Giriş

Prof.Dr. Bahadır AKTUĞ
JFM212 Python ile Mühendislik Uygulamaları

**Kaynakça bölümünde verilen kaynaklardan derlenmiştir.*

Python

- Guido Van Rossum tarafından geliştirilen Python Programlama/Betik Dilinin kökleri, Van Rossum'un önceleri çalıştığı CWI (Centrum Wiskunde & Informatica) tarafından geliştirilen ABC Programlama diline dayanmaktadır
- İlk versiyonu (versiyon 0.9.0) 1991 yılında yayınlanmıştır. İlk versiyondan itibaren tamamen nesneye yönelik bir dil olarak düşünülmüş, listeler, sözlükler gibi önemli veri yapıları ile hata ayıklama, modül yapısı sağlamaktadır.

Python

- Sonraki versiyon (1.0) 1994 yılında yayınlanmış, ve işlevsel programlama araçları lambda, map, filter, reduce gibi yenilikler getirmiştir.
- Örnek bir lambda fonksiyonu:

```
>>> f = lambda x, y : x + y
>>> f(1,1)
2
```

- Örnek bir filter/lambda fonksiyonu:

```
>>>  fib = [0,1,1,2,3,5,8,13,21,34,55]
>>> result = filter(lambda x: x % 2, fib)
>>> print result
[1, 1, 3, 5, 13, 21, 55]
```

Python

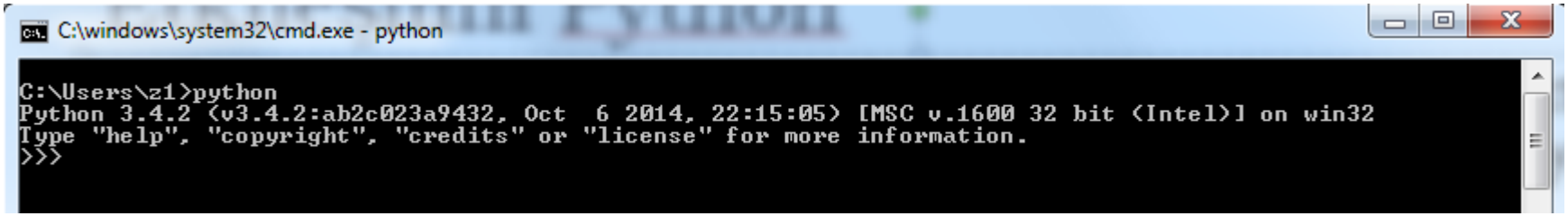
- Python 2.0 2000 yılında yayınlanmıştır. En önemli yenilikler olarak "liste tanıma (list comprehension)", "çöp toplayıcı (garbage collector)", unicode desteği sayılabilir.
- Liste tanıma örneği:

```
>>> S = [x**2 for x in range(10)]
>>> V = [2**i for i in range13)]
>>> M = [x for x in S if x % 2 == 0]
>>>
>>> print S; print V; print M
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
[1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096]
[0, 4, 16, 36, 64]
```

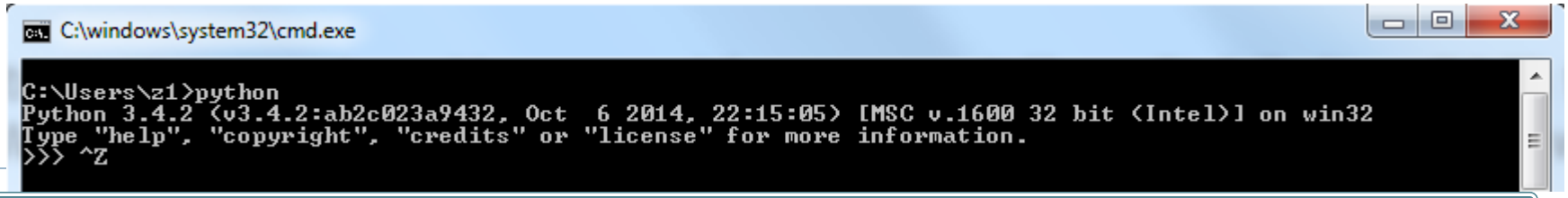
- 2008 yılına kadar Python'un 2.x şeklinde farklı versiyonları yayınlanmış olmakla birlikte, en radikal değişiklik Python 3.0 ile gelmiştir.

Etkileşimli Python

- Python programlama için farklı araçlar olmakla birlikte tüm Python sürümleri komut satırında etkileşimli işlem yapmak için güçlü bir kabuk (shell) sunmaktadır.
- Python doğru olarak yüklenmiş ise, Python kabuğuna doğrudan "python" komutu verilerek geçilebilir.

