

## 2-Synths

Bu kadar giriş yeter, artık seslere geçiş yapabiliriz.

Bu bölümde synthleri tetikleme ve manipüle etme temellerini göreceğiz. Synth synthesiser kelimesinin kısaltması. Sytnhleri kullanması biraz karışık. Fakat Sonic Pi kullanımı kolay bir hale getiriyorç

### 2.1- İlk Bipleriniz

Aşağıdaki örnek koda bakalım:

play 70

Bu kod her şeyin başlangıcı. Bu kodu kopyalayıp kod alanına (Run tuşunun altındaki beyaz alan) yapıştır ve sonra Run tuşuna bas!

-Biipp!

Etkileyici! Tekrar bas. Şimdi yine bas. Yine...

Vay be çılgın, Eminim bütün gün boyunca bu tuşa basıp durabilirsin. Fakat kendini sonsuz Beep'lerin içinde kaybetmeden önce, numarayı değiştirmeyi deneyebilirsin:

play 75

Bir fark duyabiliyor musun? Daha düşük bir numarayı dene:

play 60

Düşük rakamlar düşük perdede seslere, yüksek rakamlar yüksek perdede seslere yol açıyor. Aynı piyano gibi, piyanonun alt tarafındaki (sol tarafı) tuşlar daha düşük notalar çalarken üst tarafındaki (sağ taraf) tuşlar daha yüksek notaları çalıyor. Aslında bakarsak bu numaralar piyanonun tuşlarına denk geliyor, **play 47** demek piyanonun 47'inci notasını çal demek. Bu

da demek oluyor ki **play 48** aslında bir nota üstte (sağdan bir nota üstte). 4.oktav C, numara 60 oluyor.

## -Akorlar

Tek bir nota çalmak eğlenceli fakat birden fazla notayı aynı anda çalmak çok daha iyi: Alttaki örneği kopyalayıp yapıştır ve dene:

play 72

play 75

play 79

Senin de fark ettiğin üzere birden fazla **play** yazdığında hepsi aynı anda çalışıyor. Kendin yeni notaları dene, hangileri birlikte güzel oluyor? Hangileri kötü oluyor? Denemekten korkma, keşfet!

## -Melodi

Notaları ve akorları oynatmak eğlenceli, fakat bir melodiye ne dersin? Ya bir notayı başka bir notadan sonra oynatmak istersen o zaman napıcaksın? Tek yapman gereken arlarına **sleep** koymak!

play 72

sleep 1

play 75

sleep 1

play 79

Ne kadar güzel, küçük bir arpej. Peki **sleep 1**'deki **1** ne demek oluyor? **sleep**'ten sonra gelen numara **sleep'in süresini** belirtiyor. Şimdi arpejimizi daha hızlı yapmak istersek eğer **sleep**'lerin değerlerini daha küçültmemiz gerekiyor.

play 72

sleep 0.5

play 75

sleep 0.5

play 79

## 2.2 Synth Seçenekleri: Amp ve Pan

Ne kadar sizin hangi notayı çalmak istediğinizi ve hangi modeli başlatmak istediğinizi kontrol etse de Sonic Pi size çok geniş bir ses kontrol mekanizması sunuyor. Bu eğitimlerde çoğunun üstünden geçeceğiz ve bu tür mekanizmalar için yardım sisteminde büyük bir arşiv var. Fakat, şimdilik bunlardan en önemli ikisi olan amplitude ve pan üstünde daha fazla duracağız.

### -Seçenekler

Sonic Pi kendi synth'leri için seçenekler kavramını destekler. Seçenekler, duyduğunuz sesin özelliklerini değiştiren ve kontrol eden, kontrollerdir. Her synth, sesini hassas bir şekilde ayarlamak için kendi seçeneklerine sahiptir. Ancak **amp:** ve zarf seçenekleri (başka bir bölümde ele alınmıştır) gibi birçok ses tarafından paylaşılan ortak seçenek grupları vardır.

Seçeneklerin iki ana bölümü vardır: isimleri (kontrolün adı) ve değerleri (kontrolü ayarlamak istediğiniz değer). Örneğin **cheese:** adında bir seçeneğe sahip olabilirsiniz ve bunu 1 değerine ayarlamak isteyebilirsiniz.

