



Responsive Web Design

.. il web tascabile

Che cos'è il Responsive Web Design?

- Responsive -> reattivo

Quindi il **Design Reattivo del Web**, in poche parole, che si adatta.

Ma facciamo un passo indietro...

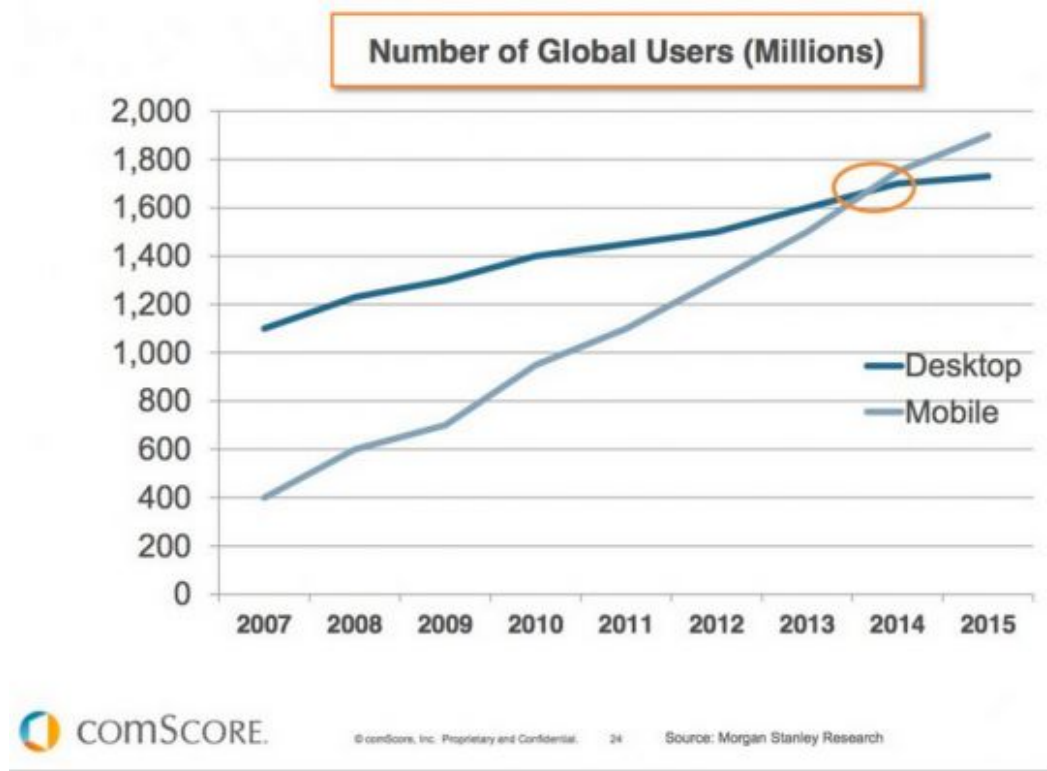


...perché tutto questo?

- Il Web è nato e cresciuto sui PC
- Monitor simili
- Navigazione tramite Mouse



- Ma c'è stato il boom tecnologico
- Laptop
- Smartphone
- Tablet



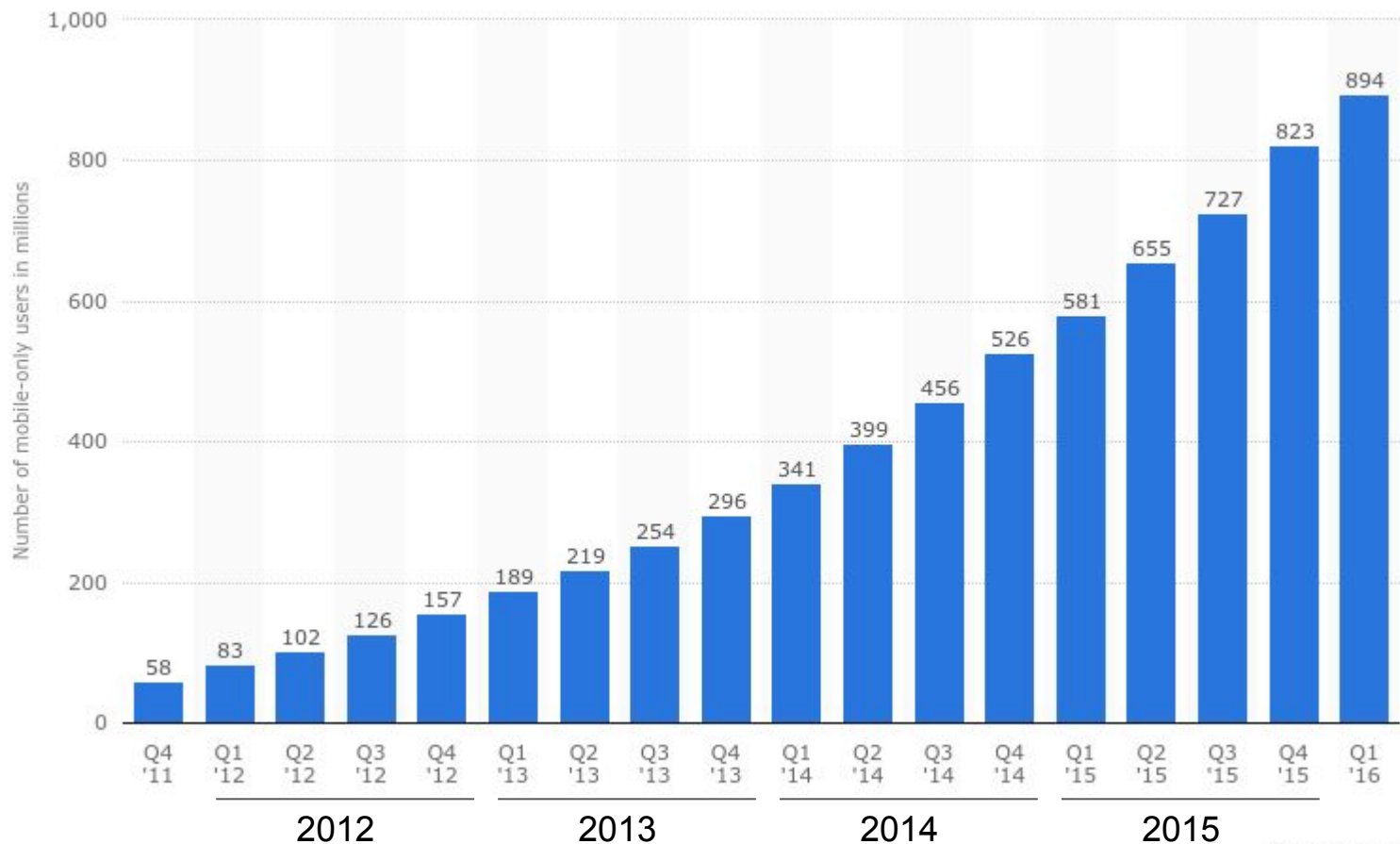
- Gli utenti mobile nel 2014 hanno superato gli utenti desktop
- Cresce la necessità di avere un sito accessibile da tutti i tipi di device e monitor



