

### 3- Örnekler

Kendi müziğinizi geliştirmenin bir başka yolu da önceden kaydedilmiş örnek sesleri kullanmaktır. Büyük hip-hop geleneğinde bu sesleri örnekler olarak adlandırıyoruz.

Eğer dışarıya çıkıp mikrofonunuzla yağmurun kanvasa çarpış sesini kaydederseniz bir örnek yaratmış olursunuz.

Sonic Pi kendi bünyesinde barındırdığı 130 farklı örnek ses ile çok eğlenceli şeyler yapmanızı sağlar. Aynı zamanda kendi seslerinizi oluşturmanızı ve değiştirmenizi de sağlar. Hadi biraz bakalım...

#### 3.1- Örnekleri çağırmak

Bip sesini çalmak sadece bir başlangıçtı. Önceden kaydedilmiş örnekleri çağırmak çok eğlencelidir. Deneyin:

sample :ambi\_lunar\_land

Sonic pi kendi içerisinde çalabileceğiniz çok fazla örnek içerir. Onları tıpkı notalardaki **play** komutunu kullandığınız gibi kullanabilirsiniz. Eğer birden fazla örnek ses çalmak istiyorsanız onları birbiri ardına yazın:

play 36

play 48

sample :ambi\_lunar\_land

sample :ambi\_drone

Eğer aralarında zaman boşluğu olmasını istiyorsanız **sleep** komutunu kullanabilirsiniz:

sample :ambi\_lunar\_land

sleep 1

play 48

sleep 0.5

play 36

sample :ambi\_drone

sleep 1

play 36

Sonic Pi bir sonraki sese başlamadan önceki sesi beklemez, buna dikkat edin. `sleep` komutu yalnızca seslerin *tetiklenmesinin* ayrılmasını sağlar. Bu fırsat, sesleri üst üste bindirerek onları katmanlandırmanıza olanak tanır. Derslerimizin ilerleyen kısımlarında seslerin *sürelerini* kontrol etmeyi öğreneceğiz.

## Örnekleri Keşfetmek

Sonic Pi’ın içinde verilen örnekleri keşfetmenin 2 farklı yolu vardır. İlk olarak, buradaki yardım sistemini kullanabilirsiniz. Bu yardım menüsü altındaki örnekler seçeneğine tıkladıktan sonra kategorinizi seçin daha sonra uygun seslerin bir listesi karşınıza çıkacaktır.

Buna alternatif olan başka bir yol da Sonic Pi’ın *otomatik tamamlama mekanizmasıdır*. `sample :ambi_` gibi basit bir başlangıç yazın ve aşağısında beliren örneklerin isimlerinden seçmeye başlayın.

- `:ambi_`
- `:bass_`
- `:elec_`
- `:perc_`
- `:guit_`
- `:drum_`
- `:misc_`
- `:bd_`

Şimdi örnekleri kendi kompozisyonunuzda birbirine karıştırın!

### 3.2- Örnek Parametreleri: Amp ve Pan

Synth'lerde gördüğümüz gibi seslerimiz parametrelerle kolayca kontrol edilebilir. Örnekler de bu parametre sistemini destekler. `amp:` ve `pan:` arkadaşlarımızı yerlerinde ziyaret edelim.

#### Örneklerin Ses Katını Yükseltme

Synth'lerde olduğu gibi örnek seslerimizin de genliğini aynı yaklaşımla değiştirebiliriz:

```
sample :ambi_lunar_land, amp: 0.5
```

#### Örnekleri Kaydırma

Sonic Pi içerisinde örnek seslerimizde `pan:` parametresini kullanabiliriz. Örnek vermek gerekirse seslerimizin hangi kulağımıza ne kadar yoğunlukta ve ne kadar süre geleceğini ayarlayabiliriz:

```
sample :loop_amen, pan: -1  
sleep 0.877  
sample :loop_amen, pan: 1
```

Son olarak bazı synth varsayımlarını `use_synth_defaults` ile ayarlarsanız bunların örnek ses tarafından yok sayılacağını da unutmamanız gerekir.

### 3.3- Örnek Sesleri Germe İşlemleri

Artık bir müzik oluşturmak için çeşitli synth'ler ve örnek sesler çalabildiğimize göre, bunları yaratıcı, ilginç ve benzersiz bir biçimde nasıl kullanacağımızı öğrenme zamanı da geldi. Öncelikle *stretch* ve *squash* örneklerimizi keşfedelim.

