

Mobile Web Cap. III



Fabio Paccosi

Giammarco Verdini

Eugenio Paluello

Valerio Belli

Vincenzo Brizzi

Nel design di un sito web, se si ha la necessità di implementare un Design Responsive, bisogna evitare di creare CONTEMPORANEAMENTE la parte Desktop e Mobile del sito. In generale è preferibile creare queste parti separatamente, preferendo un approccio Mobile-First

Lo Scenario Iniziale

Creature Comfort International è un'agenzia no-profit operante in tutto il mondo, che si occupa di aiutare a curare animali feriti o malati nelle aree colpite da calamità naturali e fornisce sostegno agli agricoltori e allevatori colpiti.

Ha un sacco di infrastrutture interne ed una significativa presenza sul Web "tradizionale" costruito su un **sistema proprietario di gestione dei contenuti** (CMS).

Il Problema

Fino a poco tempo fa l'organizzazione si basava su comunicazioni vocali.
I loro agenti come possono scaricare e condividere le informazioni di cui hanno bisogno?

L'agenzia ha persone in tutto il mondo che necessitano di rimanere in contatto mentre sono sul campo.

Nonostante abbiano già diversi tools sul loro sito web non li possono sfruttare appieno perché il sito non funziona a dovere su Mobile.

Una crescente necessità per il webMobile

Sempre più spesso il personale della Creature Comfort riscontra che, mentre si è sul campo, la più affidabile e spesso unica connessione disponibile è attraverso la rete del telefono cellulare.

**Creature Comfort ha bisogno di un sito web per cellulari:
un sito che può essere supportato da una vasta gamma di
dispositivi ed un'altrettanto vasta gamma di connessioni**

Come si fa a cambiare e modificare solo una parte del sito?

In questo caso si ha bisogno di sviluppare un sito separato per gli utenti Mobile.

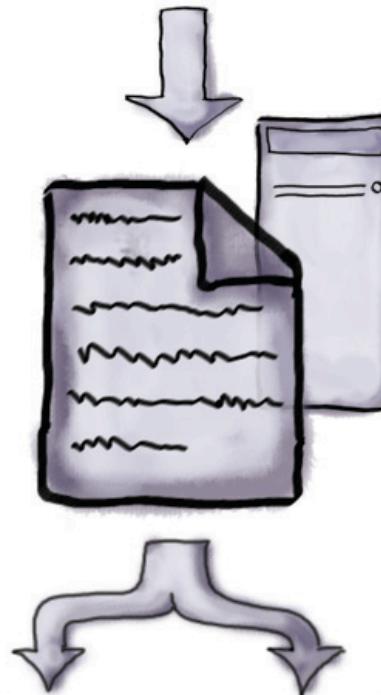
Siccome i telefoni che la maggior parte del personale dell'agenzia ha a disposizione non sono di ultima generazione e le connessioni che si riescono ad ottenere sul campo non sono così buone c'è bisogno di realizzare un sito snello, semplice e funzionale che aiuti queste persone a fare un buon lavoro

Inviare gli utenti mobili al sito strutturato per mobile

Spesso è utile avere il sito per utenti Mobile separato da quello per utenti Desktop, la necessità che si ha però è

Reindirizzare gli utenti mobile al sito apposito per i loro dispositivi quando vanno in quello Desktop. Ad esempio da [dominio.com](#) a [m.dominio.com](#)

Differenti tipi di dispositivi
e di browsers inviano
una richiesta al nostro
www.dominio.com



Uno script sul web Server
si occuperà di verificare
se tra le richieste ricevute
ce ne sia qualcuna da
dispositivo mobile



