

RESPONSIVE WEB TASARIMI VE UYGULAMALARI

BURAK TOKAK

İÇİNDEKİLER

BÖLÜM 1: RESPONSIVE WEB TASARIMI KONSEPTİ	1
Responsive Web Tasarımı Nedir?	2
Bilinmesi Gereken Kavramlar	2
Responsive Kelimesi ve Responsive Tasarım	3
Web Tarayıcıları	3
Kullanıcı Arayüzü - User Interface (UI)	3
Kullanıcı Deneyimi - User Experience (UX)	4
Handheld Aygıtlar	5
Grid (Arayüz Bölümleri)	5
Framework'ler ve Responsive Tasarım	5
Kırılma Noktaları (Breakpoints)	6
Neden Responsive Web Tasarımı	7
Mobil İnternet ve Cihaz Ekran Boyutları	8
Neler Öğrendik?	9
BÖLÜM 2: CSS VE JAVASCRIPT İLE RESPONSIVE ELEMENTLER	11
Responsive Element Nedir?	12
CSS'de Yüzde (%) Kavramı	12
CSS'de Calc() Fonksiyonu ile Responsive Elementler	16
CSS'de max-width ve min-width Özellikleri	20
JavaScript ile Yüzde (%) Kavramı	23
Neler Öğrendik?	29
BÖLÜM 3: VIEWPORT VE MEDIA KURALLARI	31
Viewport Kullanımı ve İşlevleri	32
Viewport Genişlik (Width) Tanımlayıcısı	34
Viewport Yakınlık ve Yakınlaşabilirlik (Scale) Tanımlayıcısı	35

Media Kuralları ve İşlevleri	36
Diğer Media Parametreleri	43
Neler Öğrendik?	46
BÖLÜM 4: RESPONSIVE SAYFA DÜZENİ VE UYGULAMALI ÖRNEKLER	49
Responsive Elementler ve Sayfa Düzeni	50
Sayfa Yapısını Yeniden Oranlama Tekniği	50
Önemi Az Bölümü Açılır/Kapanır Hale Getirme Tekniği	53
Responsive Olmayan Bölümü Olan ile Değiştirme Tekniği	68
Neler Öğrendik?	69
BÖLÜM 5: RESPONSIVE TİPOGRAFI VE YAZI DÜZENİ	71
Tipografi Nedir?	72
CSS ile Genel Tipografik Özellikler	73
Responsive Tipografi Mantiği	80
Neler Öğrendik?	83
BÖLÜM 6: SVG KULLANIMI VE SVG İKONLAR	85
SVG Nedir?	86
SVG'nin Klasik Resim Formatlarına Karşı Faydaları	87
SVG Syntax'ı ve Kullanımı	88
SVG için Vektörel Çizim Programları Kullanmak	94
Adobe Illustrator ve Kullanımı	95
SVG İkonları ve Kullanımları	98
Google Material Icons	99
JavaScript ile Hareketlendirilebilen SVG Elementleri	99
Neler Öğrendik?	103

BÖLÜM 7: ARAYÜZ PERFORMANSI VE HIZ OPTİMİZASYONU 105

Arayüz Performansı ve Verimliliği	106
İçerik Sıkıştırma ve Sayfa Yüklmesi	106
Verimli ve Modüler JavaScript	109
Verimli JavaScript için Genel Geçer Pratikler	110
Modüler JavaScript Nedir?	113
Neler Öğrendik?	124

BÖLÜM 8: MOBİL ÖNCELİKLİ RESPONSIVE WEB TASARIMI 127

Mobil Öncelikli Tasarım Nedir?	128
Mobil Öncelikli Tasarım Oluşturma ve Adımları	129
Mobil Öncelikli Tasarım Örneği: apple.com	134
Mobil Öncelikli Tasarım Örneği 2: fitekran.com	136
Her Zaman Mobil Öncelikli Tasarım Doğru Tercih Midir?	141
Mobil Öncelikli Tasarımda Mobil Navigasyon ve Gezilebilirlik	141
Neler Öğrendik?	143

