ONDOKUZ MAYIS ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİL-103 Programlamaya Giriş-1 Dersi 2. Hafta Ödevi

(Değişken Tanımlama, Oparatörler ve Kullanımı, String İşlemleri)

```
irb(main):001:0> 1
=> 1
irb(main):002:0> x=100
=> 100
irb(main):003:0> y=10
=> 10
irb(main):004:0> puts x-y
90
=> nil
```

1- “x” ve “y” değişkenlerine sırasıyla “100” ve “10” sayı değerlerini atadık. “x” değişkenine tanımladığımız sayı değerinden “y” değişkenine tanımladığımız sayı değerini çıkardık ve işlem sonucunu “puts” komutu ile ekrana bastırdık.

```
irb(main):005:0> 2
=> 2
irb(main):006:0> x7="Bu ruby çokeğlenceli"
=> "Bu ruby \x87oke\xA7lenceli"
irb(main):007:0> puts x7
Bu ruby çokeğlenceli
=> nil
```

2- “x7” değişkenine bir string olarak “Bu ruby çokeğlenceli” değerini atadık. Sonra “x7” değişkenine atadığımız string türü değeri “puts” komutunu kullanarak ekrana bastırdık.

```
irb(main):008:0> 3
=> 3
irb(main):009:0> 7x="Ondokuz Mayıs University"
SyntaxError: (irb):9: syntax error, unexpected tIDENTIFIER, expecting end-of-inp
ut
7x="Ondokuz Mayıs University"
  ^
        from C:/Ruby22-x64/bin/irb:11:in `<main>'
irb(main):010:0> puts 7x
SyntaxError: (irb):10: syntax error, unexpected tIDENTIFIER, expecting end-of-in
put
        from C:/Ruby22-x64/bin/irb:11:in `<main>'
```

3- Stringler sayı içerebilir fakat bir sayıyla başlayamazlar. Bu sebepten “7x” tanımlanamaz. İfadenin kendisi tanımlanamadığı için “7x” ifadesine bir değerde atayamayız.

```
irb(main):011:0> 4
=> 4
irb(main):012:0> this_is = "car"
=> "car"
irb(main):013:0> puts this_is
car
=> nil
```

4- “this_is” değişkenine “car” string değerini atadık. Sonra “this_is” değişkenine atadığımız string türü değeri “puts” komutunu kullanarak ekrana bastırdık.

```
irb(main):014:0> 5
=> 5
irb(main):015:0> this is = "car"
NoMethodError: undefined method `this' for main:Object
        from (irb):15
        from C:/Ruby22-x64/bin/irb:11:in `<main>'
irb(main):016:0> puts this is
NoMethodError: undefined method `this' for main:Object
        from (irb):16
        from C:/Ruby22-x64/bin/irb:11:in `<main>'
```

5- “this is” değişkenine “car” string türü değeri atamaya çalıştık. Ama değişkenin kelime aralarında boşluk olduğu için tanımlayamadı. Sonra “this is” değişkenine atamaya çalıştığımız değeri “puts” komutunu kullanarak ekrana bastırmaya çalıştık. Ama daha önceden değer atayamadığımız için komut hata verdi. Komut hatasının giderilmesi için değişkenin içindeki boşlukların yerine “_” (alt tire) kullanılmalıdır.

```
irb(main):017:0> 6
=> 6
irb(main):018:0> x=50
=> 50
irb(main):019:0> y=x*100
=> 5000
irb(main):020:0> x+=y
=> 5050
irb(main):021:0> puts x
5050
=> nil
```

6- “x” değişkenine “50” sayı değerini verdik. “y” değişkenine de “x*100” “x” değişkenine bağlı işlemsel değeri verdik. Sonra “x” ve “y” olarak tanımladığımız değişkenleri toplayarak “x” değişkeninin yeni değeri olarak atadık. Sonuç olarak “x” değişkeni artık “5050” sayı değerinde fakat “y” değişkeninin içerisindeki işlemde tanımlanan “x”, “50” sayı değerindedir. Yani bir değişkeni tanımladıktan sonra kullandığımız ve aslında değişkenin işlem veya işlemleri sırasında kullanılan ve sonuç değerine etki etmesi gereken değişken, başka bir değişkenmiş gibi algılandığından dolayı işleme etki etmez. Örnek olarak, eğer bu işlem sonucunda “y”nin değerini istersek, “y”nin aynı kaldığını görürüz.

```
irb(main):001:0> x=50
=> 50
irb(main):002:0> y=x*100
=> 5000
irb(main):003:0> x+=y
=> 5050
irb(main):004:0> puts x
5050
=> nil
irb(main):005:0> puts y
5000
=> nil
```

```
irb(main):022:0> 7
=> 7
irb(main):023:0> puts 100/7
14
=> nil
```