Seçenekler, virgül kullanarak çalmak için aramalara iletilir ve ardından **amp:** gibi bir seçenek adı bulunur: (iki nokta üst üste'yi unutma:) ve sonra bir boşluk ve seçeneğin değeri. Örneğin:

play 50, **cheese:** 1

(**cheese**: aslında geçerli bir seçenek değil, biz sadece onu örnek göstermek için kullandık.)

Birden çok seçeneği virgülle ayırarak geçirebilirsiniz:

```
play 50, cheese: 1, beans: 0.5
```

Seçeneklerin sırası önem arz etmemektedir.

```
play 50, beans: 0.5, cheese: 1
```

Synth tarafından tanınmayan seçenekler sadece göz ardı edilir. (Açıkça saçma tercih isimleri olan **cheese** ve **beans** gibi!)

Yanlışlıkla aynı tercihi farklı değerlerle iki kez kullanırsanız, sonuncusu kazanır. Örneğin, **beans**: burada 0,5 yerine 2 değerine sahip olacak:

```
play 50, beans: 0.5, cheese: 3, eggs: 0.1, beans: 2
```

Sonic Pi'deki pek çok şey seçenekleri kabul ediyor, bu yüzden onları nasıl kullanacağınızı öğrenmek için biraz zaman harcayın. İlk seçeneğimizle oynayalım: **amp**..

## Genlik

Genlik, bir sesin yüksekliğinin bir bilgisayar temsilidir. Yüksek genlik yüksek bir ses çıkarırken düşük genlik sessiz bir ses çıkarır. Sonic Pi, zaman ve notları temsil etmek için sayılar kullandığı gibi, genliği temsil etmek için de sayılar kullanır. 0 genliği sessizdir (hiçbir şey duyamazsınız), 1 genliği ise normal ses seviyesindedir. 2, 10, 100'e kadar daha yüksek bir genlik bile artırabilirsiniz. Ancak, tüm seslerin genliği çok yükseldiğinde, Sonic Pi'nin o sesleri sıkıştırmak için kompresör denen şeyi kullandığını unutmayın çünkü bu ses kulaklarınız için

çok yüksek. Bu genellikle sesi çamurlu ve garip yapabilir. Bu nedenle, kompresyondan kaçınmak için 0 ila 0.5 aralığında yani düşük genlikleri kullanmayı deneyin.

## Genliğe Bağlı

Bir sesin genliğini değiştirmek için **amp:** seçeneğini kullanabilirsin. Örneğin, yarı genlikte oynamak için 0.5'e geçin:

```
play 60, amp: 0.5
```

Çift genlikte çalmak için:

```
play 60, amp: 2
```

**amp:** opt, yalnızca aradığı kişiyle çalınacak aramayı değiştirir. Bu nedenle, bu örnekte, çalınacak ilk çağrı yarım ses düzeyindedir ve ikincisi varsayılamaya döner (1):

```
play 60, amp: 0.5  
sleep 0.5  
play 65
```

Tabii, farklı **amp:** değerleri de kullanabilirsiniz.

```
play 50, amp: 0.1  
sleep 0.25  
play 55, amp: 0.2  
sleep 0.25  
play 57, amp: 0.4  
sleep 0.25  
play 62, amp: 1
```

## Sesin Nerden Geleceğine Karar Vermek

Kullanmak için bir başka eğlenceli seçenek ise **pan**:sesin stereo olarak kaydırılmasını kontrol ediyor. Sesi sola kaydırmak, sol hoparlörden duyduğunuz ve sağa kaydırmak, sağ hoparlörünüzden duyduğunuz anlamına gelir. Değerlerimizde, tamamen solu temsil etmek için -1, merkezi temsil etmek için 0'ı ve tamamen sağı temsil etmek için 1 kullanırız. Elbette, sesimizin tam konumunu kontrol etmek için -1 ile arasındaki herhangi bir değeri kullanmakta özgürüz

Sol hoparlörden bip sesi çıkaralım:

```
play 60, pan: -1
```

Şimdi, sağ hoparlörden çalalım:

```
play 60, pan: 1
```

Son olarak, her ikisinden de merkezden oynatalım (varsayılan konum):

```
play 60, pan: 0
```

Şimdi, git ve seslerinin genliğini ve kaymasını değiştirerek eğlen!

