

Web 应用英文字体分析报告

Yuting Cao,^{*} Jing Liu[†], and Xiaowei Miao[‡]

Software Institute, Nanjing University, Nanjing, China

May 7, 2015

摘要

本文对 Web 应用中的流行字体，计算机屏幕对字体的展示技术进行分析，并基于此提出了对 Web 应用中字体选择的建议。

关键字：英文字体、Web 应用、分辨率

1 Web 应用流行字体

1.1 总体概况

1999 年 01 月 15 日，Microsoft 发布“TrueType core fonts for the Web”，首次提出其与 Apple Inc 共同研制的 29 种 Web 字型标准及其扩展版本，字型标准如 Figure 2 所示 [6]。

归结而言，包括：Andale Mono, Arial, Arial Black, Comic Sans MS, Courier New, Georgia, Impact, Times New Roman, Trebuchet MS, Verdana 及 Webdings 共 11 类 [9]，如 Figure 1 所示。

Andale Mono Arial **Arial Black**
Comic Sans MS Courier New Georgia
Impact Times New Roman Trebuchet MS
Verdana ▶ 📦 🚲 ♥ ⓘ ● ■ ?

Figure 1: 11 类字体样例

^{*}Email: cyt12@software.nju.edu.cn (121250005)

[†]Email: ljing12@software.nju.edu.cn (121250083)

[‡]Email: mxw12@software.nju.edu.cn (121250101)

当然，目前随着互联网技术的飞速发展，已有数百种 Web 开发字体可供选择 [8]。相应字体的分类标准也各有不同。使用最广，对字体进行系统的分类是 1954 年由 Maxmilien Vox 制定，由 Association Typographique Internationale（国际字体协会）于 1962 年修订的 Vox-AtypI 分类，将字体分为 11 大类 [5]，当然，这 11 类与 TrueType 归纳的 11 类不同。另外，法国字体艺术家 Francis Thibaudau 将字体分为四大类，Antiques（sans-serif 无衬线），Egyptiennes（slab-serif 扁平衬线），Didots，Elzevirs（triangular serifs 三角形衬线），这也是 Vox 分类的基础。

由于 Microsoft 和 Apple Inc 自 1999 年起，将其 11 类字型标准分别安装到 Internet Explorer 和 Safari 中，作为默认的浏览器字体为大众熟知，因而 Web 程序员开发大多选择这 11 类字体，逐渐成为当前 Web 主流字体。

Filename	Name	Variants	Last version	Copyright ^[9]
arial32.exe	Arial for Windows 9x, NT and Windows 2000	regular, bold, italic, bold italic	version 2.82	Monotype
Arial.sit.hqx	Arial for Apple Mac OS	regular, bold, italic, bold italic	version 2.90	Monotype
courie32.exe	Courier New for Windows 9x, NT and Windows 2000	regular, bold, italic, bold italic	version 2.82	Monotype
CourierNew.sit.hqx	Courier New for Apple Mac OS	regular, bold, italic, bold italic	version 2.61	Monotype
times32.exe	Times New Roman for Windows 9x, NT and Windows 2000	regular, bold, italic, bold italic	version 2.82	Monotype
TimesNew.sit.hqx	Times New Roman for Apple Mac OS	regular, bold, italic, bold italic	version 2.91	Monotype
arialb32.exe	Arial Black for Windows 9x, NT and Windows 2000	black	version 2.35	Monotype
ariblk.exe	Arial Black for Windows 3.1 and 3.11	black	version 2.20	Monotype
ArialBlack.sit.hqx	Arial Black for Apple Mac OS	black	version 2.35	Monotype
andale32.exe	Andale Mono for Windows 9x, NT and Windows 2000	regular	version 2.00	Monotype
mtcom.exe	Monotype.com for Windows 3.1 and 3.11 (later versions released as Andale Mono)	regular	version 1.10	Monotype
andalemono.sit.hqx	Andale Mono for Apple Mac OS	regular	version 2.00	Monotype
comic32.exe	Comic Sans MS for Windows 9x, NT and Windows 2000	regular, bold	version 2.10	Microsoft
comic.exe	Comic Sans MS for Windows 3.1 and 3.11	regular, bold	version 1.20	Microsoft
ComicSans.sit.hqx	Comic Sans MS for Apple Mac OS	regular, bold	version 2.10	Microsoft
impact32.exe	Impact for Windows 9x, NT and Windows 2000	regular	version 2.35	Monotype
impact.exe	Impact for Windows 3.1 and 3.11	regular	version 2.20	Monotype
Impact.sit.hqx	Impact for Apple Mac OS	regular	version 2.35	Monotype
georgi32.exe	Georgia for Windows 9x, NT and Windows 2000	regular, bold, italic, bold italic	version 2.05	Microsoft
georgia.exe	Georgia for Windows 3.1 and 3.11	regular, bold, italic, bold italic	version 1.00	Microsoft
Georgia.sit.hqx	Georgia for Apple Mac OS	regular, bold, italic, bold italic	version 2.05	Microsoft
trebuc32.exe	Trebuchet MS for Windows 9x, NT and Windows 2000	regular, bold, italic, bold italic	version 1.22	Microsoft
trebuc.exe	Trebuchet MS for Windows 3.1 and 3.11	regular, bold, italic, bold italic	version 1.00	Microsoft
Trebuchet.sit.hqx	Trebuchet MS for Apple Mac OS	regular, bold, italic, bold italic	version 1.15	Microsoft
verdan32.exe	Verdana for Windows 9x, NT and Windows 2000	regular, bold, italic, bold italic	version 2.35	Microsoft
verdana.exe	Verdana for Windows 3.1 and 3.11	regular, bold, italic, bold italic	version 1.01	Microsoft
Verdana.sit.hqx	Verdana for Apple Mac OS	regular, bold, italic, bold italic	version 2.35	Microsoft
webdin32.exe	Webdings for Windows 9x, NT and Windows 2000	symbol	version 1.03	Microsoft
webdings.exe	Webdings for Windows 3.1 and 3.11	symbol	version 1.01	Microsoft