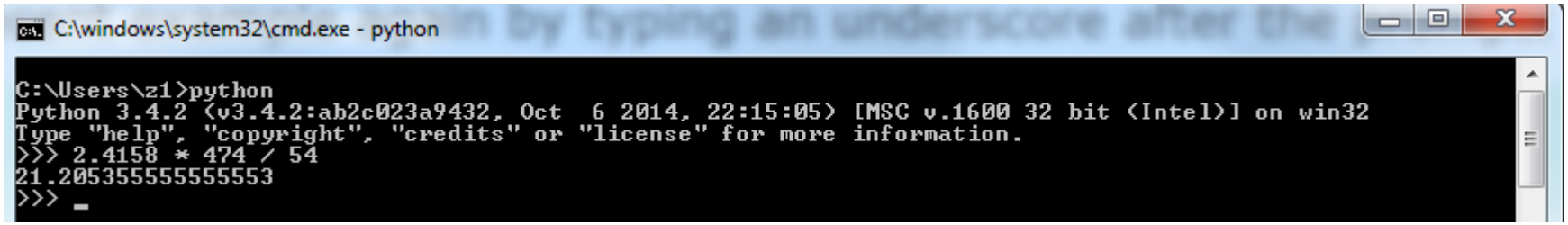


- Görüleceği üzere Python kabuğunda komut satırı >>> ile gösterilmektedir. Kabuktan çıkmak için Windows işletim sistemin CTRL+Z'ye basılır.

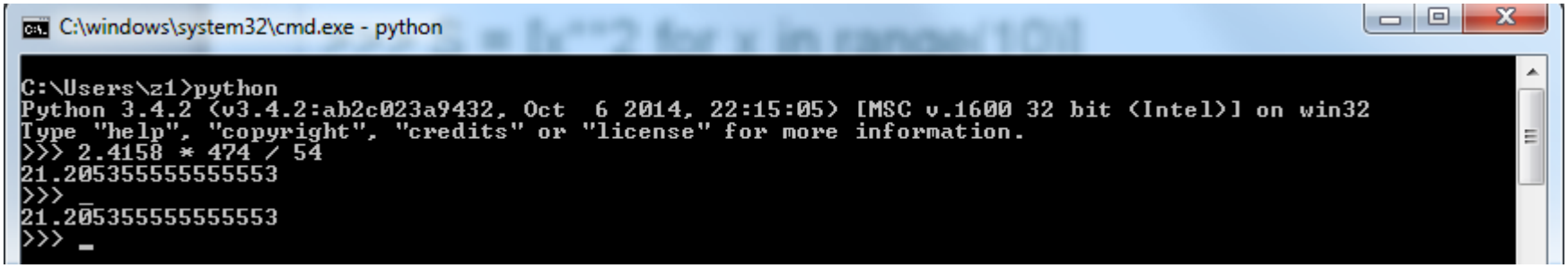


Etkileşimli Python

- Python komut satırında birçok matematiksel işlem yapılabilir.

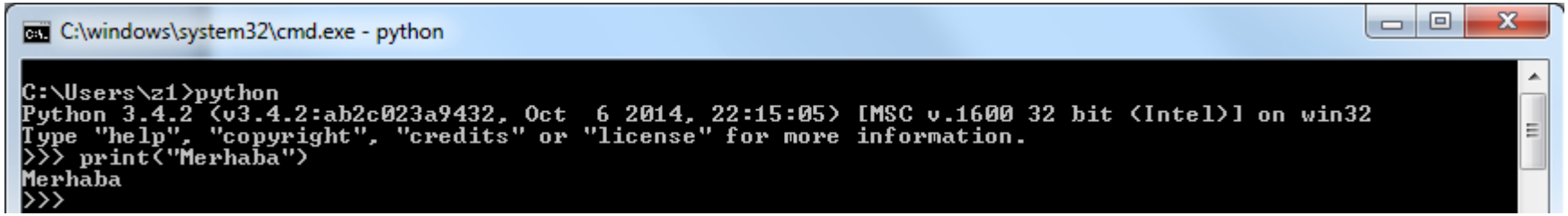


- En son elde edilen değer tekrar çağrılmak istenirse "_" operatörü kullanılabilir (Matlab'daki "ans" a benzer şekilde)

