**Kod 2: Veriler ve Değişkenler**

**Veri ve Değişkenler**

* Veri
* Veri türleri
* Değişkenler
* Lokal ve global değişkenler
* Listeler
* Veritabanları
* Kod Görevi
* Özet

**Konular:**

Bu ünitede…

* Veriler ve çeşitli veri türlerini,
* Değişkenleri ve veritabanını nasıl kullanmanız gerektiğini,
* Verileri nasıl listelere dönüştüreceğinizi,
* Veritabanına bağlı bir “yapılacaklar ” uygulaması yapmayı öğreneceksiniz.

Bu ünite çok fazla yeni bilgi içeriyor bu nedenle ilk seferde hepsini anlamayabilirsiniz. Daha iyi anlamak için tüm kod eğitimlerinin üstünden geçip kodların işlevlerini kendiniz denemelisiniz. Anlamadığınız yerleri her zaman geri dönüp tekrar edebilirsiniz.

Veriler

Hayatınızın bir bölümünde mutlaka **veri** kelimesini duymuşsunuzdur. Veri bir çok anlamda kullanılabilir, ama kodlamada **veri,**uygulamanızın çalışmak için ihtiyaç duyduğu bir takım bilgilerdir. Bu kavramı daha iyi anlamak için bir kaç örnek düşünelim.



**WhatsApp**

Bu uygulama insanları arayabilir veya mesaj atabilir. Bu uygulamanın ihtiyaç duyduğu veriler:

* Kullanıcıadı
* Arkadaşlarınızın kullanıcı adları
* Göndermek istediğiniz mesajlar
* Zaman
* Konumunuz



**Angry Birds**

Bu oyunda domuzları, kuşlar ile ateş ederek yenmeye çalışıyorsunuz. Bu uygulamanın ihtiyaç duyduğu veriler:

* Puanınız
* Her objenin puanı
* Hangi seviyede olduğunuz
* Her seviyede kaç kuşunuzun olduğu



**Slideshow Challenge App**

Bu uygulama Kod Görevi1 ‘de yaptığınız uygulama. Bu uygulamanın ihtiyaç duyduğu veriler:

* En sevdiğiniz fotoğraflarınız

Uygulamanın gereken tüm verileri alabilecek bir şekilde kodlanması gerekiyor. Bilgisayarların hiç bir şeyi yönerge almadan kendi başlarına yapamayacağını hatırlayın. App Inventor’da uygulamalar şu şekilde veriler toplayıp kullanabilirler:

1. **Programlayan** kişi bilgileri girebilir:
   * Slideshow uygulamasında siz (yani programlayan kişi) en sevdiğiniz fotoğrafları girdiniz.
   * ANgry Birds’de programlayan kişi her seviyede kaç kuş olacapı bilgisini girdi
2. **Kullanıcı** bilgiyi girebilir:
   * Whatsapp’ta kullanıcı göndermek istediği kişiyi ve mesajı giriyor.
3. **Uygulama** başka bir yere bağlanarak veri alabilir:
   * Whatsapp telefonunuza bağlanarak mesajı attığınız saati kaydeder.

Bazen uygulamalar kullanıcının girdiği bilgileri kaydeder. Örneğin, whatsapp mesajları kaydederek arkadaşınıza attığınız eski mesajları görebilmenizi sağlar. Bunu yapabilmek için programlayan kişi **veritabanı (database**) kullanır. Bu derste veritabanlarını öğreneceksiniz. Veritabanları hakkında daha detaylı olarak bir sonraki derslerde bahsedeceğiz.

Alıştırma

En yakındaki marketleri bulan bir uygulama yapacak olsaydınız uygulamanız hangi tür veriye ihtiyaç duyar?

