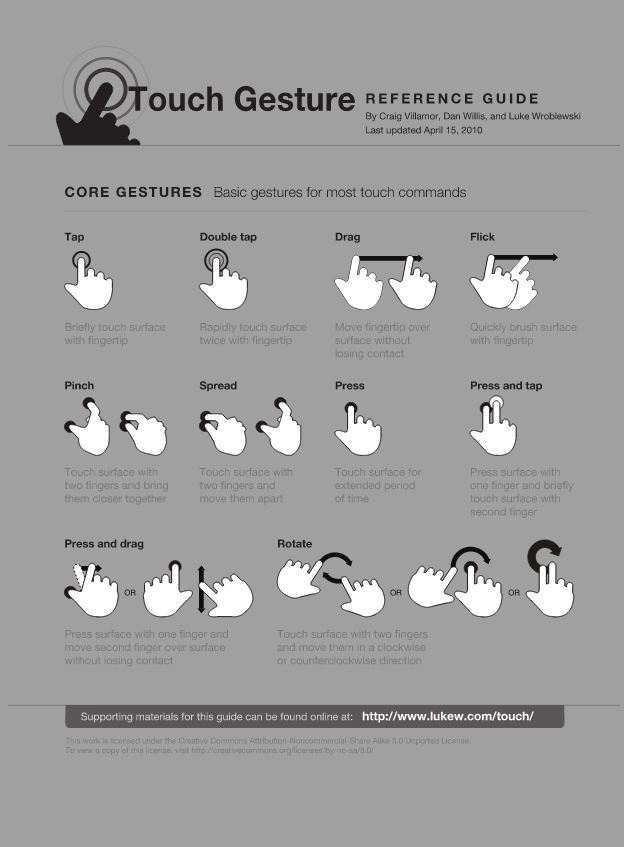
**변환 테스트 과정**

**SVG로 이미지 추출하기**

* pdf2svg로 변환합니다.  
  - 각 글자의 형상을(glyph) <defs> 태그내에 담고있습니다.  
  - 이를 use 태그에서 사용하도록 설계되어 있습니다.  
  - 나머지 그래픽은 일반 그리기 태그를 사용하여 드로잉 됩니다.  
  - 배경색은 빠져있는 상태입니다.  
  

(예)

<defs>  
 <g>  
 <symbol overflow="visible" id="glyph0-0">  
 <path style="stroke:none;" d="M 4.859375 -0.09375 L 0.640625 -0.09375 L 0.640625 -5.546875 L 4.859375 -5.546875 Z M 0.546875 -5.625 L 0.546875 0 L 4.953125 0 L 4.953125 -5.625 Z "/>  
 </symbol>  
 <symbol overflow="visible" id="glyph0-1">  
 <path style="stroke:none;" d="M 2.5 -6.40625 L 2.5 0 L 4.21875 0 L 4.21875 -6.40625 L 6.578125 -6.40625 L 6.578125 -7.859375 L 0.140625 -7.859375 L 0.140625 -6.40625 Z "/>  
 </symbol>  
 ……………………….  
 </g>  
 </defs>  
……………………….  
 <g id="surface1">  
 <g style="fill:rgb(13.729858%,12.159729%,12.548828%);fill-opacity:1;">  
 <use xlink:href="#glyph0-1" x="56.5581" y="232.5981"/>  
 <use xlink:href="#glyph0-2" x="63.2791" y="232.5981"/>  
 <use xlink:href="#glyph0-3" x="69.5931" y="232.5981"/>  
 </g>

</g>

**(참고) 구글 Swiffy 의 구조**

일반 그래픽의 드로잉 방식은 같으나 glyph 정보와 함께 폰트 정보를 함께 가지고 있어 Text 노드를 사용하여 텍스트를 표현할 수 있습니다.

<defs>  
 <defs>  
 <font horiz-adv-x="20480">  
 <font-face font-family="'WKIKGZ+HelveticaNeue-Light-313-0:0'" units-per-em="20480" ascent="14620" descent="4200" font-weight="normal" font-style="normal">  
 </font-face>  
 <glyph unicode="t" d="M 1980 13750 L 1980 10580 L 140 10580 L 140 9500 L 1960 9500 L 1960 2260 Q 1960 906 2471 403 Q 2983 -100 4273 -100 Q 4560 -100 4846 -79 Q 5133 -59 5420 -59 L 5420 1021 Q 4869 980 4319 980 Q 3626 980 3453 1340 Q 3280 1701 3280 2339 L 3280 9500 L 5420 9500 L 5420 10580 L 3270 10580 L 3270 13750 Z" horiz-adv-x="6060"/>  
 <glyph unicode="u" d="M 1200 10580 L 1200 3716 Q 1200 2774 1404 2036 Q 1609 1298 2049 786 Q 2489 274 3196 -3 Q 3902 -280 4906 -280 Q 6094 -280 7066 284 Q 8039 847 8551 1913 L 8592 1913 L 8592 0 L 9780 0 L 9780 10580 L 8489 10580 L 8489 5027 Q 8489 4228 8305 3460 Q 8120 2691 7730 2097 Q 7340 1503 6745 1144 Q 6150 786 5310 786 Q 3771 786 3145 1523 Q 2520 2261 2520 3695 L 2520 10580 Z" horiz-adv-x="10980"/>  
 </font>  
 <font horiz-adv-x="20480">  
 <font-face font-family="'WKIKGZ+HelveticaNeue-308-0:0'" units-per-em="20480" ascent="14990" descent="4280" font-weight="normal" font-style="normal"/>  
 <glyph unicode=" " d="M 0 0" horiz-adv-x="5700"/>  
 <glyph unicode="," d="M 1700 2280 L 1700 0 L 2911 0 Q 2932 -245 2880 -532 Q 2829 -819 2685 -1096 Q 2542 -1372 2305 -1608 Q 2069 -1844 1720 -1967 L 1720 -2991 Q 2911 -2643 3445 -1801 Q 3980 -960 3980 209 L 3980 2280 Z" horiz-adv-x="5700"/>  
 </font>  
 ……………………….  
 </defs>  
 </defs>  
 ……………………….  
  
