

Swift Programlama Dili ile iOS Mobil Uygulama Geliřtirme

Hakkımda

Murat Adıgüzel - 33

- Eylül 2011 - ... / Intertech Bilgi İşlem ve Pazarlama Tic. A.Ş.
- Mobil ve Dijital Bakıcılık Bölümü Mimari Danışman
- Murat.Adiguzel@intertech.com.tr

intertech



Dersler

- Pazartesi ve Cuma günleri 13.00 - 17.00
 - 13.00 - 14.15
 - 15 dk ara
 - 14.30 - 15.45
 - 15 dk ara
 - 16.00 - 17.00

Swift nedir?

- macOS, iOS, watchOS, tvOS ve daha fazlası için bir programlama dili
- Çıkış yılı 2014



iOS nedir?

- iPhone ve iPad cihazları için işletim sistemi
- Çıkış yılı 2007
- Geliştiriciler için ilk iOS SDK (Software Development Kit / Uygulama Geliştirme Kiti) 2008 yılında çıktı



2008 - 2014 ?

Objective - C

Neden Swift?

- Objective-C diline göre daha kolay ve anlaşılır, modern
- Bellek yöntemi daha kolay

Eğitimde Kullanacağımız Araç ve Platformlar

- Xcode IDE
- gist.github.com



GitHub Gist

Xcode IDE

- IDE (Integrated Development Environment) Entegre Geliştirme Ortamı
- Entegre?



Xcode IDE


- IDE (Integrated Development Environment) Entegre Geliştirme Ortamı
- Entegre?
 - Kod yazım ve kullanıcı arayüzü editörü
 - Derleyici (Builder) - Yazılan kodların ve geliştirilen kullanıcı arayüzünün makinenin anlayacağı dile (makine koduna) dönüştüren araç
 - Hata ayıklayıcı (Debugger)



gist.github.com

- Kod paylaşım platformu

GitHub Gist [All gists](#) [Back to GitHub](#)

 **madiguzel** / **Exercises-Day1.swift** Secret Edit
Last active now

<> Code Revisions 5 Embed

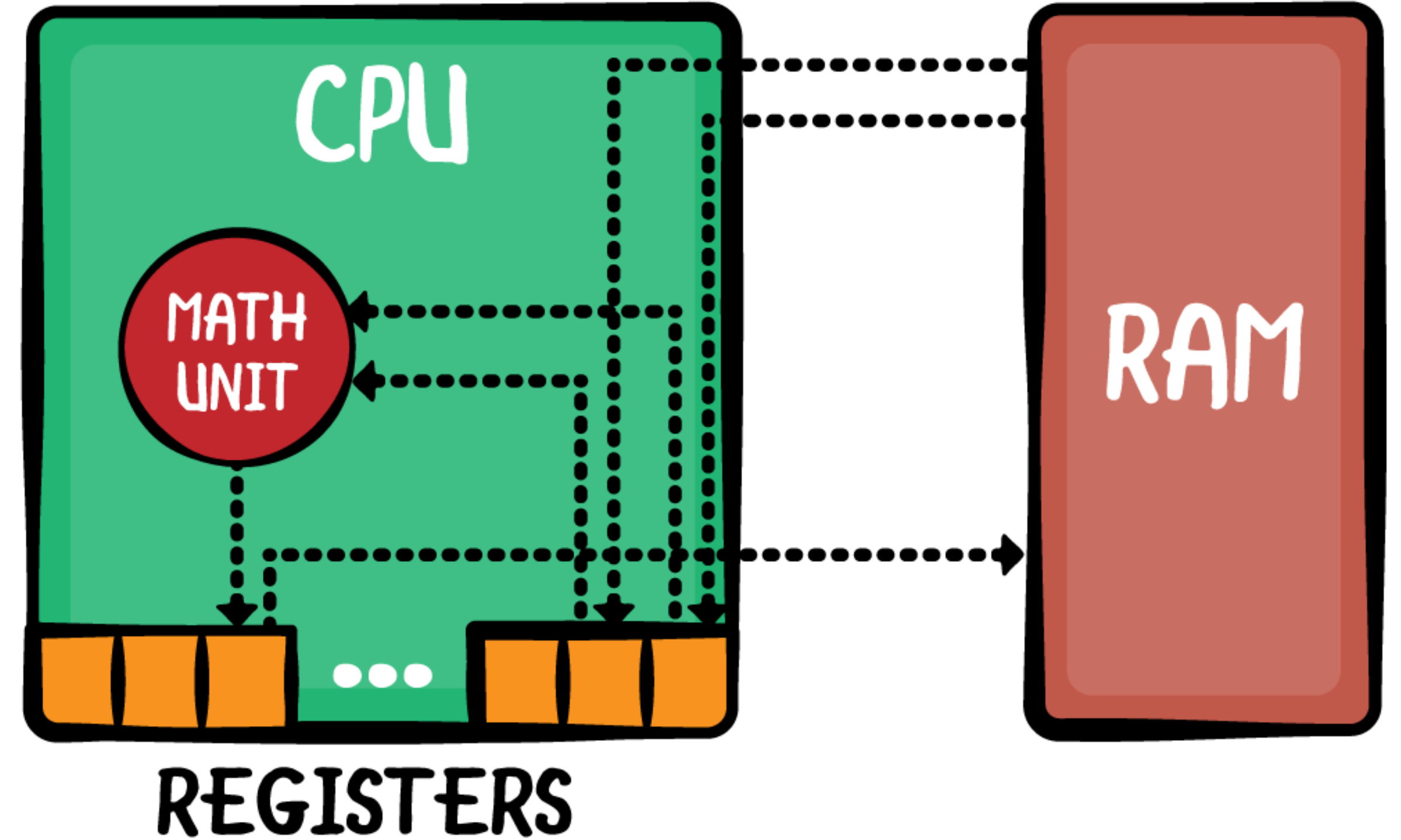
<> **Exercises-Day1.swift** Raw

```
1  let password = "HelloWorld"
2  let repeatPassword = "HelloWorld"
3
4  if password == repeatPassword {
5      print("Passwords are equal")
6  } else {
7      print("Passwords are not equal")
8  }
```