Veri Türleri

Yukarıdaki kısımda gördüğünüz gibi veriler bir çok şekilde görülebilir. Bilgisayarların hangi veri ile çalışmaları gerektiğini anlamaları için veriler farklı kategorilere ayrılır. Bu **veri türleri** olarak biliniz. Veriler üç ana başlık altında gruplandırılır: sayılar (numbers), yazılar (strings) ve booleans (doğru/ yanlış). App Inventor resimler ve renkler gibi daha karmaşık veriler kullanmanıza da izin verir.

**Sayılar (Numbers)**

Sayılardan oluşan verileri kullanmak için App Inventor’da “Math” bölümüne girebilirsiniz.

http://www.technovationchallenge.org/wp-content/uploads/2016/09/image30.png

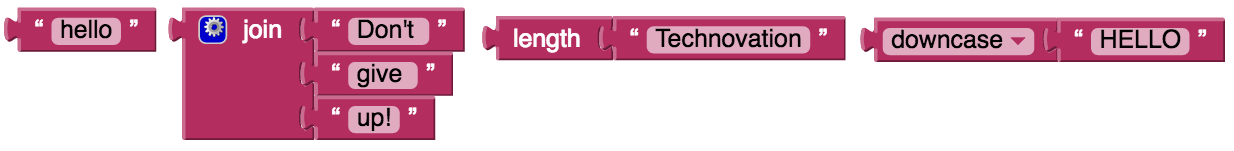
Sayı kullanmanız gerektiğinde hep “Math” bölümündeki blokları kullanmanız gerekecek. Aşağıda sayı içeren veriler kullanmanız gerekebilecek durumlar var:

* Bir oyunda oyuncunun puanını hesaplarken
* Bir kullanıcının kaç kez giriş yapmayı denediği
* Uygulamanızın bir şeyi kaç kez yapması gerektiğini saymanız gerektiğinde
* Kullanıcının belirli bir yaşta olup olmadığına bakarken

Matematik bloklarıyla yapabileceğiniz diğer şeyleri öğrenmek için MIT’in sitesine bakabilirsiniz: [Matematik Blokları](http://appinventor.mit.edu/explore/ai2/support/blocks/math.html)

**Yazılar (Strings)**

Bir string (yazı) veri türü yazıdır. App Inventor’da bloklarda Text’e giderek bir string (yazı) oluşturabilirisiniz. Aşağıda kullanabileceğiniz bazı bloklar yer almaktadır.



Kodunuza metin alanı eklemek istediğinizde  string (yazı) kullanabilirsiniz. Aşağıda uygulamanızda ne zaman yazı veri türü kullanabileceğinizin örnekleri verilmiştir.

* Kullanıcının bir kullanıcı adı veya şifre girmesini sağlamak
* Uygulamanın kullanıcıya kelimelerle konuşması
* Kullanıcıya bir mesaj veya uyarı göstermek.
* Uygulamayı bir ekran ismi vererek yeni bir ekran sayfası açmasını sağlamak için programlamak.

Devam etnmeden MIT’nin sitesinden yazılarla yapabileceğiniz her şeyi detaylı bir şekilde öğrenebilirsiniz. [Yazı Blokları](http://appinventor.mit.edu/explore/ai2/support/blocks/text.html#string)

**Boolean**

Boolean’lar doğru veya yanlış olmak üzere iki farklı veridir. Doğru “evet” anlamına yanlış “hayır” anlamına gelir. Boolean bloğu almak için “logic” kısmına girip alttaki bloklardan birini alın.



Şimdilik boolean’la rile yapabileceğiniz çok şey yok. Fakat aklınızda tutmanız gerekli çünkü ileride çok kullanacaksınız. [Kod 3](http://www.technovationchallenge.org/curriculum/code-3/)dersinde boolean’ları daha detaylı bir şekilde öğreneceksiniz.

Değişkenler

Kodlamada en önemli elementlerden biri değişkenlerdir. Bir değişken sürekli değişebilen bir değerin adıdır. Bunu*“değişmek”* kelimesinden aklınızda tutabilirsiniz. Aşağıdaki maddeler bazı değişken örnekleridir:

* Yaşınız
* Boyunuz
* Konumunuz
* Tarih ve saat

Değişkenleri daha iyi anlamak için App Inventor’da bir örnek yapalım.  Kullanıcının puan kazanıp kaybedebildiği bir oyun yaptığınızı düşünün. Bu skorun gözükmesi için ekrana bir label koyun. Oyunun başında kullanıcının sıfır puanı olmasını istiyorsunuz.