Figure 2: TrueType 字型标准

1.2 字体分析

以下是对 Web 主流字体的分析。

1.2.1 Andale Mono

Internet Explorer 4.0 的主流字体，现由 Lucida Console 替代。

1.2.2 Arial

Arial 是微软公司的网页核心字体之一，最常用的 sans serif 字体，其优点是支持正确打印并且便宜，Arial Black 是 Arial 的变体，但是，当字号很小时不容易阅读并且当大写的“I”和小写的“l”无法区别时，可以使用 Tahoma 字体替代。

苹果系统没有 Arial 这种字体，但其对应有 Helvetica 字体，它是 MAC 机上与 Arial 字体最相似的 WEB 字体，是一种非衬线字体 [4]。

这也是 Linux 和 Unix 下的 Web 常用字体。

1.2.3 Comic Sans MS

类手写，常用于纸本漫画及线上漫画（WebComic）以取代手写，风格类似于卡通式卖萌字体，但字体处理细节在“笔画均匀度”、“字间距”、“反锯齿”上处理不当 [7]。正如该字体的作者 David Kadavy 所言，其并不适用于任何情况，只适合儿童使用。

1.2.4 Courier New

Courier 是等宽的粗衬线字体，因其等宽特性可以轻易对齐字段的左右边界，成为脚本和程序设计中源代码的常用字体，Courier New 是 Courier 的变体，相对于 Courier 行距更宽。

1.2.5 Georgia

Georgia 是一种衬线字体，具有小字下仍能清新辨识的特性，可读性很好。表面上，Georgia 与 Times New Roman 相当类似，实质有很多不同。首先，在相同的字号下，Georgia 的字元比 Times New Roman 的字元略大；其次，Georgia 的字元线条较粗，衬线部分也比较钝而平。另外在数字部分也非常不同，Georgia 采用称为“不齐线数字”的数字，特色在于数字会像西文字母般有高矮大小之别。微软将 Georgia 列入网页核心字型，是视窗操作系统的内建字型之一。苹果电脑的麦金塔系统之后也跟进采用 Georgia 作为内建字型之一。

但是，数字“0”与字母“o”在 Georgia 字体下可能是显示成一模一样。

1.2.6 Impact

Impact 是一个由设计师乔佛瑞 • 李在 1965 年发表的一个无衬线字体，笔画粗、间距紧以及内部空间狭窄，正如其名 impact（压紧）所指。此字型的 x 字高相当大，几乎接近其大写字母的三分之二，因此升部相当短小，降部更是如此。因为字体较为粗犷，适合使用在 Web 页面标题上，而不常用在内文。

1.2.7 Times New Roman

Times New Roman 是 Web 开发最常用的衬线字体之一，由于其中规中矩、四平八稳的经典外观，所以经常被选择为标准字体。《泰晤士报》首次采用了 Times New Roman 后，这个优秀的字型很快地博得了大众的青睐，获得了极大的成功，后来成为 Web 开发的重要字体之一。