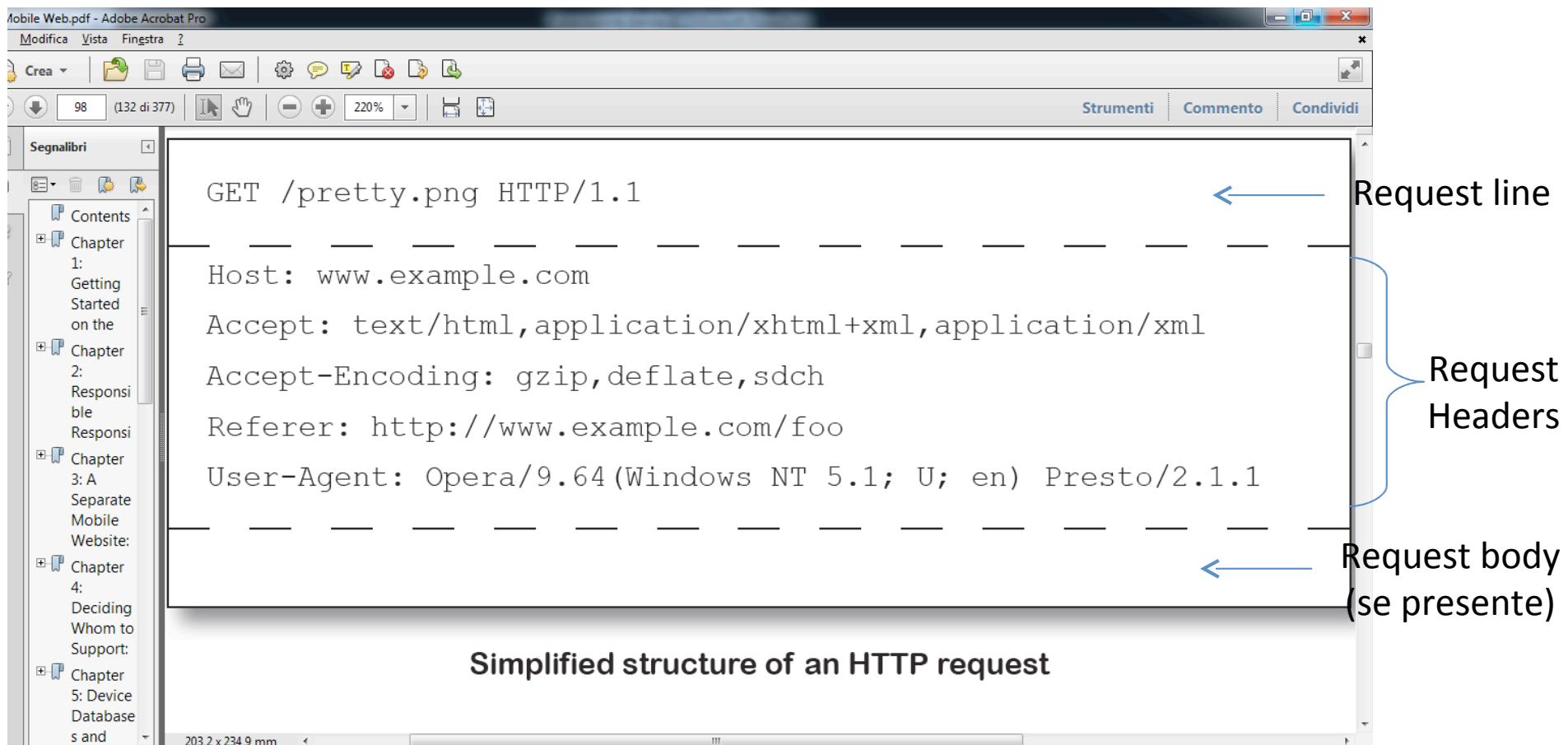
Gli utenti
mobile verranno
indirizzati qui

m.dominio.com

www.dominio.com Gli utenti desktop
Staranno nella parte
Desktop-Oriented del sito

... Gli USER AGENTS

I browser web (compresi quelli mobile) inviano un UA header (intestazione user agent) come parte della richiesta HTTP ogni volta che il browser comunica con il server (per ottenere una pagina o una qualsiasi altra risorsa web)





So, if we use user-agent sniffing to figure out which devices are mobile, isn't that just enforcing the same old bad behavior that created the user-agent mess in the first place?