 <g clip-path="url(#:24)">  
 <g clip-path="url(#:25)">  
 <text font-family="'WKIKGZ+HelveticaNeue-Bold-301-0:0'" font-size="674" x="0 411 823 1223 1610 2010 2197 2708 3095 3457 3695 4094 4356" y="0" fill-rule="nonzero" style="white-space:pre" unicode-bidi="bidi-override" transform="matrix(1 0 0 1 0 0) matrix(0.9614105224609375 0 0 1.0004730224609375 2400 2054)" fill="rgb(35,31,32)" fill-opacity="1.000">Touch Gesture</text>  
 </g>  
 </g>  
 <g clip-path="url(#:24)">  
 <g clip-path="url(#:25)">  
 <text font-family="'WKIKGZ+HelveticaNeue-Bold-301-0:0'" font-size="279" x="0 257 493 715 951 1208 1445 1708 1970 2207 2340 2608 2870 3008 3271" y="0" fill-rule="nonzero" style="white-space:pre" unicode-bidi="bidi-override" transform="matrix(1 0 0 1 0 0) matrix(1 0 0 1 7149 1908)" fill="rgb(35,31,32)" fill-opacity="1.000">REFERENCE GUIDE</text>  
 </g>  
 </g>  
 ……………………….

* 글자 그래픽 제거  
  따라서 글자에 해당되는 glyph 정보를 삭제하면 글자를 뺀 나머지 그래픽만 표시되게 됩니다.  
  일단은 <defs> 태그전체를 주석 처리한 화면입니다  
  
* glyph 심볼을 사용하는 태그 제거  
  glyph 심볼을 사용하는 태그를 제거한 후 비어있게 되는 태그까지 정리하였습니다.  
  정규 표현식을 사용하여 <use(.+)/> 를 <!--$&--> (에디트플러스에서는 <!--&-->)로 일괄 치환합니다.

**텍스트 추출하기**

폰트와 함께 폰트가 적용되는 텍스트 정보가 서로 링크되어있는 정보가 필요합니다.  
따라서 폰트만 추출된다든지 아니면 텍스트만 추출된다든지 하는 것은 의미가 없어 보입니다.