7- “100” ve “7” sayısal değerlerinin bölme işlemleri sonucunda ortaya çıkan değerin ,kesirli sayılarından hariç kısmını “puts” komutunu kullanarak ekrana bastırdık.

```
irb(main):024:0> 8
=> 8
irb(main):025:0> puts 100.0/7.0
14.285714285714286
=> nil
```

8- “100” ve “7” sayısal değerlerinin bölme işlemleri sonucunda ortaya çıkan değeri, kesirli kısmı dahil olmak üzere “puts” komutunu kullanarak ekrana bastırdık.

```
irb(main):026:0> 9
=> 9
irb(main):027:0> x=100
=> 100
irb(main):028:0> y=7
=> 7
irb(main):029:0> puts x/y
14
=> nil
```

9- “x” değişkenine “100” sayısal değerini, “y” değişkenine “7” sayısal değerini atadık. “x” ve “y” değişkenlerini bölme işlemine tâbi tuttuktan sonra, sonucu (kesirli kısmı hariç) “puts” komutu ile ekrana bastırdık.

```
irb(main):030:0> 10
=> 10
irb(main):031:0> x=100
=> 100
irb(main):032:0> y=7
=> 7
irb(main):033:0> puts x.to_f/y.to_f
14.285714285714286
=> nil
```

10- “x” değişkenine “100” sayısal değerini, “y” değişkenine “7” sayısal değerini atadık. “x” ve “y” değişkenlerinin sonunda “to_f” metodunu kullanarak sonlarındaki kesir kısmını aktif hâle getirdik. Sonra “x” ve “y” değişkenlerini bölme işlemine tâbi tuttuktan sonra “puts” komutunu kullanarak ekrana bastırdık.

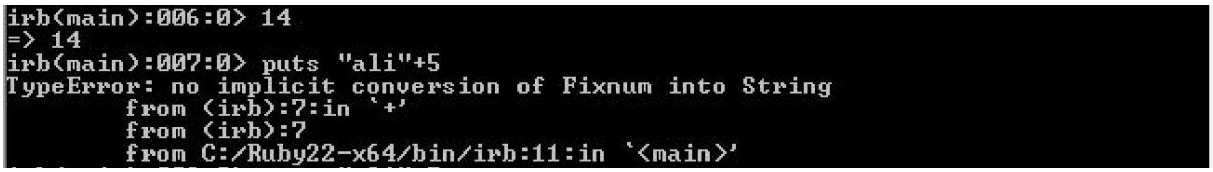
```
irb(main):034:0> 11
=> 11
irb(main):035:0> puts 5.5.to_i
5
=> nil
```

11- “5.5” kesir kısmı aktif sayısal değeri “to_i” metodunu kullanarak Float(kesir kısmı aktif sayısal değer) hâlinden Fixnum (sayısal değer) hâle getirdik.

```
irb(main):036:0> 12
=> 12
irb(main):037:0> a=5
=> 5
irb(main):038:0> b=5.5
=> 5.5
irb(main):039:0> c="OMU"
=> "OMU"
irb(main):040:0> puts a.class
Fixnum
=> nil
irb(main):041:0> puts b.class
Float
=> nil
irb(main):042:0> puts c.class
String
=> nil
irb(main):043:0>
```

12- “a”, “b” , “c” değişkenlerine değerler verdik. Bu değerler sırasıyla “ ’5’ , ‘5.5’ , ‘OMU’ ”. Sonra bu değerleri “class” metodunu kullanarak sınıflarını öğrenmiş olduk. “a” = sayısal değer, “b” = kesir kısmı aktif sayısal değer, “c” = yazısal (string) değer. Sonra “puts” u kullanarak her birini ayrı ayrı ekrana bastırdık.

13- “a” ve “b” ye birer string değeri verdik. “a” nın değeri “OMU”, “b”nin değeri “Ceng”. Toplama işlemini kullanarak ikinci strşngi birincinin arkasına ekledik. “puts” u kullanarak ekrana bastırdık. Sonra stringlerin arasına boşluk eklemek için “ ” ’u kullandık.