1.2.8 Trebuchet MS

Trebuchet MS 是一种无衬线字体，微软称 Trebuchet MS 为良好的网页设计字体，Trebuchet MS 与其他常见的无衬线字体大不相同，其显著特色包括：大写字母“M”的两端与垂直线呈 10 度角；大写字母“Q”的尾巴；大写字母“A”中的横杠较低；小写字母“e”和数字“6、9”的尾巴较短；小写字母“g”的下圆带有开口；小写字母“i、j”上面的点为圆点；小写字母“l”的尾巴呈弯曲状；金钱符号“\$”的直竖笔画不贯穿 S 字母，只出现在上端和下端；“&”符号是英文“Et”的合写；感叹号“!”下的点为圆点；斜体不单只是把原本字体倾斜，而是另外设计一套斜体的字母等。

1.2.9 Verdana

Verdana，无衬线字体，由于它在小字上仍有结构清晰端整、阅读辨识容易等高品质的表现，因而在 1996 年推出后即迅速成为许多领域所爱用的标准字型之一。

1.2.10 Webdings

Webdings 是一个 TrueType 的 dingbat 字体，于 1997 年发表，搭载在其后的 Microsoft Windows 视窗系统内。在 Unicode 7.0 发布之前，大多数的字形都没有 Unicode 的相对字。另外，由于 firefox 并不支持 Webdings，兼容性需要提升。

1.3 Web 字体的基本特性

在 CSS 中，Web 字体相关的属性如 Table 1 所示：

font-family	font-size	font-weight	font-style
font-variant	font	text-decoration	text-transform
line-height	text-indent	text-align	vertical-align
letter-spacing	word-spacing	white-space	direction

Table 1: Web 字体属性列表

1.4 字体风格及使用场合

从字体风格角度，字体主要包括 Academic Categories 及 Popular Style & Tags。

如 Figure 3 所示：

Academic Categories				
字体风格	适用/优点	不适用/缺点	举例 1	举例 2
Humanist	提供平衡	标题	Calluna	Garamond
Transitional	正文	避免过度信赖	New Baskerville	Helvetica Neue
Modern/Geometric	描述历史	正文	Bodoni	Futura

Popular Style & Tags				
字体风格	适用/优点	不适用/缺点	举例 1	举例 2
Bitmap	标题，按钮	历史	BD Brick	Text Pro
Blackletter	历史	主体	Old German	Text
Comic & Cartoon	题注，标题	大段文字	Medium	HERO
Distressed	结构	重复	Zapatista	SHNIXGUN
Futuristic	标题，主体	不一致	CHILOPOD	ZAUS
Handwriting	标题，动作	过分详细	Pacifico	Bistro Script
Stencil	插图	主体	STENCIL	Cutouts
Typewriter	代码	千篇一律	Typewriter	Mono
Western	非正式	正式	XXXXXX	BHAND

Figure 3: TrueType 字型标准

2 高细腻技术下的屏幕文字显示

2.1 术语解释

2.1.1 pt 与 px

pt 全称为 point，即“磅”，是印刷行业的常用单位；px 全称为 pixel，即“像素”，是屏幕显示的基本单位。像素本身并没有确切的大小，而磅则是一个固定的长度单位，大小为 1/72 英寸。

2.1.2 字体微调 (Font hinting)

字体微调技术多见于 TrueType 字体 [11]，是精确地定义字体所要显示的像素是一种方法，用以尽可能在小尺寸、低分辨率的情况下创建最好的字符位图形状。

TrueType 字体使用矢量图形描述字符的轮廓，并在显示时在轮廓内进行填充，而字符在屏幕上实际显示时，又需要转化为一组像素点。在转化的过程中，矢量的轮廓往往无法紧密贴合像素点的边缘（见 Figure 4）。此时若简单地对未被完全包括在轮廓内的像素进行二值化处理，显示效果将会与字体设计时的预期产生出入。在小字号中，这样的出入产生的出入将更为明显，并可能导致字符在屏幕上产生笔划粘连、残缺等问题（见 Figure 5）。使用点阵字体是解决这一问题的最佳方案，因为点阵字体精确地控制了每一个像素的明暗，可以达到最佳的显示效果。然而，点阵字体通常为某一特定字号设计，不能将字号放大成为了其最大问题。在这样的背景下，字体的 Hinting 技术应运而生。Hinting 的基本原理是在矢量字体中嵌入一些小字号的点阵字体，在字号较小时，直接将点阵结果显示在屏幕上；使用大字号时，再通过矢量图形进行渲染。Hinting 技术有效地将点阵字体与矢量字体的优势融合起来，使得 TrueType 字体无论字号大小如何变化，都能在屏幕上清晰地显示出来。