Kullanıcı ekranda sıfır yazısını görür. Ama skor kullanıcı oyunu oynadıkça değişir. Kullanıcı her puan kazanıp kaybettiğinde bu yazıyı yenilemeniz gerekir. Kullanıcı puan kazandığında aşağıdaki kodu yazmanız gerekir.



Bu kod da çalışır fakat ya kullanıcı tekrar 10 puan kazanırsa? Veya 20 puan kaybederse? O zaman skoru tekrar değiştirmeniz gerekir. Bu, bir süre sonra çok karışık bir hal alabilir.

Böyle durumlarda  bir değişken yaratıp oyuncunun skorunu takip edebilirsiniz. “score” (“skor”) adlı bir değişken yaratıp bu değişkenin her zaman kullanıcının skorunu göstermesini sağlayabilirsiniz. Kullanıcı oyunu oynadıkça “score”a puan ekleyip puan çıkarabilirsiniz. Böylece o anki skoru hesaplamakla uğraşmak zorunda kalmazsınız. App Inventor’da bunu yapmak için aşağıdaki bloğu alın.



This block tells the app that there will be a variable and asks you to give the variable a name and a value. Here’s what the block would look like for a game where the player’s score started at zero.



Bu blokla uygulamaya “score” adlı bir değişken olduğunu söylüyorsunuz. Bu değişken oyun oynandıkça değişecek. Uygulama her açıldığında uygulama “score” adlı bir değişken oluşturup değerini sıfır yapacak. Şimdi “set”(ayarla)  ve “get” (al) bloklarını kullanabiliriz. “Get” (al) bloğu bir veri almanızı yani skoru görmenizi sağlayacak. Aşağıda skoru label içinde nasıl göstereceğinizi görebilirsiniz.



“Set” (ayarla) ile değişkene yeni bir değer tanımlamanızı sağlar.



Bu kod uygulamanıza skoru kullanıcının o anki skoruna eşitlemesini sonra da buna kazanılan puanı eklemesini söyler. Eğer o anki skor 10 ise yeni skor 10 olur. Eğer o anki skor 25 ise yeni skor 35 olur. Şimdi kullanıcı her puan kazanıp kaybettiğinde skoru yenilemekle uğraşmayacaksınız. Uygulama o anki skoru alacak ve buna 10 ekleyecek.

Zor Alıştırma

En sevdiğiniz uygulamalardan birini seçin ve nerelerde değişkenler kullanıldığını bulmaya çalışın.

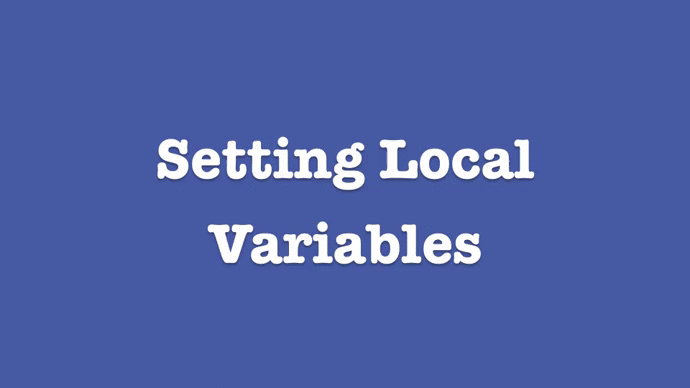
Lokal ve Global Değişkenler

Yukarıdaki kısmı yaparken değişkenin üzerinde “global” yazdığını fark etmişsinizdir. İki farklı değişken türü vardır; **lokal değişkenler**ve **global değişkenler.**Değişkenler yarattığınızda bu değişkeni kodunuzda sadece bir yerde mi yoksa farklı farklı yerlerde mi kullanacağınıza karar vermeniz gerekir.