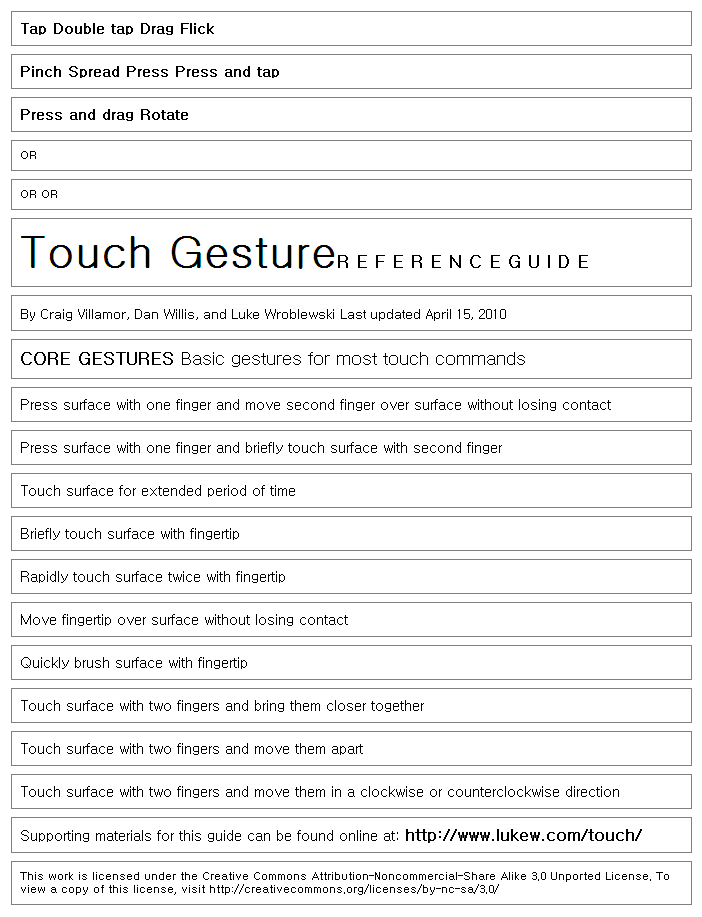
**pdf2json**

- 폰트 정보, 텍스트 정보가 서로 연결되어져서 추출됨  
- 폰트 파일이 추출이 되지 않음. 폰트 이름을 파일과 연결 지어줄 수 없음

**mupdf**

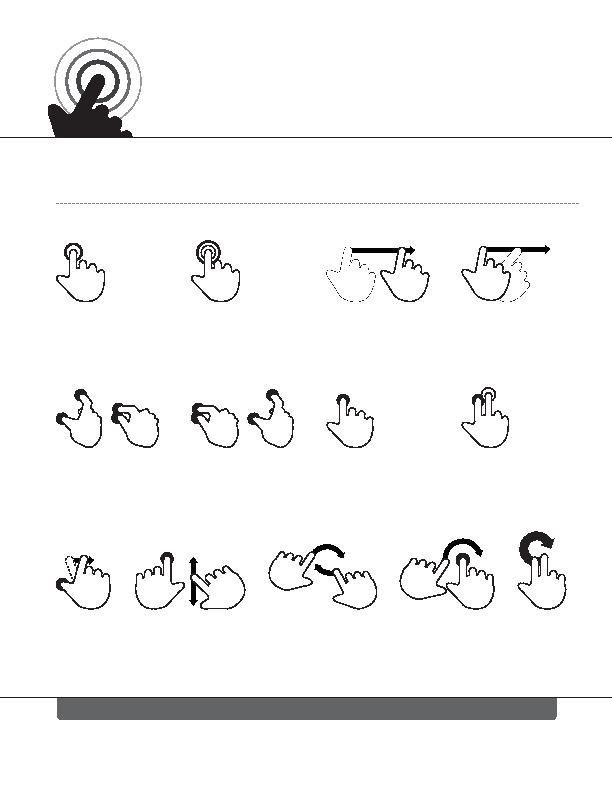
* 썸네일 추출 (mudrraw -r, -w, -h 옵션)  
  *mudraw –r 72 %d.png <pdf 파일 경로>*
* 검색 가능 텍스트 정보  
  mudrraw -t 옵션을 사용하여 ( mudraw 옵션결과(-t).txt ) 검색에 사용될 텍스트 정보를 추출합니다.  
    
  Tap Double tap Drag Flick  
  Pinch Spread Press Press and tap  
  Press and drag Rotate  
  OR  
  OR OR  
  Touch GestureR E F E R E N C E G U I D E  
  By Craig Villamor, Dan Willis, and Luke Wroblewski  
  Last updated April 15, 2010  
  CORE GESTURES Basic gestures for most touch commands
* 폰트 추출  
  폰트는 mutool.exe로 뽑아낼 수 있다.   
  폰트 이름은 추출된 폰트 파일명에서 하이픈(-) 다음 숫자를 제외한 이름이 된다.  
  (예) WKIKGZ+HelveticaNeue-Bold-0303.ttf 🡪 WKIKGZ+HelveticaNeue-Bold
* 폰트 파일와 텍스트간 연결 정보 및 화면 표시 정보  
  -tt 옵션을 사용하여 ( mudraw 옵션결과(-tt).txt ) HTML 블록 형식으로 출판되는 정보를 가공할 수 있습니다.   
  이 정보를 이용해 SVG text Element를 구성하거나 HTML Element를 추가하려면 위치정보가 필요한데 이 옵션에서는 위치정보는 나타나지 않습니다. 따라서 텍스트 배치에 문제가 있습니다.

<style>  
 body{background-color:gray;margin:12tp;}  
 div.page{background-color:white;margin:6pt;padding:6pt;}  
 div.block{border:1px solid gray;margin:6pt;padding:6pt;}  
 p{margin:0;padding:0;}  
</style>  
<style>  
 span.s10{font-family:"HelveticaNeue";font-size:8.03542pt;}  
 span.s9{font-family:"HelveticaNeue-Bold";font-size:12pt;font-weight:bold;}  
 span.s8{font-family:"HelveticaNeue";font-size:10.5pt;}  
 span.s7{font-family:"HelveticaNeue";font-size:11pt;}  
 span.s6{font-family:"HelveticaNeue-Light";font-size:14pt;}  
 span.s5{font-family:"HelveticaNeue-Bold";font-size:14pt;font-weight:bold;}  
 span.s4{font-family:"HelveticaNeue";font-size:9.45pt;}  
 span.s3{font-family:"HelveticaNeue-Bold";font-size:13.95pt;font-weight:bold;}  
 span.s2{font-family:"HelveticaNeue-Bold";font-size:33.0515pt;font-weight:bold;}  
 span.s1{font-family:"HelveticaNeue";font-size:8pt;}  
 span.s0{font-family:"HelveticaNeue-Bold";font-size:11pt;font-weight:bold;}  
</style>  
<body>  
 <div class="page">  
 <div class="block"><p>  
 <span><span class="s0">Tap Double tap Drag Flick</span></span>  
 </p></div>  
 <div class="block"><p>  
 <span><span class="s0">Pinch Spread Press Press and tap</span></span>  
 </p></div>  
 <div class="block"><p>  
 <span><span class="s0">Press and drag Rotate</span></span>  
 </p></div>  
 <div class="block"><p>  
 <span><span class="s1">OR</span></span>  
 </p></div>  
 <div class="block"><p>  
 <span><span class="s1">OR OR</span></span>  
 </p></div>  
 <div class="block"><p>  
 <span><span class="s2">Touch Gesture</span><span class="s3">R E F E R E N C E G U I D E</span></span>  
 </p></div>  
 <div class="block"><p>  
 <span><span class="s4">By Craig Villamor, Dan Willis, and Luke Wroblewski</span></span>  
 <span><span class="s4">Last updated April 15, 2010</span></span>  
 </p></div>  
 <div class="block"><p>  
 <span><span class="s5">CORE GESTURES </span><span class="s6">Basic gestures for most touch commands</span></span>  
 </p></div>  
 </div>  
……………….  
</body>