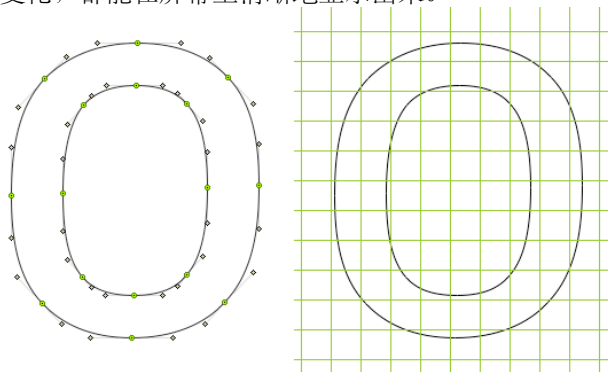


Figure 4: 像素边缘未被矢量轮廓贴合示意图 [1]

The quick brown fox jumps over the lazy dog.

Black and White rendering of an unhinted font, at 10 ppm, magnified to 200%

The quick brown fox jumps over the lazy dog.

Black and White rendering of an hinted font (Fedra Sans Screen), at 10 ppm, magnified to 200%

Figure 5: 黑白渲染模式下使用 Hinting 技术使显示效果产生差异

2.1.3 DPI

DPI 全称为 dots per inch，即每英寸容纳、打印或显示的点的数目，最初被应用于印刷行业。DPI 越大，通常意味着输出越精细，输出效果也就越好。尽管在显示器上，像素才是最终的输出对象，但 DPI 的称呼并未相应地产生变化，原因在于 PPI (pixels per inch) 已被约定俗成地用于衡量图像的采样率，成为了输入层面的度量值。近年来，在度量显示器显示画面的精细程度时也会使用 PPI，但此时，DPI 和 PPI 可以被认为是同一概念的不同称呼。

2.2 高 DPI 与高分辨率

分辨率决定了屏幕上所能显示的像素数量多少。由于任何网页内容最终都被转化为像素进行输出，分辨率也就决定了能够显示的网页元素的多少，分辨率越高，一屏幕能够容纳的内容也就越多；又由于像素本身没有大小的概念，分辨率也就无法决定网页元素显示的大小。DPI 的大小则不同。DPI 的数值体现了像素的密度水平，像素密度越大，显示单位物理长度内容使用的像素数目就越多，显示效果也就越精细。

这里需要说明的是，DPI 的提高并不是总能改善显示效果的。一旦 DPI 突破人眼极限，再高的技术指标也不能影响画面的显示效果。根据苹果公司于第三代 iPad 发布会上给出的 Retina 设计标准公式，在通常的使用距离下，手机的像素密度达到 300DPI，平板电脑的像素密度达到 260DPI，PC 与笔记本电脑的像素密度达到 200DPI 之后，人眼就无法再区分出单个像素点，画面也就不会再出现颗粒感。

现代显示器的分辨率与 DPI 并不是同步发展的。从 Table 2 中，我们可以看出，在小尺寸 4K 分辨率显示器问世之前，主流的家用户显示器，DPI 数值都在 100 左右。

尺寸与分辨率	PPI
19 寸 1440×900	90
21.5 寸 1920×1080	102
23 寸 1920×1080	96
24 寸 1920×1200	94
27 寸 1920×1080	82
27 寸 2560×1440	109
29 寸 2560×1080	96
28 寸 3840×2160	157
32 寸 3840×2160	127

Table 2: 主流显示器尺寸 PPI 对照表

2.3 高 DPI 下的屏幕文字可读性改进技术

文字的可读性最终体现在阅读给人眼带来的疲劳感上。无论应用怎样的可读性改进技术，只要能有效地降低人阅读屏幕显示文本的疲劳感，延长用户从开始阅读到被迫离开设备休息的时间，就都是成功的。应用这些可读性改进技术后，网页设计就获得了更大的自由度。在低分辨率、低 DPI 时代，为了保证文字的可读性，网页设计者往往倾向于使用一些较粗的字体，但在目前的分辨率，在一般的 DPI 下，使用比以往更细的字体仍然能实现同样清晰、优雅的文字显示效果。