İfadeler, Değişkenler ve Sabitler

Bilgisayar nasıl çalışır?

- CPU / **C**entral **P**rocessing **U**nit
(Merkezi İşlem Birimi)
- Toplama, çıkarma gibi aritmetik işlemler
- Registers (Küçük bellek birimleri)
- RAM / **R**andom **A**ccess **M**emory
(Rastgele Erişimli Bellek)

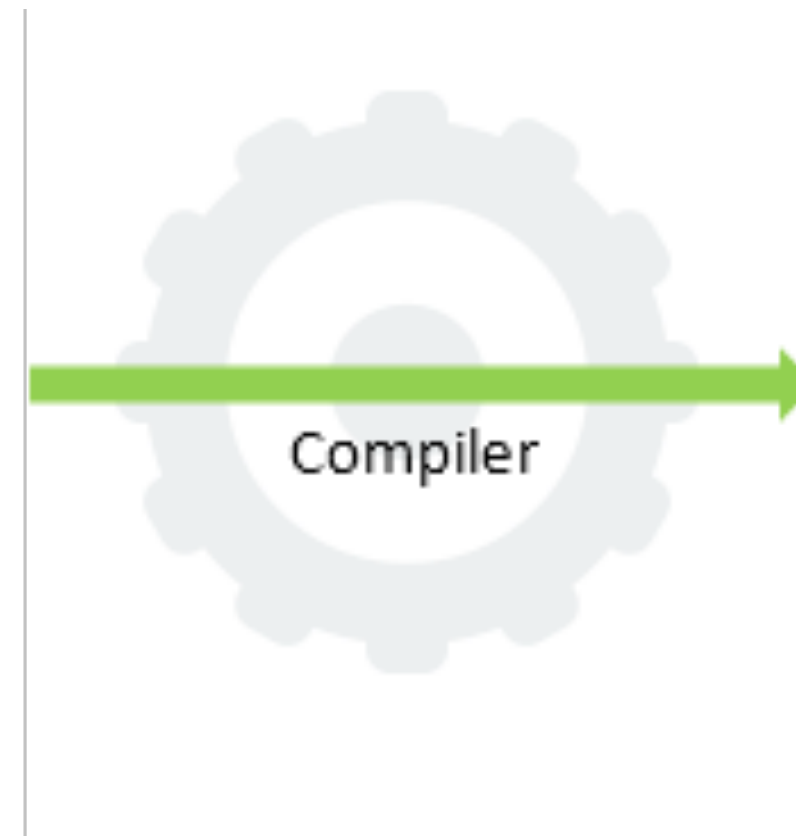


Bilgisayar nasıl çalışır?

