇地缠绕在第一条格式化的行上，并在每次行长度改变时更新。

您可以看到这选择了两个段落的第一行。

结合伪类和伪元素

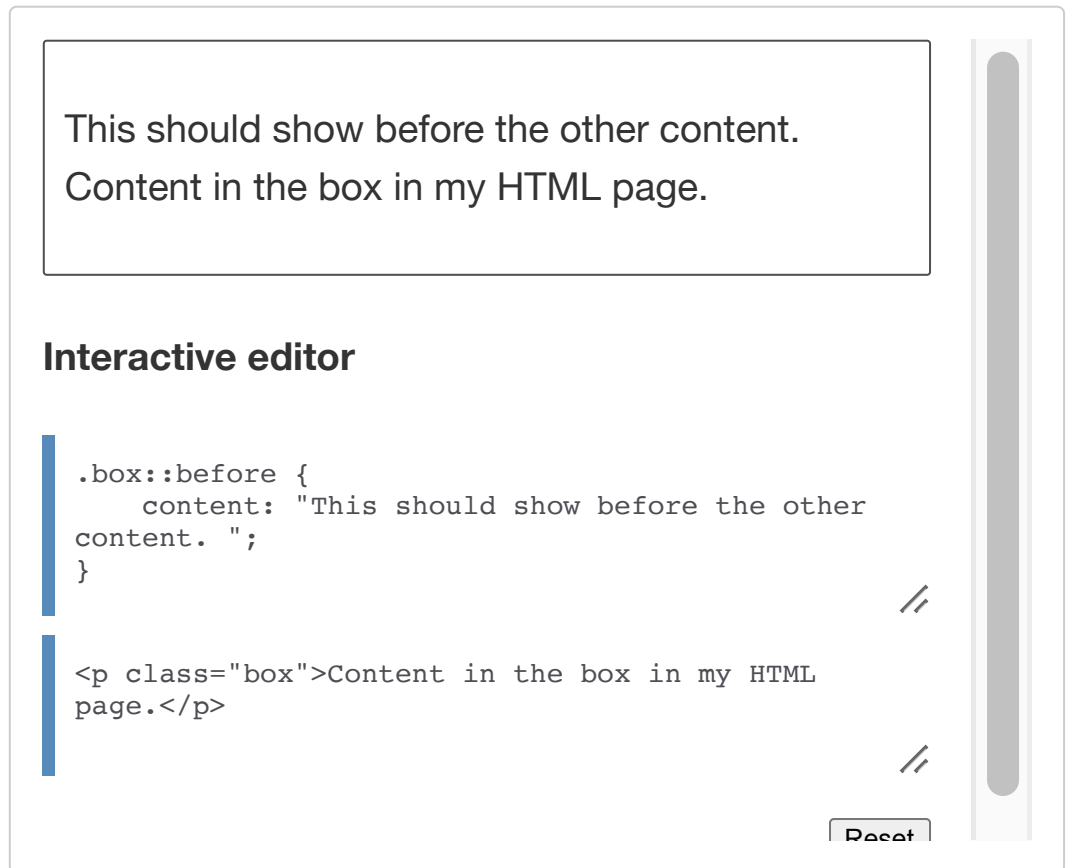
如果您想将第一段的第一行设为粗体，您可以将 `:first-child` 和 `::first-line` 选择器链接在一起。尝试编辑之前的实例，使其使用以下 CSS。 `<p>` 我们说的是我们要选择元素内部的第一个元素的第一行 `<article>`。

```
article p:first-child::first-line {  
  font-size: 120%;  
  font-weight: bold;  
}
```

使用 `::before` 和 `::after` 生成内容

有几个特殊的伪元素，它们与属性一起使用 [content](#)，以使用 CSS 将内容插入到您的文档中。

您可以使用它们来插入文本字符串，例如下面的实时示例。尝试更改属性的文本值 [content](#) 并在输出中查看它的变化。您还可以将 `::before` 伪元素更改为 `::after` 并查看在元素末尾而不是开头插入的文本。



然而，从 CSS 插入文本字符串并不是我们在网络上经常做的事情，因为某些屏幕阅读器无法访问该文本，并且将来可能很难让其他人找到和编辑。

这些伪元素更有效的用法是插入一个图标，例如下面示例中添加的小箭头，这是一个我们不希望屏幕阅读器读出的视觉指示器：

Content in the box in my HTML page.



Interactive editor

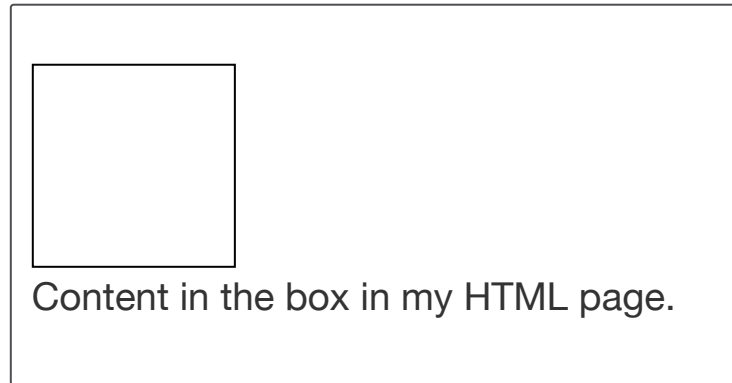
```
.box::after {  
  content: " ➡";  
}
```

```
<p class="box">Content in the box in my  
HTML page.</p>
```

Reset

这些伪元素还经常用于插入一个空字符串，然后可以像页面上的任何元素一样设置样式。

在下一个示例中，我们使用 `::before` 伪元素添加了一个空字符串。我们将其设置为 `display: block` 以便我们可以使用宽度和高度来设置样式。然后我们使用 CSS 像任何元素一样设置它的样式。您可以尝试使用 CSS 并更改它的外观和行为方式。



Interactive editor

```
.box::before {  
  content: "";  
  display: block;  
  width: 100px;  
  height: 100px;  
  background-color: rebeccapurple;  
  border: 1px solid black;  
}
```

```
<p class="box">Content in the box in my  
HTML page.</p>
```

`::before` 和 `::after` 伪元素以及属性的使用 `content` 在 CSS 中称为“生成的内容”，您会经常看到这种技术被用于各种任务。一个很好的例子是网站 [CSS Arrow Please](#)，它可以帮助您使用 CSS 生成箭头。在创建箭头时查看 CSS，您将看到正在使用的 `::before` 和 `::after` 伪元素。每当您看到这些选择器时，请查看 [content](#) 属性以查看添加到 HTML 元素的内容。

概括

在本文中，我们介绍了 CSS 伪类和伪元素，它们是特殊类型的选择器。

伪类使您能够在元素处于特定状态时将其作为目标，就好像您已经为该状态添加了一个类到 DOM 中一样。伪元素就像您向 DOM 添加了一个全新的元素一样，并使您能够对其进行样式设置。和伪元素使您能够使用 CSS 将内容插入到文档中 `::before`。 `::after`

在下一篇文章中，我们将学习[组合器](#)。

也可以看看

- [伪类参考](#)
- [伪元素参考](#)

此页面最后修改于 2023 年 2 月 23 日由[MDN 贡献者](#)提供。