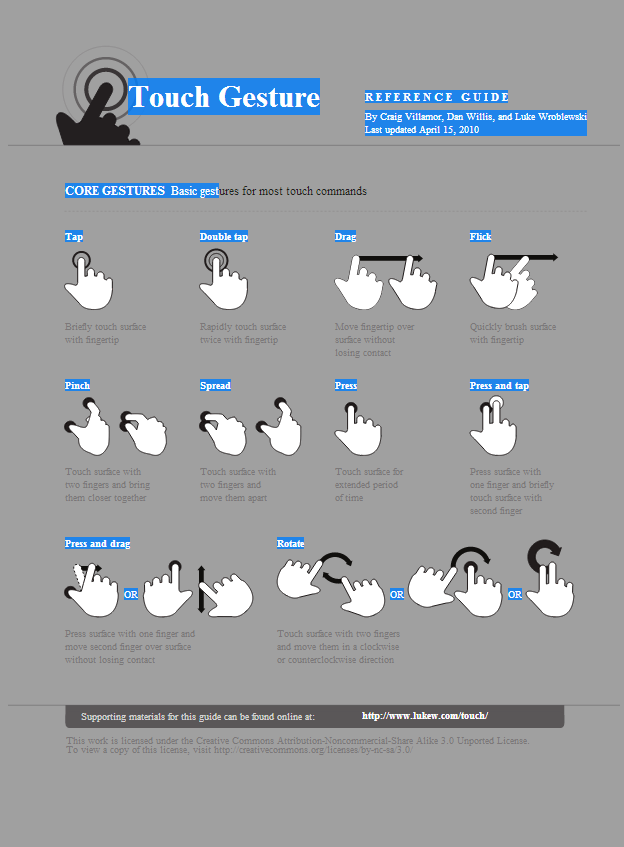
**pdftohtml**

pdf2html은 변환된 결과물이 배경 이미지와 HTML 텍스트로 분리됩니다.  
단점이라면 배경 이미지는 ghostScript를 이용해 뽑아낸 이미지로 품질이 매우 낮았습니다. 배경이 투명하지 않습니다. 또한 폰트파일을 추출할 수 없습니다.

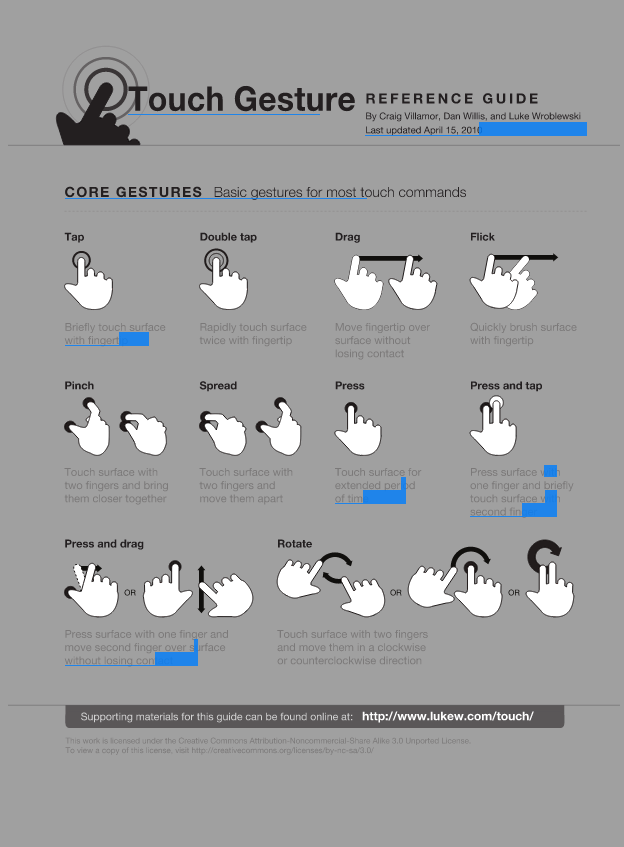


**HTML(변환된) 페이지 구성하기**

Pdf2html에서 생성된 파일에서 배경 이미지만을 제거한 후 pdf2svg에서 추출한 이미지(텍스트를 제거한)와 결합하면 고품질의 벡터 이미지와 텍스트 드래그 복사가 가능한 문서가 완성될 수 있습니다.