BÖLÜM 9: RESPONSIVE TASARIMDA KULLANICI DENEYİMİNE (UX) YAKLAŞIM 145

Kullanıcı Deneyimi (UX) Nedir?	146
Kullanıcı Deneyimi Testi	146
Kullanıcı Deneyimi Testi Teknikleri	146
Kullanıcı Deneyimine Yaklaşım	148
Neler Öğrendik?	151

BÖLÜM 10: RESS GELİŞTİRME YAKLAŞIMI 153

RESS Nedir?	154
RESS Kullanmak Verimli mi?	155
RESS ve Kullanıcı Deneyimi	156
Neler Öğrendik?	157

BÖLÜM 11: WEB TABANLI PROGRAMLAMA VE UYGULAMALARIN GELECEĞİ 159

Web Uygulaması Tanımı	160
MVC Framework'leri	160
MVC Framework Olarak AngularJS	161
Mobil Uygulamalar ve Web Teknolojileri	162
Native (Yerel) Uygulamalar	162
HTML5 (WEB) Uygulamaları	162
Hibrit Mobil Uygulamalar	163
Bir Platform Olarak WEB Tarayıcıları	165
Neler Öğrendik?	167
Son Söz	167
Dizin	168

1

RESPONSIVE WEB TASARIMI KONSEPTİ

BU BÖLÜMDE

Responsive Web Tasarımı Nedir?	2
Bilinmesi Gereken Kavramlar	2
Neden Responsive Web Tasarımı	7
Mobil İnternet ve Cihaz Ekran Boyutları	8
Neler Öğrendik?	9

Kitabımızın ilk bölümünü Responsive Web Tasarımı ve fikrinin gelişim sürecinden bahsetmeye adamanın akıllıca olabileceğini düşünüyorum. Responsive Tasarımın ne olduğunu konusunda hiç ya da çok az bilgisi olan okurlarımıza bu konseptin ne olduğunu ve varoluş nedenini tanımlamalar yaparak anlatmaya çalışacağız.

Eğer Responsive web tasarımı konusunda genel bir bilgi birikimine sahip iseniz, bu bölüm konsepti genel ve kuşbakışı bir şekilde tekrar görmenize yardımcı olabilir.

RESPONSIVE WEB TASARIMI NEDİR?

Responsive Web Tasarımı ya da yerelleştirmek gerekirse, Durumlara Duyarlı Web Tasarımı; web tasarımcıların ve geliştiricilerin, oluşturdukları internet sitelerini ve web uygulamalarını her cihaz ekranı genişliğine uyumlu bir şekilde oluşturmak istemelerinden dolayı ortaya çıkmış bir tasarım ve yazılım akımıdır.

Bir internet sitesinin Responsive bir tasarıma sahip olması, internet sitesinin her cihaz ekran boyutunda kaliteli ve tahmin edilebilir bir görüntü sunması anlamına gelir.



Aslına bakılırsa Responsive Tasarım, sanılanın aksine mobil cihazlar yaygınlaşmadan önce ortaya çıkmış bir tasarım ve programlama düzenidir. 1024x768 genişliğindeki bir ekranda görüntülenmek üzere kodlanmış bir internet sitesi, 800x600 ekran boyutuna sahip bir ekranda görüntülenirken sorunlara mahal verebiliyordu. Hatta bazı internet siteleri siteye girmeden önce ekran boyutunu kontrol edip, ona göre size internet sitesinin sahip olduğunuz ekran boyutuna özel bir versiyonunu veriyordu.

Bu konulara bu bölümün içinde daha ayrıntılı bir şekilde değineceğiz, fakat terminolojiye tam anlamıyla hakim olmayan okurlarımız için kitap devamında kullanacağımız bazı kelimeler ve kelimelerinin özet olarak anlamlarından söz edelim.

