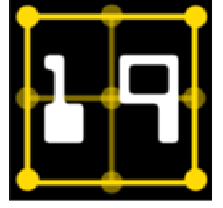




T.C. ONDOKUZ MAYIS ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



PYTHON İLE BİLGİSAYAR PROGRAMLAMAYA GİRİŞ



HAZIRLAYAN

Arş. Gör. Emre Gürbüz

ÖNSÖZ

Bu laboratuvar rehberi, Python programlama dilinin teorik olarak öğrenilmesini destekleyen laboratuvar çalışmalarında öğrencilere belirli bir rota çizmek üzere, çeşitli İngilizce ve Türkçe kaynaklardan yararlanılarak; ancak bu kaynaklara harfi harfine de bağlı kalmaksızın düzenlenmiş, Ondokuz Mayıs Üniversitesi Bilgisayar Mühendisliği Bölümü öğrencilerinin kullanımına sunulmuştur.

İki aylık bir çalışmanın ürünü olan bu rehberde; anlatımlarda ve sorularda kullanılan ifadeler mümkün olduğunca anlaşılır olarak hazırlanmaya çalışılmış, Türkçenin düzgün kullanımına özen gösterilmiştir. Ayrıca örnek kodlar da kullanılmakta olan Python sürümünün kod renklendirme stiline uygun olarak renklendirilmiş ve anlaşılabilirlikleri artırılmaya çalışılmıştır.

Gösterilen özene rağmen gözden kaçan anlamsal, biçimsel ve görsel eksiklikler içinse öğrencilerin geri bildirimleri dikkate alınarak gerekli görülen düzeltmeler yapılacaktır. Bu çalışmanın bölümümüz öğrencilerine yararlı olmasını diliyorum.

Arş. Gör. Emre Gürbüz

Ondokuz Mayıs Ünv.

Bilgisayar Müh. Bölümü

UYULMASI GEREKEN KURALLAR

- Fonksiyon, argüman ve değişken isimlerinde **küçük harf** kullanılmalı, Türkçe karakter kullanılmamalı ve gerektiğinde kelimeleri ayıtr etmek için alt çizgi (“_”) kullanılmalıdır:

ogrencisayisi	→	yanlış kullanım
ogrenciSayisi	→	yanlış kullanım
OgrenciSayisi	→	yanlış kullanım
ogrenci_sayisi	→	doğru kullanım

- Kod içerisine yazılan yorumlar, ilgili oldukları kodun üzerindeki satıra yazılmalıdır. Kod ile aynı satıra (kodun bitimine) yorum yazılmamalıdır.
- Alıştırmalarda; fonksiyonlara ve kod dosyalarına verilmesi istenen isimlerin dışında keyfi isimler verilmemelidir.
- Fonksiyon, argüman ve değişken isimlerinde, aksi belirtilmedikçe, “l” (küçük “L”) ve “I” (büyük “i”) karakterlerinin kullanımından kaçınılmalıdır (“1” sayısı ile karıştırılmaması için). Ayrıca, “0” rakamı ile karıştırılmaması için “O” (büyük “o”) karakterinin kullanımından da uzak durulmalıdır.
- Belirtilen kurallara uyulmaması halinde değerlendirme esnasında **puan kırılacaktır**.

LABORATUVAR ÇALIŞMASI 1 - Python' a Giriş

Bu Çalışmanın Amacı

Bu çalışmadaki amacımız, kullandığımız sistem ve kullanım hedefimiz için en uygun olan Python sürümünü bilgisayarımıza yükleyerek bilgisayarımızı programlamaya başlamak için hazır hale getirmektir.

Kurulum yaptıktan sonraki amacımız, kurulumunu yapmış olduğumuz, Python yazılım geliştirme aracı olan IDLE ortamında örnek programlar hazırlayarak Python programlama diline giriş yapmaktır.

Hangi Python Bizim İçin Daha Uygun?

Pek çok popüler Linux sürümünde, ayrıca Macintosh OS X 10.2 ve sonrasındaki sürümlerinde işletim sistemiyle birlikte gelmesine rağmen Windows ile beraber gelmeyen Python' u, hemen hemen her işletim sistemi (Windows, Macintosh OS, tüm Linux sürümleri, Sun Solaris vs.) ile beraber kullanmak mümkündür. Hangi Python' un bizim için daha uygun olduğu sorusunun cevabı kullandığımız işletim sisteminde yatmaktadır.