Google Mobilegeddon

- Google inserisce nell'algoritmo del suo crawler la verifica della compatibilità mobile
- Penalizzazione del ranking SEO in caso di incompatibilità
- Aggiunta della label "Mobile Friendly" nei risultati di ricerca

Utenti per mese che accedono a facebook con soli dispositivi mobile



Il mobile come *must-have*

Evoluzione del modo di progettare i siti web

- Dal singolo layout grafico a un set di layout pensato ad-hoc per ogni categoria di dispositivi (desktop, tablet, smartphone, etc)
- Da testi piccoli a testi grandi
- Grafiche a contrasto elevato
- Aumento dei contenuti a scorrimento verticale
- Scelta del device “target” principale

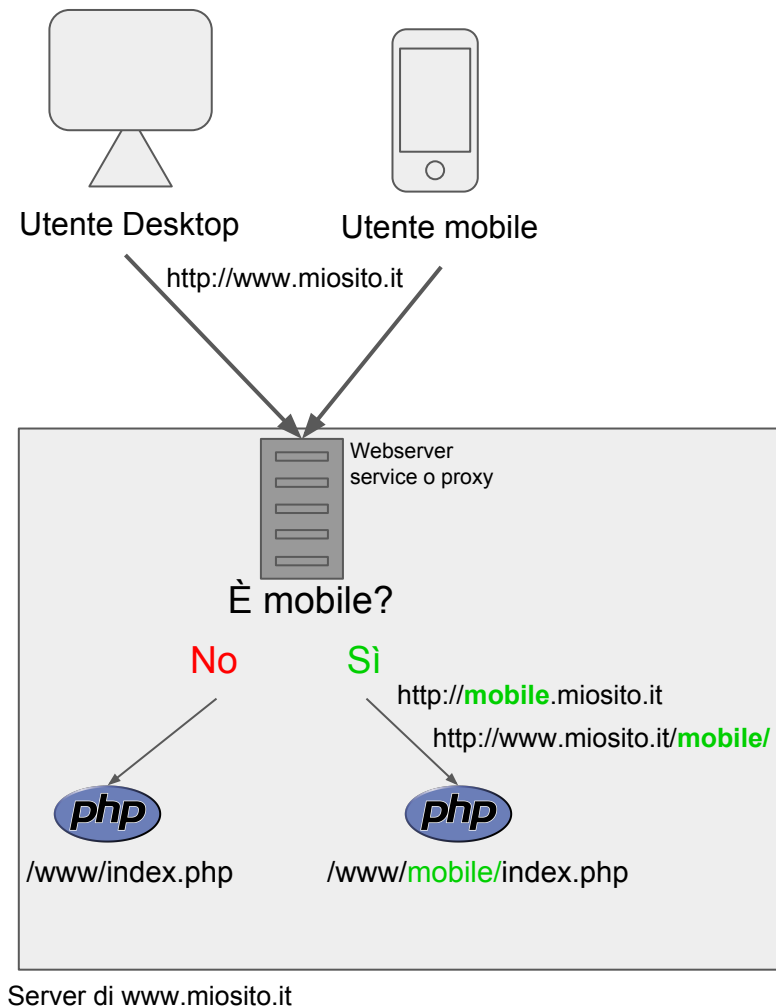
Come si realizza il mobile?

Ci sono diverse scuole di pensiero

- Separate URLs
- Dynamic Content Serving
- Responsive Design

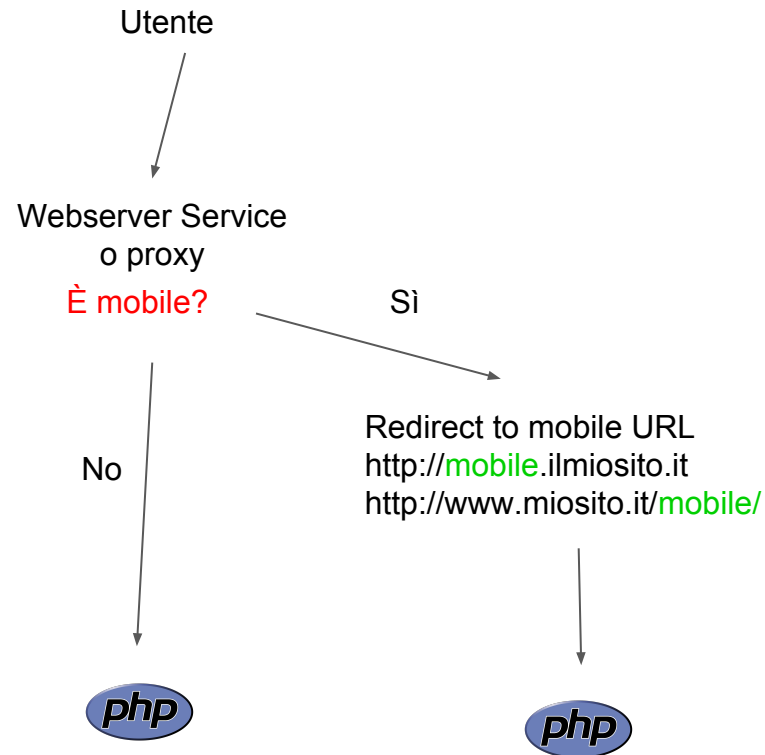
Separate URLs

- URL diverse a seconda del tipo di device
- Stessa base dei dati, diverse elaborazioni server e diversi CSS
- Controllo mobile sul server web (Apache, nginx) o sul proxy
- Riconoscimento mobile in base all'User Agent
- Diverso sottodominio o diversa directory



Server di `www.miosito.it`

Separate URLs Come funziona



Come avviene il controllo?

