

NATE CP 가이드

1. WML 페이지에서 줄바꿈 문제

- 개요

WML 페이지에서 줄바꿈은 명시적으로
 태그를 사용하여야 한다.

- 설명

그림2의 의도로 제작된 페이지에서 Card Styles 텍스트와 Select 박스 사이에 명시적인
 태그가 없어, 그림1과 같이 줄바꿈이 되지 않고 한 줄로 보여짐.

- 예제



[그림1]비정상



[그림2]정상

- sample

```
<card id="main">
  <p mode="nowrap">
    Card styles
    <select>
      <option onpick="#f1">Ordered False</option>
      <option onpick="#f2">Ordered True</option>
    </select>
    <do type = "vnd.up">
      <go href = "test1.wml" />
    </do>
  </p>
</card>
```

- sample(수정)

```

<card id="main">
  <p mode="nowrap">
    Card styles
    <br/> <- 명시적인 br 추가 필요
    <select>
      <option onpick="#f1">Ordered False</option>
      <option onpick="#f2">Ordered True</option>
    </select>

    <do type = "vnd.up">
      <go href = "test1.wml" />
    </do>
  </p>
</card>

```

2. WML 테스트페이지에서 문자열이 표시되지 않고 잘리는 현상.

- 개요

WML p 태그의 Mode는 문자열 길이가 길 경우 문장을 자르거나, 자르지 않는 것을 결정한다.

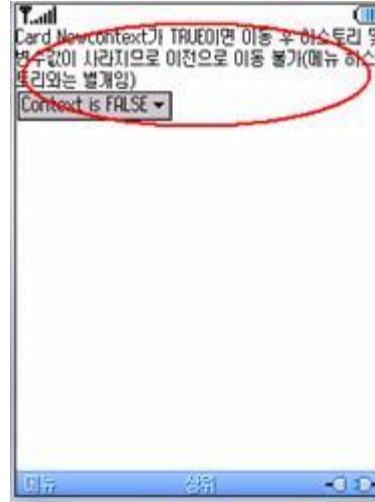
- 설명

<p mode="wrap" | "nowrap">에 따라 문자열의 길이가 길 경우 wrap mode는 문자열을 아래로 자르는 것을, nowrap mode는 문자열을 자르지 않음

- 예제



[그림3] nowrap mode



[그림4] wrap mode

- sample

```

<wml>
<card id="main">
  <p mode="nowrap">
    Card Newcontext가 TRUE이면 이동 후 히스토리 및 변수값이 사라지므로 이전으로 이동 불가(메뉴 히스토리와는 별개임)
  <select>
    <option onpick="#f1">Context is FALSE</option>

```

```

<option onpick="#f2">Context is TRUE</option>
</select>

<do type = "vnd.up">
    <go href = "test1.wml" />
</do>

</p>
</card>
.... 이하 생략함.
</wml>

```

3. WML 페이지에서 DTD 문법 오류 발생함.

- 개요

WML 페이지에서 DTD 문법 오류 발생

- 설명

<u><strike>strike</strike></u>를 지원하지 않아 DTD 오류 발생함.

DocType이 XHTML Basic이면 WML20으로 보기 때문에 DocType을 다음과 같이 XHTML MP로 바꾸면 DTD Error는 나지 않음.

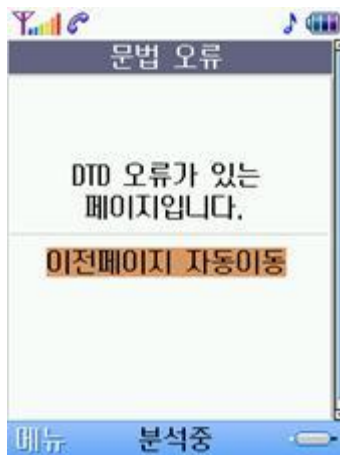
```

<?xml version="1.0" encoding="ks_c_5601-1987"?>
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile
1.1//EN" "http://www.wapforum.org/DTD/xhtml-mobile11.dtd">

```

<u><strike>strike</strike></u>의 결과가 strike 와 같이 나오지 않는 것에 대해서는 NGB3.0에서도 지원하지 않았던 부분인데, 오류가 발생하지 않도록 처리 하였으나 NGB4.0에서는 지원하지 않는 태그에 대한 오류 발생되도록 함. 따라서 strike>태그는 페이지에서 삭제 수정되어야 함.