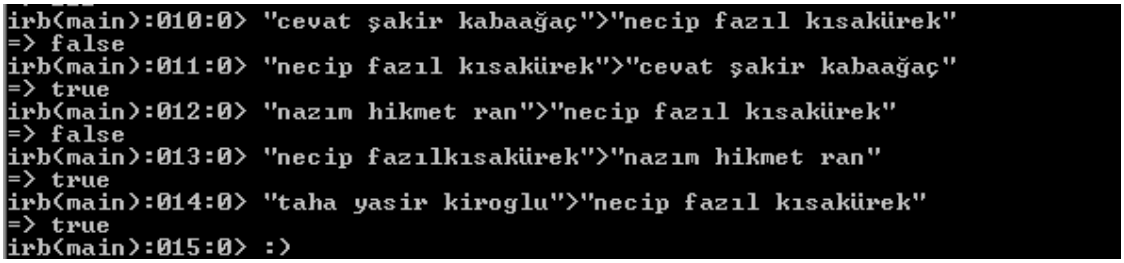


14- “ali” bir string, 5 bir sayı değeridir. Bir string ve bir sayı değeri toplanmaya çalıştık. Bu sebepten hata verdi.

15- 14. sorudaki hatayı “to_s” metodunu kullanarak giderdik. Sonra “puts” ile ekrana bastırdık.

16- “*yazdırılmak istenilen sayı” kullanılarak yazdığımız string (“ali”) istediğimiz sayıda ürettirdik. Sonra “puts” ‘la ekrana bastırdık. Bu örnekte 5 tane ürettik bastırdık.

17- Stringlerle büyüktür-küçüktür işlemi yaparılırken, doğruluk işlemi ASCII sisteminin sayısal değerlerine göre değersel olarak aynı olmayan ilk sembolleri arasında olur. Buna örnek olarak :



İşlemimizde “x”.ord = 120 ve “y”.ord =121 dir. Bu sebepten işlem hatalı olduğundan false değerini almıştır. Sonra “puts” değerini yazarak ifademizin doğruluk değerini ekrana bastık.

```
irb(main):001:0> 18
=> 18
irb(main):002:0> puts "y">"x"
true
=> nil
```

18- 17. Örnekte de bahsettiğimiz gibi “x”.ord = 120 ve “y”.ord =121 olduğu için ifade doğrudur. Sonra “puts”u kullanarak ifademizin doğruluk değerini ekrana bastık.
Eğer sembollerimizi kullanırken boşluk ifadesi varsa ASCII sistemine göre klavye işaretlerinin çoğu gibi boşlukta harflerin hepsinden küçüktür.
Örneğin :

```
irb(main):017:0> "na ber">"naber"
=> false
irb(main):018:0> "naber">"na ber"
=> true
```

Bunun sebebi:

```
irb(main):020:0> " ".ord
=> 32
irb(main):021:0> "b".ord
=> 98
```

```
irb(main):003:0> 19
=> 19
irb(main):004:0> puts "A".ord
65
=> nil
irb(main):005:0> puts 65.chr
A
=> nil
```

19- “ord” ifadesi tırnak içine yazılan ilk sembolün ASCII sistemindeki karşılığını verir. “chr” ifadesi ise verilen sayı değerinin ASCII sistemindeki sembol karsılığını verir.
Yaptığımız işlemde önce “A” sembolünün sayı karşılığını bulduk. İkinci işlemde ise 65 sayısının harf karşılığını bulduk.

```
=> 20
irb(main):007:0> puts "durmus".capitalize
Durmus
=> nil
irb(main):008:0> puts "dURMus".capitalize
Durmus
=> nil
```

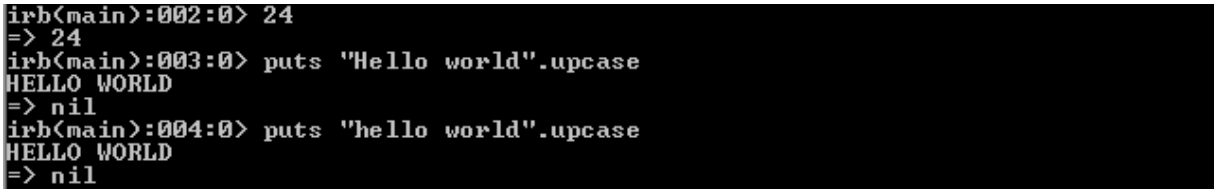
20- “capitalize” ,değer tırnağının içindeki değerin ilk sembolünü eğer harfle başlıyorsa ve kullanılan harf büyütülebiliyorsa büyütür. İlk sembol haricindeki sembolleri de eğer semboller küçültülebiliyorsa küçültür. “capitalize” harf haricindeki sembollere etki etmez.
Kelimelerimizi “capitalize” sayesinde baş harfleri hariç küçük hale getirdik. Harfler bu işlemlerden geçtikten sonra “puts”u kullanarak ürünümüzü ekrana bastık.

```
irb(main):009:0> 21
=> 21
irb(main):010:0> puts "HELLO WORLD".downcase
hello world
=> nil
irb(main):011:0> puts "Hello World".downcase
hello world
=> nil
irb(main):012:0> puts "hello world".downcase
hello world
=> nil
```