Separate URLs
Come funziona

Riga 16: si controlla che l'user agent contenga una tra una serie di stringhe pre-selezionate come mobile

Riga 19: si controlla di non essere già nel sito mobile (senza di questo controllo ci sarebbe un loop infinito)

Riga 22: in caso entrambe le condizioni siano verificate si procede al redirect

.htaccess

```
15 # Check that user agent is mobile
16 RewriteCond %{HTTP_USER_AGENT} "android|blackberry|googlebot-mobile|iemoible|ipad|iphone|ipod|opera
  mobile|palms|webos" [NC]
17
18 # Check if we're not already on the mobile site
19 RewriteCond %{REQUEST_URI} !([/]?mobile) [NC]
20
21 # Now redirect to the mobile site
22 RewriteRule ^$ mobile [R,L]
23
```

Separate URLs

Pro e Contro

Pro:

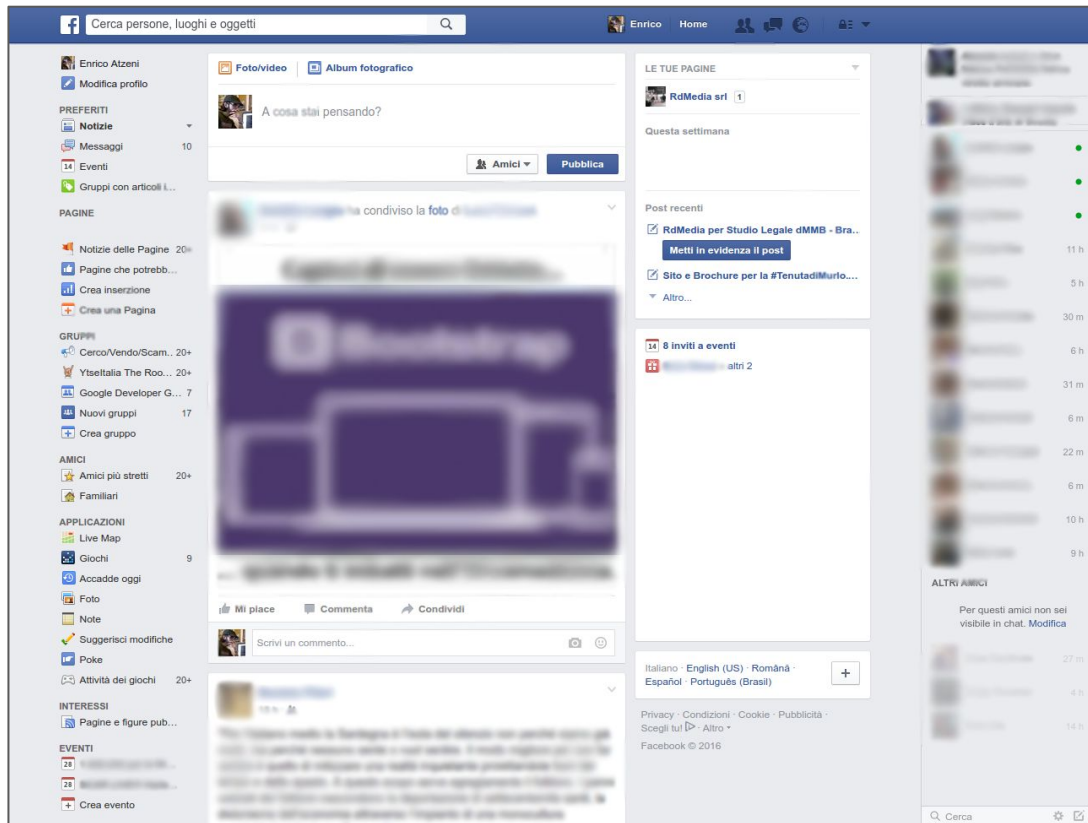
- Contenuti ottimizzati in funzione del target
- Sviluppo iniziale rapido
- Possibilità di avere un server esterno dedicato per il solo mobile
- Possibilità di switch-off

Contro:

- Gestione del redirect (aggiornamento libreria User Agent, etc..)
- Manutenzione del doppio del codice di front-end (doppia possibilità di bug, doppio del tempo impiegato)
- Minor ranking SEO per alcuni motori di ricerca

Facebook

<https://www.facebook.com/>



*Separate URLs
esempio*

<https://m.facebook.com/home.php>



Parte Prima:

Partendo dalla pagina web esistente e dal codice sorgente creare la sua versione mobile.

Il codice per i dispositivi mobile andrà inserito dentro la cartella **mobile** presente nella root del sorgente desktop.

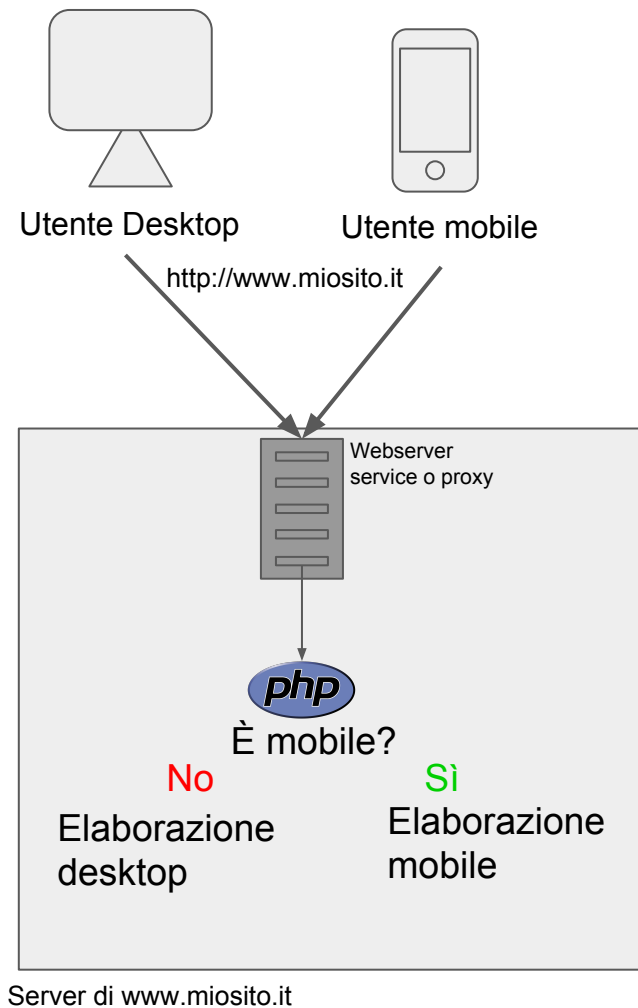
CONSIGLIO: copia il codice html e css della versione desktop poi modificalo.

Parte Seconda:

Nell'header aggiungere il testo “**Separate URLs**” prima dell'immagine nel desktop, mentre nel mobile al posto dell'immagine.