**Global** **değişkenler**kodunuzda her yerde kullanılabilirler.  Yukarıda da gördüğünüz gibi skor değişkenini farklı farklı yerlerde kullandık. Bu değişkeni kullanıcı puan kazandığında, kaybettiğinde skoru yenilemek ve label aracılığıyla gösterirken kullandınız. Bu aynı zamanda en yüksek skor tablosu yapmak için de kullanılabilir. Global değişkenlerle çalışırken hiç bir zaman bir adı iki kez kullanamazsınız. Eğer aynı adı iki veya daha fazla değişken için kullanırsanız uygulamanız hangi değişkenin hangisi olduğunu ayırt edemez ve çalışmaz. Zaten App Inventor aynı ada sahip iki farklı değişken yaratmanıza izin vermiyor. Eğer hangi değişken çeşidini kullanmanız gerektiğinden emin olamazsanız Global Değişken’i kullanın.



**Lokal Değişkenler**sadece kodunuzun bir bölümünde kullanılabilir.  Eğer bir değişkeni kodunuzun sadece bir bölümünde kullanacaksanız lokal değişken kullanın. App Inventor sadece o lokal değişkeni oluşturduğunuz bloğun içerisinde kullanmanıza izin verir ve başka yerde kullanamazsınız.  Global değişkenlerden farklı olarak lokal değişkenlerin adlarını birden fazla kez kullanabilirsiniz.



Kod Görevi 2’de ne zaman hangi değişken türünü kullanacağınıza dair örnekler göreceksiniz. Lokal ve global değişkenler hakkında daha fazla bilgi için MIT APp Inventor’un sitesinden aşağıdaki linki ziyaret edebilirsiniz.

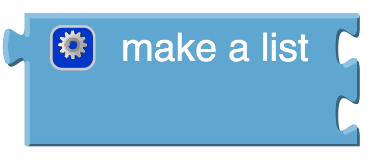
[Global ve Lokal Değişkenler](http://appinventor.mit.edu/explore/ai2/support/concepts/variables.html)

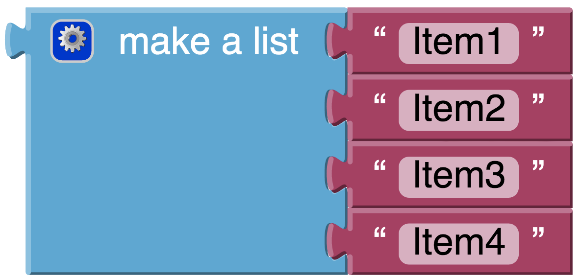
Listeler

Dünyada hayal edemeyeceğimiz kadar çok bilgi var ve bu bilgiler tek başlarına etrafta dolaşırken çok karışık bir hal alıyor. Bu nedenle bu verileri düzenlemek için çeşitli teknikler var. Verileri düzenleyerek bu verilere daha kolayca ve hızlı ulaşabiliriz. Hayatınızdaki obje ve bilgileri nasıl düzenlediğiniz hakkında örnekler düşünebiliyor musunuz? Aşağıda bazı örnekleri görebilirsiniz:

