一、Foundation前端框架

rem-calc

rem-calc($values, $base)

Converts one or more pixel values into matching rem values.

将一个或多个像素值转换为匹配的rem值。

| **Parameter** | **Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| **$values** | Number or List | None | One or more values to convert. Be sure to separate them with spaces and not commas. If you need to convert a comma-separated list, wrap the list in parentheses.  要转换的一个或多个值。一定要用空格分隔，而不是逗号。如果需要转换逗号分隔的列表，请将列表括在括号中。 |
| **$base** | Number | null | The base value to use when calculating the rem. If you're using Foundation out of the box, this is 16px. If this parameter is null, the function will reference the $global-font-size variable as the base.  计算rem时使用的基值。如果你使用的是开箱即用的基金会，这是16px。如果此参数为NULL，则函数将引用$global-font-size变量作为基数。 |

实际使用：rem-calc(要使用的PX值，不加单位)

例：rem-calc(20)

二、多媒体查询应使用;（尽量不要在JSX结构中直接使用获取屏幕宽度等方法）

@include breakpoint(medium) {

border-right: 1px solid $grey-200;

}

注括号中可以使用small medium large三种基本布局方式；

三、结构中用的通用方式，可以提至Utils中或该页面的方法中，然后，在结构中调用；

四、在JSX结构代码中不要使用处理DOM的方法（如：document.getElementsByName('radio\_location')类似的），将获取DOM的操作写到state中使用；

五、不要在JSX代码中定量性质（如：在className使用三目运算的）不要使用 ` ` (ES6的模板字符串)；如果是纯拼接字符串的则可以使用；

六、在项目中使用ID要使用 “\_”的连接符；

**七、phase-one 版本号 在bitbucket左侧点commit然后选择最上面的commit点，进去后，复制URL就OK了；**

**八、以后改到哪个页面，就把哪个页面的console.log删除；**

# Media Queries媒体查询（foundation）

**JavaScript**

CSS media queries allow us to adjust the display and orientation of content at different screen sizes.

CSS媒体查询允许我们在不同屏幕大小上调整内容的显示和定向。

Build better websites and apps, code cleaner, and become a better front-end developer with Foundation training. We're running two online webinar training sessions this month where we break down how to get the most out of Foundation and leap ahead skillwise.

建立更好的网站和应用程序，代码清洁，并成为一个更好的前端开发与基础培训。本月，我们将举办两次在线研讨会，我们将讨论如何从基础上获得最大的突破，以及如何巧妙地向前迈进。

[Get registered →](http://zurb.com/university/courses)

[获得注册](http://zurb.com/university/courses)

## Default Media Queries默认媒体查询

Foundation for Sites has three core breakpoints: （网站的基础有三个核心断点：）

* **Small：**任何屏幕
* **Medium:**任何屏幕640像素或更宽。
* **Large:**任何屏幕1024像素或更宽。

许多组件可以使用不同的屏幕尺寸进行修改。断点类. 网格是最明显的例子。在下面的代码中，左边的列在小屏幕上是六列宽的，因此.small-6. 在中型屏幕上.medium-4覆盖小样式，将列更改为四宽。

div class="grid-x grid-margin-x">

<div class="cell small-6 medium-4"></div>

<div class="cell small-6 medium-8"></div>

</div>

如果您使用的是基础的CSS版本，使用这些媒体查询来模拟三个核心断点：

*/\* Small only \*/*

*@media screen and (max-width: 39.9375em) {}*

*/\* Medium and up \*/*

*@media screen and (min-width: 40em) {}*

*/\* Medium only \*/*

*@media screen and (min-width: 40em) and (max-width: 63.9375em) {}*

*/\* Large and up \*/*

*@media screen and (min-width: 64em) {}*

*/\* Large only \*/*

*@media screen and (min-width: 64em) and (max-width: 74.9375em) {}*

## Upgrading from Foundation 5（从Foundation 5升级）

在基础5中，使用一系列名为SASS变量的断点进行访问。$small-up，$small-only，$medium-only等等。在Foundation6中，用专用的方法替换了这种媒体查询的写入方法。断点混合如下所述**在 Foundation6.3中将删除遗留变量。**