Il rilevamento del browser attraverso lo user-agent sniffing Rilevamento browser mobili da sniffing può sembrare un trucco molto poco elegante e impreciso, ma nonostante sia uno strumento "grezzo" risulta essere lo strumento migliore.

Esempi di stringhe user-agent

Mozilla/5.0 (X11; U; Linux i686; it; rv:1.9.0.10) Gecko/2009042513 Ubuntu/8.04(hardy) Firefox/3.0.10
(Firefox su Ubuntu 8.04)

Mozilla/5.0 (Macintosh; U; PPC Mac OS X; en-us) AppleWebKit/412 (KHTML, like Gecko) Safari/412
(Safari su Mac osx)

Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.56
Safari/535.11
(Google Chrome su windows 7)

Separare i siti web

La maggior parte dei siti web più importanti hanno un sito mobile separato e, per instradarci il traffico all'interno usano lo user-agent sniffing.

L' user-agent header, è l'indizio più semplice e affidabile di cui il server dispone sul richiedente. Ci sono anche altri tipi di HTTP request headers ma nessuno così presente.



OK. Say we do build a separate mobile site. How exactly are we going to see the user agents of requesting browsers...and how are we going to get the mobile traffic to the new, mobile version of Creature Comforts?



Wouldn't such a script need to go on *Creature Comforts'* existing (desktop) web server so it can detect mobile browsers and redirect them to our mobile site? I thought we couldn't make any changes to the existing site.

www.detectmobilebrowsers.com

A questo link troviamo degli script che ci permettono di distinguere un browser mobile

A screenshot of a website titled "Detect Mobile Browsers | Open source mobile phone detection". The page displays a message: "No mobile browser detected." Below this, there's a list of platforms: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_2) AppleWebKit/534.30 (KHTML, like Gecko) Chrome/12.0.740.0 Safari/534.30. A "Download Scripts" section follows, featuring several buttons for different technologies: Apache, ASP, ASP.NET, ColdFusion, C#, IIS, JSP, JavaScript, jQuery, nginx, Perl, Python, and Rails. A large blue button labeled "PHP" is highlighted with a magnifying glass. A curved arrow points from the text "The script is available for many different languages and platforms." to the "Download Scripts" section. Another curved arrow points from the text "We want the PHP version." to the "PHP" button.

The script is available for many different languages and platforms.

