

CSS font-feature-settings 50+关键字属性值完整介绍

这篇文章发布于 2018年12月9日, 星期日, 16:35, 归类于 [CSS相关](#)。阅读 2107 次, 今日 8 次 [3 条评论](#)

by zhangxinxu from <https://www.zhangxinxu.com/wordpress/?p=8238>

本文可全文转载, 个人网站无需授权, 只要保留原作者、出处以及文中链接即可, 任何网站均可摘要聚合, 商用请联系授权。

一、过去介绍的几个font相关CSS属性

之前有介绍过几个font相关的CSS属性, `font-kerning`, `font-smoothing` 以及 `font-variant`, 文章链接为: “[了解CSS属性font-kerning,font-smoothing,font-variant](#)”。

本文要介绍的则是上面链接文章最后提到的 `font-feature-settings` 属性, CSS `font-feature-settings` 语法并不复杂, 只是支持的关键字属性值何其众多, 需要长篇大论介绍, 本文会尽量简化, 但篇幅依然会很长, 提前预警。

在座的诸位没必要每一个属性值都熟知, 了解 `font-feature-settings` 可以做些什么就可以了, 有个印象, 以后遇到类似场景, 记得回到这里找就可以了。

二、font-feature-settings有什么用?

CSS `font-feature-settings` 属性控制OpenType字体中的高级排版功能, 也就是可以让OpenType字体排版更加精致, 符合实际开发需要。

兼容性很不错。

CSS font-feature-settings - CR							
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Android *
		2-3.6	4-15	3.1-3.2			
		4-14	16-20	4-6	10-12.1	3.2-6.1	
6-9		15-33	21-47	6.1-9	15-34	7-9.2	2.1-4.3
10	12-17	34-62	48-69	9.1-11.1	35-55	9.3-11.4	4.4-4.4.4
11	18	63	70	12	56	12	67
		64-65	71-73	TP			

`font-feature-settings` 要想有作用, 往往需要字体本身支持, 因此, 多自定义字体中。

语法:

```
font-feature-settings: normal | <feature-tag-value>;
```

其中:

```
<feature-tag-value> = <string> [ <integer> | on | off ]?
```

例如:

```
/* 使用默认值 */  
font-feature-settings: normal;  
  
/* 使用 OpenType 特征标签 */  
font-feature-settings: "smcp";  
font-feature-settings: "smcp" on;  
font-feature-settings: "swsh" 2;  
font-feature-settings: "smcp", "swsh" 2;
```

而本文要详细介绍的，就是上面 `"smcp"` 和 `"swsh"` 这类OpenType字体特征标签。

OpenType字体特征标签采用注册制，目前大多数的标签都是Adobe注册，也有部分出自Microsoft，比方说‘cv01’ – ‘cv99’。

三、半百的关键字索引

OpenType字体特征标签非常多，可能上百个有，我这里展示的有50+，索引列表如下：

连字与变化-Ligatures & Alternate

- liga – Common/standard ligatures – 通用/标准连字
- calt – Contextual alternates – 上下文交替
- dlig – Discretionary ligatures – 酌情连字
- smcp – Small caps – 小写字母转为小体型大写字母
- c2sc – Capitals to small caps – 大写字母转为小体型大写字母
- aalt – Access All Alternates – 所有替代形式
- swsh – Swashes – 此功能将默认字符字形替换为相应的斜线字形。
- cswsh – Contextual Swash – 语境冲突
- hist – Historical Forms – 历史形式
- hlig – Historical Ligatures – 历史连字
- locl – Localized Forms – 本地化形式
- rand – Randomize – 随机性
- cv01-cv99
- salt – Stylistic alternates – 文体变化替换默认字符字形
- sups – Superscript – 上标
- subs – Subscript – 下标
- titl – Titling Alternates – 标题替换
- size – Optical size – 可见尺寸
- ornm – Ornaments – 装饰

位置与距离-Positions

- kern – Kerning – 字距调整
- ccmp – Glyph Composition/Decomposition – 字形组成/分解

数字-Numbers

- lnum – Lining figures – 衬砌数字
- onum – Oldstyle figures – 旧样式数字
- pnum – Proportional figures – 比例数字
- tnum – Tabular figures – 表格数字