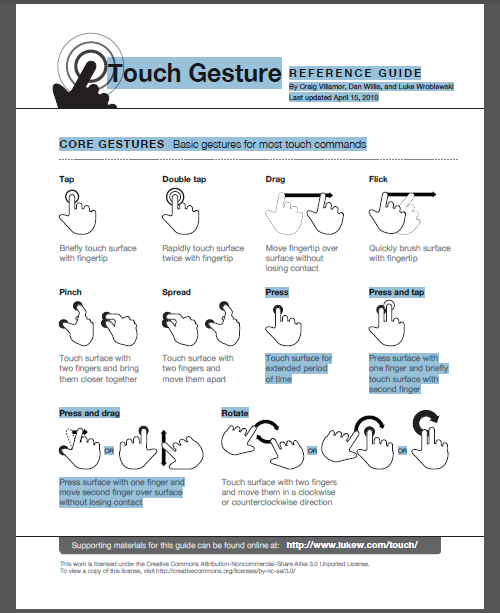
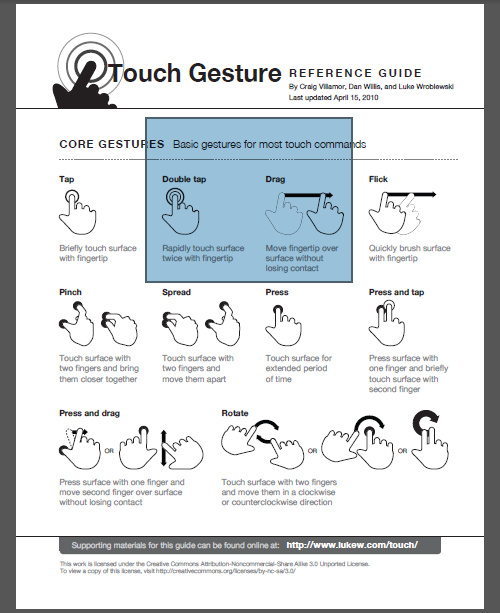
또한 텍스트를 제거하지 않은 Svg 이미지와 폰트 Opacity를 0으로 설정한 pdf2html의 결과를 합치면 폰트가 적용된 듯 보이는 문서를 얻을 수 있을 것 같았지만 (왼쪽) 선택 표시를 별도로 드로잉 해 주어야 하는 문제가 발생하였습니다.   
(선택영역 표시에도 Opacity가 적용되어 버립니다.)

오른쪽 이미지에서 보이는 것과 같이 폰트 적용에 따라 선택 영역이 일치되지 않는 문제도 있습니다.

* 문제  
  텍스트 - 텍스트 위치 정보 - 폰트 스타일 - 폰트 파일 이 네 가지 정보를 연결 시켜주는 것이 포인트
* TODO : 폰트 파일을 수동으로 연결해 주어 드로잉 화면이 일치하는지 테스트해본다.
* 테스트가 성공하면 HTML 고유의 선택기능을 사용할 수 있고 편집도 보다 쉽게 가능할 것 같다.
* 폰트 테스트 결과mutool을 이용하여 폰트를 추출할 수는 있으나 텍스트 배치는 어렵습니다.  
  브라우져 별로 폰트를 지정하는 방법이 다르고 로컬 사용자일 때에는 적용되지 않는 것 같습니다.   
  (웹에 올려놓은 폰트는 적용되나 로컬 파일로 지정된 폰트는 적용안됨)  
   @font-face{  
   font-family:'AlexBrush';  
   src: url('http://sseckun.zc.bz/test/AlexBrush-Regular-OTF.otf') format('truetype');  
   /\*  
   src: url('http://sseckun.zc.bz/test/AlexBrush-Regular-OTF.eot');  
   src: url('http://sseckun.zc.bz/test/AlexBrush-Regular-OTF.eot?iefix') format('eot'),  
   url('http://sseckun.zc.bz/test/AlexBrush-Regular-webfont.woff') format('woff'),  
   url('http://sseckun.zc.bz/test/AlexBrush-Regular-webfont.ttf') format('truetype'),  
   url('http://sseckun.zc.bz/test/AlexBrush-Regular-webfont.svg#webfontysz32sPA') format('svg');  
   \*/  
   }  
   span.s2{font-family:'AlexBrush';font-size:33.0515pt;font-weight:bold;}
* 결론은 폰트를 파일로 추출해 사용하지 말고 svg glyph 정보를 이용하는 편이 좋습니다.  
  대신 글꼴 편집은 될 수 없습니다.

