SearchForecast

February 2014 Update

Recommendations for Websites Optimization for Mobile Devices Using Mobile Centric CSS

SearchForecast

Is Mobile SEO Required Now?

- In a Google Smartphone User study, mobile searches will make up 25% of all searches in the world this year.
- A Compuware study said that over 50% of consumers would not recommend a business with a bad mobile site.
- Close to 50% of mobile searchers made a mobile purchase in the last six months.
- Over 90% of mobile searches end in some type of action…visiting a business or purchasing

SearchForecast

Mobile versus Desktop SEO

There are major differences between mobile versus desktop optimization for search engines:

• **Research mobile-specific keyword** – The traditional search experience is based on keywords users type in. With mobile a user could search using Voice Search or Google Goggles.

• **Top 3 search positions are the most important** – In traditional search, the top 3 search listings are important. In mobile, the top three positions are EVERYTHING. In fact, drop down just from the first position and your conversion plummets 90%.

• **Mobile users don't scroll** – For whatever reason, mobile users want to see everything in the screen. They don't even want to think of scrolling.

SearchForecast

Four Mobile Best Practices

1. User Opt Out
The best mobile user experience allows users to opt-out of a mobile-formatted CSS.

2. Image Rendering
Images are critical on a mobile device, but only if you render them by relative or percentage. Avoid rendering images in absolute scale, or pixel scale.

3. Develop sites that can be used across all devices
A fully optimized mobile site will outperform a site that has been reformatted for a smaller screen.

4. Link Length
Short links rule on mobile. No long URLs.

SearchForecast

Recommendation

It is recommended that the following best practice using mobile centric CSS (style sheets) is implemented.

SearchForecast

Why a Mobile Centric CSS?

- Implementing a **Mobile-centric CSS file will** recognize when a user is accessing the website from a mobile device. This then delivers a CSS version better suited to the device.

```
/* mobile 1 ---- */
@media only screen and (min-width: 1px) and (max-width: 320px) {
body:after {
        content: 'RD:Mobile 1|up to 320';
        display: none;
}
html {
        content: "RD:Mobile 1|up to 320";
}
}
```

```
/* mobile 2 ---- */
@media only screen and (min-width: 321px) and (max-width: 540px) {
body:after {
        content: 'RD:Mobile 2|321 to 540';
        display: none;
}
html {
        content: "RD:Mobile 2|321 to 540";
}
}
```

```
/* tablet ---- */
@media only screen and (min-width: 541px) and (max-width: 800px) {
body:after {
        content: 'RD:Tablet|541 to 800';
        display: none;
}
html {
        content: "RD:Tablet|541 to 800";
}
}
```

```
/* desktop ---- */
@media only screen and (min-width: 801px) {
body:after {
        content: 'RD:Desktop|801 or larger';
        display: none;
}
html {
        content: "RD:Desktop|801 or larger";
}
}
```

SearchForecast

Mobile CSS Best Practice

- When crafting CSS, try to keep things lightweight and as fluid as possible.
- Mobile devices have many different screen sizes, and that tomorrow's devices won't have the same resolutions as today's. Because screen size is an unknown, try to use the content itself to determine how the layout should adjust to its container
- Use **Separate style sheet for larger screens**
  - E.g.: Separate CSS files, style.css and enhanced.css in order to deliver basic styles for screens less than 40.5em and using media queries to serve up enhanced styles for screens larger than 40.5em.

```
<link rel="stylesheet" type="text/css" href="style.css" media="screen, handheld" />

<link rel="stylesheet" type="text/css" href="enhanced.css" media="screen  and (min-width: 40.5em)" />

<!--[if (lt IE 9)&(!IEMobile)]>
<link rel="stylesheet" type="text/css" href="enhanced.css" />
<![endif]-->
```

SearchForecast

Mobile CSS Best Practice

- Use the conditional code <!--[if (lt IE 9)&(!IEMobile)]> in order to serve up enhanced.css to non-mobile versions of IE less than version 9, which unfortunately don't support media queries.
- While this method does indeed add an HTTP request to the mix, it gives us greater flexibility over our styles.
- The example is shown using the em unit instead of px to maintain consistency with the rest of our relative units and account for user settings like zoom level. Also, the content should determine the breakpoint (here 40.5em is defined as a breakpoint) because device dimensions are too varied and are always changing so are therefore unreliable.

```
<link rel="stylesheet" type="text/css" href="style.css" media="screen, handheld" />

<link rel="stylesheet" type="text/css" href="enhanced.css" media="screen  and (min-width: 40.5em)" />

<!--[if (lt IE 9)&(!IEMobile)]>
<link rel="stylesheet" type="text/css" href="enhanced.css" />
<![endif]-->
```

SearchForecast

Mobile - First Styles

- Starting with baseline shared styles and introducing more advanced layout rules when screen size permits, keeps code simpler, smaller and more maintainable..
- The web by default is a fluid thing so try to work with it instead of against it.
- Note :Symbian browsers, Blackberry <OS 6.0, Netfront, WP7 pre-Mango, etc don't support media queries, so serving base styles by default reaches more devices and browsers.

```
 / *Large screen styles first - Avoid*/

.product-img {
  width: 50%;
  float: left;
 }
@media screen and (max-width:
40.5em) {
  .product-img {
    width: auto;
    float: none;
  }
}
```

```
 / *Large screen styles first – to
use*/

 . @media screen and (min-width:
40.5em) {
  .product-img {
    width: 50%;
    float: left;
  }
 }
```

Using CSS to Reduce Requests

- Too many HTTP requests can be a huge killer for performance, especially on mobile.
- Using CSS gradients instead of background images reduces the amount of image requests and gives more control over the design.

```
/*Using CSS gradients instead of background images*/

    header[role="banner"] {
      position: relative;
      background: #111;
      background: +linear-gradient (top, #111 0%, #222 100%);
    }
```

SearchForecast

Mobile-centric SEO practices

Googlebot-Mobile – Google inspects a site for Compact HTML or XHTML mobile. If it searches your Doc Type and doesn't find this, or worse, you block the bots, then your mobile version will not be indexed, and won't show up in mobile searches.

Call-to-Action Position – The best place to locate call-to-action buttons or icons are on the top-left. That's counter-intuitive to desktop search. Why? Put it on the top right and it may get cut off. You don't want them to have to scroll horizontal to see it.

Click Throughs – Users don't mind clicking on desktop when they know that each click takes them closer to their desired destination. Not so with mobile. Two times is the limit.

On the desktop page, **add a special link rel="alternate" tag** pointing to the corresponding mobile URL. This helps Googlebot discover the location of your site's mobile pages.

- <link rel="alternate" media="only screen and (max-width: 640px)" href="http://m.example.com/page-1"

On the mobile page, add a link rel="canonical" tag pointing to the corresponding desktop URL.

- <link rel="canonical" href="http://www.example.com/page-1" >

Mobile-centric SEO practices

Create mobile-friendly content

- After the site architecture and design is determined, your next step is to figure out how you are going to host your mobile site.
- You can find lots of options, but we suggest the best way to do it is simply park your homepage and mobile-only pages on m.domain.com subdomain or /m subfolder.
- The desktop pages you don't change to mobile you can keep at your desktop URL. Just reformat them for mobile users.
- You can also redirect, but when it comes to transcoded desktop URLs, you'll want to use canonical tags so the link juice is directed back to desktop pages.