We want the PHP version.

```

<?php
 1 $useragent=$_SERVER['HTTP_USER_AGENT'];
 2 if(preg_match('/android|avantgo|blackberry|blazer|compal|elaine|fennec|hiptop|iem
    obile|ip(hone|od)|iris|kindle|lge_|maemo|midp|mmp|opera m(ob|in)i|palm( os)?|phon
    e|p(ixi|re)|plucker|pocket|psp|symbian|treo|up\.(browser|link)|vodafone|wap|wi
    ndows (ce|phone)|xda|xino/i', $useragent))|| preg_match('/1207|6310|6590|3gso|4thp
    |50[1-6]i|770s|802s|a wa|abac|ac(er|oo)s\(|ai(ko|rn)|al(av|ca|co)|amoi|an(ex|ny
    |yw)|aptu|ar(ch|go)|as(te|us)|attw|au(di|\-m|r |s )|avan|be(ck|ll|nq)|bi(lb|rd)
    |bl(ac|az)|br(e|v)w|bumb|bw\-(n|u)|c55|\|capi|ccwa|cdm\|-|cell|chtm|cldc|cmd\|-|co
    mp(nd)|craw|da(it|ll|ng)|dbte|dc\|s|devi|dica|dmob|do(c|p)o|ds(12|\|d)|el(49|ai)
    |em(12|ul)|er(ic|k0)|esl8|ez([4-7]0|os|wal|ze)|fetc|fly(\|-_)|g1 u|g560|gen|gf\-
    5|g\|mo|go\(.w|od)|gr(ad|un)|haie|hkit|hd\-(m|p|t)|hei\|-|hi(pt|ta)|hp( i|ip)|hs\-
    c|ht(c\(\|-|_|_|\a|g|p|s|t)|tp)|hu(aw|tc)|i\-(20|go|ma)|i230|iac( |\-\|\\)|ibro|idea
    ig01|ikom|im1k|inno|ipaq|iris|ja(t|v)a|jbro|jemu|jigs|kddi|keji|kgt( |\|/)|klon|kpt
    |kwc\|-|kyo(c|k)|le(no|xi)|lg( g|\|(k|l|u)|50|54|e\|-|e\|/\|-|a-w))|libw|lynx|m1\-
    w|m3ga|m50\|/|ma(te|ui|xo)|mc(01|21|ca)|m\|-cr|me(di|rc|ri)|mi(o8|oa|ts)|mmef|mo(
    01|02|bi|de|do|t(\|-|o|v)|zz)|mt(50|p1|v )|mwbp|mywa|n10[0-2]|n20[2-3]|n30(0|2
    )|n50(0|2|5)|n7(0(0|1)|10)|ne((c|m)\|-|on|tf|wf|wg|wt)|nok(6|i)|nzph|o2im|op(ti
    |wv)|oran|owg1|p800|pan(a|d|t)|pdxg|pg(13|\|-|([1-8]|c))|phil|pire|pl(ay|uc)|pn\-
    2|po(ck|rt|se)|prox|psio|pt\|-g|qa\|-a|qc(07|12|21|32|60|\|-|2-7)|i\|-|qtek|r380|r60
    0|raks|rim9|ro(ve|zo)|s55\|/|sa(ge|ma|mm|ms|ny|va)|sc(01|h\|-|oo|p\|-)|sdk\|/|se(c\(\-
    |0|1)|47|mc|nd|ri)|sgh\|-|shar|sie(\|-|m)|sk\|-0|sl(45|id)|sm(al|ar|b3|it|t5)|so(f
    t|ny)|sp(01|h\|-|v\|-|v )|sy(01|mb)|t2(18|50)|t6(00|10|18)|ta(gt|lk)|tcl\|-|tdg\-
    |tel(i|m)|tim\|-|t\|-|mo|to(pl|sh)|ts(70|m\|-|m3|m5)|tx\|-9|up(\|.b|g1|si)|utst|v40
    0|v750|veri|vi(rg|te)|vk(40|5[0-3]|\|v)|vm40|voda|vulc|vx(52|53|60|61|70|80|8
    1|83|85|98)|w3c(\|-| )|webc|whit|wi(g_|nc|nw)|wmhb|wonu|x700|xda(\|-2|g)|yas\-
    |your|zeto|zte\|-|i', substr($useragent, 0, 4)));
 3 header('Location: http://detectmobilebrowser.com/mobile');
?>

```

1)Vengono prese le richieste del browser's user-agent string

2)Espressioni regolari

3)Vengono controllate le espressioni regolari e se corrispondono il traffico viene reindirizzato verso la Versione mobile.

E' difficile testare il proprio lavoro su dispositivi vecchi, difficilmente reperibili, per questo ci viene in aiuto il tool di Opera, che ci da la possibilità di ammirare il mockup del nostro sito...

Questo tool è un simulatore online del browser Opera Mini 4.2, molto diffuso nei dispositivi di fascia media, creato con la tecnologia **Java Applet**.

Il nostro sito su dispositivi mobile vecchi ha dei problemi...

Sharpen your pencil
Solution

The mockup looks pretty different in Opera Mini 4.2, huh? Let's look at some of the places where it differs or is weird. We found five trouble spots.

- ① The `<h1>` and `<h2>` tags are not styled and are quite small. Because of this, the background image in the header seems kind of silly.
We can't really style any of the heading elements in Opera Mini.
- ② The entire page is too wide and disappears off the right side of the screen.
This one's a doozy.
- ③ Text formatting is not getting applied very well. Italic text is not italic, and bold text is not bold.
- ④ Instead of rounded corners, we get square ones.
- ⑤ The status message uses overflow: scroll, which is an absolute no-no on all mobile devices. Content is cut off, and scrolling is not working.
Bonus points if you noticed this on the iPhone or Android already!