## 2.3-Sentezleri değiştirmek

Şu ana kadar bip sesi çıkarırken az çok eğlendik. Ama büyük ihtimalle basit bip sesleri çıkarmaktan sıkılmaya başladınız. Sonic Pi sadece bunları mı sunuyor? Tabii ki sadece bip sesi ile kalmıyor. Bu bölümde Sonic Pi'nin bize sunduğu daha ilgi çekici ses aralığını keşfedeceğiz.

## -Synths

Sonic Pi'nin synths adı altında birden fazla farklı enstrümanları vardır. Ayriyeten modeller önceden kaydedilmiş seslerken, synths sizin kontrol etme şeklinize göre yeni sesler oluşturma kapasitesine sahiptir (ki bunları daha sonra bu öğreticide keşfedeceğiz.). Sonic Pi'nin synths'leri çok güçlü ve süratlidir ve onları keşfederken ve oynarken çok eğleneceksiniz. İlk olarak geçerli sentezi kullanmak için nasıl seçmemiz gerektiğini öğrenelim.

## -Vızıltılı "saw"lar ve "prophet"ler

*Saw wave* adında eğlenceli bir ses var- hadi bir deneyelim:

```
use_synth :saw
play 38
sleep 0.25
play 50
sleep 0.25
play 62
```

Haydi başka bir ses deneyelim- the *prophet*:

```
use_synth :prophet
play 38
sleep 0.25
play 50
sleep 0.25
play 62
```

Bu iki sesi kombine etmeye ne dersiniz? İlk birini sonra da ötekini çalalım:

```
use_synth :saw
play 38
sleep 0.25
```

```
play 50
sleep 0.25
use_synth :prophet
play 57
```

Şimdi aynı sesleri birlikte çalalım(bunu aralarına duraklama komutu(sleep) koymadan yapabiliriz.):

```
use_synth :tb303
play 38
use_synth :dsaw
play 50
use_synth :prophet
play 57
```

Unutmayın ki **use\_synth** komutu sadece onun ardından yazılan **play** komutlarına etki eder. Bunu *büyük bir değiştirme*, yani yeni sesleri hangi sentezde olursa olsun hangisini gösteriyorsa kullanma, olarak düşünebilirsiniz.

## -Synths keşfetmek

Sonic Pi'ın size hangi synths oynatabileceğini görmek için alttaki menüde yardım ekranındaki(Örnekler ve Fx arasında) synth seçeneğine bak. 20 den fazla seçenek var. İşte benim birkaç favori synths:

- **:prophet**
- **:dsaw**
- **:fm**
- **:tb303**
- **:pulse**
- 

## 2.4-Zarflarla Süreç



Daha önceki bir bölümde seslerimizi ne zaman tetikleyeceğimizi kontrol etmek için uyku komutunu nasıl kullanabileceğimize baktık. Ancak, seslerimizin süresini henüz kontrol edemedik.

Seslerimizin süresini kontrol etmemiz için bize basit ama güçlü bir yöntem sağlamak için Sonic Pi, bir ADSR genlik zarfı nosyonu (bu bölümde daha sonra ADSR'nin ne anlama geldiğini anlatacağız). Bir genlik zarfı, kontrolün iki yararlı yönünü sunar:

- control over the duration of a sound
- control over the amplitude of a sound

## -Süreç

Süre, sesin dayandığı uzunluktur. Daha uzun süre, sesi daha uzun süre duyduğunuz anlamına gelir. Sonic Pi'nin seslerinin hepsinin kontrol edilebilir bir genlik zarfı vardır ve bu zarfın toplam süresi sesin süresidir. Bu nedenle, zarfı kontrol ederek süreyi kontrol edersiniz.

## -Genlik

ADSR zarfı sadece süreyi kontrol etmekle kalmaz, aynı zamanda sesin genliği üzerinde hassas kontrol sağlar. Tüm duyulabilir sesler sessizleşmeye başlar ve sonunda sessizleşir ve arada sessiz olmayan bir kısım içerir. Zarflar, sesin sessiz olmayan bölümlerinin genliğini kaydırmanıza ve tutmanıza izin verir. Bir gitar amplifikatörünün sesini nasıl açıp kapatacağı konusunda talimatlar vermek gibi. Örneğin, birinden “sessizliğe başlaması, yavaşça tam ses seviyesine yükselmesi, bir süre tutması, ardından hızlı bir şekilde sessizliğe geri dönmesini” isteyebilirsiniz. Sonic Pi, bu davranışı zarfların yardımıyla programlamanıza izin verir.