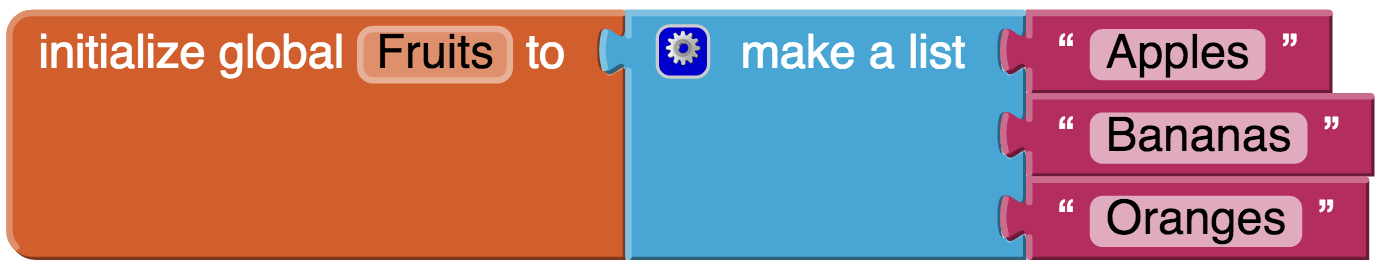
* Arkadaşlarınızın telefon numaralarını telefon rehberinize kaydederek
* Kalemlerinizi kalem kutunuza koyarak
* Kıyafetlerinizi dolaba koyarak

Programlamacılar listeler kullanarak uygulamalarındaki verileri düzenlerler. Listeler içinde birden fazla veri barındırabilir ve bu verileri kolayca aktarabilir. Daha önceden hepimiz alışverişe giderken bir liste yapmışızdır, App Inventor’daki listeler de bunlara çok benzer. App Inventor’da bloklar kısmına gidip “lists” bölümüne giriniz.





  
Listeler bilgileri tutarken bize çok yardımcı olur. Bir listedeki her maddenin bir *“index”*i vardır. “index” size o maddenin listede kaçıncı sırada olduğunu verir. Listedeki ilk maddenin “index”i bir, ikincisinin 2’dir. Ve bu böyle devam eder.

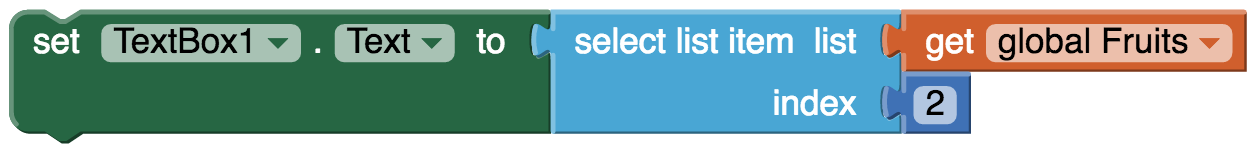


Uygulamanız bu bilgiyi şu şekilde görür:

**Liste adı: Meyveler (Fruits)**

* Apples (index=1)
* Bananas (index=2)
* Oranges (index=3)

Uygulamanızda her zaman tüm listeyi almak yerine bazen bir listenin içerisinden bir madde de alabilrisiniz. Örneğin “banana”(muz) yazan maddeyi bir label ile görüntülemek istiyorsanız uygulamanıza index’i 2 olan maddeyi almasını söylemelisiniz.



Listelerdeki şeylerin yerlerini değiştirebilir, silebilir, bir şey ekleyebilir veya bir maddeyi değiştirebilirsiniz. Örneğin “kiviler” ve “üzümler” maddelerini listenize eklemeyi unuttuysanız aşağıdaki blokları kullanarak ekleyebilirsiniz.