BİLİNMESİ GEREKEN KAVRAMLAR

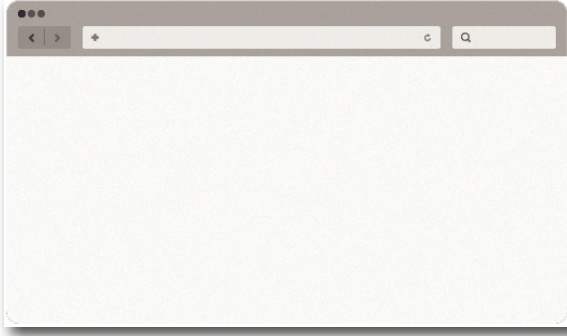
Terminoloji, her yığın bilgi konseptine sahip olan konuda olduğu gibi, web geliştiriciliği konusunda da büyük bir önem taşıyor. Bu başlık altında Responsive Tasarım ile ilgili olan web jargonunda bilinmesi gereken kavramlara bir göz atalım.

RESPONSIVE KELİMESİ VE RESPONSIVE TASARIM

Responsive, sözlük karşılığı **duyarlı** olsa da, aslına bakılırsa web için **Durumlara Duyarlı** olarak çevrilmesi gereken bir kavramdır. Responsive bir tasarım, farklı durumlara duyarlıdır. Örneğin; tasarımın görüntülediği ekranın genişliği, yüksekliği, dokunmatik olup olmaması, retinaya sahip olup olmaması cihazın elde taşınıp taşınmıyor olması **durumlara** verilebilecek örneklerden bazıları olabilir.

WEB TARAYICILARI

Web tarayıcıları, internet kullanıcısı ve kodlar arasında bir arayüz oluşturan, internet sitesinin kaynağını yorumlayabilen, gerektiğinde kullanıcıdan girdiler alan bir program-uygulamadır. Google Chrome, Microsoft Edge, Firefox veya Safari çok kullanılan web tarayıcılarına örnek olarak verilebilir.



Klasik bir masaüstü ya da dizüstü bilgisayarda bulunabildiği gibi, elde tutulabilen mobil cihazlarda da bulunan web tarayıcıları, çoğu aygıtta standart olarak bulunan yazılımların başında gelmektedir.

NOT

Web tarayıcıları günümüzde çok daha gelişmiş hale gelmiş ve çoğu ek özelliği yanında sunan birer platforma dönüşmüştür. Bu konu ile ilgili ayrıntılı bilgiyi kitabın devamında paylaşacağız.

KULLANICI ARAYÜZÜ - USER INTERFACE (UI)

Kullanıcı arayüzü ya da yabancı tabiri ile **User Interface**, kullanıcıların bir teknolojik aygıt üzerinde yapmak istediği işlemi gerçekleştirmek için kullanıcıya sunulmuş bir tür sistemdir. Akıllı telefonlarda bir uygulamayı açmak istediğimizi varsayalım, uygulamalarının anasayfada listelenmiş olması ve bu listelenmişlik durumu bir **Kullanıcı Arayüz (UI)** elemanıdır.

2

CSS VE JAVASCRIPT İLE RESPONSIVE ELEMENTLER

BU BÖLÜMDE

Responsive Element Nedir?
Neler Öğrendik?

12
29

Kitabımızın bu bölümünde birkaç kısa ısınma turu ile kod yazmaya başlıyoruz, Responsive elementlerin tasarımdaki rolünü ve nasıl kodlara döküldüğünü (implement) edildiğini ve tasarım bileşenlerinin media kuralları ile nasıl yönetilebileceğini örnek kodlar ile tartışacağız.

Basit HTML ve CSS özelliklerini adres ederek, CSS'in sunduğu Responsive tasarım kitine genel bir bakış atacağız. Bunun yanında bu kitin JavaScript alternatifi ile birkaç kod örneğini bu bölümde sunacak ve nasıl çalıştığı konusunda açıklamalar yapacağız.

RESPONSIVE ELEMENT NEDİR?

Tasarımlar bölümlerden, parçalardan, objelerden ya da teknik anlamıyla elementlerden (bileşenlerden) oluşur. Eğer tasarımınızın tümü veya çoğu elementleri ekran boyutuna duyarlıysa, genel anlamda konuştuğumuzda tasarımınız da ekran boyutuna duyarlı olur. Ekran genişliği ve yüksekliğine göre durum değiştiren elementlere tasarımların **Responsive Elementleri** denilir.