## Örnek Seslerin Temsili

Örnek sesler onları yeniden üretmek için hoparlör konisinin hareketlerini temsil eden numaralarla önceden kaydedilmiş seslerdir. Hoparlör konisi içeri ve dışarı belirli sürelerde hareket edebilir ve sayılar koninin ne kadar içeri ya da dışarıda bulunacağını denetler. Kayıtlı bir sesi bozulmaksızın çoğaltmak için örnek sesimizin saniyede binlerce numara depolaması gereklidir! İnanılmaz! Sonic Pi bu numara listesini alarak doğru sıralamada ve yoğunlukta hoparlörünüzü ses çıkartması için besler. Bununla birlikte sesi değiştirmek amacıyla hoparlörün beslenme hızını değiştirmek de eğlenceli olabilir.

## Oranların Değiştirilmesi

Sonic Pi'nin ambiyans seslerinden biri olan `:ambi_choir` sesini ele alalım. Varsayılan değerleri değiştirmek için `rate:` özelliğini kullanabiliriz:

```
sample :ambi_choir, rate: 1
```

```
sample :ambi_choir, rate: 0.5
```

Oh, neler oldu öyle? Hem örnek sesimizin oynaması 2 kat daha uzun sürdü hem de sesimizin oktavı düştü. Haydi bunu biraz daha detaylı bir biçimde inceleyelim.

## Hadi Uzatalım!

Amen Break üzerinde değişiklik yapılması oldukça zevkli bir fonksiyondur.

```
sample :loop_amen
```

```
sample :loop_amen, rate: 0.5
```

```
sample :loop_amen, rate: 1.5
```

```
sample :loop_amen, rate: -1
```

Ah, sonuncu örnekte tersten mi çaldı o? Şimdi Farklı şeyler denemeye devam edin ve kendinizi müziğe kaptırın!

### 3.4- Zarflanmış Örnek Sesler

Bir ADSR zarfı kullanarak bir örnek sesin genliğini ve uzunluğunu değiştirmek mümkündür. Ancak, synth'lerde bulunan ADSR zarfları normalden biraz daha farklı çalışır.

#### Amen Zarfı

```
sample :loop_amen
```

Haydi şimdi **attack**: parametresini kullanalım:

```
sample :loop_amen, attack: 1
```

Daha kısa bir geçiş için, daha kısa bir attack değeri seçin

```
sample :loop_amen, attack: 0.3
```

#### Otomatik Sürdürme

**Sustain** parametresi sayesinde kullandığımız seslerin uzunluğunu yani çalma sürelerini ayarlayabiliriz.

#### Soluk Çıkışlar

Burada hem **attack:** hem de **release:** parametrelerini beraber kullanabiliriz.

```
sample :loop_amen, attack: 0.75, release: 0.75
```

Buradaki değerler ile oynayarak müziğinizi keşfetmeye devam edin!

### 3.5-Kısmi Örnekler

Bu kısımda Sonic Pi'In modellerini daha da çok keşfedeceğiz. Nasıl kullanıldığını göstermiştik.

Örneğin:

```
sample :loop_amen
```

Daha sonra hızlarını nasıl değiştirdiğimiz gördük:

```
sample :loop_amen, rate: 0.5
```

Daha hafifletmek için:

```
sample :loop_amen, rate: 0.5, attack: 1
```

Aynı zamanda başlangıcının **sustain:** komutu ile nasıl başlayacağının yanında attack ve release komutlarını ayarlamak için de kullanılabilir.

```
sample :loop_amen, rate: 2, attack: 0.01, sustain: 0, release:  
0.35
```

Fakat bunun yanında modellerin nerden başlayacağını söyleyebilsek güzel olmaz mıydı?

### Başlangıç Noktası Seçme

0 'ın başlangıç 1'in son ve 0.5'in yarı yol olduğu, 0 ve 1 arasında sıradan bir başlangıç noktası seçmek mümkün. Alttaki kodda olduğu gibi:

```
sample :loop_amen, start: 0.5
```

,

Son çeyreğine neredesiniz:

```
sample :loop_amen, start: 0.75
```

### Son Noktayı Seçmek

Benzer bir şekilde 0 ile 1 arasında sıradan bir bitiş noktası seçilebilir:

```
sample :loop_amen, finish: 0.5
```

### Başlangıç ve Bitiş Belirleme

Modelin sadece bir kısmı için:

```
sample :loop_amen, start: 0.4, finish: 0.6
```

Peki başlangıç noktasının bitişinden daha büyük seçersek ne olur? Tersten çalmaya başlar:

```
sample :loop_amen, start: 0.6, finish: 0.4
```