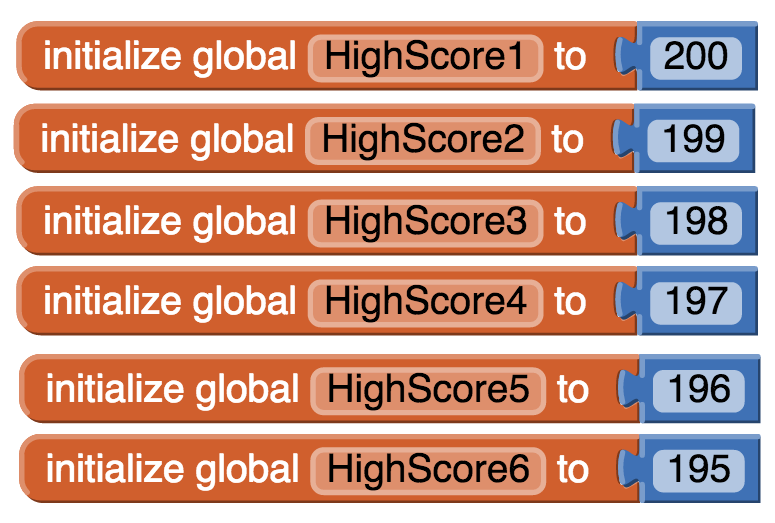
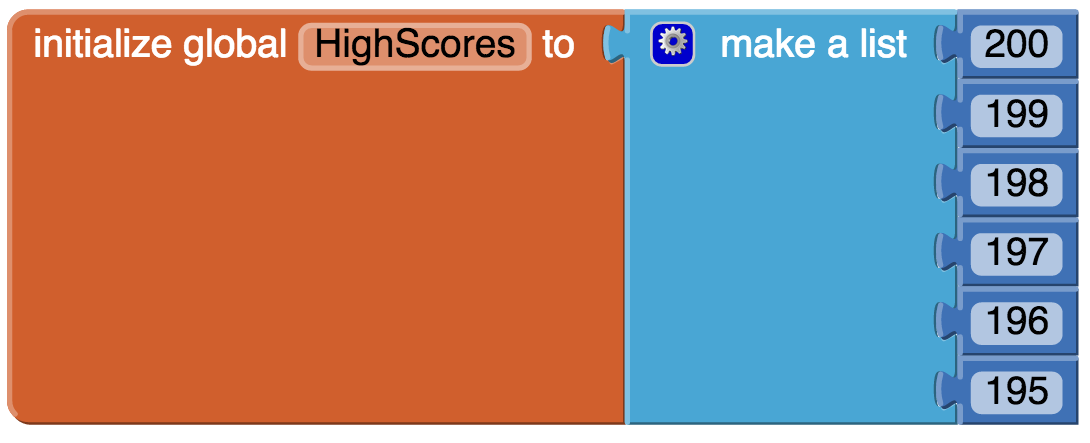


Şimdi uygulamanız listeyi şu şekilde görecek.

* Elmalar (Indeks = 1)
* Muzlar (Indeks = 2)
* Portakallar (Indeks = 3)
* Kiviler (Indeks = 4)
* Üzümler (Indeks = 5)

Elinizde birden fazla veri olduğunda ve bu verileri bir değişken altında toplamak istediğinizde listelerden yararlanmalısınız. Örneğin, bir oyundaki en yüksek 10 skoru görüntülemek istiyorsanız HighScore1 (YüksekSkor1), HighScore2…HighScore10 gibi 10 tane değişken oluşturabilirsiniz.

İkinci ve dah akısa bir yol olarak HighScores (YüksekSkorlar) adlı bir liste oluşturup bunun içine 10 yüksek skoru koyabilirsiniz ve bunları index’lerini kullanarak görüntüleyebilirsiniz. Liste kullanarak hem yeni skorlar ekleyip çıkarabilir hem de yerlerini değiştirebilirsiniz. Böylece oldukça zaman kazanırsınız.

 (enyüksekskor1)  
 (ok)  
 (enyüksekskor2)

MIT’nin sitesinden bu konu hakkında daha fazla bilgi edinebilirsiniz:  [listeler](http://appinventor.mit.edu/explore/ai2/support/blocks/lists.html#string)

*Not: Bir sürü programa dilinde listeler “arrays” olarak adlandırılır ve index’ler 1 değil 0’dan başlar. Bu nedenle yeni bir programlama dili öğrenirken kurallarına baktığınıza emin olun.*