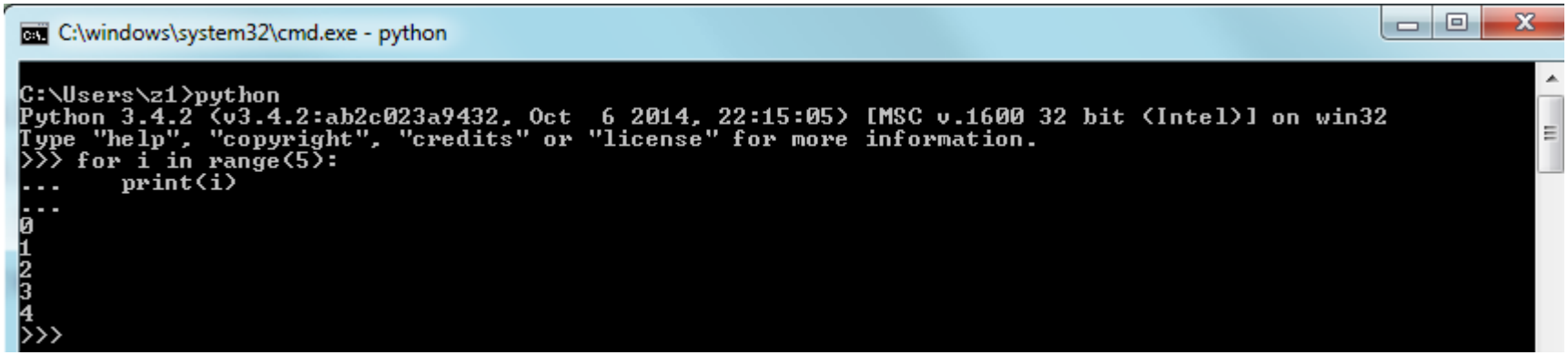


Etkileşimli Python

- Komut satırında komutlar doğrudan çalıştırılabilir ve sonuç ekranda görüntülenebilir.

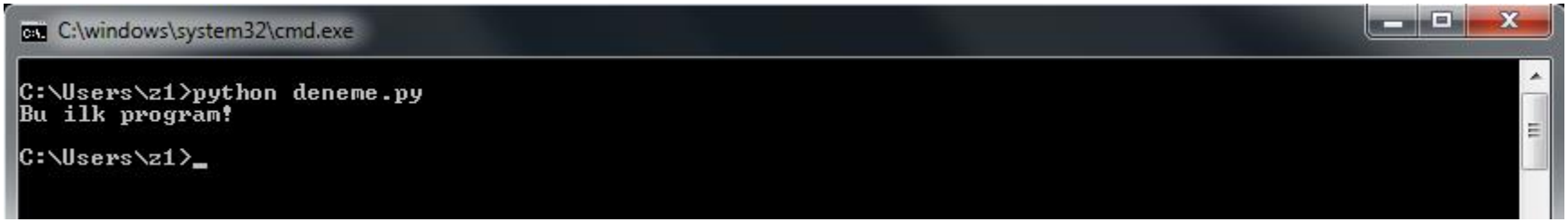


- Komut satırında iken birden fazla satır içeren komutlar "…" ile otomatik olarak ayrılır.



Python Programının Çalıştırılması

- Python komutları bir program dosyası haline getirilerek dışarıdan çalıştırılabilir. Özellikle uzun programlarda bu yöntem tercih edilir.
- Şimdi aşağıdaki Python komutunu "deneme.py" ismiyle bir dosyaya yazalım:
  - print("Bu ilk program!")
- Daha sonra DOS komut satırında (Python komut satırı değil!) aşağıdaki komutu verelim.



- Görüleceği üzere, bir Python programı doğrudan kullanılan işletim sisteminin kabuğu üzerinden çalıştırılabilmektedir.

Derleyici / Yorumlayıcı

- **Derleyiciler**

Derleyiciler, bir programlama dilinin kaynak kodunu, hedef platformun (kullanılan bilgisayar mimarisi ve işletim sistemi) diline çeviren bilgisayar programlarıdır. Yüksek seviye bir dilde hazırlanmış kaynak kodunu, düşük seviyedeki bir dile (assembly veya makine kodu) dönüştürürler.

- **Yorumlayıcılar**

Yorumlayıcılar, bir programlama dilinde hazırlanmış kaynak bilgisayar kodlarını çalıştıran programlardır. Yorumlayıcılar, kaynak kodunu doğrudan çalıştırabildiklerini gibi, kaynak kodunu önce ara bir formata dönüştürüp daha sonra da çalıştırabilirler.