Font Hinting: 不可否认，字体微调技术在低分辨率、小尺寸屏幕的时代非常有效。然而，随着分辨率和 DPI 数值的提升，TrueType 字体的 Hinting 技术虽然仍有效果，但效果已不十分明显。一方面，高分辨率和高 DPI 使得显示同样实际大小的字符时，可以进行渲染的像素数目大幅增加，不使用 Hinting 已经可以达到良好的显示效果；另一方面，由于目前并没有有效的自动化 Hinting 工具，字符的微调需要设计师的人工控制。有数据显示，即使只对 256 个 ASCII 字符进行二值化微调，一名有经验的设计师也需要工作 80 小时 [3]。因此，Hinting 技术在高 DPI 时代已然成为鸡肋，吃力不讨好，只有在与其它显示技术结合的情况下，Hinting 才能重新发挥其应有的作用。

ClearType&DirectWrite: ClearType 技术是针对 LCD 显示设备开发的平滑显示技术。与 CRT 显示器不同，LCD 显示设备中的一个像素可以被分解为 3 个次像素，分别显示红、绿、蓝三种颜色。基于人眼的视觉系统对颜色的细节错误不敏感，而对亮度的细节错误更敏感的事实，ClearType 技术在字符边缘使用一些彩色像素点来增强显示效果 [2]。（ClearType 技术的简单实现机制见 Figure 6）。

DirectWrite 技术与 ClearType 类似，不同之处在于，ClearType 技术只在水平方向平滑的字符边缘的显示效果，而 DirectWrite 技术在水平和垂直两个方向都能使字符的显示更加平滑。

需要说明的是，ClearType 与 DirectWrite 技术只对拥有 Hinting 信息的字体产生效果。

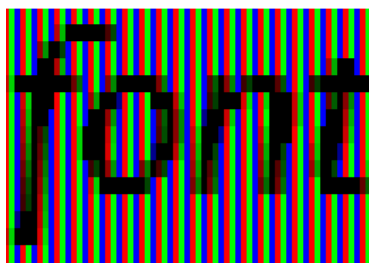


Figure 6: LCD 显示屏像素点极度放大后的 ClearType 显示效果

Retina: 工作方式与游戏中常见的抗锯齿技术十分类似。以 MacBook Pro

with Retina Display 为例，工作时，显卡渲染出 2880×1800 的像素，然后以四个像素为一组，输出显示器上一个实际的像素。这样，用户所看到的文字、图像大小与实际使用 1440×900 分辨率时的保持一致，但精细度得到了明显的提高。

2.4 不同情况下的字体显示情况

2.4.1 Font Hinting

在前文中，字体微调在二值渲染模式下的显示效果已经得到了展示。此处将展示字体微调技术在灰度渲染模式下产生的效果。灰度渲染模式将未被 TrueType 字体的轮廓全部包围的像素以一定亮度的灰色进行填充，可以避免简单的黑白渲染带来的字符残缺问题，但字符粘连的问题仍然不能有效解决。从 Figure 7 中可以看出，在不使用 Hinting 技术时，小字号的文字整体发虚，势必加剧用眼疲劳；使用 Hinting 技术后，字符的主干部分黑色更深，文字显得更“实”，给用户带来了更好的阅读体验。

The quick brown fox jumps over the lazy dog.

Grey-scale rendering of an unhinted font, at 10 ppm, magnified to 200%

The quick brown fox jumps over the lazy dog.

Grey-scale rendering of an hinted font (Fedra Sans Screen), at 10 ppm, magnified to 200%

Figure 7: 灰度渲染模式下 Hinting 技术造成的显示效果差异

2.4.2 ClearType&DirectWrite

如 Figure 8 所示，开启 ClearType 技术后，文字周边的锯齿状轮廓几乎不可见，边缘更加平滑。之所以在放大的状态下，开启 ClearType 后的文字显得发虚，是因为 ClearType 技术对次像素的处理经过图像放大的插值计算后产生了一些负面效果，在原大小下，文字边缘仍然是清晰的。

The quick brown fox jumps over the lazy dog.

Without ClearType rendering of a hinted font (Arial), at 10 pt, magnified to 200%.

The quick brown fox jumps over the lazy dog.

With ClearType rendering of a hinted font (Arial), at 10 pt, magnified to 200%.