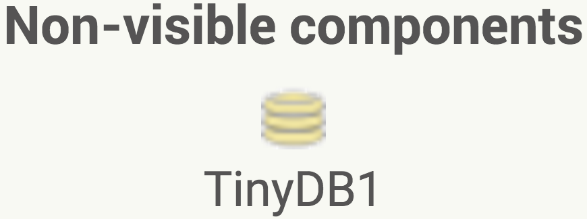
**Veritabanları (Databases)**

Değişkenler, kullanıcı uygulamayı kullanırken veri depolamak için olan yöntemlerden biri ama eğer kullanıcı uygulamayı kapatıp açınca da depolanan verilerin korunmasını isterseniz bir **veritabanı (database)** kullanmalısınız. Veritabanları daha sonradan kullanmak üzere uygulama kapalıyken bile veri toplayabilirler. Veritabanlarını sonradan veri alabileceğiniz bir depo odası gibi düşünebilirsiniz. Aşağıda her gün kullandığımız bazı veri tabanı örnekleri var.

* Telefon rehberi- arkadaşlarımızın numaralarını daha sonradan kullanmak için kaydederiz
* Mesajlaşma uygulamaları- bu uygulamaların çoğu daha sonradan mesajları okuyabilmek için kaydederler
* Defter- daha sonradan okumak üzere defterinize yazılar yazarsınız.

App Inventor’da **Tiny DB**ile kullanıcının telefonuna bilgileri kaydedebilirsiniz ve kullanıcı her uygulamaya girdiğinde bu verileri geri alabilirsiniz. Ama Tiny DB verileri sadece bir cihaza kaydeder yani lokal bir veritabanıdır. İki farklı cihaz arası veri akışını sağlamaz. Bir kullanıcı TinyDB ile bir veri kaydettiğinde bu veri sadece onun telefonunda kayıtlı olacaktır. Farklı cihazlar arası bilgi alışverişini nasıl yapacağınızı [Kod](https://technovationchallenge.org/curriculum/kod-2-veriler-ve-degiskenler/www.technovationchallenge.org/curriculum/code-4/)[4: Mobil Özelliklerin Kullanımı ve İnternet ile Bağlantı](http://www.technovationchallenge.org/curriculum/code-4/)dersinde öğreneceksiniz.

Tiny DB kullanmak için App Inventor’un “design” (tasarım) kısmından TinyDB alın ve ekranınızın altına sürükleyin. Bu görünmez bir elementtir. Kullanıcı bunu görmez.

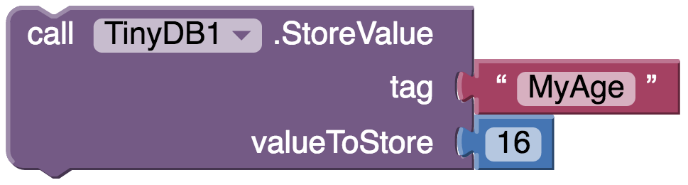
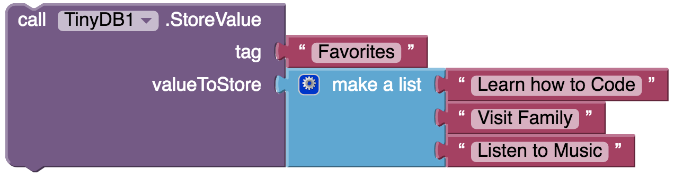


TinyDB’e bir görev vermek için “call” (çağır) bloğu kullanılır. TinyDB’i çağırdığınızda onda veri depolayabilir veya ondan veri alabilirsiniz. TinyDB’de bir veriler “tag”ler (etiketler) altında depolanır. Bu etiketler verileri geri alırken de kullanılır. Bu da değişken adları gibidir. Eğer iki farklı veriye de aynı etiketi verirseniz App Inventor eski veriyi siler ve yerine yenisini yazar. Bu veritabanınızdaki bilgileri yenilemek istediğinizde işinize yarayabilir ama onun haricindeki durumlarda aynı etiketi iki kez kullanmadığınızdan emin olmalısınız. Aşağıdaki TinyDB’in nasıl kullanıldığını görebilirsiniz.


İlk blok, TinyDB’de değerlerin nasıl saklanacağını gösterir. Bu durumda, meyveler listemizi daha önce “Gıda” etiketi ile sakladık. İkinci blok, meyve listesinin veritabanından nasıl alınacağını gösterir. Etiketin, tüm büyük harfler dahil olmak üzere, verilerin saklandığı sırada olduğu gibi tam olarak yazılması gerekir.