- `ordn` – Ordinals – 序数字
- `frac` – Fractions – 分数
- `afrc` – Alternative Fractions – 变成分数形式
- `dnom` – Denominator form – 分母形式
- `numr` – Numerator form – 分子形式
- `zero` – Slashed Zero – 斜杠零
- `mgrk` – Mathematical Greek – 数学中的希腊字符

文体集-**Stylistic sets**

- `ss01-ss20` – Stylistic sets 01–20 – 文体集01-20

东亚文字-**East Asian script**

宽度变化-**Width variants**

- `pwid` – Proportional Widths – 比例宽度
- `palt` – Proportional Alternate Widths – 改变宽度占据比例
- `pkna` – Proportional Kana – 比例假名
- `fwid` – Full Widths – 全宽
- `hwid` – Half Widths – 半宽
- `halt` – Alternate Half Widths – 变为半宽
- `twid` – Third Widths – 三分之一宽
- `qwid` – Quarter Widths – 四分之一宽

文化变体-**Cultural variants**

- `jp78` – JIS78 Forms – JIS78形式
- `jp83` – JIS83 Forms – JIS83形式
- `jp90` – JIS90 Forms – JIS90形式
- `jp04` – JIS2004 Forms – JIS2004形式
- `trad` – Traditional Forms – 传统形式
- `expt` – Expert Forms – 专业形式
- `hojo` – Hojo Kanji Forms – Hojo汉字形式
- `nlck` – NLC Kanji Forms – NLC汉字形式
- `ruby` – Ruby Notation Forms – Ruby表示形式
- `hkna` – Horizontal Kana Alternates – 水平假名替代
- `nalt` – Alternate Annotation Forms – 备用注释形式
- `ital` – Italics – 斜体

垂直特征-**Vertical features**

- `vkern` – Vertical Kerning – 垂直字距
- `vert` – Vertical Alternates – 垂直交替
- `vpal` – Proportional Alternate Vertical Metrics – 比例替代垂直度量标准
- `vhal` – Alternate Vertical Half Metrics – 替代垂直半度量
- `vkna` – Vertical Kana Alternates – 垂直文本的替代日语假名形式

其他-**others**

- `others` – Others Feature tags – 其他功能标签

以上OpenType字体特征标签我不会全部介绍，例如一些日文相关的设定我们用不到，只会一带而过，你若有兴趣，可以去文末的参考资料继续深入。

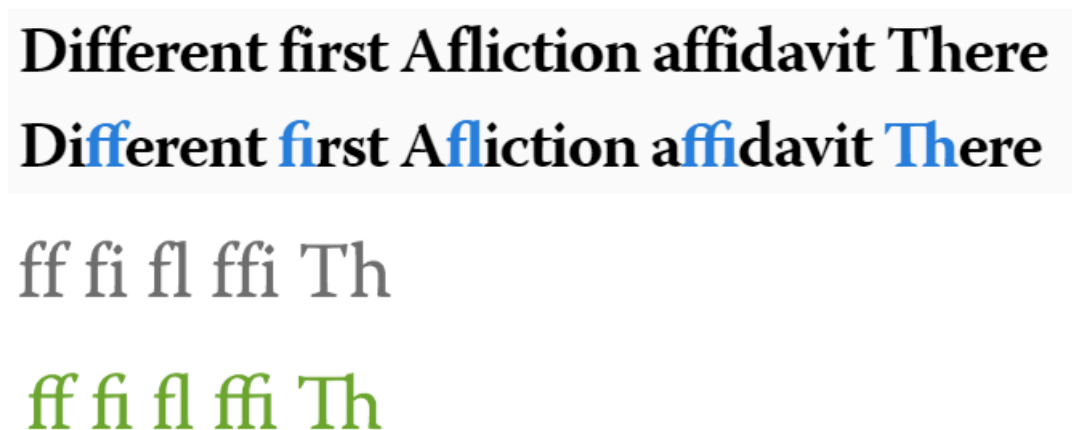
四、关键字详细语法和案例

☀ **Standard Ligatures: liga**

主要针对英文书写，可以让多个单独的字符表现为连续书写，俗称连字。

可以表现此特征的字体： Adobe Warnock系列字体

下面是两个不同出处的效果对比：



使用CSS代码如下：

```
.liga { font-feature-settings: "liga", "clig"; }
```

和 `"clig"` 一同使用了。

此标签设计为默认可用，也就是无需设置CSS就连字效果。但是实际Chrome下默认却是禁用的，而Safari下却不能被禁用。

☀ **Contextual Alternates: calt**

包含上下交替的连字行为。看下面图就知道什么意思了：



可以表现此特征的字体： Adobe caflisch-script系列字体

使用示意：

```
.calt {  
  font-feature-settings: "calt";  
  font-variant-ligatures: common-ligatures;  
}
```