Özetlemek gerekirse, daha önce gördüğümüz gibi, 0 genliği sessizlik ve 1 genliği normal sestir.

Şimdi sırayla zarfların her bir kısmına bakalım.

## -Serbest Bırakma Evresi

Zarfın varsayılan olarak kullanılan tek kısmı yayınlanma süresidir. Bu, synth'in sesinin azalması için geçen zamandır. Tüm sentezlerin 1 serbest bırakılma süresi vardır, bu, varsayılan olarak 1 vuruş süresine sahip oldukları anlamına gelir (varsayılan 60'ta BPM 1 saniyedir):

```
play 70
```

Not 1 saniye boyunca sesli olarak duyulur. Devam et ve zamanı geç :-) Bu daha uzun süren bir süreç için kısa bir bölüm:

```
play 70, release: 1
```

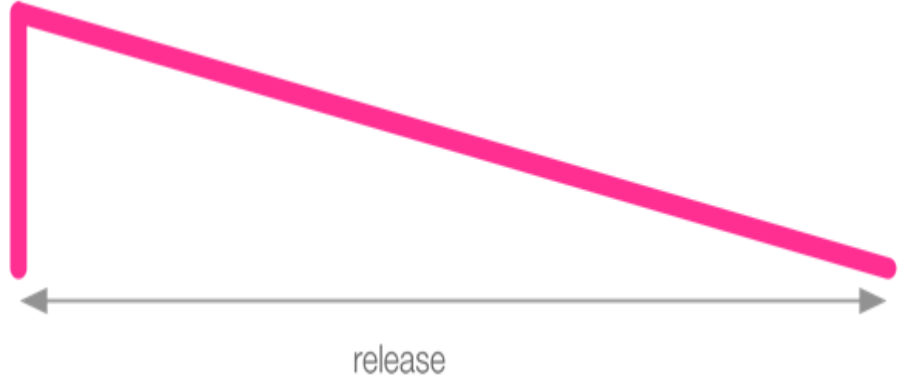
Bunun nasıl aynı şekilde ses çıkardığına dikkat edin (ses bir saniye sürer). Ancak, sürecin değerini değiştirerek süreyi değiştirmek artık çok kolay

```
play 60, release: 2
```

Çok küçük bir süreç değeri kullanarak synth sesini çok kısa yapabiliriz:

```
play 60, release: 0.2
```

Sürecin süresi, serbest bırakma evresi olarak adlandırılır ve varsayılan olarak doğrusal bir geçiştir (yani düz bir çizgi). Aşağıdaki diyagram bu geçişi göstermektedir:



Diyagramın en solundaki dikey çizgi, sesin 0 genlikle başladığını ancak derhal tam genliğe kadar çıktığını gösterir (bu, daha sonra ele alacağımız saldırı aşamasıdır). Tam genliğe geldikten sonra lineer bir şekilde sıfıra doğru hareket eder ve **release:** tarafından belirtilen süreyi alır. Daha uzun salma süreleri daha uzun sentetik az ses çıkışları üretir.

Bu nedenle, bırakma süresini değiştirerek sesinizin süresini değiştirebilirsiniz. Müziğinize serbest bırakma süresi ekleyerek bir oyun oynayın.