- 예제



[그림5] NGB4.0(E5)



[그림6] NGB3.0(E4)

- sample

```

<?xml version="1.0" encoding="ks_c_5601-1987"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
"http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<!-- SKTelecom NGB test page. by RoCode Inc.-->
<html>

```

```

<body>

```

```

<div>[설명:u ]<br/>underline</div>
<div>[시험]<br/>
<br/>

<u>

<a href="./testpage.xhtml">링크</a><br/>
<abbr>abbr</abbr><br/>
<acronym>acronym</acronym><br/>
<b>b</b><br/>
<big>big</big><br/>
<cite>cite</cite><br/>
<code>code</code><br/>
<dfn>dfn</dfn><br/>
<em>em</em><br/>
<i>i</i><br/>
<br/>
<kbd>kbd</kbd><br/>
<samp>samp</samp><br/>
<small>small</small><br/>
<del>strike</del><br/>
<strong>strong</strong><br/>
<var>var</var><br/>

</u>

</div>
<div>
    <a href="tr.xhtml">이전</a>|
    <a href="index17.xhtml">목록</a>
</div>

</body>
</html>

```

4. a>태그에서의 밑줄 표시되는 현상.

- 개요
 - <a> 태그에 underline이 생기는 현상
- 설명
 - WEB(HTML/XHTML MP) 규격의 지원으로 WEB 상의 <a> 태그의 default는 underline 이 그려짐
 - NGB3.0의 WAP(WML1.x/WML2.x/ XHTML basic)규격은 <a>에 대한 Under Line표시 여부가 규격상 정의 되어있지 않아 NGB3.0에서는 표시하지 않고 있음
- 예제



[그림7] NGB3.0(E4)



[그림8] NGB4.0(E5)

- sample

<a> 태그에 밑줄이 생기지 않도록 하기 위해서는 아래와 같이 <a> 태그의 style 속성에 “text-decoration:none”을 주면 <a> 태그에 밑줄이 생기지 않음
 TEST

5. URL Link에 full url 쓰는 문제

- 개요

각 태그의 URL link에 full url 보다는 server name 을 빼고 상대 path 와 file name 만 사용한다.

- 설명

anchor tag 또는 여러 번 반복적으로 사용되는 Image File의 url link에 full url을 쓸 경우 속도 저하의 요인이 된다.

- sample

```
<body>
<a href="http://mobile.embider.com/dev/testpage.html"> full url </a>

</body>
```

- sample(수정)

```
<body>
<a href="testpage.html"> full url </a>

</body>
```

6. Image File 갯수 문제

- 개요

Image 파일의 개수를 최대한 줄이도록 한다.

- 설명

자주 변경되는 Image가 아니라면 위치상으로 떨어져 있는 이미지라도 background 속성을 사용해 하나로 만들어 잘라 붙이도록 한다. Image File의 개수가 많은 것은 속도 저하의 요인이 된다.

아래 [그림 9]같은 경우 쉽게 Image 파일이 두 개가 나올 수 있는 것이지만, 고정 영역이기 때문에 하나로 만든 뒤 [그림 10]처럼 표현하였다.

- 예제



[그림9]Image File



[그림10]적용된 콘텐츠 화면

- sample : Image File 3개

```



```

- sample(수정) : Image File 2개(1개-고정, 1개-가변)

```
<div style="height:25;background-image:URL(image/photo_title.png);">
    PH000012
</div>

<div style="height:19;background-image:URL(image/photo_title.png);
    background-position-x:0;background-position-y:-25;">
</div>
```

7. Table 태그보다는 Div 태그 사용을 권장

- 개요

table 태그는 연산량 및 예외 동작이 많기 때문에 가급적 사용을 자제해야 한다.

- 설명

Table 태그는 내부적으로 연산량이 많고, 예외처리가 많다. 따라서 table을 굳이 안 쓰고 간단히 해결할 수 있다면 table태그의 사용을 피하는 것이 좋다.

- sample

```
<table>
```

```
<tr><td>(0,0)</td><td>(1,0)</td></tr>
<tr><td>(0,1)</td><td>(1,1)</td></tr>
</table>
```

- sample(수정)

```
<div>
  <span>(0,0)</span>&nbsp;<span>(1,0)</span><br>
  <span>(0,1)</span>&nbsp;<span>(1,1)</span>
</div>
```

8. 코드 유지보수를 위한 주석은 제거할 것

- 개요

주석은 페이지의 용량을 증가시킬 뿐만 아니라, 불필요한 연산을 필요로 하므로 실제 서비스 컨텐츠에서는 주석을 제거하도록 한다.

- 설명

일반적으로 컨텐츠 상에 유지보수를 위해 주석(<!-- -->)을 다는 것이 정설이다. 하지만, 모바일 환경에서는 이런 주석들이 차지하는 용량자체도 신경을 써야 한다. 또한 주석도 연산과정에서 예외로 처리되기 때문에 성능에 영향을 미치게 된다. 따라서 개발 단계가 아닌 실제 서비스에 들어가기 전에 컨텐츠 상에서 불필요한 주석은 모두 제거하도록 한다.