Figure 8: ClearType 技术造成的显示效果差异

2.4.3 Retina

如 Figure 9，左右的对比结果显然。在某非 Retina 设备中，图标和文字周边的像素点与锯齿十分明显，而在相同实际物理大小的情况下，搭载了 Retina 技术的 iPad 显示效果则相当出彩，图标和文字的边缘都十分平滑，以肉眼无法察觉出像素点的存在。



Figure 9: 非 Retina 与 Retina 显示设备的显示效果对比

3 Web 应用字体使用建议

基于以上分析，结合 [10] 中的讨论，我们提出了 Web 应用字体选择的建议。

3.1 善于使用字体组合

使用不同字体的组合可以让文本的差异性给用户带来更好地体验，无论是从逻辑上更容易分类，还是从视觉上更方便感知内容角度，字体组合都是非常重要的一个设计方法。关于字体组合有以下几条建议：

- 避免相似性产生的冲突如 Figure 10 相似的但是又有区别的字体放在同一块区域，尤其是在大小一致的情况下，很容易彼此造成干扰，产生不好的视觉体验，可能导致认知困难。因为人眼需要花费额外精力去区别两者，这一点要避免。

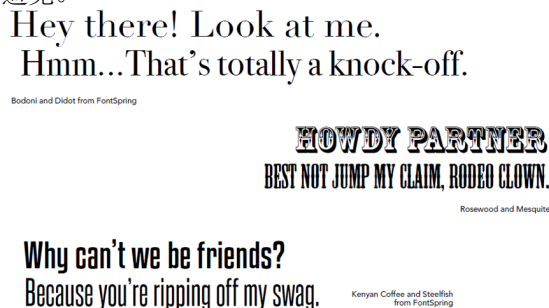


Figure 10: 相似字体产生冲突

- 朴素与时尚兼具

适度使用一些时尚的字体，可以使得文本具有更强的吸引力，但是也同时要结合简朴的字体，正所谓淡妆浓抹总相宜，达到一种简朴和时尚的平衡会是一个很好的选择，如下 Figure 11。

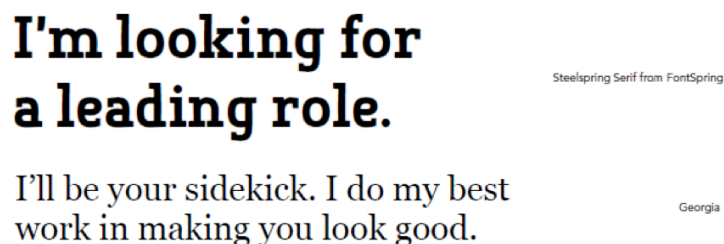


Figure 11: 朴素与时尚结合示例

- 不要忘记使用斜体

斜体是一种简单的变换，但往往得到不错的效果，其接近手写体的风格，对文本有一定强调作用，也增强了文本的艺术性，例如 Figure 12 所示。



Figure 12: 斜体的使用

- 标题与正文不同

对于标题和正文采用不同的风格可以有效的突出标题，使用户很容易的得到关键内容，而且在保证文本整体一致性（正文）的基础上丰富了文本的视觉体验。如 Figure 13 所示。

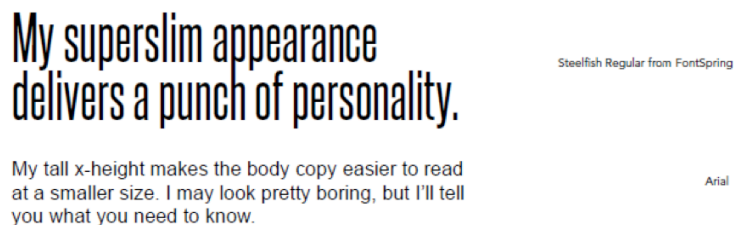


Figure 13: 斜体的使用

3.2 利用对比反差建立字体层次

3.2.1 列出需要表达的所有层次的信息

Web 网页设计中往往有以下的信息块以展现字体。他们的功能不同，往往对应于不同的信息层次，利用字体层次分别对应不同的信息层次是一种比较好的处理办法。常见的信息块包括：

- Navigation （导航）
- Headline （标题）
- Sub-Headline for main content （正文的副标题）
- Introductory （引言）
- Sidebar headline （边栏标题）
- Sidebar body copy （边栏正文）
- Link text （链接文本）
- Call to action （操作提示）
- Contact information （联系信息）
- Footer links （下边栏链接）

