

Python Programming

Hellenic Mediterranean University

Lecture 3

Dr. Alina Eqtami

Lecture Roadmap

Σήμερα:

- Quick recall από Lecture 2
- Strings ως νέο data type
- String comparison και case sensitivity
- String methods: `startswith`, `lower`
- Membership με `in`
- `print()` και string concatenation
- `input()` και interaction με τον χρήστη
- Mini programs και πολλά trace examples
- Rule-based chatbot
- Guessing game και validation λεκτικής εισόδου

Στόχος: Να περνάμε από state tracing σε interactive programming.

Why Lecture 3?

Lecture 2: αριθμοί, λογική, έλεγχος ροής.

Lecture 3: κείμενο, είσοδος χρήστη, κανόνες και μικρά interactive programs.

Ιδέα κεφαλαίου:

- Ο υπολογιστής δεν είναι “έξυπνος”.
- Ακολουθεί κανόνες.
- Αν οι κανόνες είναι καλά σχεδιασμένοι, το πρόγραμμα φαίνεται έξυπνο.

Quick Recall from Lecture 2

Θυμόμαστε μόνο όσα θα χρειαστούμε σήμερα:

- Comparison operators: `==` `!=` `<` `<=` `>` `>=`
- Boolean values: `True`, `False`
- Logical operators: `and`, `or`, `not`
- Selection: `if`, `elif`, `else`
- Repetition: `for`, `while`
- Accumulation / counter patterns

Σήμερα τα ίδια εργαλεία θα δουλέψουν πάνω σε **strings**.

Recall Example: Boolean Logic

```
x = 7
print(x > 5 and x < 10)
print(x < 5 or x == 7)
print(not(x == 7))
```

Ερώτηση: Μπορούμε να κάνουμε παρόμοιους ελέγχους και σε strings;

Recall Example: if as Decision

```
score = 82

if score >= 90:
    print("A")
elif score >= 80:
    print("B")
else:
    print("C or below")
```

Σήμερα το ίδιο μοτίβο θα χρησιμοποιηθεί για λέξεις και προτάσεις.

Transition: From Numbers to Text

Μέχρι τώρα δουλεύαμε κυρίως με:

- `int`
- `float`
- `bool`

Σήμερα προσθέτουμε:

`string`

String = ακολουθία χαρακτήρων.

The String Data Type

```
print("Hello")  
print('world')  
print("Python Programming")  
print("123")
```

Σημείο-κλειδί:

- "123" είναι string
- 123 είναι int

Naming Strings

```
name = "Alice"  
city = "Heraklion"  
message = "Welcome to Python"  
  
print(name)  
print(city)  
print(message)
```

Η μεταβλητή δείχνει σε string value, όπως πριν έδειχνε σε αριθμητική τιμή.

Single Quotes vs Double Quotes

```
a = "Hello"  
b = 'Hello'  
print(a)  
print(b)
```

Και τα δύο είναι σωστά.

Χρειαζόμαστε και τα δύο κυρίως για περιπτώσεις όπως:

```
print("Let's go")  
print('She said, "Hi"')
```

String Example Set

```
s1 = "robot"  
s2 = "A"  
s3 = "This is a sentence."  
s4 = "42"
```

Ερώτηση προς φοιτητές:

- Ποιο από αυτά είναι “λέξη”;
- Ποιο είναι “χαρακτήρας”;
- Ποιο μοιάζει με αριθμό αλλά δεν είναι αριθμός;

String Equality

```
print("cat" == "cat")  
print("cat" == "dog")  
print("cat" != "dog")
```

Οι συγκρίσεις δεν είναι μόνο για αριθμούς.
Επιστρέφουν πάλι:

True ή False

Case Sensitivity

```
print("Monday" == "Monday")  
print("Monday" == "monday")  
print("WHO" == "who")
```

Σημαντικό:

- Τα κεφαλαία και τα πεζά δεν θεωρούνται ίδια.
- Άρα οι συγκρίσεις string είναι **case sensitive**.

Strings inside Boolean Expressions

```
day = "Monday"

print(day == "Monday")
print(day == "Friday")
print(day != "Sunday")
```

Το pattern είναι το ίδιο όπως με αριθμούς.

Compound Conditions with Strings

```
day = "Saturday"

print(day == "Saturday" or day == "Sunday")
print(day != "Monday" and day != "Tuesday")
```

Άρα μπορούμε να γράψουμε rules και για λεκτική πληροφορία.

Selection with Strings

```
color = "green"

if color == "red":
    print("Stop")
elif color == "green":
    print("Go")
else:
    print("Unknown")
```

Το `if` δεν ενδιαφέρεται αν το `condition` προέρχεται από αριθμούς ή strings.

Mini Exercise 1

```
weather = "rain"

if weather == "rain":
    print("Umbrella")
else:
    print("Sunglasses")
```

Ερώτηση: Τι θα τυπωθεί;

Δεν θέλουμε μόνο να συγκρίνουμε ολόκληρα strings.
Θέλουμε και να ρωτάμε:

- Ξεκινά με μια λέξη;
- Περιέχει μια λέξη;
- Πώς το κάνουμε case-insensitive;

Εδώ μπαίνουν τα string methods.