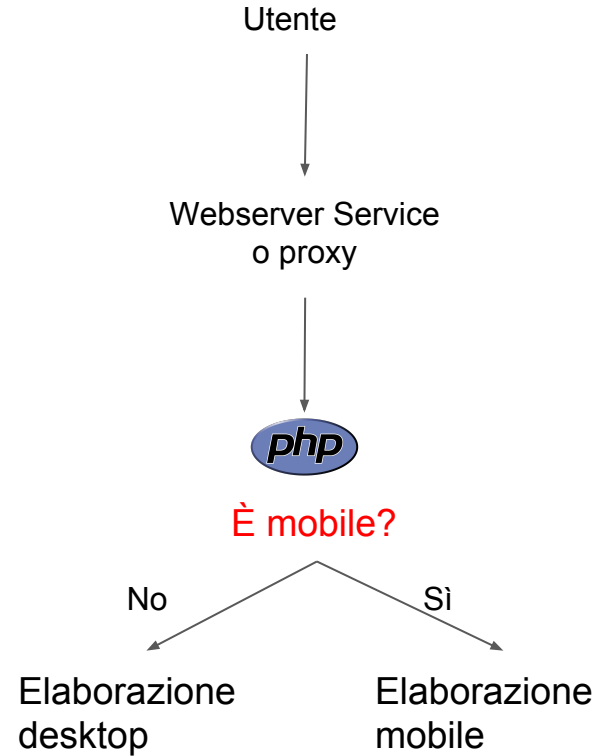
Dynamic Content Serving

- Stessa URL per tutti i dispositivi
- Stessa base dei dati, diverse elaborazioni server e diversi CSS
- Controllo mobile sul PHP (o ASP)
- Riconoscimento mobile in base all'User Agent



Dynamic Content Serving

Come funziona



Come avviene il controllo?

- Tramite una funzione PHP (o ASP)
- Preferibilmente tramite una classe
- Si confronta l'user agent del client con una serie di stringhe preselezionate
- In caso positivo si restituisce **true**, e il codice potrà quindi differenziare le esecuzioni
- In caso si sia utilizzata una classe:
 - Nel costruttore si effettua il controllo e si memorizza il risultato in un attributo `is_mobile` privato
 - Tramite un metodo pubblico si permette di ottenere il valore dell'attributo `is_mobile` privato

Come avviene il controllo?

```
3 class MyMobileDetector{
4
5     // properties
6     private $isCurrentUserMobile;
7     private $mobileIdentifierPattern = '/(android|blackberry|iemobile|ipad|iphone|ipod|opera mobile|palms|webos)/i';
8
9     //methods
10    function MyMobileDetector(){
11        //this is the constructor
12
13        // checking if the user agent contains Mobile identifiers
14        if(preg_match($this->mobileIdentifierPattern, $_SERVER['HTTP_USER_AGENT'])){
15            $this->isCurrentUserMobile = true;
16        }else{
17            $this->isCurrentUserMobile = false;
18        }
19    }
20
21
22    public function isMobile(){
23        return $this->isCurrentUserMobile;
24    }
25
26
27 }
```

Insieme di stringhe che identificano user agent mobile

Controllo dello user agent

Assegnazione valore all'attributo privato

Metodo pubblico per ottenere il valore dell'attributo privato

Dynamic Content Serving

Pro e Contro

Pro:

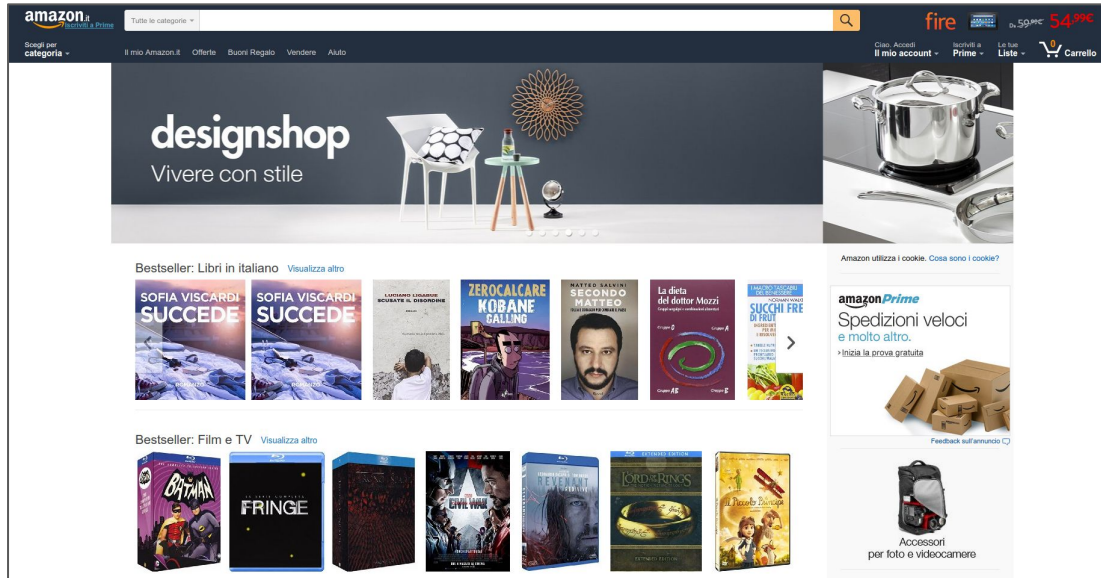
- Contenuti ottimizzati in funzione del target
- Separazione dei soli contenuti specifici
- Possibilità di switch-off

Contro:

- Gestione degli User Agent (aggiornamento libreria User Agent, etc..)
- Codice poco pulito
- Conflitto con i cache server

Amazon

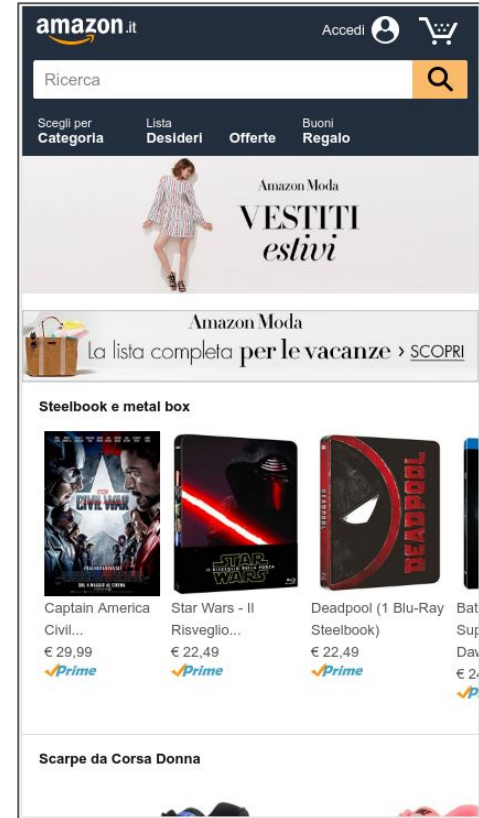
<https://www.amazon.it/>



Dynamic Content Serving

Esempio

<https://www.amazon.it/>



Parte Prima:

Partendo dalla pagina web esistente e dal codice sorgente PHP creare la sua versione mobile differenziando i blocchi di codice con i **conditional statements** (**if()**) a seconda del tipo di device appoggiandosi sulla classe PHP **MyMobileDetector.class.php** e sul suo metodo **\$mobileDetectorIstance->isMobile()**.

```
<?php
if($mobileDetectorIstance->isMobile()){
    // here html code for mobile
    echo '<h2>Versione Mobile</h2>';
}else{
    // here html code for desktop
    echo '<h2>Versione Desktop</h2>';
}
?>
```