Birkaç örnek ile Responsive elementlerin tasarımınızı ekran boyutuna duyarlı hale getirme konusunda nasıl bir rol oynadığına göz atalım.

CSS'DE YÜZDE (%) KAVRAMI

CSS'de yüzde kavramı, tasarımların Responsive elementlerini oluştururken en çok kullanacağımız boyut değeridir. Bir örnek ile CSS üzerinde nasıl kullanıldığını görelim;

HTML Kodu

```
<html>
<head>
  <meta charset="utf-8" />
  <title>Responsive Design</title>
  <link rel="stylesheet" href="style.css"/>
</head>
<body>
  
</body>
</html>
```

CSS Kodu

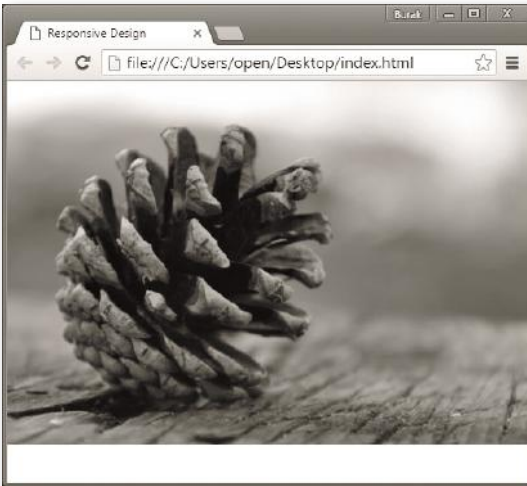
```
* { margin:0px; padding:0px; }
```



İlk görselimizde HTML çıktısını görebilirsiniz. **kozalak.jpg** resim dosyası, tarayıcıdan daha büyük boyutlara sahip olduğundan kaydırma çubuklarımız görünecek ve resmin tarayıcı boyutuna sığabilecek kadar olan bölümü görünür olacak.

CSS Kodu

```
* { margin:0px; padding:0px; }  
#responsiveResim { width:100%; }
```





Resimdeki ilk diyagramda ekran genişliği ve sitenin taşıyıcı elementinin genişliği birbirine eşit durumdadır. Bunun yanında 2. diyagramda ekran boyutunu 1000px'in üzerine çıkarttığımda, genel taşıyıcının 1000px değerinde sabitlendiğini ve daha da artmadığını görebilirsiniz. Bunu `max-width:1000px;` özelliği ile sağladık. Bu özelliğin ve yöntemin birçok internet sitesinde (Facebook, Instagram vb.) kullanıldığını görebilirsiniz.

JAVASCRIPT İLE YÜZDE (%) KAVRAMI

Kitabın bu bölümü boyunca Responsive Design yaklaşımının CSS'in genellikle yeni özellikleri ile implement edilmesi konusunda bize yardımcı olabilecek özelliklerini öğrendik. Asıl amacımız ekran boyutuna duyarlılıktır. Bu yüzden JavaScript ile de benzer uygulamalar gerçekleştirebiliriz.

NOT

CSS, çoğunlukla daha verimli bir sonuç vereceğinden ve kullanımı daha rahat olduğundan bu tür boyut tanımlamalarını ve hesaplamalarını CSS üzerinde yapmak daha mantıklı, fakat bazı durumlarda JavaScript kullanmak zorunda kalabiliyoruz.

Gelin isterseniz birkaç basit örnek ile yüzde kavramını JavaScript üzerinde uygulama yapalım...

HTML Kodu

```
<html>
<head>
  <meta charset="utf-8" />
  <title>Responsive Design</title>
  <link rel="stylesheet" href="style.css"/>
  <script type="text/javascript" src="script.js"></script>
</head>
```

4

RESPONSIVE SAYFA DÜZENİ VE UYGULAMALI ÖRNEKLER

BU BÖLÜMDE

Responsive Elementler ve Sayfa Düzeni 50
Neler Öğrendik? 69

Responsive Design'ın "Web Tasarımı" için belkide en önemli konularından birisi sayfa düzenini Responsive bir şekilde oluşturmaktır. Kitabımızın bu bölümünde Responsive bir sayfa düzenini, önceki bölümlerde öğrendiğimiz bilgiler ile nasıl oluşturabileceğimizden bahsedeceğiz.