- Derleyici (Compiler)

```
var name = "Murat Adıgüzel"  
print("Hello \(name)!")
```

Kaynak Kodu



```
0001011001  
0010110110  
1010001001  
0010100010  
1001011001
```

Makine Komutu

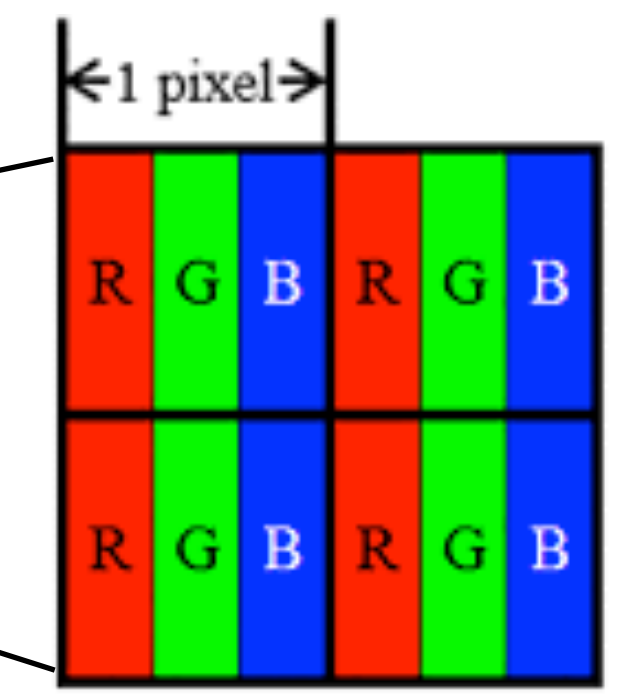
Bilgisayar nasıl çalışır?

Sayılar ile ifade

- Metin bloğu (String)

“Hello, Swift!”

- Resim



KIRMIZI

Red %100

Green % 0

Blue %0

Bilgisayar nasıl çalışır?

Sayılar ile ifade

- Sayılar (Decimals)

423

1000	100	10	1
0	4	2	3

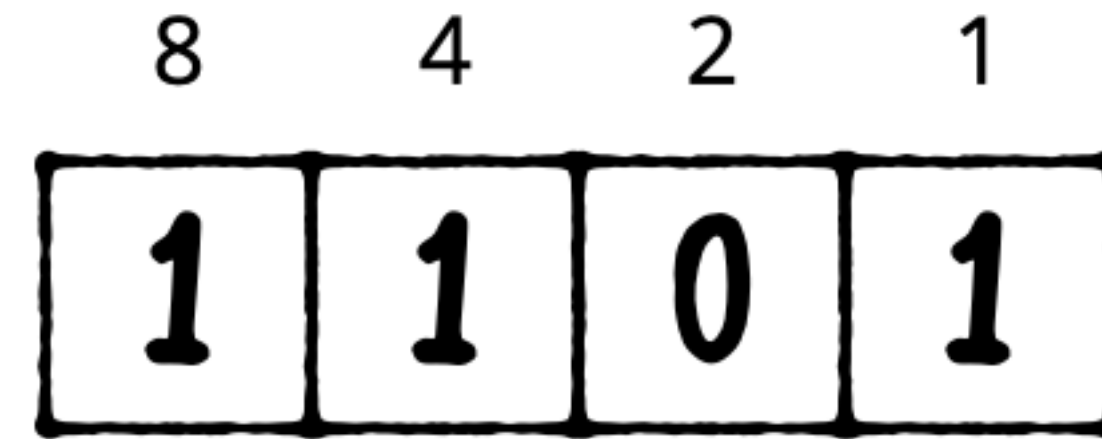
$$(0 * 1000) + (4 * 100) + (2 * 10) + (3 * 1) = 423$$

Bilgisayar nasıl çalışır?

Sayılar ile ifade

- İkili Sayılar (Binary numbers)

1101



$$(1 * 8) + (1 * 4) + (0 * 2) + (1 * 1) = 13$$

423₍₁₀₎ \longrightarrow **110100111**₍₂₎

$$(1 * 256) + (1 * 128) + (0 * 64) + (1 * 32) + (0 * 16) + (0 * 8) + (1 * 4) + (1 * 2) + (1 * 1) = 423$$

Bilgisayar nasıl çalışır?

Sayılar ile ifade

- İkili Sayı Terimleri
 - * Bit 0 veya 1
 - * Byte = 8 bit 0011 1010
 - * Nibble = 4 bit 1011
- CPU'daki register'lar genellikle 32 bit veya 64 bit uzunluğundadırlar.

Bilgisayar nasıl çalışır?