3.2.2 基于不同信息层次的字体对比处理

对于不同的信息层次采用不同的字体，有两种思路一是使用尺寸、颜色、粗细和形状产生对比，如 Figure 14 二是对特定的数据元素使用差异化字体，如 Figure 15。

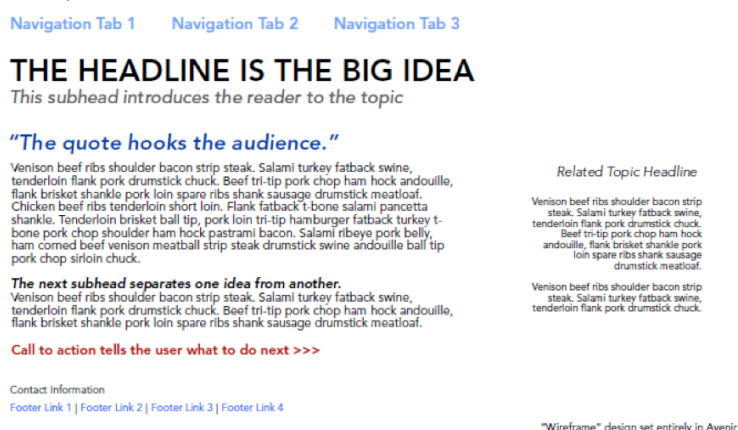


Figure 14: 字体层次化差异处理示例一

THE HEADLINE IS THE BIG IDEA

This subhead introduces the reader to the topic

"The quote hooks the audience."

Venison beef ribs shoulder bacon strip steak. Salami turkey fatback swine, tenderloin flank pork drumstick chuck. Beef tri-tip pork chop ham hock andouille, flank brisket shankle pork loin spare ribs shank sausage drumstick meatloaf. Chicken beef ribs tenderloin short loin. Flank fatback t-bone salami pancetta shankle. Tenderloin brisket ball tip, pork loin tri-tip hamburger fatback turkey t-bone pork chop shoulder ham hock pestrami bacon. Salami ribeye pork belly, ham corned beef venison meatball strip steak drumstick swine andouille ball tip pork chop sirloin chuck.

The next subhead separates one idea from another.

Venison beef ribs shoulder bacon strip steak. Salami turkey fatback swine, tenderloin flank pork drumstick chuck. Beef tri-tip pork chop ham hock andouille, flank brisket shankle pork loin spare ribs shank sausage drumstick meatloaf.

Call to action tells the user what to do next >>>

Related Topic Headline

Venison beef ribs shoulder bacon strip steak. Salami turkey fatback swine, tenderloin flank pork drumstick chuck. Beef tri-tip pork chop ham hock andouille, flank brisket shankle pork loin spare ribs shank sausage drumstick meatloaf.

Venison beef ribs shoulder bacon strip steak. Salami turkey fatback swine, tenderloin flank pork drumstick chuck.

Contact Information

Footer Link 1 | Footer Link 2 | Footer Link 3 | Footer Link 4

"Heirarchy" design set in:
Calibri, Muffraw, Museo, Vanburg,
Franklin Gothic Book, Museo Slab,
Georgia, and Rockwell

Figure 15: 字体层次化差异处理示例二

3.3 处理难以完全控制的情况

设计者需要意识到，无论怎样进行设计，Web 页面最终需要在客户端的浏览器上展现效果，字体的选择很多情况下因此不能起到决定性的作用，反而是客户端的操作系统环境和浏览器配置占据了较大的比重。

3.3.1 浏览器和操作系统的字体支持

Figure 16 中显示了 2014 年全年各个浏览器的市场份额，Figure 17 给出了主流浏览器对字体的支持情况。时至今日，依旧没有一个对于各个浏览器的一个统一而足够优雅的字体的选择解决方案，因此设计者在字体设计过程中，需要更多的考量浏览器对字体的支持情况以及不同操作系统之间字体的差异性。

