Εισαγωγή στην Επιστήμη των Υπολογιστών

Λεξικό

```
>>> eng2sp = dict()
>>> print eng2sp
{}
>>> eng2sp['one'] = 'uno‘
>>> print eng2sp
{'one': 'uno'}
```

Λεξικό (2)

```
>>> eng2sp = {'one': 'uno', 'two': 'dos', 'three': 'tres'}
>>> print eng2sp
{'one': 'uno', 'three': 'tres', 'two': 'dos'}
>>> print eng2sp['two']
'dos'
>>> 'one' in eng2sp
True
>>> 'uno' in eng2sp
False
```

Λεξικό (3)

```
>>> vals = eng2sp.values()
>>> 'uno' in vals
True
```

Λεξικό (4)

```
>>> d = {'a':0, 'b':1, 'c':2}
>>> t = d.items()
>>> print t
[('a', 0), ('c', 2), ('b', 1)]
>>> t = [('a', 0), ('c', 2), ('b', 1)]
>>> d = dict(t)
>>> print d
{'a': 0, 'c': 2, 'b': 1}
```

Λεξικό (5)

```
c = { 'GOOG' : 490.10, 'IBM' : 91.50, 'AAPL' : 123.15 }
for key in c:
        print key, c[key]
```

GOOG 490.1

AAPL 123.15

IBM 91.5

Χρήση συνάρτησης

```
def histogram(s):
        d = dict()
        for c in s:
                if c not in d:
                        d[c] = 1
                else:
                        d[c] += 1
        return d
>>> h = histogram('brontosaurus')
>>> print h
{'a': 1, 'b': 1, 'o': 2, 'n': 1, 's': 2, 'r': 2, 'u': 2, 't': 1}
```

Πλειάδες

```
>>> t = tuple('lupins')
>>> print t
('l', 'u', 'p', 'i', 'n', 's')
>>> t = ('a', 'b', 'c', 'd', 'e')
>>> print t[0]
'a‘
>>> print t[1:3]
('b', 'c')
```

Πλειάδες (2)

```
>>> t[0] = 'A'
TypeError: object doesn't support item assignment
>>> t = ('A',) + t[1:]
>>> print t
('A', 'b', 'c', 'd', 'e')
```

Χρήση συναρτήσεων

```
def sort_by_length(words):
        t = []
        for word in words:
                t.append((len(word), word))

        t.sort(reverse=True)

        res = []
        for length, word in t:
                res.append(word)
        return res
```

Χρήση συναρτήσεων (2)

```
>>> rec = ('Smith','John',(6,23,68)) # This is a tuple
>>> lastName,firstName,birthdate = rec # Unpacking the tuple
>>> print firstName
John
>>> birthYear = birthdate[2]
>>> print birthYear
68
>>> name = rec[1] + ' ' + rec[0]
>>> print name
John Smith
>>> print rec[0:2]
('Smith', 'John')
```

Πίνακες

```
>>> a = [[1, 2, 3], \
[4, 5, 6], \
[7, 8, 9]]
>>> print a[1]
[4, 5, 6]
>>> print a[1][2]
6
```

Πίνακες (2)

```
>>> from numpy import *
>>> a = zeros((3,3),dtype=int)
>>> print a
[[0 0 0]
[0 0 0]
[0 0 0]]
```

Πίνακες (3)

```
>>> from numpy import array,sqrt,sin
>>> a = array([1.0, 4.0, 9.0, 16.0])
>>> print sqrt(a)
[ 1. 2. 3. 4.]
>>> print sin(a)
[ 0.84147098 -0.7568025 0.41211849 -
0.28790332]
```

Πίνακες (4)

```
>>> from numpy import *
>>> A = array([[4,-2,1],[-2,4,-2],[1,-2,3]],dtype=float)
>>> b = array([1,4,3],dtype=float)
>>> print diagonal(A) # Principal diagonal
[ 4. 4. 3.]
>>> print diagonal(A,1) # First subdiagonal
[-2. -2.]
>>> print trace(A) # Sum of diagonal elements
11.0
```

Πίνακες (5)

```
>>> print identity(3) # Identity matrix
[[ 1. 0. 0.]
[ 0. 1. 0.]
[ 0. 0. 1.]]
```

Πράξεις Πινάκων

```
from numpy import *
x = array([7,3])
y = array([2,1])
A = array([[1,2],[3,2]])
B = array([[1,1],[2,2]])
# Dot product
print "dot(x,y) =\n",dot(x,y) # {x}.{y}
print "dot(A,x) =\n",dot(A,x) # [A]{x}
print "dot(A,B) =\n",dot(A,B) # [A][B]
```

Πράξεις Πινάκων (2)

dot(x,y) =

17

dot(A,x) =

[13 27]

dot(A,B) =

[[5 5]

[7 7]]

Χρήση συναρτήσεων χειρισμού πινάκων

```
>>> from numpy import array
>>> from numpy.linalg import inv,solve
>>> A = array([[ 4.0, -2.0, 1.0], \
[-2.0, 4.0, -2.0], \
[ 1.0, -2.0, 3.0]])
>>> b = array([1.0, 4.0, 2.0])
>>> print inv(A) # Matrix inverse
[[ 0.33333333 0.16666667 0. ]
[ 0.16666667 0.45833333 0.25 ]
[ 0. 0.25 0.5 ]]
>>> print solve(A,b) # Solve [A]{x} = {b}
[ 1. , 2.5, 2. ]
```

Ευκλείδεια απόσταση

```
def distance(x1, y1, x2, y2):
    dx = x2 - x1
    dy = y2 - y1
    dsquared = dx**2 + dy**2
    return math.sqrt(dsquared)
```

Με for

```
def dist(X, Y):
    Z = [(x - y)**2 for x, y in zip(X, Y)]
    return sum(Z)**0.5
```