Bölüme başlamadan önce eğer kendinize güvenmiyorsanız CSS'in temel özelliklerine kısa bir göz atmanızda fayda var gibi görünüyor.

RESPONSIVE ELEMENTLER VE SAYFA DÜZENİ

Responsive elementleri anlattığımız bölümlerde, eğer bir sayfadaki tüm elementler Responsive ile teknik olarak sayfa da tarayıcıya duyarlı olur demiştik. Sayfa düzenini (layout) elementler oluşturur. Öyleyse sayfa düzenini Responsive yapmak için yapmamız gereken tek şey sayfa düzenini oluşturan bu elementleri Responsive birer element haline getirmektir.

Çok basit ve önceki bölümlerde de göstermiş olduğumuz 2 kolonlu bir sayfa düzenini boyutlara göre bir örnek olarak modifiye edelim:

SAYFA YAPISINI YENİDEN ORANLAMA TEKNİĞİ

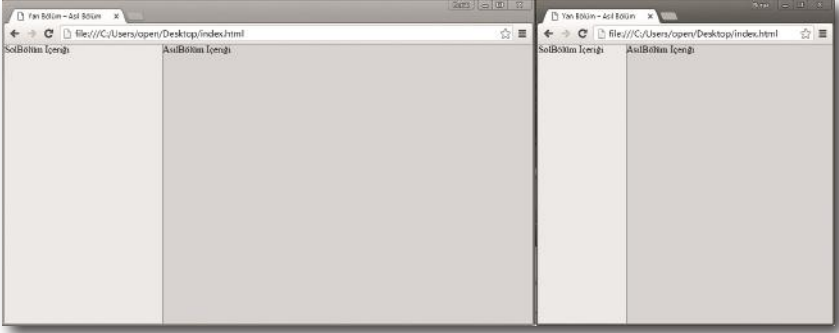
Çoğu blog sitesinde görmeye alışkın olduğumuz bir sayfa düzeni olan **Yan Bölüm - Asıl Bölüm Yapısı** örneğini en basit şekilde kodlara dökmeye çalışalım. Bu örneğimizde sayfa yapısını yeniden oranlama tekniğini kullanacağız.

HTML Kodu

```
<html>
<head>
  <meta charset="utf-8" />
  <title>Yan Bölüm - Asıl Bölüm</title>
  <meta name="viewport" content="width=device-width">
  <link rel="stylesheet" href="style.css"/>
</head>
<body>
  <div class="tasiyici">
    <div class="solbolum">SolBölüm İçeriği</div>
    <div class="asilbolum">AsılBölüm İçeriği</div>
  </div>
</body>
</html>
```

CSS Kodu

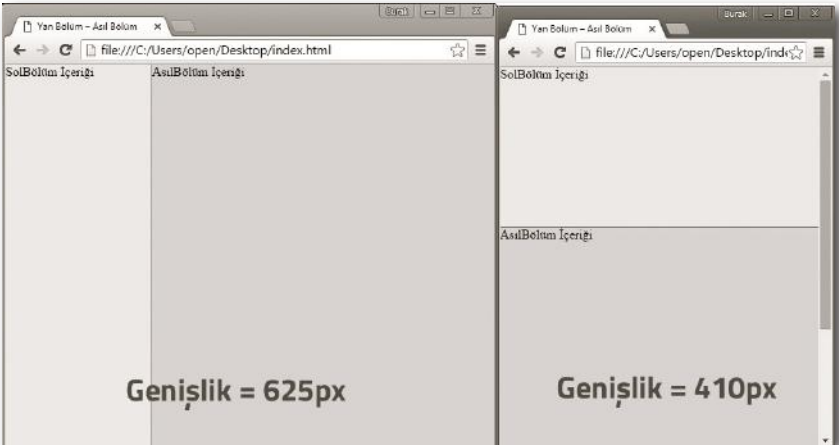
```
* { margin:0px; padding:0px; }
.tasiyici { position:relative; width:100%; max-width:1000px; margin:auto; }
.solbolum { width:calc(30% - 1px); background:#eee; height:100%;
  border-right: 1px solid #888; float:left; }
.asilbolum { width:70%; background:#ddd; float:right; height:100%; }
.temizle { clear:both; }
```