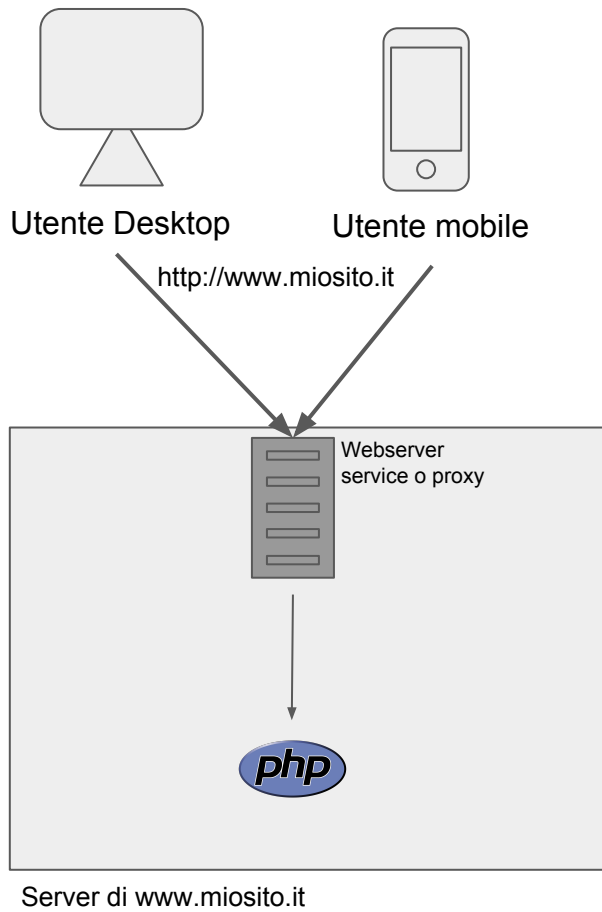
CONSIGLIO: utilizzando il metodo **isMobile()** caricare un file CSS creato appositamente per mobile.

Parte Seconda:

Rimuovi la sidebar dal mobile

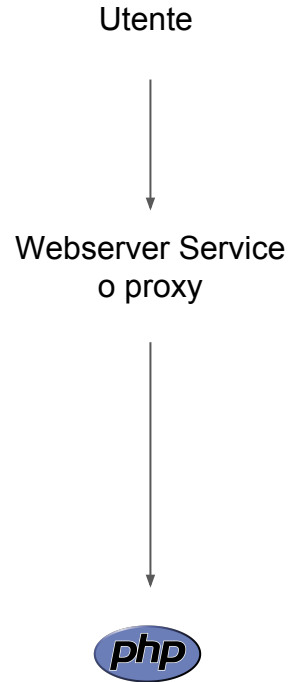
Responsive Design

- Stessa URL per tutti i dispositivi
- Stessa base dei dati, stessa elaborazione server e stesso CSS
- Controllo mobile sul browser
- Riconoscimento mobile in base alla risoluzione dello schermo



Responsive Design

Come funziona



Come avviene il controllo?

- Nessun controllo
- Si basa sulle dimensioni della finestra del browser (**viewport**)



Responsive Design

Pro e Contro

Pro:

- Unico sorgente per entrambi i device
- Manutenzione del codice efficiente e veloce
- Tutto il mobile in un CSS

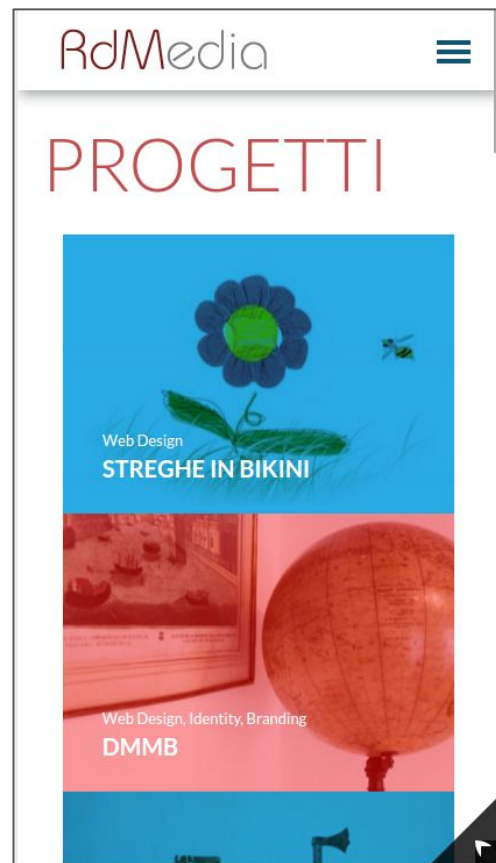
Contro:

- Costo maggiore di sviluppo (tempo, energie)
- Contenuti non sempre ottimizzati per il mobile (video, widget e plugin pesanti, etc..)

RdMedia SRL

Responsive Deisign *Esempio*

<http://www.rdmedia.it/>



CSS3 e le media queries

```
4  
5 @media [device] [and] [resolution] {  
6     [rules]  
7 }
```

- Regole specifiche per device (screen, print, etc)
- Regole specifiche per risoluzione (max-width, min-height, etc)
- 100% compatibile con i browser

Media queries, l'altra possibilità: diversi CSS

```
1 <link rel="stylesheet" type="text/css" href="style.css" />
2
3 <link rel="stylesheet" type="text/css" href="style_mobile.css"
  media="screen and (max-width:800px)" />
```

Media queries: **min-width** o **max-width**?

Min-width: dal mobile al desktop

Max-width: dal desktop al mobile

CSS3 media-queries: Esempio

```
11 .one-third-column{
12     width: 33.3%;
13 }
14 @media screen and(max-width: 800px){
15     .one-third-column{
16         width: 50%;
17     }
18 }
```

screen.width <= 800

.one-third-column	.one-third-column
.one-third-column	

screen.width > 800

.one-third-column	.one-third-column	.one-third-column

Parte Prima:

Partendo dalla pagina web esistente creare la sua versione mobile usando le **media queries** nel file **style.css** già esistente.

Parte Seconda:

Aggiungere la voce “**il nostro obiettivo**” nel menù tra “**dove siamo**” e “**chi siamo**”

Quindi.. Quale uso?