Bunu daha iyi anlamak için bir örnek üzerinde yürüyelim. Daha sonra kullanmak için bir veritabanında üç şeyi saklamanız gerektiğini varsayalım. Birincisi son bölümdeki Meyve listemizdir, bunlardan biri sizin yaşınız, diğeri yapmak için en sevdiğiniz şeylerin listesi. Veritabanına şu şekilde üç çağrı yaparsınız:

Artık veritabanınızda üç giriş var ve bunlar neye benziyor:

Etiket Adı: “Yemek”

Veri:

* “Elmalar” (indeks = 1)
* “Muzlar” (index = 2)
* “Portakallar” (indeks = 3)

Etiket Adı: “Benim Yaşım”

Veri

* 16

Etiket Adı: “Favori”

Veri:

* “Kodlamayı Öğrenin” (indeks = 1)
* “Ailenizi ziyaret edin” (indeks = 2)
* “Müzik dinleyin” (indeks = 3)

Şimdi, en sevdiğiniz şeyleri almak istediğinizde, böyle bir çağrı yapın:



Bu bloğu kullandığınızda “Kodun Nasıl Kullanılacağını Öğrenin Müzik Dinle’yi Öğrenin”. Şimdi yaşınızı almak istediğinizi söyleyelim, böylece böyle bir çağrı yapın:



Bu bloğu kullandığınızda boş bir dize (string) alırsınız: “”. Bunun nedeni veritabanınızda “benimadım” etiketinin bulunmamasıdır! Bununla birlikte, “BenimAdım” etiketi mevcut. Veritabanınız etiketi tanımıyordu, size boş bir dize olan “eğerDeğerOradaDeğilseEtiketle” (“valueIfTagNotThere)” in yanındaki bloğu gösterecek şekilde varsayılan olarak ayarlandı. Bu dize istediğiniz herhangi bir şey söyleyebilirsiniz. Hatta kullanıcınıza bir hata mesajı bile yapabilirsiniz.

Birden fazla ekranlı bir uygulama oluşturursanız (slayt gösterisi uygulamanız gibi), bileşenler ve değişkenler her ekran arasında birbirleriyle konuşamaz. Uygulamanızın bir ekranından diğerine bilgi aktarmak için TinyDB kullanmanız gerekecektir. Daha fazla bilgi için bu sayfayı ziyaret edin: [MIT Ekranları.](http://appinventor.mit.edu/explore/ai2/concepts.html#Screens)

Bu ünitedeki tüm içeriği doldurduğunuz için tebrikler! Çok fazla bilgi vardı, bu yüzden şu an her şeyi anlamadıysanız sorun yok. Uygulamanızı oluştururken her zaman bu birime başvurabilirsiniz. Kod Mücadelesi, şimdiye kadar öğrendiğiniz her şeyi dahil etmek için tasarlanmıştır, bu yüzden bir deneyin!

**Kod 2 Mücadelesi: Bir Yapılacaklar Listesi Hazırlayın**

Açıklama:

Kullanıcıların öğelere girip çıkarmasına izin veren bir yapılacaklar listesi yapın. Uygulamanız, kullanıcının uygulamayı her açışında öğeleri hatırlamalıdır. Mücadeleyi denemeden önce [ListView](http://ai2.appinventor.mit.edu/reference/components/userinterface.html" \l "ListView" \t "_blank) (liste görünümü)’ün nasıl kullanılacağını öğrenmek isteyebilirsiniz!

Talimatları okumadan bu mücadeleyi kendiniz yapabilmek için elinizden geldiği kadar uğraşın. Unutmayın, bu talimatlar, sorunun çözülebileceği tek yolu değildir! Örnek uygulamamızı Google Play Store’da da indirebilirsiniz. Mutlu kodlamalar!