此标签设计为默认可用，也就是无需设置CSS就连字效果。但是实际Chrome下默认却是禁用的，而Safari下却不能被禁用。

☀ **Discretionary Ligatures: dlig**

顾名思义，酌情连字，不过嘛，看效果，不是酌情，而是强行，哈哈哈哈……（这里有我的笑声……）

。

可以表现此特征的字体：  Adobe Warnock系列字体

此标签设计为默认不可用，需要CSS开启：

```
.dlig { font-feature-settings: "dlig"; }
```

☀ Small Caps: smcp

此功能可将小写字母转换为小型大写字母。

可以表现此特征的字体：  Adobe Warnock系列字体以及  苹果San Francisco字体。

此标签设计为默认不可用，需要CSS开启：

```
.smcp {  
  font-variant-caps: small-caps;  
  font-feature-settings: "smcp";  
}
```

☀ Capitals to Small Caps: c2sc

此功能可以将大写字母转换为小型大写字母。



可以表现此特征的字体：  Adobe Warnock系列字体以及  苹果San Francisco字体

此标签设计为默认不可用，需要CSS开启：

```
.c2sc {  
  font-variant-caps: all-small-caps;  
  font-feature-settings: "c2sc", "smcp";  
}
```

☀ Access All Alternates: aalt

这是个比较特别的特征，所有字体中包含的可替换的特征，全部都向用户呈现。

此效果在所有支持ligation和alternate形式的字体上都可以呈现，例如  Google的Sorts Mill Goudy和  Google的Yanone Kaffeesatz等。

```
.aalt { font-feature-settings: "aalt"; }
```

☀ Swashes: swsh

此功能将默认字符字形替换为相应的斜线字形。

可以表现此特征的字体:  Adobe bickham-script系列字体

使用CSS示意:

```
.swsh { font-feature-settings: "swsh"; }
```

☀ Contextual Swash: cswh

根据字母周围的字符将字母转换为虚线版本。

可以表现此特征的字体:  Adobe Bello系列字体

使用CSS示意:

```
.cswh { font-feature-settings: "cswh"; }
```

☀ Historical Forms: hist

由用户自行决定是否应用的过时形式的字符。

可以表现此特征的字体:  Google的Sorts Mill Goudy字体和 Google的Cardo字体等。

使用CSS示意:

```
.hist { font-feature-settings: "hist"; }
```

☀ Historical Ligatures: hlig

由用户自行决定是否应用的过时的连字书写形式。

可以表现此特征的字体:  Google的Cardo等字体。

使用CSS示意:

```
.hlig { font-feature-settings: "hlig"; }
```

☀ Localised Forms: locl

用于在广泛地理区域上编写多种语言的任何脚本都开发了特定字母的本地化变体形式,供各个文学社区使用。例如,保加利亚和塞尔维亚字母表中的许多字母具有与俄罗斯对应字母,但形式彼此不同。在某些情况下,本地化形式仅与脚本“规范”略有不同,在其他情况下,形式是完全不同的。此功能可以将本地化的字形替换为默认形式。

可以表现此特征的字体:  苹果San Francisco字体。

此特征标签默认设计可用，但实际表现各个浏览器有差异。使用CSS示意：

```
.loc1 { font-feature-settings: "loc1"; }
```

☀ Localised Forms: rand

用随机形式替换字符（用来模拟手写）。

可以表现此特征的字体：FF Kosmik和Punk Nova和MF FetaMont。

表现示意如下截图，仔细观察可以看到下面一行连续字母大小参差，上面一行是没有应用该标签特性的效果：

使用CSS示意：

```
.rand { font-feature-settings: "rand"; }
```

如果想要关闭随机，代码如下：

```
.rand--off { font-feature-settings: "rand" 0, "calt" 0; }
```

