

CSS 自定义属性

作者: [Rabbitzzc](#)

原文链接: <https://ld246.com/article/1589729872396>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

带有前缀 `--` 的属性名，比如 `--example--nam`，表示的是带有值的自定义属性，其可以通过 `var` 函在全文档范围内复用的。

CSS 自定义属性是可以级联的：每一个自定义属性可以多次出现，并且变量的值将会借助级联算法和定义属性值运算出来。

CSS 自定义属性的使用

有点CSS预处理SCSS的味道

```
<p id="firstParagraph" >
  hacpai
</p>
<p id="secondParagraph" >
  hacpai
</p>
<div id="container" >
  <p id="thirdParagraph" >
    hacpai
  </p>
</div>

:root {
  --first-color: #488cff;
  --second-color: #ffff8c;
}

#firstParagraph {
  background-color: var(--first-color);
  color: var(--second-color);
}

#secondParagraph {
  background-color: var(--second-color);
  color: var(--first-color);
}

#container {
  --first-color: #48ff32;
}

#thirdParagraph {
  background-color: var(--first-color);
  color: var(--second-color);
}
```

splitting.js 仓库的实现

[splitting.js](#) 是一个JavaScript库，其中包含一组小型内置插件，这些插件旨在以多种方式（例如单词字符，子节点等）拆分（分割）元素。概括来说，其实就是将字符串拆解为单个字符，并同时给字符置相应的数据属性以及一些自定义属性，基础的逻辑大概如下：

```
<!DOCTYPE html>
```

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    html {
      height: 100%;
    }
    body {
      margin: 0;
      padding: 0;
      height: 100%;
      display: flex;
      align-items: center;
      justify-content: center;
      flex-wrap: wrap;
    }
    p {
      margin: 10px 0;
      font-size: 30px;
      width: 100%;
      /* text-align: center; */
    }

    .scalpel-char {
      padding-left: 10px;
      transition: all 0.5s;
    }

    .scalpel-char:first-child {
      padding-left: 0;
    }
    .scalpel-word:hover .scalpel-char {
      padding-left: 0;
    }

  </style>

</head>
<body>
  <p id="hac1" data-scalpel>HacPai</p>
  <script>
    // 判断哪些单词设置了数据属性: scalpel (手术刀)
    let hac1 = document.querySelectorAll('[data-scalpel]')
    const words1 = typeof hac1.innerText === 'string' ? hac1.innerText.split('') : []

    const decoratorWords = (words) => {
      return words.reduce((pre, cur, index) => {
        return pre + `<span class="scalpel-char" data-char="${cur}" style="--char-index:${index};">${cur}</span>`
      }, '')
    }
    hac1.innerHTML = `<span class="scalpel-word" data-word="${hac1.innerText}" style="--

```

```
word-index:0;">` + decoratorWords(words1) + "</span>"
  hac1.style['--word-total'] = 1
  hac1.style['--char-total'] = words1.length

</script>
</body>
</html>
```

后续有时间再看看一些具体的细节吧