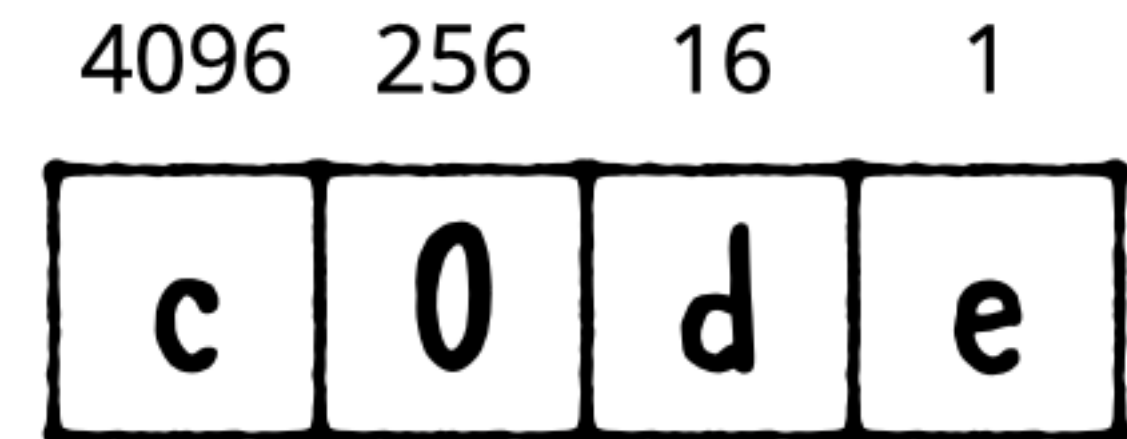
Sayılar ile ifade

- Onaltılık Sayılar (Hexadecimals)

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

$$(12 * 4096) + (0 * 256) + (13 * 16) + (14 * 1) = 49374$$



Bilgisayar nasıl çalışır?

Sayılar ile ifade

- Onaltılık Sayılar (Hexadecimals)

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

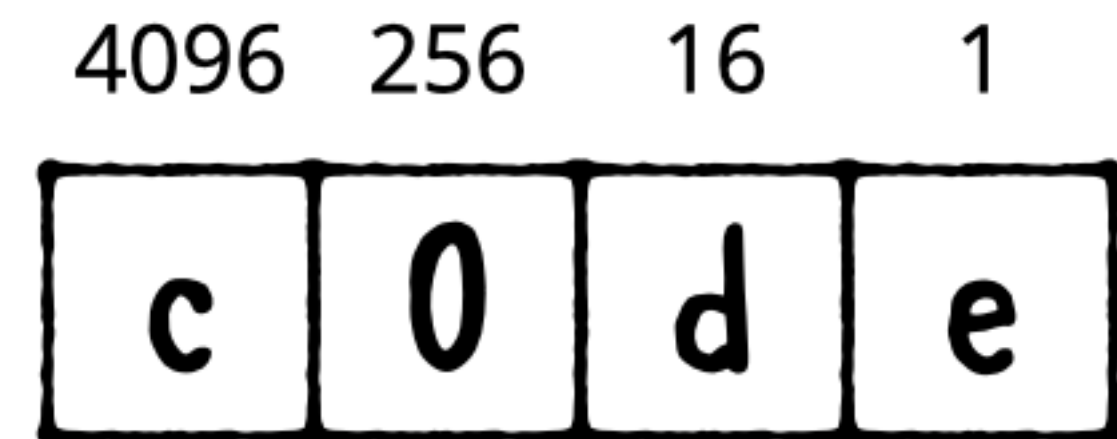
c = 1100

0 = 0000

d = 1101

e = 1110

1100 0000 1101 1110₍₂₎



Kod nasıl çalışır?

Pseudo Code

1. Adım - Sabit diskten fotoğraf yükleyin
2. Adım - Fotoğrafı 400 piksel genişliğe ve 300 piksel yüksekliğe yeniden boyutlandırın.
3. Adım - Fotoğrafa sepya filtre uygulayın.
4. Adım - Fotoğrafı yazdırın.

Kod nasıl çalışır?

Algoritma

1. Adım - Sabit diskten fotoğraf yükleyin
2. Adım - Fotoğrafı 400 piksel genişliğe ve 300 piksel yüksekliğe yeniden boyutlandırın.
3. Adım - Fotoğrafa sepya filtre uygulayın.
4. Adım - Fotoğrafı yazdırın.

Xcode Playground

İfadeler, Değişkenler ve Sabitler

Egzersizler

1. **myAge** adında, **Int** türünde bir sabit tanımayıp bu sabite yaşınızı atayın.
2. **averageAge** adında bir değişken tanımlayın. İlk değer olarak bu değişkene yaşınızı atayın. Sonra, bu değişkene sizin ve benim yaşımın ortalamasını atayın. (Yaşım 33)
3. **testNumber** adında bir sabit oluşturun ve onu istediğiniz bir tamsayı atayın. Ardından, **evenOdd** adında başka bir sabit oluşturun ve bunu **testNumber** modulo 2'ye eşit olarak ayarlayın. Şimdi **testNumber**'ı çeşitli sayılarla değiştirin. **evenOdd** sabitinin aldığı değerler nasıl değişiyor?
4. **answer** adında bir değişken oluşturun ve 0 ilk değerini verin. Bunu 1 artırın. Buna 10 ekleyin. 10 ile çarpın. Ardından sağa 3 kaydırın. Tüm bu işlemlerden sonra yanıt nedir?