☀ Character Variant 1 – 99: cv01-cv99

单个字符的多个变体（可能不适用于许多其他字符）请参阅大量文档的参考。

其他略~

☀ Stylistic Alternates: salt

此功能使用变化的样式替换默认字符字形。

可以表现此特征的字体： Adobe的Bree系统字体。

使用示意：

```
.salt { font-feature-settings: "salt"; }
```

☀ Subscript: subs

此功能可以使用下标字形替换默认字形，或者可以将字形替换与定位调整组合以进行正确放置。

可以表现此特征的字体： Adobe Warnock系列字体

使用示意：

```
.subs { font-feature-settings: "subs"; }
```

☀ Superscript: sups

此功能用优质数字（主要用于脚注指示）替换衬里或旧式数字，并用高级字母替换小写字母（主要用于缩写的法语标题）。


可以表现此特征的字体： Adobe Hypatia Sans系列字体

使用示意：

```
.sup { font-feature-settings: "sup"; }
```

☀ Titling Alternates: titl

此功能将默认字形替换为专门为标题设计的相应形式。这些可以是全身和（或）更大的身体，并且可以调整以便以更大的尺寸观看。

可以表现此特征的字体： Google的Allan字体

CSS使用示意：

```
.titl { font-feature-settings: "titl"; }
```

☀ Optical size: size

不是查找：功能表向应用程序提供有关字体外观和意图的信息，以帮助进行字体选择。

可以表现此特征的字体： Google的Source Sans Pro字体

其他略。

☀ Ornaments: ornm

这是一个双功能功能，它使用两种输入方法让用户可以访问字体中的装饰字形（例如fleurons，dingbats和border元素）。一种方法用全套可用装饰字符替换项目符号字符；另一个用分配给它们的装饰字符替换特定的“低级ASCII”字符。第一种方法支持普通用户或浏览用户；第二个支持高级用户。

可以表现此特征的字体：Clara

CSS使用示意：

```
.ornm { font-feature-settings: "ornm"; }
```

☀ Kerning: kern

此功能可调整字形之间的空间量，通常可在字形之间提供可见的一致间距。虽然设计良好的字体总体上具有一致的字形间距，但是一些字形组合需要调整以提高易读性。请注意，此功能可能适用于两个以上字形的运行，并且不会用于等宽字体。另请注意，此功能不适用于垂直设置的文本。

大部分OpenType字体都支持此特征，因此，是一个可以在web中大范围示意的属性值。

效果示意：

CSS使用示意，两行CSS分别对应上面截图的上下两行：

```
.kern--off { font-feature-settings: normal; font-kerning: none; }  
.kern { font-feature-settings: "kern"; }
```

☀ Glyph Composition / Decomposition: ccmp

为了最小化字形交替的数量，有时需要将字符分解为两个字形。另外，为了更好的字形处理，最好将两个字符组合成单个字形。该特征允许这种组合/分解。该特征应作为处理的第一个特征处理，并且只应在调用时处理。

根据某些资料显示，此特征浏览器还在实现中，因此，其他略.....

☀ Lining Figures: lnum

此功能将数字字形从默认或旧式数字更改为衬里数字。请注意，某些字体可能包含衬里图形作为其默认图形样式，在这种情况下启用此功能可能不会影响字形的样式。

可以表现此特征的字体： Adobe Warnock系列字体和 Adobe的Hypatia Sans系列字体。

CSS使用示意：

```
.lnum { font-feature-settings: "lnum"; }
```

☀ Oldstyle Figures: onum

此功能将数字字形从默认或衬里数字更改为旧式数字。请注意，某些字体可能包含旧样式作为其默认图形样式，在这种情况下启用此功能可能不会影响字形样式。

可以表现此特征的字体： Adobe Warnock系列字体和 Adobe的Hypatia Sans系列字体。

应用此特征，数字可以表现为上下高低错落排列。

CSS使用示意：

```
.onum { font-feature-settings: "onum"; }
```

☀ Proportional Figures: pnum

原本默认字形的宽度是表格宽度，也就是各个字形有着统一的宽度表现，此功能可以把原本的宽度表现变成基于字形本身形状下的特定宽度表现。请注意，某些字体默认情况下其宽度就是按照数字本身形状来的，在这种情况下启用此功能可能不会影响字形的宽度。还有，等宽字体不受此功能影响。