Windows' ta Python

Ücretsiz olarak temin edilebilen ve açık kaynak kodlu olan Python' un en güncel halini indirerek bilgisayarımıza kurmak için aşağıdaki adımları takip etmeliyiz:

1. Python Windows yükleyicisini indirmek için, <http://www.python.org/download/releases/> adresine girelim.
2. Python 2.6.4' e tıklayalım.
3. Çıkan sayfanın altında, Windows için kurulum dosyasını indirelim (32 Bit' lik mimariye sahip bilgisayarlar için “Windows x86 MSI Installer”, 64 Bit' lik mimariye sahip olanlar içinse “Windows AMD 64 MSI Installer” indirilecektir).
4. “msi” uzantılı kurulum programını çalıştıralım, “ileri” diyerek kurulumu tamamlayalım.

5. **‘Başlat → Programlar’** altında “Python” u seçip, “IDLE (Python GUI)” a tıklayalım.
“Python Shell” penceresi açılacaktır. Kurulum tamamlanmıştır.

Linux’ ta Python (Ubuntu 9.10)

İşletim sisteminin kurulumu ile ilk başta bilgisayarımıza yüklenmiş olmamasına rağmen program ekleme özelliğini kullanarak Ubuntu’ yu yükleyebiliriz. Bunun için aşağıdaki işlemleri sırayla takip edelim:

1. **‘Uygulamalar → Ubuntu Yazılım Merkezi’** seçimini yapalım.
2. Sağ üstte çıkan arama alanına “idle” yazalım.
3. “IDLE (Python-2.6)” yı seçerek sağ tarafında çıkan oka basalım. Gelen ekrandaki “Kur” düğmesine basarak kurulmasını sağlayalım.
4. Kurulum tamamlandıktan sonra, “Uygulamalar” altında “Programlama” bölümü oluşacaktır. **‘Uygulamalar → Programlama → IDLE’** seçeneğine tıklayalım.
“Python Shell” penceresi açılacaktır. Kurulum tamamlanmıştır.

Python’ a Giriş

Python programları bir yorumlayıcı (interpreter) tarafından çalıştırıldıkları için Python, “yorumlanan programlama dilleri” kapsamına girer. Python yorumlayıcısını kullanmanın iki farklı yolu vardır: **etkileşimli olarak** (interactive mode) ve **betik kullanarak** (script mode).
[betik: yazılı olan şey]

Etkileşimli kullanımda, biz kodumuzu yazdıktan sonra yorumlayıcı bize sonucu döndürür:

```
>>> (6 * 2) - (21 / 3) + 4
9
```

Komut ekranındaki >>> işareti bize yorumlayıcının hazır durumda olduğunu bildirmektedir.

Betik kullanarak program yazdığımızda ise yazılan program bir metin dosyasında saklanır. Bu dosyayı çalıştırdığımız zaman içerisinde bulunan tüm kod yorumlanır ve sonucu döndürülür. Kabul edilen bir gösterim biçimi olarak, Python programlarının yer aldığı betik dosyalarına **“py”** uzantısı verilir.

Bir betik dosyası oluşturarak içerisine bir Python programı yazmak için şu aşamaları takip edebiliriz:

1. **Python Shell** üzerinde '**File → New Window**' diyerek boş bir sayfa açarız.
2. Bu sayfaya, uygun sözdizimiyle kodlanmış bir Python kodu yazdıktan sonra, sayfa üzerinden '**File → Save**' diyerek dosyamıza bir isim verir ve kaydederiz (Uzantısının “py” olmasına dikkat etmeliyiz.).
3. Sayfamız açıkken üst taraftaki menüden '**Run → Run Module**' seçerek ya da **F5** tuşuna basarak dosyadaki kodun yorumlanması ve sonucun dönmesini sağlarız.
4. Bu pencere kapalıyken de Python Shell ekranından bu pencereyi tekrar açarak çalıştırabiliriz. Bunun içinse, '**File → Open**' diyerek kaydetmiş olduğumuz dosyayı gösterir ve bu dosyalara ait kodların bulunduğu ekranı yeniden açarız. Daha sonra, 3. maddeyi tekrar ederiz.

Python Hakkında Yardım Alma

Python ile programlama yaparken, bilemediğimiz ya da emin olamadığımız konularda bize yardımcı olabilecek olan kaynaklardan yararlanmamız; Python' un programlama dünyasına daha başarılı bir giriş yapmamız, dili daha doğru ve çabuk bir biçimde öğrenmemiz ve daha başarılı programlar geliştirmemiz konularında bize büyük katkı sağlayacaktır.