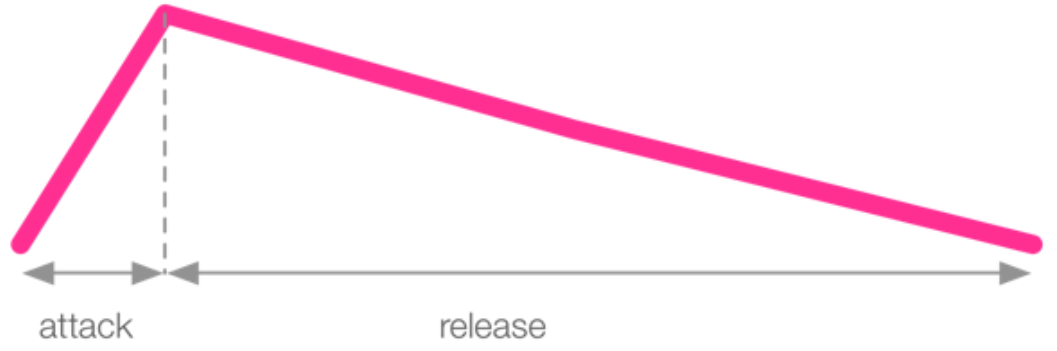
### -Atak Evresi

Varsayılan olarak, atak evresi tüm synth'lerde 0'dır, bu da hemen 0 genlikten 1'e geçmeleri anlamına gelir. Bu, synth'a ilk vurmalı bir ses verir. Ancak, sesini kısmak isteyebilirsiniz. Bu durum **attack:** seçeneği ile başarılabılır. Bazı seslerde kısmayı deneyin:

```
play 60, attack: 2
sleep 3
play 65, attack: 0.5
```

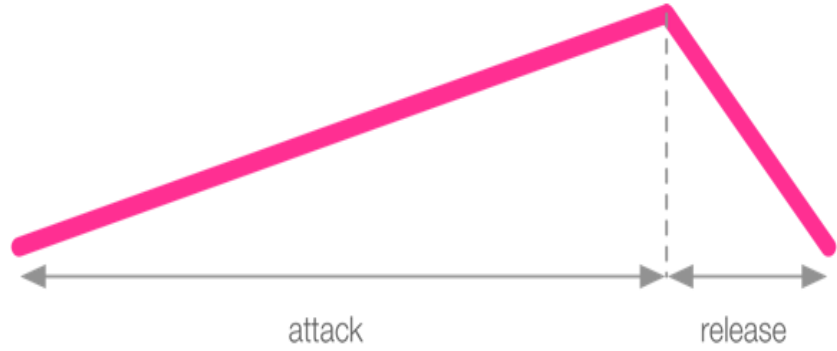
Birçok seçeneği aynı anda kullanabilirsiniz. Örneğin kısa bir atak ve uzun bir serbest bırakmayı deneyin:

```
play 60, attack: 0.7, release: 4
```



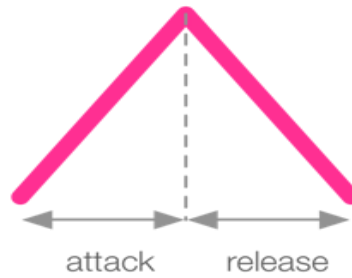
Tabii ki, bunları farklı sırada yapabilirsiniz.

```
play 60, attack: 4, release: 0.7
```



Son olarak kısa sesler için kısa ataklara ve serbest bırakma sürelerine sahip olabilirsiniz.

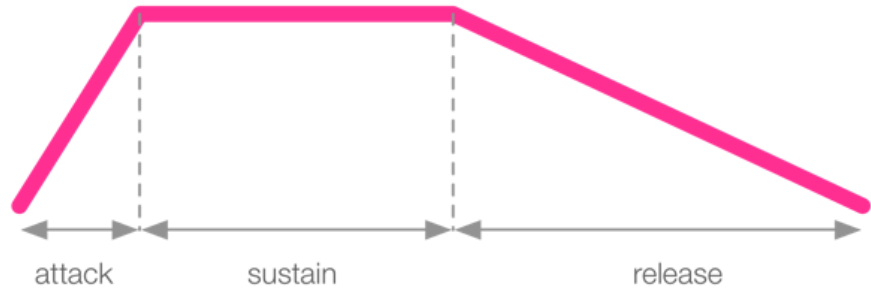
```
play 60, attack: 0.5, release: 0.5
```



**-Sürdürme Evresi**

Atak ve serbest bırakma zamanlarını belirtmeye ek olarak, sürdürme aşamasını kontrol etmek için bir sürdürme süresi de belirleyebilirsiniz. Bu, sesin atak ve serbest bırakma evreleri arasında maksimum genlikte tutulduğu zamandır.

```
play 60, attack: 0.3, sustain: 1, release: 1
```