可以表现此特征的字体： Adobe的Hypatia Sans系列字体。

效果示意如下：

上图中的垂直线是为了方便大家看清楚每个数字宽度而画出来的，实际上是没有这些竖直线的。

CSS使用示意：

```
.pnum { font-feature-settings: "pnum"; }
```

☀ Tabular Figures: tnum

此功能和 "pnum" 正好是对立的。"tnum" 是让比例宽度显示的数字变成按照表格宽度显示，对比下面的图就知道意思了：

尤其观察数字 1 占据的宽度，"pnum" 只有一点点宽度，按照字形比例来的，而这里的 "tnum" 则宽度和后面的一致，按照表格宽度来的。

可以表现此特征的字体：📄 Adobe的Hypatia Sans系列字体。

CSS使用示意：

```
.tnum { font-feature-settings: "tnum"; }
```

☀ Ordinals: ordn

表现为序列数字。

可以表现此特征的字体：📄 Adobe Warnock系列字体。

CSS使用示意：

```
.ordn { font-feature-settings: "ordn"; }
```

☀ Fractions: frac

此功能将斜线分隔的数字替换为普通（对角线）分数。

可以表现此特征的字体：📄 Google的Ubuntu Mono字体、📄 Adobe Warnock系列字体以及🍏 苹果San Francisco字体。

效果示意：

CSS使用示意：

```
.frac { font-feature-settings: "frac"; }
```

☀ Alternative Fractions: afrc

将由斜线分隔的数字转换为交替叠加分数形式，对比下图可以感知到和 "frac" 分数表现形式的不同。

可以表现此特征的字体：  Google的Ubuntu Mono字体。

CSS使用示意：

```
.afrc { font-feature-settings: "afrc"; }
```

☀ Denominator: dnom

转换为适当的分数分母形式。

可以表现此特征的字体：  Google的Ubuntu Mono字体和  Adobe Hypatia Sans系列字体。

浏览器真实表现截图：

CSS使用示意：

```
.dnom { font-feature-settings: "dnom"; }
```

☀ Numerator: numr

转换为适当的分数分子形式。

可以表现此特征的字体：  Google的Ubuntu Mono字体。

Chrome浏览器下可能表现为：

CSS使用示意：

```
.numr { font-feature-settings: "numr"; }
```

☀ Slashed Zero: zero

0变成带斜杠的0。

可以表现此特征的字体：  Adobe Warnock系列字体、  Adobe Hypatia Sans系列字体以及  Google的Exo字体等。

CSS代码示意：

```
.zero { font-feature-settings: "zero"; }
```

☀ Mathematical Greek: mgrk

将希腊字形的标准印刷形式替换为数学符号中常用的相应形式（这是希腊字母表的一个子集）。

可以表现此特征的字体：  Fira Sans、FF DIN、FF Meta。

```
.mgrk { font-feature-settings: "mgrk"; }
```

☀ Stylistic Set 1 – 20: ss01-ss20

此功能使用其他样式替换默认字符字形集，可以看成是上面诸多字形特征的另外一个数组名称，范围从ss01-ss20，可以有20个代称。这种多种风格集中在一起的字形，可以设计成在视觉上有特定规律、或以特定方式交互或以其他比较协调的方式，这样，一字排开呈现的时候效果看上去会很舒服。另请参见：stylistic alternates (salt)，旨在提供单个字形的风格替代。

要使用 `font-feature-settings` 启用它，请使用 `ss##` 作为功能标记，其中##是01-20中的任意两位数组组合。请注意，字体以完全任意和自定义的方式使用样式集。例如，两种不同的字体可以使用相同的特征（如ss01）来替换完全不同的字形集；或者，特定字体可以使用ss01和ss03来替换字形集，但忽略ss02。某些类型的生产这会提供有关如何组织功能的文档。