Ayrıca, Python Shell ekranında “**help()**” komutunu kullanarak da yardım almamız mümkündür. Bunun için, parantez içerisine, tırnak içinde yardım almak istediğimiz konuyu yazabiliriz. Örnek olarak, matematikle ilgili fonksiyonlar içeren “math” sınıfı ile ilgili yardım almak için komut satırına aşağıdaki ifade yazılabilir:

```
>>> help('math')
```

Print Komutu

Python Shell ekranına yazı yazdırmak için kullanılan komuttur. Ekrana **metin** yazdırmak için, **print** komutundan sonra 1 boşluk bırakıp, tek tırnaklar ya da çift tırnaklar arasında istenen metni yazmak yeterli olacaktır. Python programlama dilinde ekrana “Programlamayı seviyorum.” yazdırabilmek için, aşağıdaki kod kullanılabilir (Python Shell’ de):

```
>>> print 'Programlamayı seviyorum.'
Programlamayı seviyorum.
>>> print Programlamayı seviyorum.
***Hata Mesajı***
```

print komutu ile ekrana **sayı** yazdırmak da mümkündür. Sayıları, tırnak işareti kullanmadan yazdırabiliriz:

```
>>> print 29101923
29101923
```

Daha fazla bilgi için <http://www.istihza.com/py2/pdf/Python2x.pdf> adresindeki dokümanın “1.3 print Komutu” bölümünde yer alan açıklamaları ve örnekleri inceleyiniz.

Bir metni yazdırırken metnin bir kısmının bir alt satıra geçmesini isteyebiliriz. Bunun için, **print** komutuna verdiğimiz metnin içerisinde “\n” karakterlerini aşağıdaki biçimde kullanmamız gerekir (“\n” karakterlerinin yazdırma işlemi sırasında görünmemesine dikkat ediniz):

```
>>> print 'birinci_satir\nikinci_satir'
birinci_satir
ikinci_satir
```

Metin editörlerinde metni hizalamak için “**tab**” karakteri kullanılır. Bu karakteri **print** komutu ile kullanmak da mümkündür. Metin içerisine “\t” karakterleri ile yerleştirilen **tab** karakterinin kullanım örneği aşağıdaki gibidir:

```
>>> print 'ocak\tsubat\tmart\nnisan\tmayis\thaziran'
ocak      subat      mart
nisan     mayis     haziran
```

Uyarı

Python Shell ekranında kod yazarken; gerek **print** komutunun kullanımında, gerekse diğer komutların kullanımında, “>>>” işaretlerinden sonra yalnızca 1 karakter boşluk bırakılmalıdır:

```
>>> print 'Programlamayı seviyorum.'  
Programlamayı seviyorum.  
>>>     print 'Programlamayı seviyorum.'  
***Hata Mesajı***
```

“for” komutu ile (İlerleyen bölümlerde ele alınacaktır.) belirli bir işlemi arka arkaya birden çok kere yaptırmak mümkündür. Bu işlemler sırasında **print** komutu kullanılıyorsa ve bu komut ile yazdırılanların aynı satırda yer alması isteniyorsa “,” karakterinden yararlanılabilir. Aynı satıra arka arkaya 5 kere “Merhaba” yazdırılmasını gösteren örnek kullanım aşağıdadır:

```
>>> for i in range(0,5):  
     print 'Merhaba',  
  
Merhaba Merhaba Merhaba Merhaba Merhaba
```


Pratik Uygulamalar

Python Shell ekranında kod yazarken, daha önceden yazmış olduğumuz bir kodu tekrar yazmamız gerekebilir. Kodu tekrar klavye ile yazmak yerine “**ALT + P**” tuş bileşimini kullanarak, komut satırına yazmış olduğumuz komutları en yeniden en eskiye doğru geri çağırabiliriz. Örneğin,

```
>>> (3 - 1) * (4 + 7) / (28 - (18 / 2)) + 40 / (22 - 14)
6
>>> 6 * 18 / (7 + (60 / (12 + 18)))
12
>>>
...
...
```

şeklinde uzunca işlemler yaptıktan sonra en üstteki ifadeyi tekrar kullanmamız gerekebilir. Karışık bir Python Shell ekranında arayıp, bulup, kopyalayıp komut satırına yapıştırmaktansa, tekrar bulana kadar “**ALT + P**” ye basmak daha pratik olacaktır. Bu tuş bileşimine yeteri kadar bastıktan sonra aynı ifade elde edilebilecektir:

```
>>> (3 - 1) * (4 + 7) / (28 - (18 / 2)) + 40 / (22 - 14)
```

Python’ da bir değişkene atama yaptıktan sonra, o değişkenin adını kullanarak içerdiği değere ulaşabilir, bu değeri ekrana basabiliriz. “_” karakteri de değeri en son ekrana basılan değişkenin değerini elde etmekte kullanılabilir:

```
>>> a = 5
>>> a
5
>>> b = 6
>>> b
6
>>> c = _ + 7
>>> c
13
```

“_” işareti, değeri en son ekrana basılan “b” değişkeninin değerini tuttuğu için, toplama işleminin sonucunda “c” nin değeri 13 olmuştur.