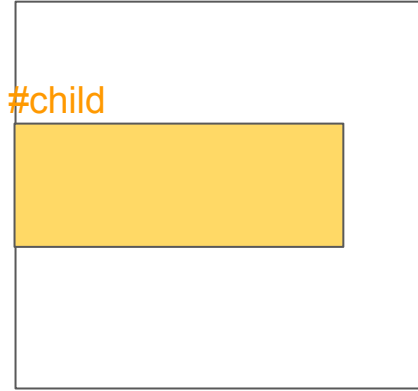
- Scegli in base al progetto
 - Contenuto mobile = contenuto desktop => Responsive Design
 - Contenuto mobile molto simile al contenuto desktop => Dynamic Content Serving
 - Contenuto mobile != contenuto desktop => Separate URLs
- Scegli in base al tempo
- Scegli in base al codice già esistente (se ce ne fosse)
- Deve piacere a te

Consigli pratici per un layout responsive

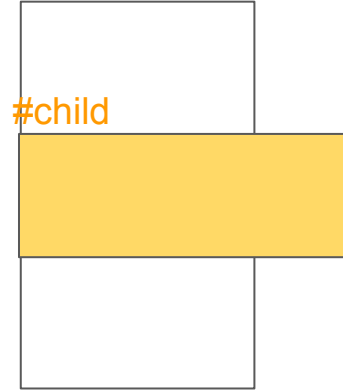
100%

- È una percentuale
- Si adatta alle dimensioni del contenitore
- Sarà il vostro migliore amico
- Progettate in percentuale

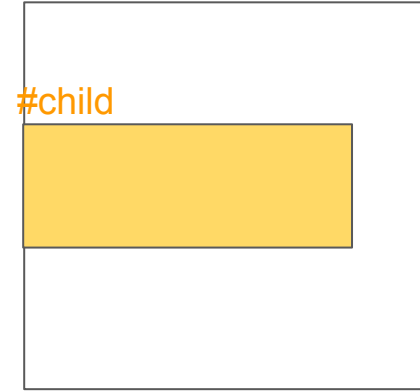
Browser window



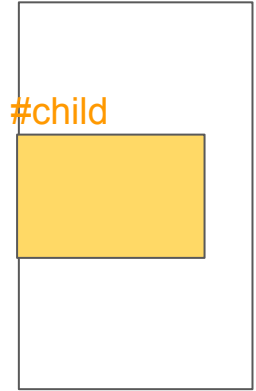
Browser window



Browser window



Browser window



```
112  
113 #child{  
114     width: 400px;  
115 }
```

```
113 #child{  
114     width: 80%;  
115 }
```

Immagini SEMPRE proporzionali!



New York City

An Amazing City

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus nunc ut leo vehicula pretium. Ut enim ante, rhoncus volutpat finibus molestie, tincidunt et lectus. Proin libero neque, elementum eu sollicitudin vitae, porttitor eu lacus.

Nulla elementum nibh non felis varius, fermentum maximus ipsum tristique. Sed feugiat massa est, tempor vehicula libero pretium a. Nunc sit amet ex sit amet odio molestie semper. Nam eu vehicula diam hendrerit viverra ex. Etiam rhoncus leo eu scelerisque tempus. Vestibulum auctor magna id

```

```



New York City

An Amazing City

Lorem ipsum dolor sit amet, consectetur

```
img{  
  width: 400px;  
}
```



New York City

An Amazing City

```
img{  
  width: 400px;  
  height: auto;  
}
```

Accorpa le regole comuni in classi CSS

```
#block-1{
  width: 35%;
  background: red;
}
#block-2{
  width: 35%;
  font-size: 23px;
  background: green;
}
#block-3{
  width: 35%;
  font-size: 21px;
}
```

```
<div id="block-1">Lorem ipsum, sfondo rosso</div>
<div id="block-2">Lorem ipsum, sfondo verde, testo grande</div>
<div id="block-3">Lorem ipsum, testo piccolo</div>
```

```
.block-width{
  width: 35%;
}
#block-1{
  background: red;
}
#block-2{
  font-size: 23px;
  background: green;
}
#block-3{
  font-size: 21px;
}
```

```
<div id="block-1" class="block-width">Lorem ipsum, sfondo rosso</div>
<div id="block-2" class="block-width">Lorem ipsum, sfondo verde, testo grande</div>
<div id="block-3" class="block-width">Lorem ipsum, testo piccolo</div>
```

Buone Pratiche...

Esempi

Guide

Forum

Blog

Documentazione

Community

News

Altri siti



Cerca su Google o digita l'URL



Progettare un layout complesso

- Dividetelo in blocchi (header, content, sidebar, footer)
- Immaginate il responsive
- Pensatelo prima in landscape poi in portrait
- Pensate a cosa si aspetta di trovarsi l'utente
- Pochi colori (in media 3)
- Bozza del layout html
- Realizza prima l'html e poi il css



[HTML](#)
[CSS](#)
[JavaScript](#)
[PHP](#)
[ASP](#)
[Ajax](#)
[jQuery](#)
[AngularJS](#)



Lorem ipsum dolor sit amet, consectetur
dipiscing elit. Proin vitae orci vel nibh molestie
iverra. Nulla vulputate augue pulvinar,
suere neque et, blandit nunc. Donec
condimentum sem eu purus cursus egestas. Vivamus est
odio, lobortis et tempor in, hendrerit in ipsum. Aliquam erat
volutpat.

Quisque at nisi vel quam viverra vehicula. Aenean
vulputate auctor semper.
Curabitur at vehicula urna. Cras convallis tempus risus,
eget vulputate nibh fermentum vel.

Sed rhoncus est sit amet turpis malesuada, non interdum
nunc porta.



[Home](#) - [Chi Siamo](#) - [Prodotti](#) - [Contatti](#)

[HTML](#)
[CSS](#)
[JavaScript](#)
[PHP](#)
[ASP](#)
[Ajax](#)
[jQuery](#)
[AngularJS](#)



Lorem ipsum dolor sit amet, consectetur
dipiscing elit. Proin vitae orci vel nibh molestie
iverra. Nulla vulputate augue pulvinar,
suere neque et, blandit nunc. Donec
condimentum sem eu purus cursus egestas. Vivamus est
odio, lobortis et tempor in, hendrerit in ipsum. Aliquam erat
volutpat.

Quisque at nisi vel quam viverra vehicula. Aenean
vulputate auctor semper.
Curabitur at vehicula urna. Cras convallis tempus risus,
eget vulputate nibh fermentum vel.