要升级现有的媒体查询，请将规则集替换为：

@media #{$medium-only} {}

用这个：

@include breakpoint(medium only) {}

## Changing the Breakpoints（更改断点）

如果您使用的是基础的Sass版本，则可以更改默认断点。断点的名称及其宽度存储在设置文件中的$breakpoints变量中。

$breakpoints: ( small: 0px, medium: 640px, large: 1024px, xlarge: 1200px, xxlarge: 1440px,);

尽管上面的值是以像素为单位的，但它们最终会转换为EMS，以便在媒体查询中使用。

Changing the widths of any of the breakpoints is as easy as changing the pixel values in this map. Note that here there are two extra breakpoints:xlargeandxxlarge. We don't use these for any components, and also don't output any CSS classes that use them by default.

改变任何断点的宽度与改变该映射中的像素值一样容易。注意这里有两个额外的断点：xlarge和xxlarge. 对于任何组件，我们不使用这些组件，也不输出默认使用它们的任何CSS类。

You can change that by modifying the$breakpoint-classesvariable in your settings file. This is a list of breakpoint names. Adding or removing names from the list will change the CSS class output. It looks like this by default:

您可以通过修改$breakpoint-classes设置文件中的变量。这是一个断点名称列表。从列表中添加或删除名称将更改CSS类输出。默认情况下是这样的：

$breakpoint-classes: (small medium large);

For example, to get.xlargeclasses in your CSS, for use in the grid, Menu, and more, just add it to the end of the list:

例如，得到.xlarge CSS中的类，用于网格、菜单等，只需将其添加到列表的末尾：

$breakpoint-classes: (small medium large xlarge);

## Sass

### The Breakpoint Mixin（断点混合）

我们的breakpoint()混合使得编写媒体查询变得很容易。可以使用命名的断点，也可以使用自定义像素、rem或em值。

要使用混合，请用@include调用它，然后在花括号内包含您想要的CSS内容。

.element {

// Only affects medium screens and larger

@include breakpoint(medium) {

// All CSS in here goes inside the media query

}

}

媒体查询的行为可以通过在断点值之后添加关键字down 和 only来更改，用空格分隔。

.element {

// Only affects medium screens and smaller

@include breakpoint(medium down) { }

// Only affects medium screens, not small or large

@include breakpoint(medium only) { }

}

也可以传入自定义值。可以输入像素、rem或em值-所有值都会在末尾转换为em。

.element {

// Converted to 20em

@include breakpoint(320px) { }

// Unitless values are assumed to be pixels

@include breakpoint(320) { }

// Converted to 40em

@include breakpoint(40rem) { }

}

最后，有三个不是基于宽度的特殊媒体查询：portrait，landscape和retina。将这些关键字与breakpoint()混合使用将输出设备方向或像素密度的媒体查询，而不是屏幕宽度。

.element {

@include breakpoint(landscape) {

// CSS for landscape-oriented devices only

}

@include breakpoint(retina) {

// CSS for high-resolution displays only

}

}

### Breakpoint Function（断点函数）

breakpoint()混合的功能来自内部函数，也称为breakpoint()。如果要编写自己的媒体查询，可以使用breakpoint()。函数直接访问混合的逻辑。

@media screen and #{breakpoint(medium)} {

// Medium and up styles

}

这可以用于将多个媒体查询组合在一起。

@media screen and #{breakpoint(medium)} and #{breakpoint(xlarge down)} {

// Medium to extra large styles

}

## JavaScript

### 使用媒体查询

Foundation JavaScript包括一组用于处理媒体查询的帮助函数。它们都在Foundation.MediaQueryobject上。

MediaQuery实用程序使用Sass断点设置，并需要导入Foundation CSS。对于SASS用户，您需要包括foundation-everything()or foundation-global-styles().

获取当前断点的名称MediaQuery.current

Foundation.MediaQuery.current // => 'small', 'medium', etc.