Python’ da betik dosyasında kod yazarken, kodun belirli yerlerine açıklama, hatırlatma gibi amaçlarla **yorum** yazmamız gerekebilir. Python’ da tek satıra yorum yazmak için **diyez** (“#”), birden çok satıra yorum yazmak içinse **üç tırnak** (“ """ ”) kullanılabilir (Python Shell

ekranında da bu komutları denemek mümkündür). Bir betik dosyasında yorum yapılmasına dair örnek aşağıdadır:

```
print 'Diyez öncesi...'

# Diyez ile yapılan yorum

print 'Diyez sonrası üç tırnak öncesi...'

"""
üç
tırnak
ile
yapılan
yorum
"""

print 'Üç tırnak sonrası...'
```

Bu dosya açıkken F5' e bastığımızda, kodun yorumlar yokmuşçasına çalıştığını görürüz:

```
>>>
Diyez öncesi...
Diyez sonrası üç tırnak öncesi...
Üç tırnak sonrası...
>>>
```

Python' da normalde birden fazla satıra yazdığımız kodları tek bir satıra yazmamız da mümkündür. Bunun için, “;” karakterini kullanmamız gerekir:

```
>>> print 'O'; print 'M'; print 'U'
O
M
U
```

print komutu kullanılarak şekil çizmek gibi farklı uygulamalar da yapılabilir (betik dosyasında). Ekranı, “*” karakterlerinden oluşan büyük bir “E” harfi yazdırmak için hazırlanmış örnek betik dosyası aşağıdadır:

```
print '*****'  
print '*****'  
print '**'  
print '**'  
print '*****'  
print '*****'  
print '**'  
print '**'  
print '*****'  
print '*****'
```

Bu kodu bir betik dosyasına kaydederek F5' e bastığımızda Python Shell ekranında göreceğimiz ekran çıktısı aşağıdaki gibi olacaktır:

```
>>>  
*****  
*****  
**  
**  
*****  
*****  
**  
**  
*****  
*****
```

Alıřtırmalar

Alıřtırma – 1

Görev

Bir betik dosyası oluşturarak içeşisine, Python Shell ekranına “**Merhaba Dünya**” yazısını yazdıracak kodu uygun biçimde yazınız, “**Lab01_merhaba.py**” isminde kaydediniz ve çalıştırarak sonucu gözlemleyiniz.

İpucu

“Print Komutu” bölümünü inceleyiniz.

Sonuç

Gerçekleřtiriminizi (Python kodunuzu) ve / veya karşılařtıđınız problemleri ařađıdaki kutunun içeşisine yazınız.

Alıştırma – 2

Görev

Betik dosyası kullanmadan (Python Shell komut satırına), ekrana “**Python’ u seviyorum.**” cümlesini yazdıracak bir kod yazınız. Yazacağınız komut ve size döndürülecek olan sonuç, aşağıdaki gibi olmalıdır:

```
>>> print yazacağınız kısım  
Python' u seviyorum.
```

Benzer biçimde, ekrana “**Python programlama dilinin adı "piton" yılanından gelmez.**” cümlesini yazdıracak bir kod yazınız. Yazacağınız komut ve size döndürülecek olan sonuç, aşağıdaki gibi olmalıdır:

```
>>> print yazacağınız kısım  
Python programlama dilinin adı "piton" yılanından gelmez.
```

UYARI: Her iki işlemde de sonuçların, **tek tırnaklar ve çift tırnaklar da dâhil**, kutular içerisinde belirtilenlerle **aynı** olması gerekmektedir.

İpucu

<http://www.istihza.com/py2/pdf/Python2x.pdf> adresinde yer alan Python kılavuzlarından, kullandığınız Python sürümüne uygun olanı indirerek içerisinde “**print**” komutunun açıklandığı bölümleri incelemeniz size yardımcı olacaktır.

Sonuç

Gözleminizin sonucunu ve / veya karşılaştığınız problemleri aşağıdaki kutunun içerisine yazınız. Bu konuda birden fazla çözüm olup – olmadığını belirterek birden fazla çözüm olması durumunda alternatif çözümleri de gösteriniz.

Alıştırma – 3

Görev

print komutu kullanarak, ekrana en az 8 satır yüksekliğinde ve okunaklı bir biçimde “**O.M.U.**” yazısını yazdırınız (Noktaların yükseklik ve genişlikleri de en az 2 birim olmalıdır.). Kodunuzu “**Lab01_print.py**” isimli betik dosyasına kaydediniz.

İpucu

“Print Komutu” bölümünü inceleyiniz.

Sonuç

Gerçekleştirmenizi ve / veya karşılaştığınız problemleri aşağıdaki kutunun içerisine yazınız.