Örneğimizin çıktısında 1200px ve 800px genişlikleri için kodlarımızın çıktısını görebilirsiniz. Örneğimizin sayfa düzeninde 1000px'i geçtiğinde ekran boyutunun sabitlendiğini görebilirsiniz. Bunun yanında ekran boyutu küçüldükçe ekran boyutunun %30'unu genişlik olarak alan `.solbolum` div'i istenilenden çok daha küçük bir hal alacak. Bu durumu düzeltmek için 500px için bir kırılma noktası atayalım;

CSS Kodu

```
/* Önceki Örnek ile aynı CSS kodları */
@media all and (max-width: 500px){
    .solbolum { width: 100%; border-right:0px; max-height:200px;
                border-bottom:1px solid #888; float:none; }
    .asilbolum { width: 100%; float:none; }
}
```



6

SVG KULLANIMI VE SVG İKONLAR

BU BÖLÜMDE

SVG Nedir?

86

Neler Öğrendik?

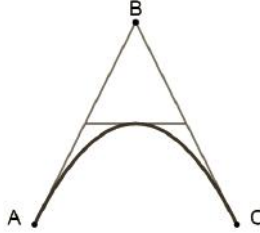
103

Responsive web tasarımında, çok farklı ve fazla sayıda aygıt üzerinde çalıştığınızdan dolayı performans konusu önemli bir yer alır. Resimlerin yüklenmesi ve yönetilmesi açısından tasarımsal bazı sorunları ortadan kaldırmak için vektörel grafikler ve SVG kullanıyoruz.

Kitabın bu bölümünde vektörel imajları klasik resim formatında sunmaktansa XML şeklinde sunarak trafikten ve performanstan kazanç sağlamamıza ön ayak olacak SVG formatından ve SVG'nin en fazla kullanılan uygulaması olan ikonlarının nasıl kullanılacağına göz atacağız.

SVG NEDİR?

SVG (Scalable Vector Graphics), XML'in söz dizimi (syntax) üzerine oturtulmuş HTML içine gömülebilir bir tür boyutu isteğe göre değiştirilebilir vektör grafik formatıdır. HTML5'in gelişmesi ile ileride büyük bir kullanım alanı oluşabilir. Sistem bağımsız bir geliştirme grubu tarafından geliştiriliyor. SVG, grafiklerin geometrik olarak kodlanmış, düşük boyutlu data ile birbirine bağımlı grafikler oluşturmamıza olanak sağlıyor.



Referans resmimizde gördüğünüz grafik SVG kodları ile oluşturulmuş ve bu öge istenilen boyutta piksellenme olmadan büyütülebilir halde.

SVG'nin mantığını anlatmak için şu tür bir örnek verebiliriz. Örneğin; internet sitenizde vektör tabanlı örnek veriyorum Adobe Illustrator'da tasarladığınız bir grafiği kullanacaksınız, bunu PNG ya da benzeri bir resim formatında render edip sayfanıza eklemeniz gerekiyor. Fakat bildiğiniz gibi sayfaya bir resim eklemek her zaman sayfanın açılış hızını yükseltir. HTML ve CSS'in gelişmesiyle artık göze hoş görünen resim kullanılmayan web sayfaları oluşunca, bir grup insan demiş ki, neden bir vektör grafiklerini XML benzeri bir syntax içinde oluşturuyoruz? Bu fikirle ortaya çıkan bu grafik formatı, durumun karmaşıklığına göre, sayfaya embed image olarak resim eklemekten daha hızlı sonuç vermektedir.



Yukarıda gördüğünüz ekran görüntüsünü **CodyHouse.co** isimli bir web sitesinin ana sayfasından aldım. CSS konusunda bilgisi olan arkadaşlar, eğer SVG'den habersizse bunun klasik bir vektörel PNG resmi olarak arkaplan (background) ayarlandığını düşünebilir. Fakat bu çalışmada hiçbir klasik resim formatı kullanılmamış. Gördüğünüz bilgisayar, tablet, kitaplar hatta ve hatta kaktüs, SVG ile kodlanmış ve içerisinde herhangi bir resim formatı içermiyor. Hızlı olmasının yanında SVG'inin en güzel yanlarından birisi, örneğin isterseniz tek bir değişkeni değiştirerek bilgisayarın arkaplan rengini değiştirebilirsiniz.