## Hız ile Düzenleme

Modellerin sadece bir kısmını daha yavaş ya da daha hızlı yapmak ister misin? Şunu bir dene:

```
sample :loop_amen, start: 0.5, finish: 0.7, rate: 0.2
```

## Zarflar ile Düzenleme

Sonunda bunların hepsini ADSR zarfları ile daha ilgi çekici bir hale getirebiliriz.

```
sample :loop_amen, start: 0.5, finish: 0.8, rate: -0.2, attack:  
0.3, release: 1
```

## 3.6-Harici Modeller

Eğer Sonic Pi içerisinde bulunan modeller yetersiz geliyorsa kendi kayıtlarınızı da dahil etmek mümkün. Bunun için parçanızın taşınabilirliğini kontrol etmek durumundayız.

### -Taşınabilirlik

Eğer eserinizi sadece Sonic Pi bünyesindeki modeller ile yaparsanız bunu arkadaşlarınızla dahi paylaşp keyfini çıkarabilirsiniz.

Fakat, kendi parça modelleriniz kullanırsanız, arkadaşlarınızla bu parça modeli de paylaşp eğlenceye devam edebilirsiniz. Göz önünde bulundurulması gereken bir özellik.

### -Yerel Modeller



Şu şekilde WAV, AIFF veya FLAC tipi dosyaları parça modeller olarak kullanabilirsiniz:

*# Raspberry Pi, Mac, Linux*

sample *"/Users/sam/Desktop/my-sound.wav"*

*# Windows*

sample *"C:/Users/sam/Desktop/my-sound.wav"*

Sonic Pi otomatik olarak modeli yükleyi çalacaktır. Aynı zamanda standart komutları kullanarak modeli düzenleyebilirsiniz.

*# Raspberry Pi, Mac, Linux*

sample *"/Users/sam/Desktop/my-sound.wav", rate: 0.5, amp: 0.3*

*# Windows*

sample *"C:/Users/sam/Desktop/my-sound.wav", rate: 0.5, amp: 0.3*

### 3.7-Model Paketleri

**NOT: BU kısım Sonic Pi ile uyumlu parça modelleri yaratanlar ve satın alanlar için düzenlenmiştir. Sonic Pi bünyesindeki modeller yeterliyse bu kısmı geçebilirsiniz.**

Büyük model dosyaları ile çalışmak biraz sıkıntılı bir durum yaratabilir. Örneğin:

*/path/to/my/samples/*

Dosyasında şu modeller olsun:

- *100\_A#\_melody1.wav*
- *100\_A#\_melody2.wav*
- *100\_A#\_melody3.wav*
- *120\_A#\_melody4.wav*
- *120\_Bb\_guit1.wav*

- 120\_Bb\_piano1.wav

Piyano modelini kullanmak için şu yolu deneyebilirsin:

```
sample "/path/to/my/samples/120_Bb_piano1.wav"
```

Sonra gitar modeli için:

```
sample "/path/to/my/samples/120_Bb_guit.wav"
```

Burada dosya ve model isminiz bildiğimiz bir durum var, ya bunları bilmeseydik.

## -Model Paketlerini İndexleme

Eğer ilk modeli duymak istersen:

```
sample "/path/to/my/samples/", 0
```

Değişken kullanarak daha basit bir yol izlenebilir.

```
samps = "/path/to/my/samples/"  
sample samps, 0
```

Şimdi ikinci modeli duymak için:

```
samps = "/path/to/my/samples/"  
sample samps, 1
```

Artık modellerin isimlerine gerek duymadığımızı fark ettin mi? Hatta dosyadaki model sayısından daha fazla bir sayı girersen de içlerinden birini çalmaya devam edecektir.

## -Model Paketleri Filtreleme

Genellikle indeksleme yeterli olduğu halde modelleri düzenlemek ve sıralamak için daha çok güce ihtiyacımız var. Tesadüfi, birçok model paketleri yararlı bilgiler içeriyor. Örneğin:

- 100\_A#\_melody1.wav
- 100\_A#\_melody2.wav
- 100\_A#\_melody3.wav
- 120\_A#\_melody4.wav
- 120\_Bb\_guit1.wav
- 120\_Bb\_piano1.wav
- 

Bu isimlerde birden fazla bilgi olduğunu fark ettin mi? 100 ve 120 sayıları modellerin bpm değerini göstermekte. Aynı zamanda A# ve Bb notalarında çalındığını göstermekte.