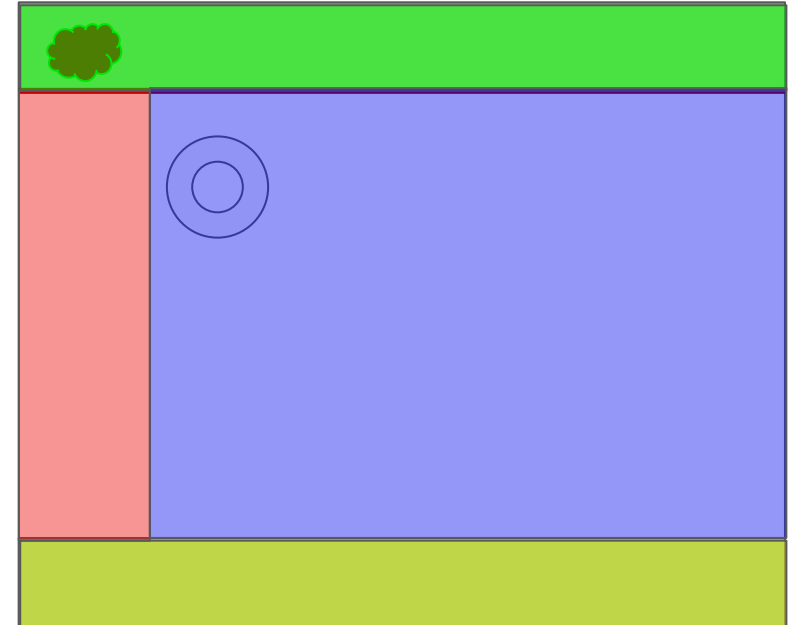
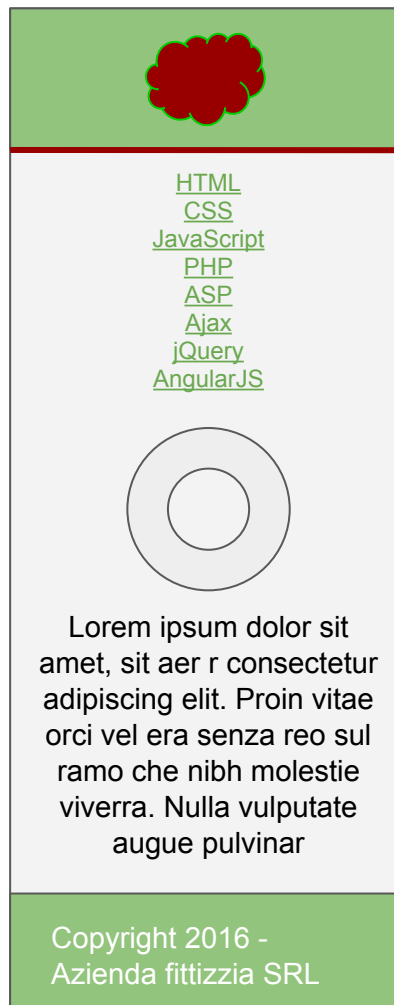
Sed rhoncus est sit amet turpis malesuada, non interdum
nunc porta.

Copyright 2016 - Azienda fittizzia SRL

Buone Pratiche...
...in pratica

Buone Pratiche...

...in pratica



Parte Prima:

Basandosi sulla grafica desktop realizzare la pagina web.

Parte Seconda:

Basandosi sulla grafica mobile realizzare la versione mobile della pagina web desktop appena creata.
Potete usare il metodo che preferite.

Nota per Separate URLs:

rinominare il file “**rename-me.htaccess**” in “**.htaccess**”

Nota per Dynamic Content Serving:

Includere la libreria nel file index.php

Strumenti per sviluppare

- Un buon text-editor con completamento automatico
- Un ambiente locale di test
- Carta e penna



Compatibilità fra i browser

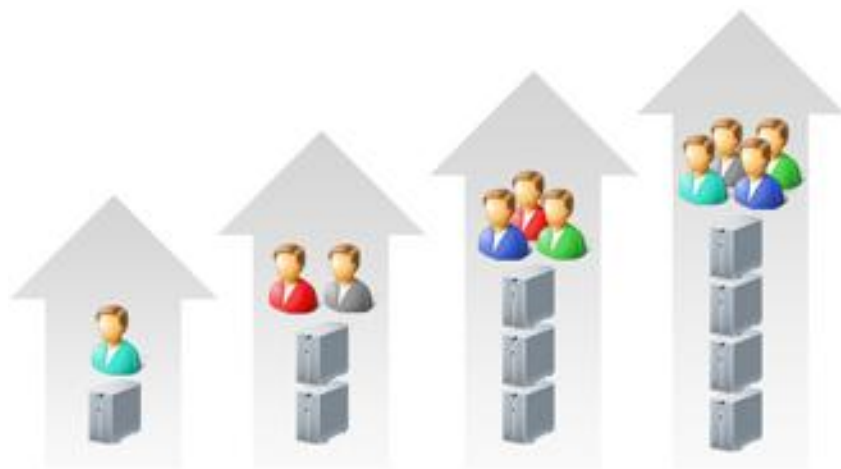
- CSS3 non del tutto implementato
- Browser mobile stock meno completi
- Ogni browser interpreta a modo suo il CSS
- Floating, posizionamenti fissi, overlay, trasparenze
- Diversi calcoli per le dimensioni (width, height, etc)
- Risolvendo su un browser rischi di creare un problema sugli altri

Niente panico

- Progetta un layout il più semplice possibile
- Se diventa complicato probabilmente c'è qualche errore
- C'è sempre un metodo alternativo, scovalo

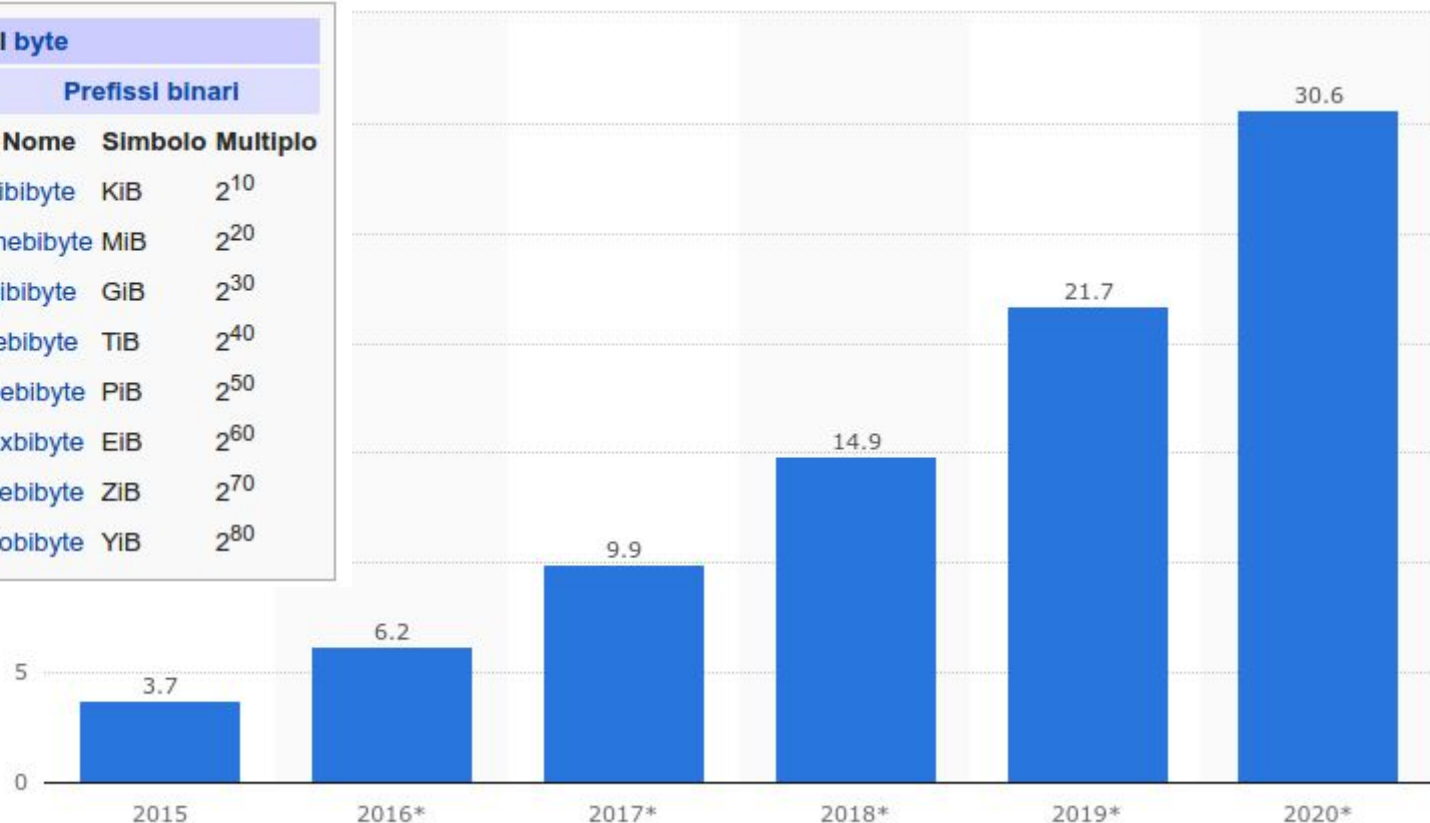
Semplice è funzionale, perché?