SVG'NİN KLASİK RESİM FORMATLARINA KARŞI FAYDALARI

Peki neden PNG gibi klasik bir resim formatıyla resimlerimizi kullanmak yerine vektörel öğeler için SVG kodlama düzenini kullanıyoruz?

Çok Daha Hızlı!

Her ne kadar gelişen internet sistemleri sayesinde sayfaların yüklenme hızı arasında hissedilir bir değişiklik hissedilmeyecek olmasına rağmen çok fazla kişinin uğradığı sayfalarda SVG sistemi kullanılarak sunucunun trafiği ve sayfa yüklenme hızında gözle görülür bir değişme sağlayabiliyorsunuz. SVG'nin en büyük albenisi de zaten budur.



İsim	Boyut	Tür	Değişme
file.png	52.451	PNG resmi	11.10.2015 16:35
file.svg	4.313	SVG Belgesi	11.10.2015 16:35

Referans resminde içerisinde websafe olmayan bir font ile yazılmış 5 harflik 300x150 boyutlarında bir logo bulunuyor. Bu logo Adobe Illustrator üzerinde hem PNG hem de SVG olarak çıktı alındı. 10x'dan büyük bir boyut farkı bulunduğunu görebilirsiniz.

NOT

Peki SVG bunu nasıl sağlıyor? PNG ve benzeri klasik resim formatları piksel tabanlı çalışır, her piksel için renk kodları tutar, bunun yanında SVG vektörel çalışır ve grafik ortamdaki vektörlerin yalnızca konumu, açısı ve rengini tutar. Bu da büyük bir verimlilik sağlar.

8

MOBİL ÖNCELİKLİ RESPONSIVE WEB TASARIMI

BU BÖLÜMDE

Mobil Öncelikli Tasarım Nedir?	128
Neler Öğrendik?	143

Web Tasarımcıları, tasarımlarını genellikle büyük bir ekranda oluşturduklarından, tasarımların genellikle **masaüstü öncelikli** bir hal aldığını, masaüstünde web sitesinin sağladığı bazı özelliklerin, daha küçük ekranlara sahip mobil aygıtlarda erişilebilir olmadığını gözlemlemiştir.

Bu bölümde, bu durumu ortadan kaldıran ve kullanıcılara her platformda aynı işlevi sunmamıza olanak sağlayan bir tasarım yaklaşımı olan mobil öncelikli web tasarımından bahsedeceğiz.

MOBİL ÖNCELİKLİ TASARIM NEDİR?

Mobil öncelikli tasarım ya da **mobile-first** yaklaşımı, tasarımcının tasarımını oluştururken, küçük ekrana sahip cihazlardan yola çıkması ile tasarımını ekran boyutuna göre uyarlanabilir hale getirmesi durumuna denilir. Bu durumun masaüstünden mobilde değil de mobilden masaüstüne doğru geliyor olmasının en büyük nedeni mobil kullanıcılarında büyük bir artış yaşanmasıdır.

Yapılan bağımsız araştırmalara göre internet kullanıcılarının ortalama %25 kadarı yalnızca mobil internet kullanmaktadır ve bu sayı gitgide her yıl ivmeli bir şekilde artmaktadır. Bu durum tasarımcıların teknolojinin gelişmesi ile masaüstü öncelikli tasarımlardan çok mobil öncelikli tasarımları tercih etmesine yol açtı.



NOT

Yalnızca mobil internet kullanan %25'lik kesim istatistiği şu şekilde yorumlanabilir. Kişisel bilgisayar sahibi ya da erişimi olan internet kullanıcıların büyük bir çoğunluğu mobil internet de kullanmakta fakat bunun yanında mobil internet kullanan kullanıcıların %25'i kişisel bilgisayar ile internet kullanmıyor.