若要查看屏幕当前是否为某个断点或更大的断点，请使用MediaQuery.atLeast.

if (Foundation.MediaQuery.atLeast('medium')) {

// True if medium or large

// False if small

}

若要查看屏幕当前是否为某个断点，请使用MediaQuery.is.

if (Foundation.MediaQuery.is('small only')) {

// True if small

// False if medium or large

}

若要获取断点的媒体查询，请使用MediaQuery.get.

Foundation.MediaQuery.get('medium') // => only screen and (min-width: 640px)

### 监视断点变化

每次断点更改时，媒体查询助手都会在窗口上广播一个事件。我们在内部将其与Interchange等插件一起使用，以检测断点的变化。您也可以自己订阅活动。

$(window).on('changed.zf.mediaquery', function(event, newSize, oldSize) {

// newSize is the name of the now-current breakpoint, oldSize is the previous breakpoint

});

## SASS参考文献

### 变量

可以使用项目中的这些Sass变量来自定义此组件的默认样式。[settings file](https://foundation.zurb.com/sites/docs/sass.html#the-settings-file).

| **姓名** | **类型** | **默认值** | **描述** |
| --- | --- | --- | --- |
| **$breakpoints** | Map | small：0 medium: 640px large：1024px xlarge：1200PX xxlarge: 1440PX | 命名断点的列表。您可以将这些与breakpoint()混合使用，以快速创建媒体查询。 |
| **$print-breakpoint** | Keyword | large | 要将breakpoint作为媒体类型包含在其中的最大命名断点 |
| **$breakpoint-classes** | List | small  medium  large | 此列表中的所有名称都将作为CSS中的类输出，如.small-12、.Medium-6等。此列表中的每个值也必须位于$Breakpoints map中。 |

### Mixins混合

我们使用这些mixins来构建这个组件的最终CSS输出。您可以自己使用混合来构建您自己的类结构，而不是我们的组件。。

#### breakpoint

@includebreakpoint($value) { }

将媒体查询包装在您放入混合中的内容周围。此混合接受多个值：

* 如果传递了一个字符串，Mixin将在$breakpoints map中查找它，并在那里使用媒体查询。
* 如果传递了像素值，则将使用$global-font-size值作为基础将其转换为em值。
* 如果传递rem值，则单位将更改为em。
* 如果传递em值，则该值将按原样使用。

| **Parameter(参数)** | **Type(类型)** | **Default Value(默认值)** | **Description(描述)** |
| --- | --- | --- | --- |
|  | | | |
| **$value** | Keyword or Number | None | Breakpoint name, or px, rem, or em value to process.  要处理的断点名称或px、rem或em值。 |

### Functions

#### Breakpoint(断点)

breakpoint($val)

生成与输入值匹配的媒体查询字符串。请参阅breakpoint()混合的文档，以了解可能的输入是什么。

| **参数** | **类型** | **默认值** | **描述** |
| --- | --- | --- | --- |
| **$val** | Keyword or Number  关键字或数字 | small | Breakpoint name, or px, rem, or em value to process.  要处理的断点名称或px、rem或em值。 |

## JavaScript Reference

### 初始化

下列文件必须包含在JavaScript中以使用此插件：

* foundation.core.js

### 方法

#### atLeast

$('#element').foundation('atLeast', size);

检查屏幕是否至少与断点一样宽。

| **Name** | **类型** | **描述** |
| --- | --- | --- |
| **size** | String | Name of the breakpoint to check.  要检查的断点名称 |

#### is

$('#element').foundation('is', size);

检查屏幕是否与断点匹配。

| **Name** | **类型** | **描述** |
| --- | --- | --- |
| **size** | String | Name of the breakpoint to check, either 'small only' or 'small'. Omitting 'only' falls back to using atLeast() method.  要检查的断点的名称，可以是**“small only”**或**“small”**。省略**“only”**回到使用**atLeast()**方法。 |

#### get

$('#element').foundation('get', size);

获取断点的媒体查询

| **姓名** | **类型** | **描述** |
| --- | --- | --- |
| **size** | String | Name of the breakpoint to get.  要获取断点的名称 |