可以表现此特征的字体： Adobe Hypatia Sans系列字体、 苹果San Francisco字体等很多字体。

效果略，演示略，有兴趣可以参见最后的文档。

☀ Proportional Widths: pwid

字形宽度，东亚文字占据宽度按照字形来算。通常用于CJKV字体中的拉丁字符，但也可用于日语字体中的假名。中文字体都是等宽，因此，用不到。放个图示意下吧：

☀ Proportional Alternate Widths: palt

此功能可重新设置设计为全宽度宽度的字形，使其适合单独（或多或少成比例）的水平宽度。这与pwid的不同之处在于它不能替代新的字形（GPOS，而不是GSUB功能）。用户可能更喜欢等宽的形式，或者可能只是想确保字形适合并且不在垂直设置中旋转。

☀ Proportional Kana: pkna

此功能使用比例字形替换在均匀宽度（半宽或全宽）上设置的假名和假名相关字形。它类似于比例宽度功能，但仅影响假名。

日语体系中有效，不展开，略~

☀ Full Widths: fwid

此功能使用在完整（通常为em）宽度上设置的字形替换在其他宽度上设置的字形。在CJKV字体中，这可能包括“低ASCII”拉丁字符和各种符号。在欧洲字体中，此功能使用等宽字形替换按比例间隔的字形，这些字形通常设置为0.6em的宽度。

```
.fwid { font-feature-settings: "fwid"; }
```

☀ Half Widths: hwid

某些字符占据em一半宽度。图略，可对比下面的1/4宽脑补。

☀ Alternate Half Widths: halt

此功能重新设置了设计为全宽度宽度的字形，使它们适合半宽宽度。这与 `"hwid"` 的不同之处在于它不能替代为新的字形。

例如默认情况下，有些标点虽然字形很小，但依然占据1em宽度，就显得很稀疏，例如：「作者：张鑫旭」。这里的冒号和「之间距离就很开，如果表现为半宽度，则甚好。

CSS使用：

```
.halt { font-feature-settings: "halt"; }
```

☀ Third Widths: twid

1/3宽。

```
.twid { font-feature-settings: "twid"; }
```

☀ Quarter Widths: qwid

1/4宽。

```
.qwid { font-feature-settings: "qwid"; }
```

☀ JIS78 Forms: jp78

日语字形某规范形式，我们用不到，略~

☀ JIS83 Forms: jp83

日语字形某规范形式，我们用不到，略~

☀ JIS90 Forms: jp90


日语字形某规范形式，我们用不到，略~

☀ JIS2004 Forms: jp04

日语字形某规范形式，我们用不到，略~

☀ Traditional Forms: trad

用相应的“繁体”形式取代“简体”的中国汉字或日本汉字形式。

在  Adobe的Kozuka Mincho系统日文字体中有效，中文字体目前没有看到谁对此有所设计。

CSS使用：

```
.trad { font-feature-settings: "trad"; }
```

☀ **Expert Forms: expt**

日语中有效，不必了解。

☀ **Hojo Kanji Forms: hojo**

日语中有效，不必了解。

☀ **NLC Kanji Forms: nlck**

日语中有效，不必了解。

☀ **Ruby Notation Forms: ruby**

日语体系中有效，不展开，略~

☀ **Horizontal Kana Alternates: hkna**

此功能将标准假名替换为专门为仅水平书写而设计的形式。这是一种印刷优化方法，可以改善贴合度，使颜色更均匀。

日语专用，我们汉字用不到，不展开~

☀ **Alternate Annotation Forms: nalt**

此功能用各种符号形式替换默认字形（例如，放置在空心圆圈或实心圆圈，正方形，圆括号，菱形或圆形框中的字形）。在某些情况下，注释可能已经存在，但是用户可能想要不同的注释形式。

上面截图，依次使用的CSS如下：

```
.nalt { font-feature-settings: "nalt"; }  
.nalt-1 { font-feature-settings: "nalt" 1; }  
.nalt-2 { font-feature-settings: "nalt" 2; }  
.nalt-3 { font-feature-settings: "nalt" 3; }  
.nalt-4 { font-feature-settings: "nalt" 4; }  
.nalt-5 { font-feature-settings: "nalt" 5; }  
.nalt-6 { font-feature-settings: "nalt" 6; }  
.nalt-7 { font-feature-settings: "nalt" 7; }
```