- Codice più pulito e più leggibile
- Codice meglio organizzato
- Logica divisa in moduli
- Meno codice complessivo, minor traffico di rete richiesto
- Più facile da testare
- L'applicazione diventa scalabile



Traffico globale da dispositivi mobile in Exabyte al mese

Multipli del byte					
Prefissi SI			Prefissi binari		
Nome	Simbolo	Multiplo	Nome	Simbolo	Multiplo
kilobyte	kB	10^3	kibibyte	KiB	2^{10}
megabyte	MB	10^6	mebibyte	MiB	2^{20}
gigabyte	GB	10^9	gibibyte	GiB	2^{30}
terabyte	TB	10^{12}	tebibyte	TiB	2^{40}
petabyte	PB	10^{15}	pebibyte	PiB	2^{50}
exabyte	EB	10^{18}	exbibyte	EiB	2^{60}
zettabyte	ZB	10^{21}	zebibyte	ZiB	2^{70}
yottabyte	YB	10^{24}	yobibyte	YiB	2^{80}



Su quali browser faccio i test?

Desktop:

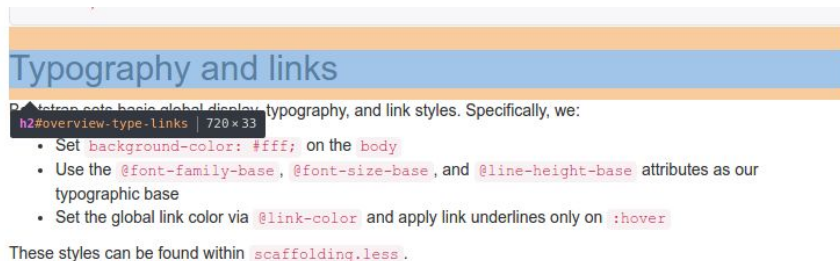
- Chrome
- Firefox
- Safari
- Internet Explorer (**Edge**)

Mobile

- Android Browser stock
- Chrome
- Safari in iOS
- Firefox

Strumenti per il debug

- Fortunatamente ci vengono forniti dai principali browser
- Si aprono premendo F12
- O si può installare un'estensione del browser, come Firebug



Normalize.css

For improved cross-browser rendering, we use [Normalize.css](#), a project by [Nicolas Gallagher](#) and [Jonathan Neal](#).

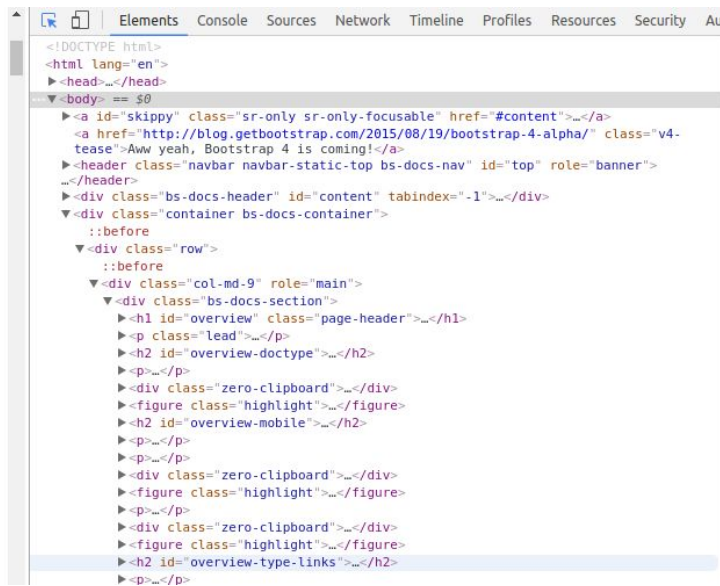
Containers

Bootstrap requires a containing element to wrap site contents and house our grid system. You may choose one of two containers to use in your projects. Note that, due to `padding` and more, neither container is nestable.

Use `.container` for a responsive fixed width container.

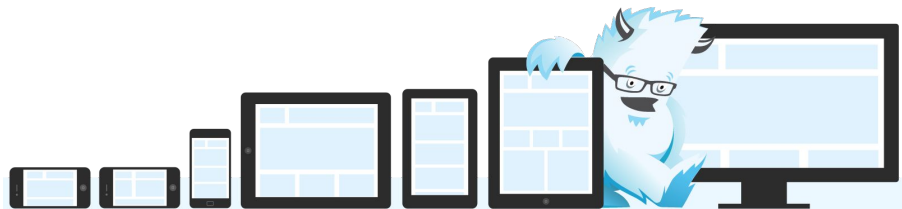
```
<div class="container">
```

Copy



Frameworks CSS / HTML

- Esistono vari framework open-source (Bootstrap, Foundation, etc)
- Layout preconfezionati
- Veloci da usare
- Personalizzabili, ma fino ad un certo punto
- “Stampo” di fabbrica





Buon developing!!!

Contacts:



eatzeni@rdmedia.it



<https://www.facebook.com/rdmediasrl/>



<https://www.linkedin.com/company/rdmedia>

RdMedia

<http://www.rdmedia.it>