İfadeler, Değişkenler ve Sabitler

Egzersizler

5. **dogs** adlı bir **Int** değişkeni tanımlayın ve bunu sahip olduğunuz köpeklerin sayısına eşit olarak ayarlayın. Sonra yeni bir köpek yavrusu aldığınızı ve değişkenleri birer birer artırdığınızı düşünün.
6. Kodun derlenebilmesi için ilk satırda **var** mı **let** mi kullanmalıyız?

```
age: Int = 16
```

```
print(age)
```

```
age = 30
```

```
print(age)
```

İfadeler, Değişkenler ve Sabitler

Egzersizler

7. Verilen **x** ve **y** değerine göre aşağıdaki üç sabitin değerleri nedir?

```
let x: Int = 46
```

```
let y: Int = 10
```

```
let answer1: Int = (x * 100) + y
```

```
let answer2: Int = (x * 100) + (y * 100)
```

```
let answer3: Int = (x * 100) + (y / 10)
```

İfadeler, Değişkenler ve Sabitler

Egzersizler

8. Aşağıdaki ifadenin değerinin değişmeyeceği şekilde parantezler ekleyin.

$$8 - 4 * 2 + 6 / 3 * 4$$

9. **Double** türünde **rating1**, **rating2** ve **rating3** adlı üç sabit tanımlayın ve her birine bir değer atayın. Üçünün ortalamasını hesaplayın ve sonucu **average** adlı bir sabit değerde saklayın.

10. Elektrikli bir cihazın gücü, voltajın akımla çarpılmasıyla hesaplanır. **voltage** adında **Double** türünde bir sabit tanımlayın ve ona bir değer atayın. Ardından, **Double** türünde, **current** adında bir sabit tanımlayın ve ona bir değer atayın. Son olarak, **power** adında **Double** türünde bir sabit tanımlayıp voltaj ve akım değerleri ile gücü hesaplayın.

İfadeler, Değişkenler ve Sabitler

Egzersizler

11. Elektrikli bir cihazın direnci gücün akımın karesine bölünmesi ile hesaplanır. Direnci hesaplayın ve bunu **Double** türünde **resistance** adında bir sabitte saklayın.
12. **arc4random()** işlevini kullanarak rastgele bir tam sayı oluşturabilirsiniz. Bu, **0** ile **4294967295** arasında herhangi bir sayı oluşturur. Bu rastgele sayıyı istediğiniz aralığa kesmek için modulo (%) operatörünü kullanabilirsiniz.

Sabit bir **randomNumber** tanımlayın ve ona **arc4random()** ile oluşturulan rastgele bir sayı atayın. Sonra **diceRoll** adlı bir sabiti tanımlayın ve **1** ile **6** arasında rastgele bir sayı oluşturmak için bulduğunuz rastgele sayıyı kullanın. (İpucu: **arc4random()** 'a erişmek için **import Foundation** satırını eklemeniz gerekir.)

İfadeler, Değişkenler ve Sabitler

Egzersizler

13. İkinci dereceden denklem, $a \cdot x^2 + b \cdot x + c = 0$ biçimindedir. Bunu sağlayan x değerleri,

$$x = (-b + \text{sqrt}(b^2 - 4 * a * c)) / (2 * a) \text{ ve}$$

$$x = (-b - \text{sqrt}(b^2 - 4 * a * c)) / (2 * a)$$

denklemleri kullanılarak çözülebilir.

Double türünde **a**, **b** ve **c** adlı üç sabit tanımlayın. Ardından, yukarıdaki denklemleri kullanarak **x** için iki değeri hesaplayın. Sonuçları, **Double** türünde **root1** ve **root2** adlı sabitlerde saklayın.

İfadeler, Değişkenler ve Sabitler

Özet

1. Bilgisayarlar, en temel düzeyde basit matematik yaparlar.
2. Bir programlama dili, derleyicinin CPU'nun çalıştırabileceği komutlara dönüştürdüğü kodu yazmanıza izin verir.
3. Bilgisayarlar, binary olarak da bilinen 2'lik tabandaki sayılar üzerinde çalışır.
4. Playground, kodun nasıl çalıştığı hakkında anında geri bildirim sağlayarak Swift kodunu hızlı ve verimli bir şekilde yazmanıza ve test etmenize olanak tanır.
5. Kod yorumları `//` ile başlayan bir satırla veya `/*` ve `*/` ile ayrılmış birden çok satırla gösterilir.