Yalnızca mobil internet kullanıcılarının karşılaştığı en büyük sorunlardan bir tanesi, bazı web sitelerinin mobil versiyonlarının, klasik versiyonları ile aynı işlevsel özelliklere sahip olmaması. Bu tür durumlara karşılaşmamak için mobil öncelikli tasarımlar en mantıklı ve işlevsel seçenek gibi durmakta.

Mobil öncelikli tasarımın sağladığı faydalar;

- » **Her ekran boyutu için eşit işlevsellik:** Yalnızca mobil internet kullanan internet kullanıcılarının de web sitesinden, klasik internet kullanıcısı ile aynı randımanı alabilmesini sağlar ve kolaylaştırır.
- » **Daha hızlı arayüz/websitesi:** Mobil öncelikli tasarlanmış bir web sitesi masaüstü öncelikli tasarlanmış bir web sitesine göre genellikle daha sade ve işlev odaklıdır. Bu da kodların daha az olması ve sitenin daha hızlı yüklenmesi anlamına gelir. Bu durum veri trafiğinin azalmasına da katkıda bulunmaktadır.
- » **Tanımlırlığı artırır:** Bazı internet sitelerinin mobil ve masaüstü versiyonları arasında hem konsept hem de yapı olarak büyük farklar bulunur. Mobilde bir in-

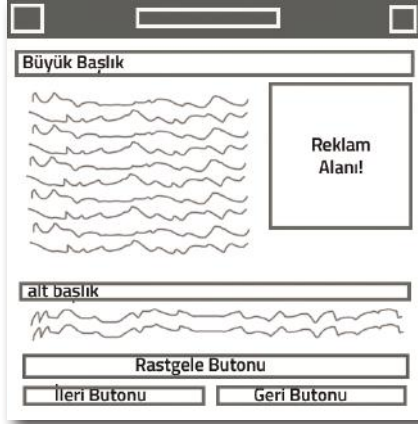
ternet sitesini keşfeden internet kullanıcısı, aynı internet sitesi ile masaüstünde karşılaştığında siteyi tanır ve memnun kaldı ise tekrar kullanır. Bu durum markalaşmaya da büyük bir yardım sağlamaktadır.

MOBİL ÖNCELİKLİ TASARIM OLUŞTURMA VE ADIMLARI

Mobil öncelikli tasarımın, estetikten çok işleve odaklandığından bahsetmiştik. Bunun yanında ekran boyutu arttıkça kullanıcıya sunulan içeriğin değerini değiştirmeden gösterilebilir hale getiriyor olmamızın mobil öncelikli tasarımın en büyük albenilerinden biri olduğunu örnekler ile sunduk. Mobil öncelikli tasarımlar oluşturulurken dikkat edilmesi gereken tek nokta ekran boyutu da değil. Hazırlanan herhangi bir Responsive tasarımda olduğu gibi, oluşturulan arayüzün/tasarımın, aygıtın desteklediği özellikler ile paralel olarak geliştirilmesi büyük bir önem taşıyor. Mobil öncelikli tasarımlarınızı oluştururken dikkat etmeniz gereken bazı adımlar şöyle sıralanabilir:

1. Arayüzde basit taslak tasarımlar ile çalışmaya başlayın!

Arayüzünüzü oluştururken, estetik kaygısı ön planda, karmaşık ve fazla imaj dosyası içeren bir arayüz ile çalışmaya başlamayın. Bunun yerine, kullanıcı tecrübesine odaklanarak tasarımların taslak hali üzerinde çalışmalarınızı geliştirin, bunun üzerine sonrasında modifikasyonlar ile estetik olarak istediğiniz görünümü rahatlıkla verebilirsiniz.



Ben bu kitabı yazarken çalıştığım en son iş için şöyle bir tasarım hazırladım. Bu tasarımda müşteri tarafından önceden belirlenmiş işlevler göz önüne alarak ekranda öğelerin nerede durmasını istediğimi HTML ve CSS yardımı ile konumlandırdım. Burada dikkat ettiyseniz taslak sadece öğelerin boyutlarını ve konumlarını içeriyor.