**텍스트 선택 기능 구현**

* 영역 지정에 의한 텍스트 선택 방식   
  - 텍스트 및 텍스트 위치 정보는 쉽게 구할 수 있으니 SVG 파일에 텍스트 선택 기능을 추가하는 것도 하나의 방법일 수도 있을 것 같습니다. (아래 이미지는 PDF 문서에서 텍스트 선택하는 방식임)  
   
* sgv에서 제거된 텍스트 정보를 활용한다. 텍스트 레이어로 따로 분리합니다.  
  (svg 원본에서 정규 표현식을 사용하여 <path\s.+/> 를 <!--$&--> (에디트플러스에서는 <path .+/>를 <!--&-->)로 일괄 치환합니다.)  
  

**TODO :**

* TODO : mudraw 옵션결과(-ttt) 와 mudraw 옵션결과(-x) 를 분석하여 직접 위치 계산 및 드로잉을 해보아야 할 것 같음  
  이 파일에는 폰트 정보, 영역 크기 및 위치, 라인 단위 및 한글자 단위로 정보가 구성되어 있다.  
  이 정보로 드로잉한 텍스트가 svg 이미지와 크기가 일치하여야 한다.
* 테스트가 성공하면 성공한 결과대로 텍스트를 드로잉하여 보여줄 수 있고 위치 및 영역 크기에 대한 데이터도 있으므로 선택표시도 해줄 수 있을 것 같다.

**pdf.js 참고**

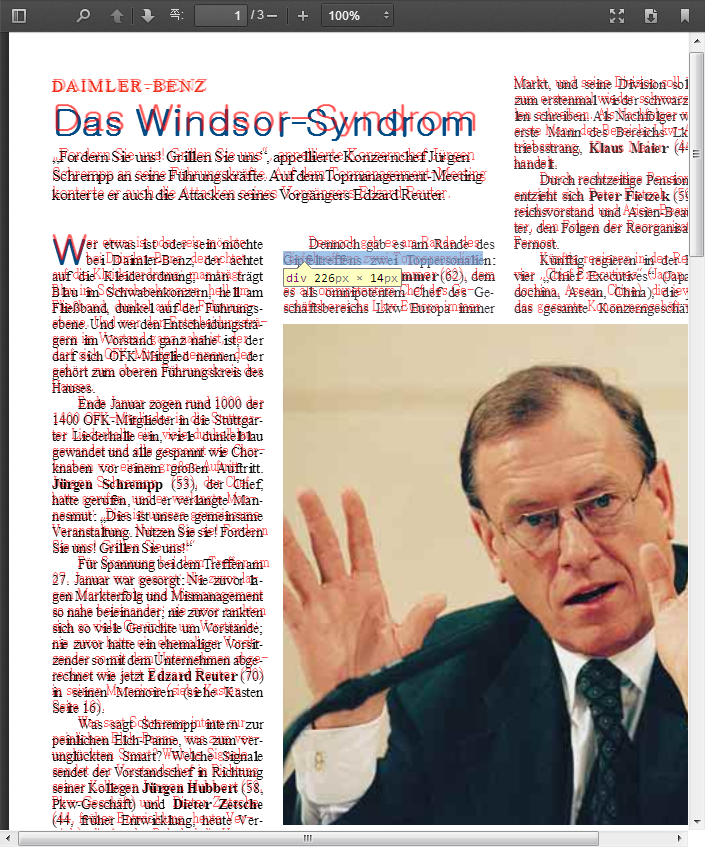
<https://github.com/mozilla/pdf.js>



< 원본 PDF 문서에서의 텍스트 선택(좌)과 Pdf.js 를 이용한 변환시 텍스트 선택(우)의 차이 >

차이의 원인은 실제 텍스트는 투명하게 처리되어있고 보이는 텍스트는 이미지이기 때문인데 투명한 텍스트에는 폰트가 적용되지 않아 장평, 자간등에 차이가 생기게 되므로 보여지는 선택 영역에서도 차이가 생기게 된다.

아래 이미지는 선택 기능을 위해 투명 처리된 글자를 빨간색으로 표기한 것이다. 위치 및 모양, 크기가 보여지는 텍스트와 약간의 차이가 있다.