Bu bilgileri sadece istediklerimizi çalmak amacıyla kullanabiliriz. Örneğin:

```
samps = "/path/to/my/samples/"
sample samps, "120"
```

Bu kod bizi ilk eşleşen modele götürecektir yani gitar1 modeline.

```
samps = "/path/to/my/samples/"
sample samps, "120", 1
```

Bu kod ise bize 120 bpm'deki 2. Eşleşmeyi sağlayacak.

```
samps = "/path/to/my/samples/"
sample samps, "120", "A#"
```

Hatta az sonra görüleceği üzere birden fazla filtre kullanılabilir.

```
samps = "/path/to/my/samples/"  
sample samps, "120", "Bb", 1, lpf: 70, amp: 2
```

## -Kaynaklar

Model filtresi pre-arg sistemi 2 tür bilgiyi algılıyor: kaynaklar ve filtreler. Kaynaklar potansiyel adaylar için kullanılan bilgi olarak gösterilebilir. Bir kaynak 2 formda olabilir.

1. “/path/to/samples”– bizi geçerli dosyaya ulaştıran bir yol
2. “/path/to/samples/foo.wav” – bizi geçerli modeller ulaştıran bir yol

**sample** komutu önce tüm kaynakları toplayarak bir muhtemel aday listesi oluşturuyor. Bu liste önce geçerli yolları ekliyor daha sonra geçerli **.flac**, **.aif**, **.aiff**, **.wav**, **.wave** dosyalarını ekliyor. Örneğin:

```
samps = "/path/to/my/samples/"  
samps2 = "/path/to/my/samples2/"  
path = "/path/to/my/samples3/foo.wav"  
  
sample samps, samps2, path, 0
```

Varsayılan olarak, yalnızca bir dizindeki örnek dosyalar aday listesine toplanır. Bazen aramak ve filtrelemek istediğiniz birçok iç içe geçmiş model klasörünüz olabilir. Bu nedenle, yolun sonuna **\*\*** ekleyerek, belirli bir klasörün tüm alt klasörlerindeki tüm örnekler için özyinelemeli bir arama yapabilirsiniz.

```
samps = "/path/to/nested/samples/**"  
sample samps, 0
```

Son olarak, ilk önce kaynakların gitmesi gerektiğini unutmayın. Herhangi bir kaynak belirtilmezse, yerleşik örneklem grubu birlikte çalışılacak adayların varsayılan listesi olarak seçilecektir.

## -Filtreler

Bir aday listeniz varsa şu filtreleme türlerini kullanabilirsiniz:

**"foo"** Dizeler, dosya adındaki alt dize oluşumuna (eksi dizin yolu ve uzantı) filtre uygular.

**/fo[oO]/**Normal ifadeler, dosya adının (eksi dizin yolu ve uzantısı) desen eşleşmesine filtre uygular.

**:foo** - Anahtar kelimeler adayları, anahtar kelimenin dosya adıyla doğrudan eşleşip eşleşmediği konusunda filtreleyecektir (eksi dizin yolu ve uzantısı).

**lambda{[a] ... }** - Bir argüman olan işlemler aday filtre veya jeneratör işlevi olarak değerlendirilir. Mevcut adayların listesinden geçirilecek ve yeni bir aday listesi (örnek dosyalara uygulanan geçerli yolların bir listesi) geri göndermesi gerekir.

**1** - Rakamlar bu indekse sahip adayı seçecektir (gerekirse bir halka gibi sarma).

## -Karışımlar

Son olarak, kaynak veya filtre yerleştirebileceğiniz her yerde listeleri kullanabilirsiniz. Liste otomatik olarak düzleştirilecek ve içerik düzenli kaynaklar ve filtreler olarak değerlendirilecektir. Bu nedenle aşağıdaki **sample** çağrıları anlamsal olarak eşdeğerdir:

```
sample "/path/to/dir", "100", "C#"
sample ["/path/to/dir", "100", "C#"]
sample "/path/to/dir", ["100", "C#"]
sample ["/path/to/dir", ["100", ["C#"]]]
```

## -Sarma

Bu, model paketleri işlemek ve kullanmak için gerçek güce ihtiyaç duyan insanlar için gelişmiş bir bölümdü. Bu bölümün çoğu çok fazla anlam ifade etmiyorsa, endişelenmeyin. Henüz bu işlevselliğin hiçbirine ihtiyacınız yok. Ancak, ne zaman ihtiyacınız olduğunu bilirsiniz ve büyük örneklem dizinleriyle çalışmaya başladığınızda geri dönüp tekrar okuyabilirsiniz.