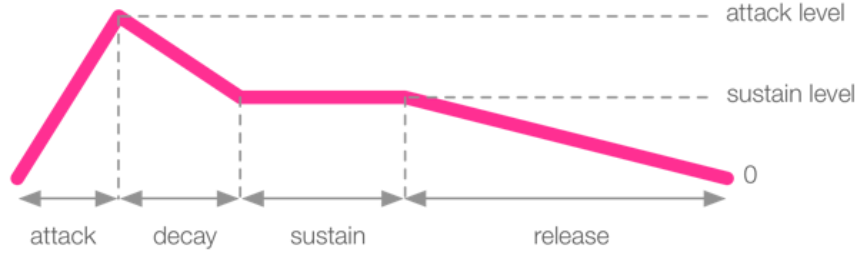


Sürdürme süresi, isteğe bağlı bir serbest bırakma evresine girmeden önce karışımda var olmasını istediğiniz önemli sesler için kullanışlıdır. Tabii ki, hem **attack**: hem de **release**: seçeneklerini 0'a ayarlamak tamamen geçerli ve sadece sürekli olarak solmaya veya sese solmayacak şekilde kullanmak için sürdürme seçeneğini kullanın. Ancak, uyarı olarak, serbest bırakmada 0 değeri seste tıklamalar üretebilir ve 0.2 gibi çok küçük bir değer kullanmak genellikle daha iyidir.

## -Zayıflama Evresi

Ekstra kontrol seviyesi için ayrıca bir zayıflama süresi belirleyebilirsiniz. Bu, zarfın atak ve sürdürme evreleri arasına uyan ve genliklerin **attack\_level**: konumundan **decay\_level**: (açıkça ayarlanmadıkça, **sustain\_level**: ayarlanacağı) konumuna düşeceği bir zaman aralığı belirten bir aşamasıdır. Varsayılan olarak, **decay**: seçeneği 0'dır. Fakat hem atak hem de sürdürmenin varsayılanı 1'dir, bu nedenle herhangi bir etkiye sahip olmak için zayıflama zamanları için bunları belirtmeniz gerekir:

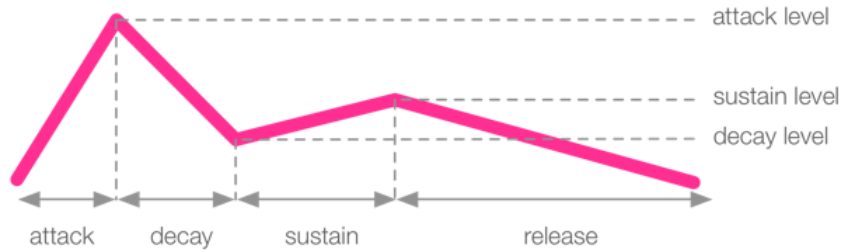
```
play 60, attack: 0.1, attack_level: 1, decay: 0.2,  
sustain_level: 0.4, sustain: 1, release: 0.5
```



## -Zayıflama Seviyesi

Son bir not olarak, ne kadar **decay\_level:** seçeneğinin varsayılanı **sustain\_level:** seçeneği ile aynı olsa da, onları zarfların üstündeki tam kontrolü sağlamak için farklı değerlere ayarlayabilirsiniz. Bu size örnekteki gibi zarflar yaratmanızda yardımcı olur.

```
play 60, attack: 0.1, attack_level: 1, decay: 0.2, decay_level: 0.3,  
sustain: 1, sustain_level: 0.4, release: 0.5
```



**decay\_level:** seçeneğini **sustain\_level:** seçeneğinden daha yükseğe kurmak da mümkün.

## -ADSR Zarfları

Özetlemek gerekirse:

Atak: (**attack:**) 0 genlikten **attack\_level** genliğine kadar geçen süre.

Zayıflama: (decay:) attack\_level genliğinden decay\_level genliğine kadark geçen süre.

Sürdürme: (sustain:) decay\_level genliğinden sustain\_level genliğine kadar geçen süre.

Serbest Bırakma: (release:) sustain\_level genliğinden 0'a kadarki geçen süre.

Bir sesin süresinin, bu aşamaların her birinin zamanlarının toplamı olduğunu not etmek önemlidir. Bu nedenle, aşağıdaki ses  $0.5 + 1 + 2 + 0.5 = 4$  vuruş süresine sahip olacaktır:

```
play 60, attack: 0.5, attack_level: 1, decay: 1, sustain_level: 0.4, sustain: 2, release: 0.5
```

Şimdi zarfları seslerinizde kullanarak keyfini çıkarabilirsiniz.