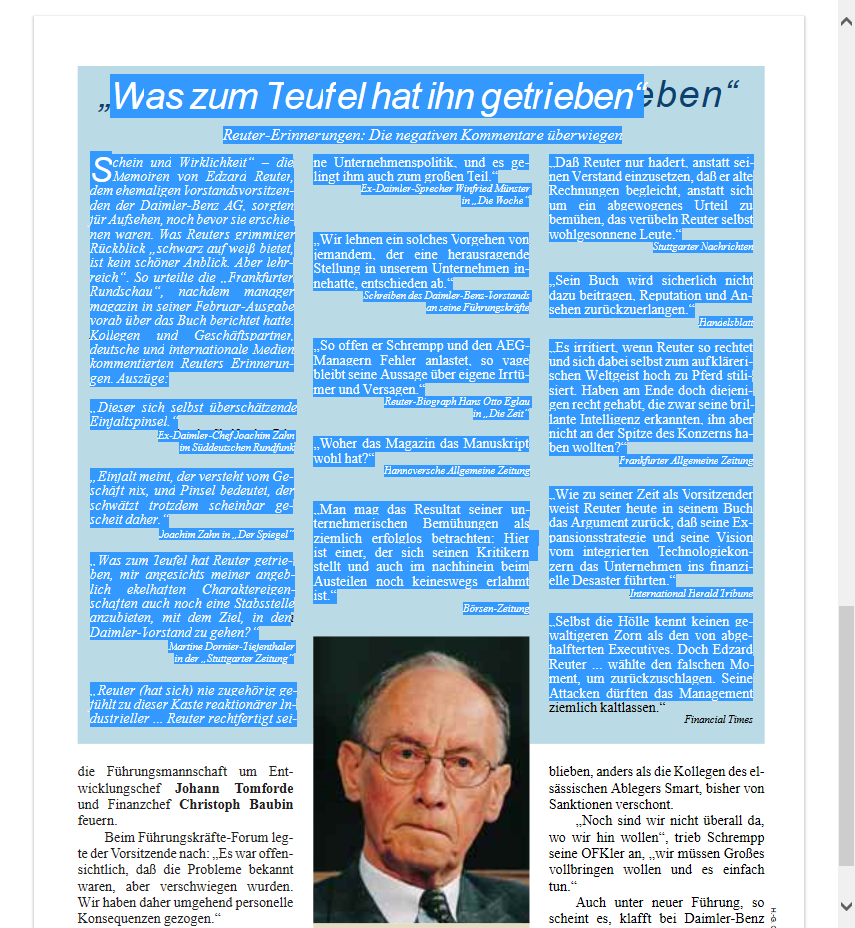
 

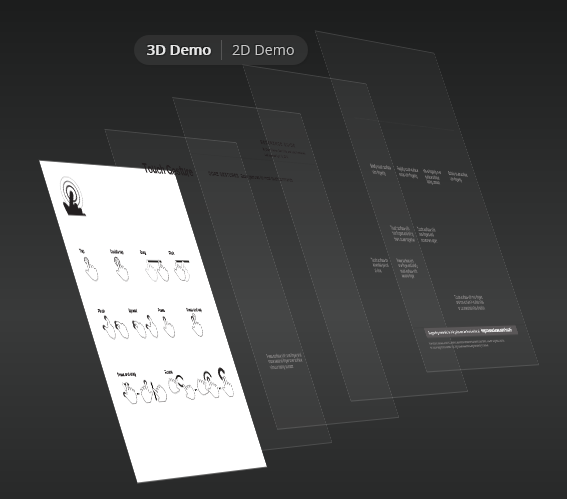
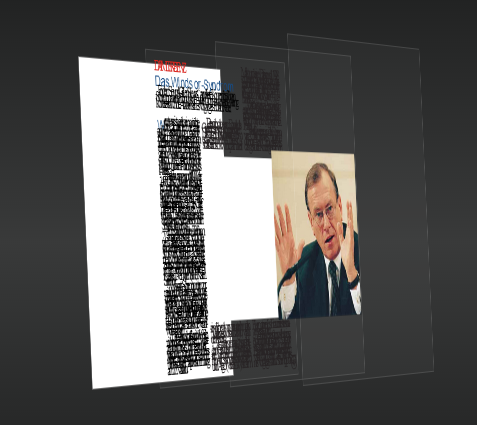
**Crocodoc.com 참고**