Derleyici / Yorumlayıcı

- Python'un bir yorumlayıcı olduğu düşünülmekle birlikte, aslında Python hem bir yorumlayıcı hem de bir derleyicidir.
- Sanılanın aksine, Python kaynak kodunu çalıştırmak için doğrudan makine koduna dönüştürmek yerine önce ara bir dile dönüştürür.
- Oluşturulan bu ara dil «Python Sanal Makinesi (PVM)» olarak bilinen bir ortamda çalıştırılır. Bu durum Java'daki Java Sanal Makinesi ya da .NET ortamına benzemektedir.
- Python'un bu iki aşamalı yapısı nedeniyle, doğrudan Python ile doğrudan JVM'de ve .NET'de çalışan program oluşturmayı sağlayan geliştirme ortamları bulunmaktadır (Jython, IronPython)
- Genel olarak derlenmiş kaynak kodunun çalıştırılması daha hızlıdır. Python gerekli kısımları otomatik olarak derler.

Python Kodlarının Derlenmesi

- Ancak yine de derlemek istenirse;

```
>>> import py_compile
>>> py_compile.compile('test.py') veya
```

- veya

```
>>> python -m py_compile test.py
```

- Her iki yöntem ile de çalışılan dizinde «__pycache__», isimli bir dizin oluşturulur ve bu dizin içinde «test.cpython-34.pyc» yer alır. Kullanılan versiyona göre dosya ismi farklı olabilir.

Python Kodlarının Derlenmesi

- Ancak yine de derlemek istenirse;

```
>>> import py_compile
>>> py_compile.compile('test.py') veya
```

- veya

```
>>> python -m py_compile test.py
```

- Her iki yöntem ile de çalışılan dizinde «__pycache__», isimli bir dizin oluşturulur ve bu dizin içinde «test.cpython-34.pyc» yer alır. Kullanılan versiyona göre dosya ismi farklı olabilir.

- Derleme işlemlerini Python otomatik olarak yapar. Yeni bir kaynak kodu çalıştırılmak istendiğinde Python öncelikle çalıştırılacak kodun kaynak kodunun bulunup bulunmadığını kontrol eder.
- Derlenmiş kodlar «.pyc» olarak dizinde bulunurlar.
- Bulursa, «byte code» adı verilen ara kodu yükler ve bu işlem programların çalışmasını hızlandırır.
- Bulamazsa yeni derlenmiş kod oluşturur.
- En son olarak, yüklenen ara kod «Python Sanal Makinesi» içinde çalıştırılır.

Python ve Girintiler

- Bir program ve betikte belirli bir komut grubuna program bloğu adı verilir.
- Programlama dillerinde bir çok program bloğuna ihtiyaç duyulur (fonksiyonlar, kontrol ve döngü komutları (if/for gibi)).
- Program bloklarının başlangıç ve bitişlerinin belirlenmesi için farklı yöntemler kullanılır. Örneğin;
  - Pascal Programla dilinde begin/end
  - C/C++/C#/Java dillelrinde {}
  - Bash/Bourne Shell dilinde do/done
- Program blokları iç içe veya bağımsız olabilir.
- Python'da blokların başlangıç ve bitişlerinin belirlenmesi için «girinti» kullanılır.
- Girinti miktarının nasıl/ne kadar olacağı kullanıcıya bağlı olmakla birlikte, tutarlı olması zorunludur. Diğer bir deyimle aynı blok içerisinde girinti miktarı tüm komutlar için aynı olmak zorundadır.

Kaynakça

1 Wentworth, P., Elkner, J., Downey, A.B., Meyers, C. (2014). How to Think Like a Computer Scientist: Learning with Python (3nd edition).
2 Pilgrim, M. (2014). Dive into Python 3 by. Free online version: DiveIntoPython3.org ISBN: 978-1430224150.
3 Summerfield, M. (2014) Programming in Python 3 2nd ed (PIP3) : - Addison Wesley ISBN: 0-321-68056-1.
4 Summerfield, M. (2014) Programming in Python 3 2nd ed (PIP3) : - Addison Wesley ISBN: 0-321-68056-1.
5 Jones E, Oliphant E, Peterson P, et al. SciPy: Open Source Scientific Tools for Python, 2001-, http://www.scipy.org/.
6 Millman, K.J., Aivazis, M. (2011). Python for Scientists and Engineers, Computing in Science & Engineering, 13, 9-12.
7 John D. Hunter (2007). Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95.
8 Travis E. Oliphant (2007). Python for Scientific Computing, Computing in Science & Engineering, 9, 10-20.
9 Goodrich, M.T., Tamassia, R., Goldwasser, M.H. (2013). Data Structures and Algorithms in Python, Wiley.
10 http://www.diveintopython.net/
11 https://docs.python.org/3/tutorial/
12 http://www.python-course.eu
13 https://developers.google.com/edu/python/
14 http://learnpythonthehardway.org/book/