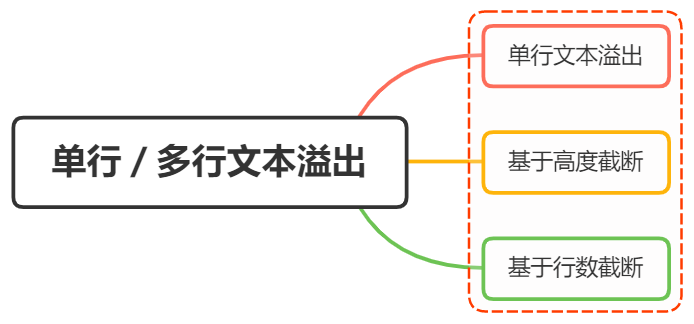
# 面试官：如何实现单行／多行文本溢出的省略样式？



## 一、前言

在日常开发展示页面，如果一段文本的数量过长，受制于元素宽度的因素，有可能不能完全显示，为了提高用户的使用体验，这个时候就需要我们把溢出的文本显示成省略号

对于文本的溢出，我们可以分成两种形式：

* 单行文本溢出
* 多行文本溢出

## 二、实现方式

### 单行文本溢出省略

理解也很简单，即文本在一行内显示，超出部分以省略号的形式展现

实现方式也很简单，涉及的css属性有：

* text-overflow：规定当文本溢出时，显示省略符号来代表被修剪的文本
* white-space：设置文字在一行显示，不能换行
* overflow：文字长度超出限定宽度，则隐藏超出的内容

overflow设为hidden，普通情况用在块级元素的外层隐藏内部溢出元素，或者配合下面两个属性实现文本溢出省略

white-space:nowrap，作用是设置文本不换行，是overflow:hidden和text-overflow：ellipsis生效的基础

text-overflow属性值有如下：

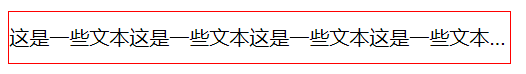
* clip：当对象内文本溢出部分裁切掉
* ellipsis：当对象内文本溢出时显示省略标记（...）

text-overflow只有在设置了overflow:hidden和white-space:nowrap才能够生效的

举个例子

<style>  
 p{  
 overflow: hidden;  
 line-height: 40px;  
 width:400px;  
 height:40px;  
 border:1px solid red;  
 text-overflow: ellipsis;  
 white-space: nowrap;  
 }  
</style>  
<p 这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本</p >

效果如下：



可以看到，设置单行文本溢出较为简单，并且省略号显示的位置较好

### 多行文本溢出省略

多行文本溢出的时候，我们可以分为两种情况：

* 基于高度截断
* 基于行数截断

#### 基于高度截断

#### 伪元素 + 定位

核心的css代码结构如下：

* position: relative：为伪元素绝对定位
* overflow: hidden：文本溢出限定的宽度就隐藏内容）
* position: absolute：给省略号绝对定位
* line-height: 20px：结合元素高度,高度固定的情况下,设定行高, 控制显示行数
* height: 40px：设定当前元素高度
* ::after {} ：设置省略号样式

代码如下所示：

<style>  
 .demo {  
 position: relative;  
 line-height: 20px;  
 height: 40px;  
 overflow: hidden;  
 }  
 .demo::after {  
 content: "...";  
 position: absolute;  
 bottom: 0;  
 right: 0;  
 padding: 0 20px 0 10px;  
 }  
</style>  
  
<body>  
 <div class='demo'>这是一段很长的文本</div>  
</body>

实现原理很好理解，就是通过伪元素绝对定位到行尾并遮住文字，再通过 overflow: hidden 隐藏多余文字

这种实现具有以下优点：

* 兼容性好，对各大主流浏览器有好的支持
* 响应式截断，根据不同宽度做出调整

一般文本存在英文的时候，可以设置word-break: break-all使一个单词能够在换行时进行拆分

#### 基于行数截断

纯css实现也非常简单，核心的css代码如下：

* -webkit-line-clamp: 2：用来限制在一个块元素显示的文本的行数，为了实现该效果，它需要组合其他的WebKit属性）
* display: -webkit-box：和1结合使用，将对象作为弹性伸缩盒子模型显示
* -webkit-box-orient: vertical：和1结合使用 ，设置或检索伸缩盒对象的子元素的排列方式
* overflow: hidden：文本溢出限定的宽度就隐藏内容
* text-overflow: ellipsis：多行文本的情况下，用省略号“…”隐藏溢出范围的文本

<style>  
 p {  
 width: 400px;  
 border-radius: 1px solid red;  
 -webkit-line-clamp: 2;  
 display: -webkit-box;  
 -webkit-box-orient: vertical;  
 overflow: hidden;  
 text-overflow: ellipsis;  
 }  
</style>  
<p>  
 这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本  
 这是一些文本这是一些文本这是一些文本这是一些文本这是一些文本  
</p >

可以看到，上述使用了webkit的CSS属性扩展，所以兼容浏览器范围是PC端的webkit内核的浏览器，由于移动端大多数是使用webkit，所以移动端常用该形式

需要注意的是，如果文本为一段很长的英文或者数字，则需要添加word-wrap: break-word属性

还能通过使用javascript实现配合css，实现代码如下所示：

css结构如下：

p {  
 position: relative;  
 width: 400px;  
 line-height: 20px;  
 overflow: hidden;  
  
}  
.p-after:after{  
 content: "...";   
 position: absolute;   
 bottom: 0;   
 right: 0;   
 padding-left: 40px;  
 background: -webkit-linear-gradient(left, transparent, #fff 55%);  
 background: -moz-linear-gradient(left, transparent, #fff 55%);  
 background: -o-linear-gradient(left, transparent, #fff 55%);  
 background: linear-gradient(to right, transparent, #fff 55%);  
}

javascript代码如下：

$(function(){  
 //获取文本的行高，并获取文本的高度，假设我们规定的行数是五行，那么对超过行数的部分进行限制高度，并加上省略号  
 $('p').each(function(i, obj){  
 var lineHeight = parseInt($(this).css("line-height"));  
 var height = parseInt($(this).height());  
 if((height / lineHeight) >3 ){  
 $(this).addClass("p-after")  
 $(this).css("height","60px");  
 }else{  
 $(this).removeClass("p-after");  
 }  
 });  
})

## 参考文献

* https://www.zoo.team/article/text-overflow
* https://segmentfault.com/a/1190000017078153