İfadeler, Değişkenler ve Sabitler

Özet

6. Kodunuzu belgelemek için yorumları kullanırsınız.
7. Konsola veri yazdırmak için **print** işlevini kullanabilirsiniz.
8. Aritmetik operatörler

Toplama: +

Çıkarma: -

Çarpma: *

Bölme: /

Modulo: %

İfadeler, Değişkenler ve Sabitler

Özet

9. Swift'te **min**, **max**, **squareRoot**, **sin**, **cos** gibi çeşitli matematiksel işlevler bulunur.
10. Sabitler ve değişkenler verilere isim verir.
11. Bir sabit belirledikten sonra, verilerini değiştiremezsiniz, ancak bir değişkenin verilerini istediğiniz zaman değiştirebilirsiniz.
12. Kendinizi ve iş arkadaşlarınızı daha sonra baş ağrısından kurtarmak için her zaman değişkenlere ve sabitlere anlamlı isimler verin

İfadeler, Değişkenler ve Sabitler

Özet

13. Aritmetik işlem gerçekleştirmek ve ardından değişkene geri atamak için operatörler:

Toplama ve atama: $+=$

Çıkarma ve atama: $-=$

Çarpma ve atama: $*=$

Bölme ve atama: $/=$

Türler ve İşlemler

Türler ve İşlemler

Egzersizler

1. **age1** adında bir sabit oluşturun ve buna **42** değerini atayın. **age2** adlı bir sabit oluşturun ve buna **21** değerini atayın. Her ikisinin türünün **Int** olarak doğru bir şekilde çıkarıldığını Option-tıklama kullanarak kontrol edin.
2. **avg1** adında bir sabit oluşturun ve $(\text{age1} + \text{age2}) / 2$ işlemini kullanarak bu sabite **age1** ve **age2** ortalamasını atayın. **avg1**'in türünü ve sonucunu kontrol etmek için Option-tıklamayı kullanın. Neden yanlış?
3. İfadede **age1** ve **age2**'yi **Double** türüne dönüştürerek yukarıdaki hatayı düzeltin. **avg1**'in türünü ve sonucunu kontrol etmek için Option-tıklamayı kullanın. Şimdi neden doğru?

Türler ve İşlemler

String

Türler ve İşlemler

String

Unicode

c	a	f	e
99	97	102	101

Türler ve İşlemler

String

Unicode

c	a	f	e
99	97	102	101

code point / code position



Türler ve İşlemler

String

Unicode

c	a	f	é
99	97	102	233

Türler ve İşlemler

String

Unicode



电	脑	编	程
30005	33041	32534	31243

“Bilgisayar Programlama”

Türler ve İşlemler

String

Unicode

	
128169	128512

Türler ve İşlemler

Egzersizler

1. `firstName` adında bir `String` sabiti oluşturun ve ona adınızı atayın. Ayrıca, `lastName` adında bir `String` sabiti oluşturun ve buna soyadınızı atayın.
2. `firstName` ve `lastName` sabitlerini bir boşlukla ayırıp birbirine ekleyerek `fullName` adlı bir string sabiti oluşturun.
3. Interpolation kullanarak, kendinizi tanıtan bir `String` oluşturmak için `fullName` sabitini kullanan `myDetails` adında bir `String` sabiti oluşturun. Sonuçta şuna benzer bir değer elde etmeniz gerekiyor: **"Merhaba, benim adım Murat Adıgüzel."**

Türler ve İşlemler

Egzersizler

1. Üç **Int** ve ardından bir **Double** değer tutabilecek bir **tuple** tanımlayın. Bunu bir tarihi (gün, ay, yıl) ve ardından o tarih için ortalama bir sıcaklığı ifade etmek için kullanın.
2. Bileşenleri adlandırmak için **tuple**'ı değiştirin. Onlara içerdikleri verilerle ilgili adlar verin: **day**, **month**, **year** ve **averageTemperature**.
3. Bir satırda, gün ve ortalama sıcaklık değerlerini iki sabit halinde okuyun. Ay ve yılı yok saymak için alt çizgiyi kullanmanız gerekecektir.
4. Şimdiye kadar sadece sabit **tuple**'lar gördünüz. Ancak değişken **tuple** da oluşturabilirsiniz. İkinci egzersizde **let** yerine **var** kullanarak oluşturduğunuz **tuple**'ı bir değişken yapın ve ortalama sıcaklığa yeni bir değer atayın.