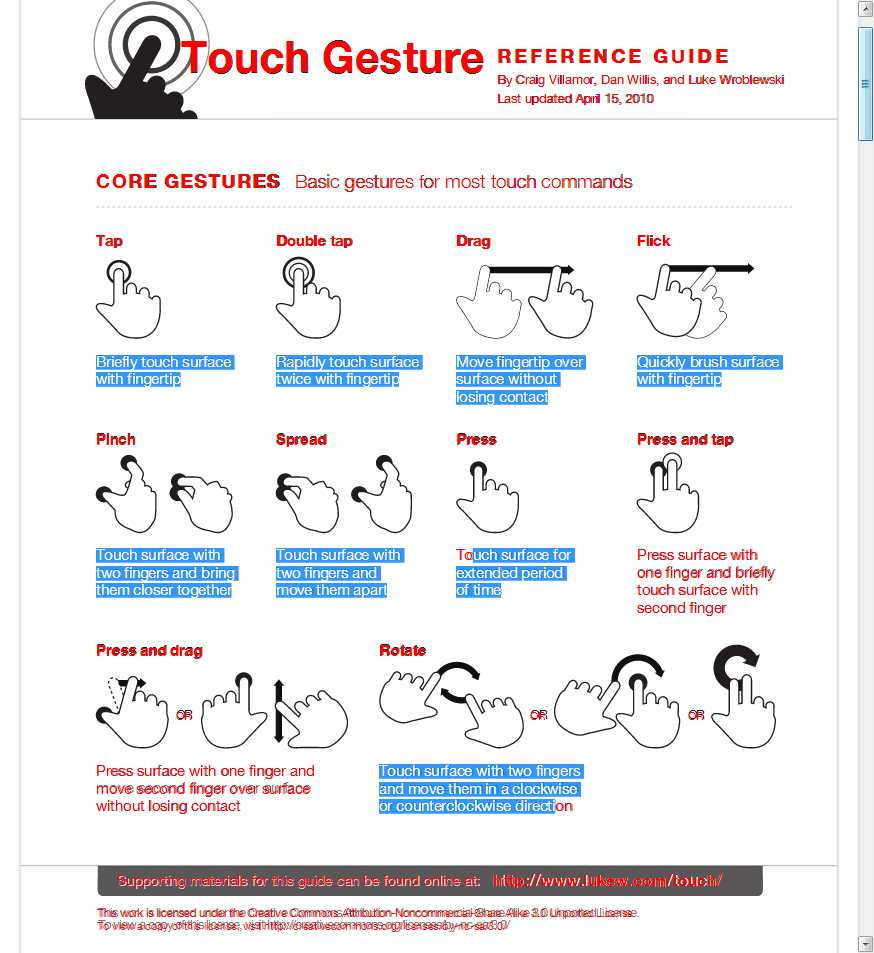
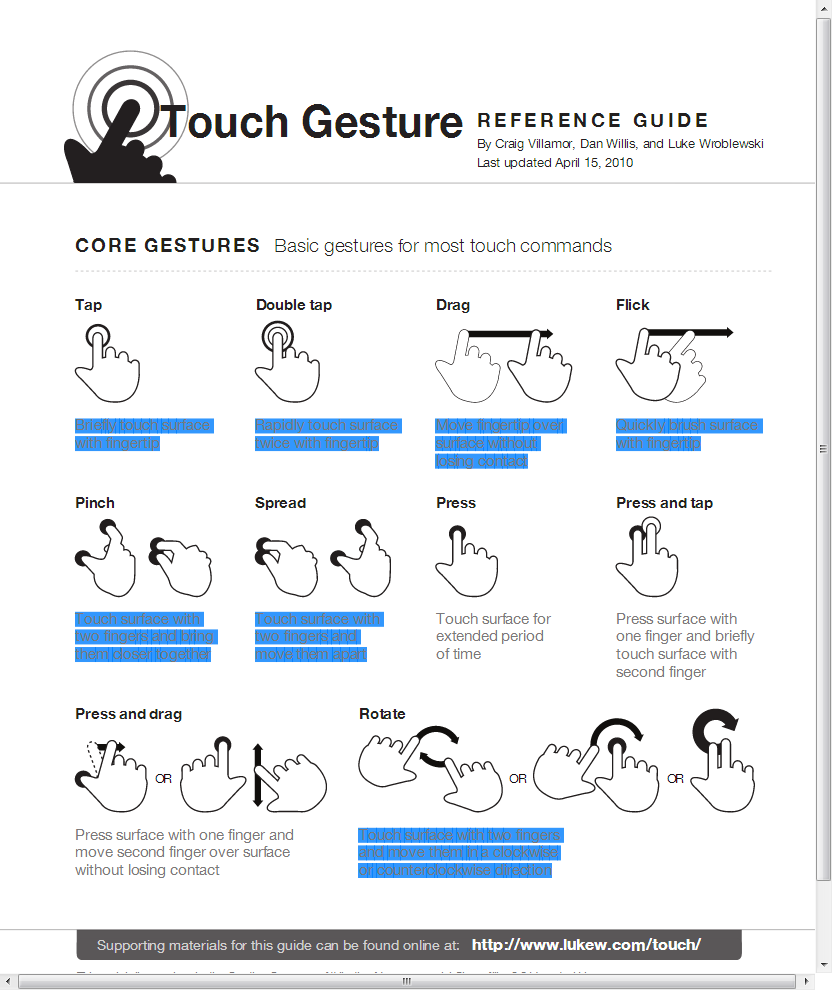
<https://preview.crocodoc.com/>

<https://preview.crocodoc.com/> 에서 pdf문서를 업로드하고 변환된 결과물을 다운받을 수 있다.

역시 폰트 문제를 해결하지 못하고 있는 듯 하다.

    
  
원본 PDF (좌), 웹에서 본 변환 문서(우)  
   
웹에서 본 문서(svg + div) - 빨간색 글자가 숨겨져 있는 텍스트  
  
다운받은 문서 (우, svg, text) – svg에 text shem 만으로 구성되어 있다. 따라서 선택영역이 보이는 부분과 일치 한다.



   
svg + div 텍스트 (좌), svg 텍스트(우)  
  
**poppler 0.22 (pdftohtml)**

poppler 0.22 버전에 있는 pdftohtml에서는 폰트가 추출된다.

스타일시트에서 @fontface로 연결만 시켜주면 된다. 다만 로컬에서 폰트 적용이 잘 안되는 것 같다. (corcodoc.com, idrsolutions.com 의 문서는 폰트가 잘 적용되어 있으니 참고 할것)

<http://blog.alivate.com.au/poppler-windows/>

**pdf2htmlEX**

**TODO : 이미지 요소 분리하여 추출하기 (테스트)**

**SVG 참고**

<http://msdn.microsoft.com/ko-kr/library/ie/bg124132(v=vs.85).aspx>