☀ **Italics: ital**

某些字体（例如Adobe的Pro日语字体）某些字符同时包含罗马字形和斜体字形。此功能将罗马字形替换为相应的斜体字形。

CSS示意：

```
.ital { font-feature-settings: "ital"; }
```

☀ **Vertical Kerning: vkern**

此功能可调整字形之间的空间量，通常可在字形之间提供视觉上一致的间距。虽然设计良好的字体总体上具有一致的字形间距，但是一些字形组合需要调整以提高易读性。除了垂直方向的标准调整之外，该功能还可以通过设备表提供与尺寸相关的字距调整数据，在X文本方向上提供“交叉流”字距调整，并且可以调整字形位置，而与提前调整无关。请注意，此功能可能适用于两个以上字形的运行，并且不会用于等宽字体。另请注意，此功能仅适用于垂直设置的文本。

```
.vkern { font-feature-settings: "vkern"; }
```

☀ **Vertical Alternates: vert**

在垂直书写模式下，此功能使用针对垂直书写调整的变体替换默认表单。虽然大多数CJKV字形在垂直书写模式下保持垂直，但有些字体采用不同的形式（通常是旋转和重新定位）。此功能涵盖的字形对应于通常在低端DTP应用程序中旋转的字符集。

```
.vert { font-feature-settings: "vert"; }
```

☀ **Proportional Alternate Vertical Metrics: vpal**

此功能可重新设置设计为全高度的字形，使其适合单个（或多或少比例）垂直高度。这与valt的不同之处在于它不能替代新的字形（GPOS，而不是GSUB特征）。用户可能更喜欢等宽的形式，或者可能只是想确保字形适合。

```
.vpal { font-feature-settings: "vpal"; }
```

☀ **Alternate Vertical Half Metrics: vhal**

此功能可重新设置设计为全高度的字形，使其适合半高。

```
.vhal { font-feature-settings: "vhal"; }
```

☀ **Vertical Kana Alternates: vkna**

此功能使用专门为垂直书写设计的表单替换标准假名。这是一种印刷优化，可以改善贴合度，使颜色更均匀。

```
.vkna { font-feature-settings: "vkna"; }
```

☀ **其他: others**

还有很多其他特征关键字，浏览器尚未来得及跟进，有兴趣可以参见这里：[Feature tags](#)

四、写在最后点内容

Stop ! Stop ! 不能再多说什么了，这页面滚动高度要直逼10,000像素+了。

还是说一句吧：前段时间，设计师给我提了个需求，输入法输入的是简体，呈现的是繁体。理论上讲，font-feature-settings可以实现。然而，尴尬在于，没有中文字体在Design的时候留下这个功能特征，有能力没材料。

font-feature-settings虽然可以让字体呈现更加精致，更加丰富。但，如果没有优秀设计师设计出对应的字体，则，一切都是镜中水月——虚的。英文和数字数量有限，因此，设计成本低，才有了丰富多彩的展现。而中文字体，常见汉字就好几千，成本高，也是没办法的时候。

支持各种font-feature-settings特性的日文字体“Kozuka Mincho Pr6N”是字体设计师Masahiko Kozuka（小塚昌彦）设计的，不知道，什么时候有位优秀的中国设计师设计出一款字体，包含丰富的feature。

参考文档

- [Syntax for OpenType features in CSS](#)
- [The Complete CSS Demo for OpenType Features](#)
- [Registered features – definitions and implementations](#)
- [MDN font-feature-settings](#)

《CSS世界》签名版独家发售，包邮，可指定寄语，[点击显示购买码](#)

（本篇完） // 想要打赏？[点击这里](#)。有话说？[点击这里](#)。



« CSS蛋糕应用之：数据上报和HTML验证

使用wavesurfer.js显示mp3 audio音频的波形图 »

猜你喜欢

- [了解CSS属性font-kerning,font-smoothing,font-variant](#)
- [真正了解CSS3背景下的@font face规则](#)
- [页面可用性之浏览器默认字体与CSS中文字体](#)
- [CSS font关键字属性值的简单研究](#)
- [翻译 - CSS高峰会议内容精选](#)
- [Google字体API使用简单示例](#)
- [CSS unicode-range特定字符使用font-face自定义字体](#)
- [CSS font-family常见中文字体对应的英文名称](#)
- [小tips: 使用JS检测用户是否安装某font-family字体](#)
- [分享三个纯CSS实现26个英文字母的案例](#)
- [CSS3 transition实现超酷图片墙动画效果](#)