Sito
Creature
Conforts

Non tutti hanno uno smartphone, quindi se vogliamo che il nostro sito sia supportato dai browser dei dispositivi più vecchi, dobbiamo avere alcune accortezze...

```
<!DOCTYPE html>
```



Questo doctype è la migliore scelta per siti web moderni e affidabili, ma non nel nostro caso, dove dobbiamo interagire con dispositivi antiquati...

Ci viene in aiuto un altro standard, **XHTML**, che è una versione semplificata di **HTML**, progettata per essere eseguita bene sui browser mobile che sono progettati per supportarlo.

Guardiamo la nostra pagina ***index_mobile.html*** e questa dichiarazione **<!doctype html>**

Ma questo non basta per supportare la maggior parte dei dispositivi...

Ecco quindi il nostro nuovo amico, **XHTML-MP**, dove MP sta per mobile profile, è un doctype sviluppata specificatamente per supportare i telefoni di medio livello con browser mediocri.

Let's convert the mockup page to be an XHTML-MP document. Edit the ***index_mobile.html*** file and replace the HTML5 DOCTYPE (top line of the file) with the following two lines:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.2//EN"  
"http://www.openmobilealliance.org/tech/DTD/xhtml-mobile12.dtd">
```

XHTML-MP is a kind of XML, so we need this XML declaration, too.

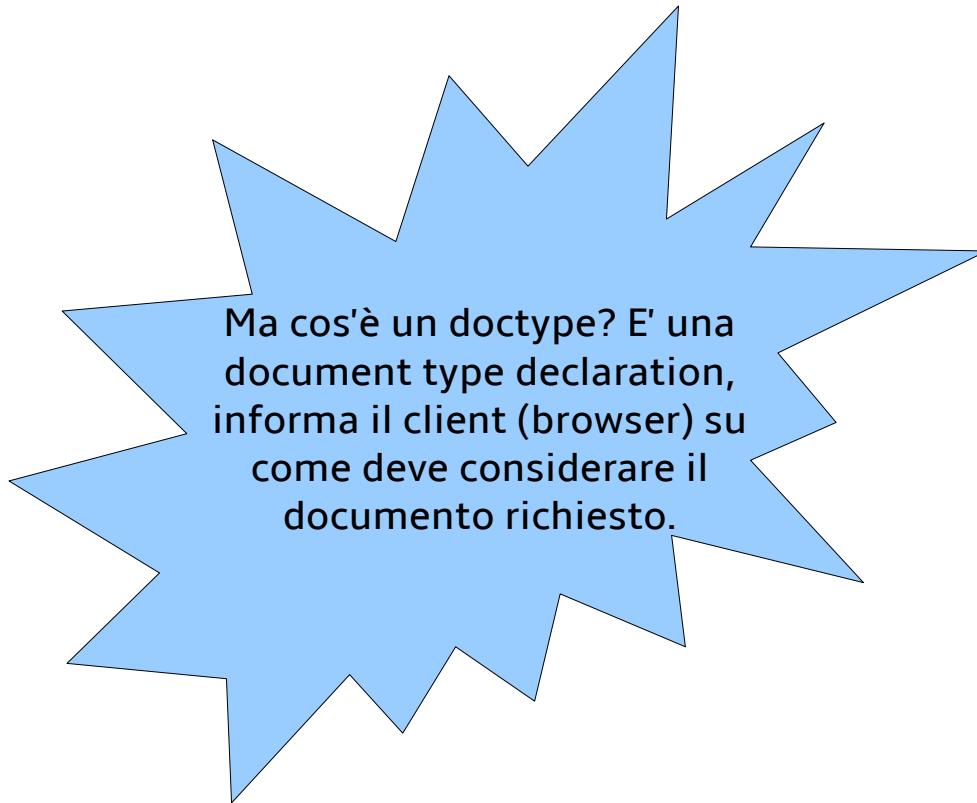
This is the DOCTYPE for XHTML-MP version 1.2. That's the most recent version.

And yes, we know we said "two lines," but it doesn't all fit onto the page.



index_mobile.html

Abbiamo semplicemente cambiato la dichiarazione del doctype, ma non posso utilizzare semplicemente l' html5? Sempre più nuovi telefoni lo supportano, ma nel nostro caso, utilizzando telefoni di fascia media, cambiando la dichiarazione abbiamo ottenuto un notevole cambiamento: nel nostro sito, ad esempio, il contenuto è ora proporzionato allo schermo del dispositivo.



Hey, look at that!
The content fits on
the screen now.

Tutto ciò garantisce che il nostro sito venga visualizzato correttamente anche su moltissimi browser mobile di vecchi dispositivi, sebbene la maggior parte dei browser moderni non soffrono troppo (tranne la perdita di alcune funzionalità) quando incontrano un documento in html5...

Se poi la funzionalità richiesta non è supportata in **xhtml-mp** c'è una ragione!! Questo **doctype** offre infatti tutto ciò che, semplicemente, rende il nostro sito ampiamente compatibile dalla maggior parte dei dispositivi in circolazione.

Queste cose sono molto belle, ma **xhtml-mp 1.2** è stato sostituito da **xhtml-basic** nel 2008, mi dispiace...