Türler ve İşlemler

Daha Fazla Sayı Türü

Type	Minimum value	Maximum value	Storage size
Int8	-128	127	1
UInt8	0	255	1
Int16	-32768	32767	2
UInt16	0	65535	2
Int32	-2147483648	2147483647	4
UInt32	0	4294967295	4
Int64	-9223372036854775808	9223372036854775807	8
UInt64	0	18446744073709551615	8

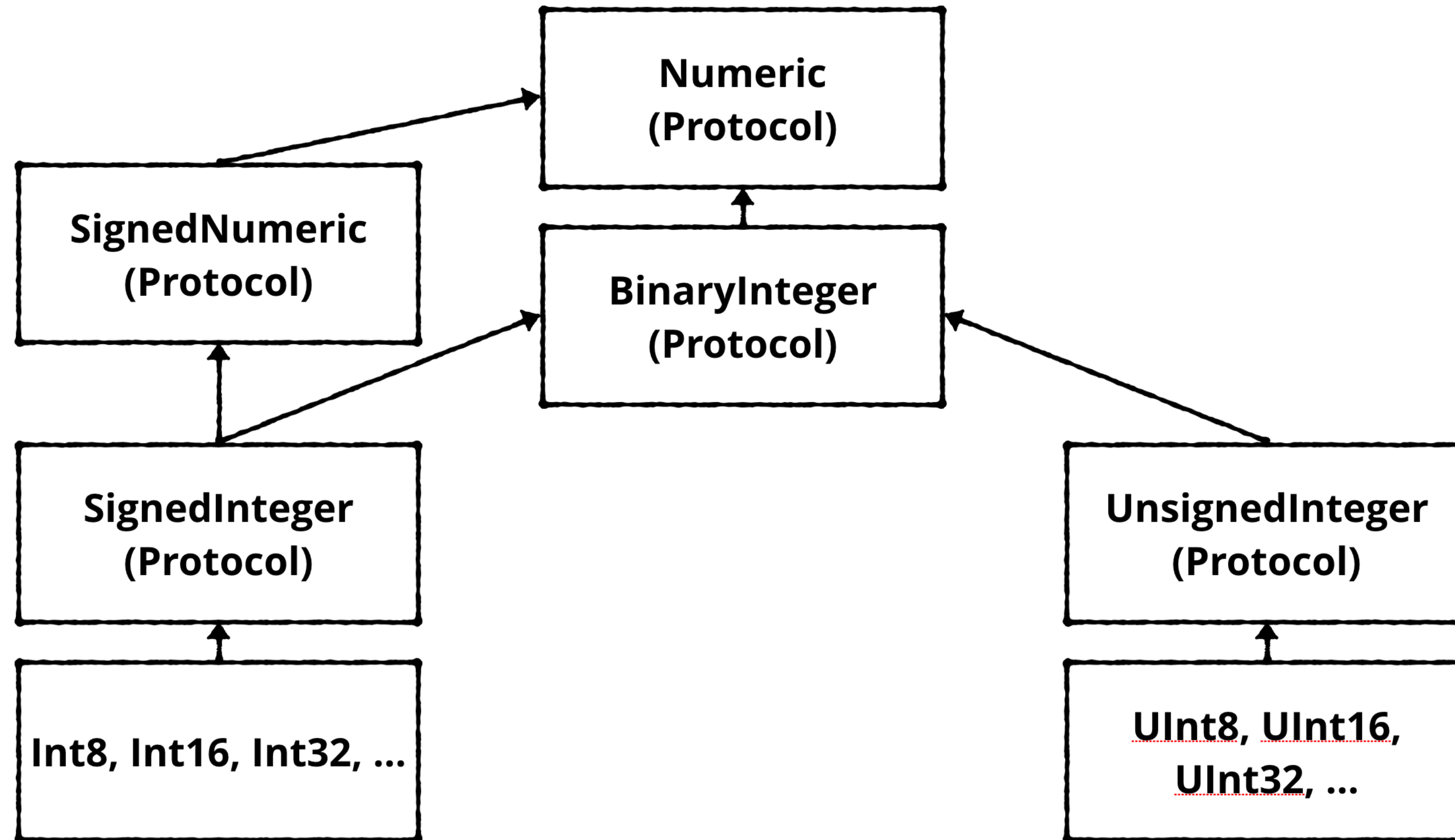
Türler ve İşlemler

Daha Fazla Sayı Türü

Type	Minimum value	Maximum value	Precision	Storage size
Float	1.175494E-38	3.402823E+38	6 digits	4
Double	2.225073e-308	1.797693E+308	15 digits	8

Türler ve İşlemler

Protokol (Önizleme)



Türler ve İşlemler

Egzersizler

1. **coordinates** olarak adlandırılan bir sabit oluşturun ve ona **3** ve **5** içeren bir **tuple** atayın.
2. Satır (**row**) ve sütun (**column**) bileşeni olan **namedCoordinate** adında bir sabit oluşturun.
3. Aşağıdakilerden hangileri geçerli ifadelerdir?

```
let character: Character = "Dog"
```

```
let character: Character = "🐕"
```

```
let string: String = "Dog"
```

```
let string: String = "🐕"
```