分享到：[+](#) [QQ](#) [微信](#) [微博](#) [贴吧](#) [收藏](#) [1](#)

标签：[css相关](#), [font](#), [font-feature-settings](#), [font-kerning](#), [font-smoothing](#), [font-variant](#), [中文字体](#), [字体](#)

发表评论（目前3条评论）

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

1. 逆袭哥说道：

2018年12月12日 15:16

路过帮顶，写的很好！

[回复](#)



2. 巴里切罗说道：

2018年12月11日 21:52

在 mac 下试了一下，苹方和冬青黑都可用 trad，但是比较机械，完全是一对一的字形直翻

[回复](#)



张 鑫旭说道：

2018年12月12日 10:56

好的，感谢反馈。

[回复](#)



最新文章

- » [常见的CSS图形绘制合集](#)
- » [粉丝群第1期CSS小测点评与答疑](#)
- » [分享三个纯CSS实现26个英文字母的案例](#)
- » [小tips: 纯CSS实现打字动画效果](#)
- » [CSS/CSS3 box-decoration-break属性简介](#)
- » [CSS :placeholder-shown伪类实现Material Design占位符交互效果](#)
- » [从天猫某活动视频不必要的3次请求说起](#)
- » [CSS vector-effect与SVG stroke描边缩放](#)
- » [CSS ::backdrop伪元素是干嘛用的？](#)
- » [周知：CSS -webkit-伪元素选择器不再导致整行无效](#)

今日热门

- » [常见的CSS图形绘制合集](#) ⁽¹⁷⁸⁾
- » [粉丝群第1期CSS小测点评与答疑](#) ⁽¹¹²⁾
- » [未来必热：SVG Sprite技术介绍](#) ⁽¹¹¹⁾
- » [HTML5终极备忘大全（图片版+文字版）](#) ⁽⁸⁵⁾
- » [让所有浏览器支持HTML5 video视频标签](#) ⁽⁸³⁾
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) ⁽⁸⁰⁾
- » [CSS3下的147个颜色名称及对应颜色值](#) ⁽⁷⁸⁾
- » [小tips: 纯CSS实现打字动画效果](#) ⁽⁷²⁾

- » [写给自己看的display: flex布局教程](#) ⁽⁶⁹⁾
- » [分享三个纯CSS实现26个英文字母的案例](#) ⁽⁶⁹⁾

今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) ⁽⁷⁶⁾
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) ⁽⁶⁴⁾
- » [看，for..in和for..of在那里吵架！](#) ⁽⁶⁰⁾
- » [是时候好好安利下LuLu UI框架了！](#) ⁽⁴⁷⁾
- » [原来浏览器原生支持JS Base64编码解码](#) ⁽³⁵⁾
- » [妙法攻略：渐变虚框及边框滚动动画的纯CSS实现](#) ⁽³³⁾
- » [炫酷H5中序列图片视频化播放的高性能实现](#) ⁽³¹⁾
- » [CSS scroll-behavior和JS scrollIntoView让页面滚动平滑](#) ⁽³⁰⁾
- » [windows系统下批量删除OS X系统.DS_Store文件](#) ⁽²⁶⁾
- » [写给自己看的display: flex布局教程](#) ⁽²⁶⁾

猜你喜欢

- [了解CSS属性font-kerning,font-smoothing,font-variant](#)
- [真正了解CSS3背景下的@font face规则](#)
- [页面可用性之浏览器默认字体与CSS中文字体](#)
- [CSS font关键字属性值的简单研究](#)
- [翻译 - CSS高峰会议内容精选](#)
- [Google字体API使用简单示例](#)
- [CSS unicode-range特定字符使用font-face自定义字体](#)
- [CSS font-family常见中文字体对应的英文名称](#)
- [小tips: 使用JS检测用户是否安装某font-family字体](#)
- [分享三个纯CSS实现26个英文字母的案例](#)
- [CSS3 transition实现超酷图片墙动画效果](#)