Ma non temete! Tutto ciò che è stato fatto in xhtml-mp è perfettamente compatibile con il nuovo standard per mobile che offre soltanto più funzionalità!

CONVALIDARE IL CODICE

una volta modificato il doctype del sito da HTML a XHTML è necessario accertarsi che la sintassi sia corretta e valida, per questo motivo è necessario testare il codice:
<http://validator.w3.org>

 **Markup Validation Service**
Check the markup (HTML, XHTML, ...) of Web documents

Jump To: Notes and Potential Issues Validation Output

Errors found while checking this document as XHTML Basic 1.1!

Result:	5 Errors, 2 warning(s)
Source :	<?xml version="1.0" encoding="UTF-8" ?> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.1//EN" "http://www.w3.org/TR/xhtml-basic/xhtml-basic11.dtd"> <html> <head> <title>Creature Comforts Agent Portal</title> <meta http-equiv="Content-Type" content="text/html; charset=utf-8" /> <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1" /> </head> <body> <div class="header"> <h1>Creature Comforts</h1>
Encoding :	utf-8 <input type="button" value="(detect automatically)"/>
Doctype :	XHTML Basic 1.1 <input type="button" value="(detect automatically)"/>
Root Element:	html

Come si poteva immaginare c'è sempre qualcosa che va storto, vediamo quali sono stati gli errori:

- 1) I tags <thead> e <tbody> non sono supportati da XHTML Basic e non potranno quindi essere utilizzati, è dunque necessario farne a meno correggendo il codice della table.
- 2) Alcuni tag non sono nidificati correttamente, per ditrazione più succedere di chiudere dei tag in ordine sbagliato. Sia ai programmatore esperti che quelli alle prime righe di codice può succedere.

Con questi errori corretti si puo tentare nuovamente di convalidare il codice incrociando le dita.

CSS Mobile Profile

Il CSS MP è utilizzato specificamente per lo stile di pagine XHTML Basic ed ha come target principale i dispositivi mobili di fascia media-bassa. Nel CSS MP si trovano le stesse funzionalità del CSS2 ed anche alcune del CSS3, ma non tutte.

In teoria tutti i browser che implementano il CSS MP dovrebbero supportare tutte le sue funzionalità, ma in realtà non è sempre così. Sta quindi al programmatore accertarsi che i diversi browser interpretino correttamente il codice ed in caso contrario prendere delle contromisure. Come se ciò non bastasse diversi valori e proprietà del CSS sono considerati opzionali in CSS MP, il che vuol dire che gli sviluppatori dei browser possono decidere di supportarle - oppure no.