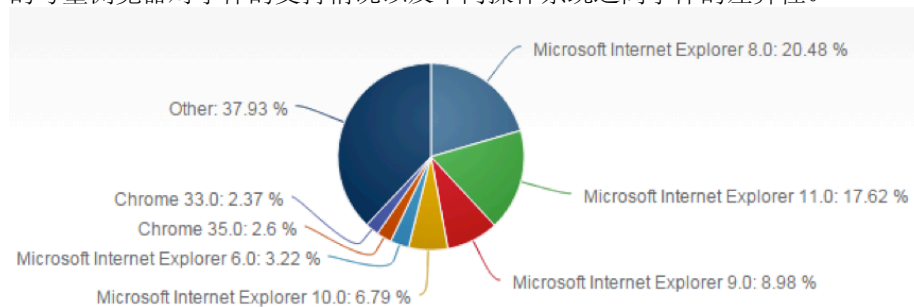


Figure 16: 2014-2015 各个浏览器市场份额

Browser	TrueType	WOFF	EOT	SVG
IE 5-8			Yes	
IE 9	Limited	Yes	Yes	
Firefox 3.5	Yes			
Firefox 3.6+	Yes	Yes		
Safari 3.1+	Yes			Yes
Chrome 6+	Yes	Yes		Yes
Opera 10+	Yes	Coming Soon		Yes

Figure 17: 主流浏览器对字体的支持情况

3.3.2 Web 安全字体

基于兼容性的考虑，可以选择使用对所有主流操作系统支持的字体集合，这样可以在字体显示发那个面提供较好的安全性，缺点是会在显示效果方面有一定的牺牲。

截止到 2011 年的 19 种 Web 安全字体包括，Arial, Arial Black , Arial Narrow, Verdana, Georgia, Times New Roman, Trebuchet MS, Courier, Courier New, Impact, Comic Sans MS, Tahoma, Lucida Sans Unicode, Garamond, MS Sans Serif, MS Serif, Palatino Linotype, Symbol, Bookman Old Style。

3.3.3 使用 @font-face

Bulletproof @font-face 语法以及进阶版 Footspring @font-face 语法，用于诱导浏览器使用它们支持的最佳的（或者某些情况下唯一的）web 字体格式。写法如下所示

```
@font-face{
  font-family:'MyFontFamily';
  src:url('myfont-webfont.eot?#iefix') format('')
    url('myfont-webfont.woff') format('woff')
    url('myfont-webfont.ttf') format('truetype')
    url('myfont-webfont.svg#svgFontName') format('svg');
}
```

3.3.4 字体尺寸控制

- 描述尺寸的标准有如下几种：
 - 绝对尺寸关键字：包括 xxx-small, x-small, small, large, x-large and xx-large.
 - em：衡量宽高（尤其是高度），em-square 标准是基于字体设计的（也叫做 em-box）

- 相对尺寸关键字: `larger` 和 `smaller`。基于父对象的尺寸进行变化。
- 比例关键字: 一种较为可靠的使用比例描述文本尺寸的方法。
- 对于字体尺寸的控制, 建议是结合比例和 `em` 作为标准使用。具体是 [9] 中所指出的方法, 在正文中使用比例关键字使 `body` 部分的文本轻微缩小, 当设计者想使一些独立文本比周围文本偏大的情况下, 使用 `em`。例子如下

3.3.5 创建理想的字体栈

浏览器将会从左到右地寻找字体直到它找到一个用户计算机中存在的字体。比如它会查找 Georgia 字体, 如果没有找到, 它将会查找 Times 字体等等。需要注意的时同一个字体栈下的所有字体应该有相同的(或相似的)高宽比。设计者还需要确保考虑了不同的操作系统。同时需要有大量的备选字体以供浏览器最终可以找到能够满足的字体。

3.4 测试实际效果

设计者的设计结果最终要在用户的浏览器上展示效果, 所以需要再设计时进行充分的测试, 充分掌握选择的字体, 以及字体栈中的替代字体可能产生的表现。

References

- [1] <http://blog.typekit.com/2013/05/01/hi-dpi-typography/>.
- [2] <http://research.microsoft.com/en-us/projects/cleartype/>.
- [3] <https://www.typosetheque.com/articles/hinting>.
- [4] Bob Baumele. Improving appearance of arial font on the macintosh. 1999.
- [5] VoxATypI classification. http://en.wikipedia.org/wiki/voxatypi_classification.
- [6] Microsoft Corporation. Truetype core fonts for the web. 2001.
- [7] David Kadavy. *Design for Hackers: Reverse Engineering Beauty*. Wiley, USA, 2011.
- [8] Full library of typekit. <https://typekit.com/fonts>.
- [9] Jennifer Niederst Robbins. *Web Design in a Nutshell, Third Edition*. O'Reilly Media, Inc., 2006.

- [10] Mick Winters. How to choose fonts how to choose fonts(a practical guide to web-optimized typefaces).
- [11] 舒忠梅, 胡金柱, 左亚尧. 浅析 truetype 中的 hinting 原理及相关技术. 计算机应用研究, 7:25-26, 30, 1999.