21- “downcase”i kullanarak harfler ister küçük olsun, ister büyük olsun hepsini küçük hale getirdik. Sonra “puts”u kullanarak ekrana bastık. “downcase” harflerin haricindeki sembollere dokunmuyor ,semboller oldukları gibi kalıyorlar.

22- “chop”u kullanarak kelimenin son sembolünü sildik. Sonra “puts”u kullanarak ekrana bastık. “chop” kelimenin sonuna koymadığımız sürece “\n”i ve tırnak işaretini(“”) görmemekte ve onlara etki etmemektedir.

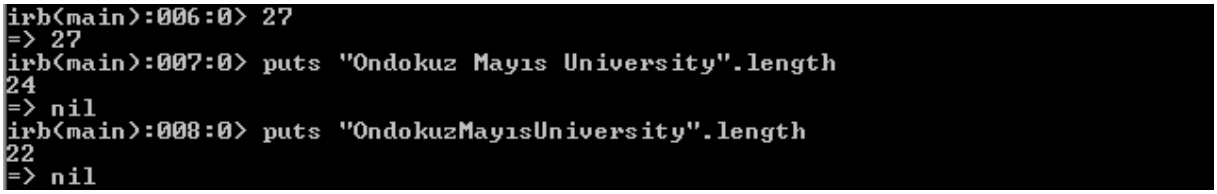
23- “next”i kullanarak stringin son harfini değiştirdik. Sonra “puts”u kullanrak ekrana bastık.



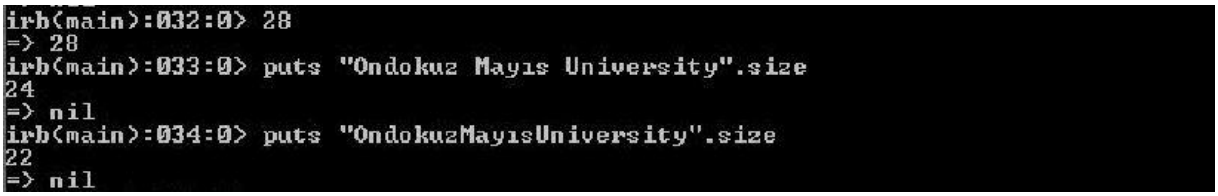
24- “upcase”i kullanarak harfler ister küçük olsun, ister büyük olsun hepsini büyük hale getirdik. Sonra “puts”u kullanarak ekrana bastık. “upcase” harflerin haricindeki sembollere dokunmuyor ,semboller oldukları gibi kalıyorlar.

25- “swapcase” i kullanarak kelimelerdeki harflar büyükse küçük, küçükse büyük hâle getirdik.En son “puts” u kullanarak ekrana bastık.

26- “upcase”i kullanarak bütün harflari büyük yaptık. “reverse” yi kullanarak “DURMUS” kelimesini ters çevirdik ve “SUMRUD” yaptık. “next” le son harfini değiştirerek “E” yaptık. Sonra “chop”u kullanarak son harfi (D) sildik. Son olarak “SUMRU” kelimesini “puts”u kullanarak ekrana bastık. Bu örnek için eğer “chop”u kullanacaksak “next”i kullanmaya gerek yoktur. Çünkü zaten sileceksek değiştirmenin anlamı yoktur.



27- “length” yi kullanarak verilen stringlerde kullanılan sembol sayısını bulduk. İki örnek arasındaki farklılıktan da anlayabileceğimiz gibi boşluk bir sembol sayılmaktadır. “puts”u kullanarak da ekrana bastırdık.



28- “size” yi kullanarak verilen stringlerde kullanılan sembol sayısını bulduk. İki örnek arasındaki farklılıktan da anlayabileceğimiz gibi boşluk bir sembol sayılmaktadır. “puts”u kullanarak da ekrana bastırdık.