Türler ve İşlemler

Egzersizler

4. Aşağıdaki kod derlenir mi?

```
let tuple = (day: 15, month: 8, year: 2021)  
  
let day = tuple.Day
```

5. Aşağıdaki kodda sorun nedir?

```
let name = "Murat"  
  
name += " Adıgüzel"
```

Türler ve İşlemler

Egzersizler

6. **value** isimli sabitin değeri nedir?

```
let tuple = (100, 1.5, 10)
```

```
let value = tuple.1
```

7. **month** isimli sabitin değeri nedir?

```
let tuple = (day: 15, month: 8, year: 2021)
```

```
let month = tuple.month
```

Türler ve İşlemler

Egzersizler

8. **summary** isimli sabitin değeri nedir?

```
let number = 10
```

```
let multiplier = 5
```

```
let summary = "\ (number) multiplied by \ (multiplier)  
equals \ (number * multiplier)"
```

Türler ve İşlemler

Egzersizler

9. **a** ve **b**'nin toplamı eksi **c**'nin değerini **d** adlı sabite atayınız?

```
let a = 4
```

```
let b: Int32 = 100
```

```
let c: UInt8 = 12
```

10. **Double.pi** ve **Float.pi** arasındaki sayısal fark nedir?

Türler ve İşlemler

Özet

1. Tür dönüştürme (type conversion), bir türdeki değerleri diğerine dönüştürmenize olanak tanır.
2. Karışık türlerle (+, -, *, /) gibi temel aritmetik operatörler kullanırken tür dönüşümü gereklidir.
3. Tür çıkarımı (type inference), Swift zaten biliyorsa, türü belirtmemenize olanak tanır.
4. Unicode, karakterleri sayılarla eşleme standardıdır.
5. Unicode'daki tek bir eşlemeye code point veya code position denir.

Türler ve İşlemler

Özet

6. **Character** veri türü tek karakterleri depolar. **String** veri türü, karakter koleksiyonlarını veya dizeleri depolar.
7. Toplama operatörünü kullanarak **String** türündeki değerleri birleştirebilirsiniz.
8. Yerinde bir **String** oluşturmak için string interpolation kullanabilirsiniz.
9. Verileri tek bir veri türünde gruplamak için tuple kullanabilirsiniz.
10. Tuple'lar adsız veya adlandırılmış olabilir. Tuple içindeki değerlere isimsiz tuple'lar için dizin numaraları, isimlendirilmiş tuple'lar için verdiğiniz isimler ile erişebilirsiniz.

Türler ve İşlemler

Özet

- 11.Farklı depolama ve hassaslık yeteneklerine sahip çok sayıda sayısal tür vardır.
- 12.Type alias ile Swift'te hazır türleri kullanarak kendi türlerinizi oluşturabilirsiniz.
- 13.Protokoller, birden çok türün paylaştığı ortak işlemleri açıklarlar.

Temel Kontrol Akışı

Temel Kontrol Akışı

Egzersizler

1. **myAge** adında bir sabit oluşturun ve buna yaşınızı atayın. Ardından, yaşın **13** ile **19** yaş aralığındaki birini gösterip göstermediğini belirlemek için boolean mantığını kullanan **isTeenager** adlı bir sabit oluşturun.
2. **theirAge** adlı başka bir sabit oluşturun ve **33** olan yaşına ayarlayın. Ardından, hem sizin hem de benim genç olup olmadığımızı belirlemek için boolean mantığını kullanan **bothTeenagers** adlı bir sabit oluşturun.
3. **trainee** adlı bir sabit oluşturun ve buna isminizi atayın. **educator** adında bir sabit oluşturun ve buna benim adımlı atayın. **trainee** ve **educator** sabitlerinin değerlerinin eşit olup olmadığını belirlemek için **String** eşitliğini kullanan **traineeIsEducator** adlı bir sabit oluşturun.
4. **trainee**'nin **educator**'dan önce gelip gelmediğini belirlemek için **String** karşılaştırmasını kullanan **traineeBeforeEducator** adlı bir sabit oluşturun.

Temel Kontrol Akışı

Egzersizler (if - ternary operator)

1. **myAge** adında bir sabit oluşturun ve buna yaşınızı atayın. Yaşınız **13** ile **19** arasındaysa "**Teenager**", **13** ile **19** arasında değilse "**Not a Teenager**" yazdırmak için bir **if** ifadesi yazın.
2. **answer** adında bir sabit oluşturun ve bunu yukarıdaki egzersiz ile sonuçlar aynı olacak şekilde ternary operator kullanarak yeniden yazın.