

# Φύλλο Εργασίας: Χρήση Αρχείων στην Python

Τάξη: 12 ετών

Θέμα: Εισαγωγή στα αρχεία στην Python

Σκοπός: Να μάθουμε πώς να ανοίγουμε, να διαβάζουμε και να γράφουμε δεδομένα σε αρχεία χρησιμοποιώντας τη γλώσσα προγραμματισμού Python.

## Τι είναι τα Αρχεία;

Στην Python, ένα **αρχείο** είναι ένα δοχείο που περιέχει δεδομένα. Μπορεί να είναι κείμενο, αριθμοί, ή ακόμη και πιο σύνθετα δεδομένα. Χρησιμοποιούμε αρχεία για να αποθηκεύουμε πληροφορίες που μπορεί να θέλουμε να χρησιμοποιήσουμε αργότερα, όπως ένα ημερολόγιο, δεδομένα ενός παιχνιδιού ή ακόμα και μια λίστα με βαθμούς μαθητών.

## Πώς να ανοίξουμε ένα αρχείο στην Python

Για να εργαστούμε με αρχεία, χρησιμοποιούμε την εντολή **open()**. Αυτή η εντολή μας επιτρέπει να ανοίξουμε ένα αρχείο και να το διαβάσουμε ή να το γράψουμε.

Η εντολή **open()** χρειάζεται δύο πράγματα:

1. **Το όνομα του αρχείου** (το αρχείο που θέλουμε να ανοίξουμε).
2. **Τον τρόπο λειτουργίας** (τι θέλουμε να κάνουμε με το αρχείο: ανάγνωση, εγγραφή κ.λπ.).

## Τρόποι Λειτουργίας Αρχείων:

- **r**: Ανάγνωση (read). Ανοίγει το αρχείο μόνο για να το διαβάσουμε.
- **w**: Εγγραφή (write). Δημιουργεί ένα νέο αρχείο ή αντικαθιστά ένα υπάρχον με το νέο περιεχόμενο.
- **a**: Προσθήκη (append). Προσθέτει περιεχόμενο στο τέλος του αρχείου χωρίς να διαγράψει τα υπάρχοντα δεδομένα.

## Διαδικασία για Άνοιγμα και Κλείσιμο Αρχείου

Όταν ανοίγουμε ένα αρχείο, είναι σημαντικό να το **κλείσουμε** όταν τελειώνουμε, για να διασφαλίσουμε ότι τα δεδομένα γράφονται σωστά.

### Παράδειγμα:

```
# Άνοιγμα αρχείου για ανάγνωση
file = open("example.txt", "r")

# Κλείσιμο αρχείου μετά την ανάγνωση
file.close()
```

## Ανάγνωση από Αρχείο

Μπορούμε να διαβάσουμε δεδομένα από ένα αρχείο με διάφορους τρόπους:

- **read()**: Διαβάζει ολόκληρο το αρχείο.
- **readline()**: Διαβάζει μία γραμμή κάθε φορά.
- **readlines()**: Διαβάζει όλες τις γραμμές και τις επιστρέφει ως λίστα.

### Παράδειγμα:

```
# Άνοιγμα αρχείου για ανάγνωση
file = open("example.txt", "r")

# Ανάγνωση όλων των περιεχομένων του αρχείου
content = file.read()
print(content)

# Κλείσιμο του αρχείου
file.close()
```

### Παράδειγμα με **readline()**:

```
file = open("example.txt", "r")

# Ανάγνωση μίας γραμμής από το αρχείο
line = file.readline()
print(line)

file.close()
```

## Εγγραφή σε Αρχείο

Για να γράψουμε δεδομένα σε ένα αρχείο, πρέπει να το ανοίξουμε σε λειτουργία εγγραφής (**w** ή **a**).

**Παράδειγμα:**

```
# Άνοιγμα αρχείου για εγγραφή
file = open("example.txt", "w")

# Εγγραφή δεδομένων στο αρχείο
file.write("Hello, World!\n")
file.write("Αυτό είναι ένα παράδειγμα.")

# Κλείσιμο του αρχείου
file.close()
```

## Προσθήκη σε Αρχείο (Append)

Αν δεν θέλουμε να αντικαταστήσουμε το περιεχόμενο του αρχείου, αλλά να προσθέσουμε νέα δεδομένα, χρησιμοποιούμε τη λειτουργία **a**.

**Παράδειγμα:**

```
# Άνοιγμα αρχείου για προσθήκη
file = open("example.txt", "a")

# Προσθήκη νέων δεδομένων στο τέλος του αρχείου
file.write("\nΠροσθέτουμε αυτή τη γραμμή στο τέλος.")

# Κλείσιμο του αρχείου
file.close()
```

## Χρήση του **with** για Αρχεία

Η Python μας επιτρέπει να ανοίγουμε αρχεία με τη λέξη-κλειδί **with**. Με αυτόν τον τρόπο, το αρχείο κλείνει αυτόματα όταν τελειώσουμε.

### Παράδειγμα:

```
# Άνοιγμα αρχείου για ανάγνωση με χρήση του with
with open("example.txt", "r") as file:
    content = file.read()
    print(content)
# Το αρχείο κλείνει αυτόματα μόλις ολοκληρωθεί το μπλοκ with.
```

## Ασκήσεις

### Άσκηση 1: Ανάγνωση Αρχείου

Γράψε ένα πρόγραμμα που να ανοίγει ένα αρχείο που ονομάζεται **"data.txt"** και να εκτυπώνει τα περιεχόμενα του αρχείου στην οθόνη.

### Άσκηση 2: Εγγραφή Αρχείου

Γράψε ένα πρόγραμμα που να ανοίγει ένα αρχείο και να γράφει μέσα του το κείμενο **"Καλημέρα, κόσμε!"**.

### Άσκηση 3: Προσθήκη σε Αρχείο

Γράψε ένα πρόγραμμα που να ανοίγει το αρχείο **"log.txt"** και να προσθέτει την τρέχουσα ημερομηνία και ώρα στο τέλος του αρχείου.

## Συμπέρασμα

Τα αρχεία στην Python είναι ένας σημαντικός τρόπος για να αποθηκεύουμε και να διαχειριζόμαστε δεδομένα. Μάθαμε πώς να ανοίγουμε αρχεία, να διαβάζουμε και να γράφουμε σε αυτά, και πώς να χρησιμοποιούμε την εντολή **with** για να διασφαλίσουμε ότι τα αρχεία μας κλείνουν σωστά. Με την εξάσκηση, θα μπορέσουμε να δημιουργήσουμε πιο σύνθετα προγράμματα που θα αποθηκεύουν και θα διαχειρίζονται δεδομένα εύκολα και αποδοτικά.