- sample

```
<script>
<!--
  Document.writeln("Test");
-->
</script>
<!-- Link들 -->
<a href="http://wap.nate.com">NATE</a><br>
```

- sample(수정)

```
<script>
<!--
  Document.writeln("Test");
-->
</script>
<= 불필요한 주석 제거
<a href="http://wap.nate.com">NATE</a><br>
```

9. CSS Selector가 적용되는 tag가 많은 경우는 신중히

- 개요

inline 형식의 style sheet 대신 selector로 처리할 경우 페이지 로딩 속도에 영향을 줄 수 있다.

- 설명

inline(tag 상에 style attribute로 쓰는 방법)형식의 style sheet 대신 selector를 사용할 경우 text의 용량은 줄어 드는 대신 tag들이 많아지면 연산 속도가 급격히 떨어질 수 있다. inline 으로 1차 작업을 하고 selector 로 변경해서 속도가 떨어지지 않는지 확인 후 사용해야 함

selector 를 써서 용량을 줄인 효과로 다운로드 속도가 개선 되지만 역효과로 연산속도가 느려지기 때문에 항상 확인 후 사용해야 함

- sample

```
<style type=css/text>
.myStyle
{
  background-color:#FF0000;
  width:100;
  height:100;
}
</style>
..생략..
<div class=myStyle>...</div>
<div class=myStyle>...</div>
...생략..
<div class=myStyle>...</div>
```

- sample(수정)

같은 스타일의 div가 몇 개가 있는지에 따라 inline스타일이 빠를 수도 있고, selector 를 쓰는 방법이 빠를 수도 있다.

```
<div style="background-color:#FF0000;width:100;height:100">...</div>
<div style="background-color:#FF0000;width:100;height:100">...</div>
...생략..
<div style="background-color:#FF0000;width:100;height:100">...</div>
```

10. Image 데이터는 미리 접근해둔다.

- 개요

페이지 내에 이미지가 사용될 경우, 가능한 페이지 상단에 이미지를 사용하는 코드를 넣어두면 페이지 로딩 속도가 개선될 수 있다.

- 설명

이미지는 상대적으로 용량이 큰 편이다. 따라서 미리 이미지 데이터를 망(network)에 요청해 놓고, 이미지 데이터를 기다리는 동안 페이지의 분석을 진행할 수 있도록 구성해 주면 사용된 이미지의 용량 및 개수에 따라 페이지 로딩 속도가 개선될 수 있다.

- sample

```
<html>
<body>
  
  
</body>
</html>
```

- sample(수정)

실제로 이미지가 필요한 부분보다, 미리 더미(dummy)로 해당 이미지를 접근한다. 예제에 사용된 dummyImage는 단순히 이미지 데이터 요청을 위해 임시로 생성할 뿐, 실제로 사용되지는 않는다.

```
<html>
<script>
```



```

Var dummyImage=new Image();
dummyImage.src="image/test.png";
dummyImage.src="image/test2.png";
</script>
<body>
  
  
</body>
</html>

```

11. 동적인 Mobile Flash는 Div안에 넣어서 사용하도록 한다.

- 개요

Mobile Flash를 사용하는 경우 object 태그를 div 태그로 다시 한번 감싸줄 것을 권장한다.

- 설명

Mobile Flash 스스로가 다시 그려질(draw) 필요가 있을 경우, 그와 이웃한 sibling들도 다시 그려지는 경우가 있다. 이때 object 태그와 sibling관계의 있는 요소들도 다시 그려지는 것은 불필요한 연산이다. 따라서 object태그만을 담고 있는 div태그를 만들어 불필요한 연산을 줄일 수 있다.

- sample

아래의 경우 <div> 태그에 둘러싸인 <a> 태그들도 Mobile Flash 자체의 변경 때문에 다시 그려질 수 있다.

```

<object classid=clsname:mfinlinerender2 data=flash/m.dmf type=application/plugin />
</div>
  <a href="">object의 sibling의 child입니다.</a>
  <a href="">object의 sibling의 child입니다.</a>
  <a href="">object의 sibling의 child입니다.</a>
</div>

```

- sample(수정)

아래와 같이 object 태그의 sibling이었던 div를 object와 분리함으로서 문제를 해결할 수 있다.

```

<div>
  <object classid=clsname:mfinlinerender2 data=flash/m.dmf
  type=application/plugin />
</div>
<div>
  <a href="">object의 parent의 sibling의 child입니다.</a>
  <a href="">object의 parent의 sibling의 child입니다.</a>
  <a href="">object의 parent의 sibling의 child입니다.</a>
</div>

```