Method 1: startswith

```
question = "Who are you?"  
  
print(question.startswith("Who"))  
print(question.startswith("What"))
```

Ερμηνεία:

- True αν το string ξεκινά με αυτό το pattern
- αλλιώς False

startswith in an if Statement

```
question = "Where is the lab?"  
  
if question.startswith("Where"):  
    print("Location question")  
else:  
    print("Something else")
```

Method 2: lower

```
question = "WHO ARE YOU?"  
print(question.lower())
```

Χρήση:

- μετατρέπει γράμματα σε lowercase
- κάνει πιο εύκολη την case-insensitive σύγκριση

Case-Insensitive Check

```
question = "WHY is Python useful?"  
question = question.lower()  
  
print(question)  
print(question.startswith("why"))
```

Pattern:

normalize first, compare second

Membership with in

```
question = "What is your name?"  
  
print("name" in question)  
print("robot" in question)
```

Ο τελεστής `in` ελέγχει αν ένα `string` περιέχεται μέσα σε άλλο `string`.

Combining startswith and in

```
question = "How are you feeling today?"  
question = question.lower()  
  
print(question.startswith("how"))  
print("feeling" in question)
```

Αυτό οδηγεί σε πιο εκφραστικούς κανόνες.

Compound Rule Example

```
question = "How are you feeling today?"
question = question.lower()

if question.startswith("how") and "feeling" in question:
    print("Special response")
else:
    print("Default response")
```

Εδώ αρχίζει να φαίνεται η “νοημοσύνη” του rule-based program.

Mini Exercise 2

```
msg = "What is your name?"  
msg = msg.lower()  
  
print(msg.startswith("what"))  
print("name" in msg)
```

Ερώτηση: Ποια είναι τα δύο outputs;

The print() Function Revisited

Το ξέρουμε ήδη, αλλά σήμερα θα το δούμε αλλιώς.
Δεν τυπώνουμε μόνο αριθμούς.

Τυπώνουμε:

- μηνύματα
- prompts για τον χρήστη
- τελικά αποτελέσματα με συνδυασμό strings και αριθμών

Printing Strings and Variables

```
name = "Maria"  
print("Hello")  
print(name)  
print("Welcome to class")
```

Η έξοδος είναι μέρος της επικοινωνίας προγράμματος - χρήστη.

String Concatenation

```
name = "Maria"  
print("Hello, " + name)
```

Ο τελεστής + σε strings σημαίνει:

ένωση strings

όχι αριθμητική πρόσθεση.

Concatenation Needs Strings

```
age = 20  
print("Age: " + age)
```

Αυτό είναι λάθος.

Γιατί;

Επειδή προσπαθούμε να ενώσουμε `string` με `int`.

Using str()

```
age = 20  
print("Age: " + str(age))
```

str() μετατρέπει αριθμό σε string ώστε να μπορεί να συμμετέχει σε concatenation.

Example: Building a Message

```
score = 92  
print("Your score is " + str(score))
```

Μικρό αλλά πολύ βασικό pattern για interactive προγράμματα.

Mini Exercise 3

```
a = 5
b = 7
print("Sum = " + str(a + b))
```

Ερώτηση: Τι τυπώνεται;

Getting Input from the User

Μέχρι τώρα οι τιμές ήταν hard-coded.

Τώρα θέλουμε το πρόγραμμα να περιμένει δεδομένα από τον χρήστη.

Για αυτό χρησιμοποιούμε:

```
input()
```

Basic input()

```
name = input("What is your name? ")  
print("Hello, " + name)
```

Η `input()` επιστρέφει **string**.

Important: Input Returns a String

```
age = input("How old are you? ")  
print(age)  
print(type(age))
```

Ακόμα και αν ο χρήστης γράψει 25, η `input()` επιστρέφει string.

Converting Input to Number

```
age = int(input("How old are you? "))  
print(age + 1)
```

Pattern:

- input first
- convert second
- use numerically third

Mini Program: Greeting

```
name = input("Enter your name: ")
city = input("Enter your city: ")

print("Hello, " + name)
print("You live in " + city)
```

Απλό πρόγραμμα, αλλά ήδη interactive.

Mini Program: Average of 3 Scores

```
t1 = int(input("Enter first score: "))
t2 = int(input("Enter second score: "))
t3 = int(input("Enter third score: "))

avg = (t1 + t2 + t3) / 3
print("Average = " + str(avg))
```

Συνδυασμός `input()`, conversion και output formatting.

Mini Program: Pass / Fail

```
score = int(input("Enter score: "))  
  
if score >= 50:  
    print("Pass")  
else:  
    print("Fail")
```

Τώρα ο χρήστης επηρεάζει τη ροή του προγράμματος.

Mini Program: Weekend Detector

```
day = input("Enter day: ")
day = day.lower()

if day == "saturday" or day == "sunday":
    print("Weekend")
else:
    print("Weekday")
```

Παράδειγμα case-insensitive λεκτικού ελέγχου.

Mini Program: Simple Email Check

```
email = input("Enter email: ")  
  
if "@" in email and "." in email:  
    print("Looks valid")  
else:  
    print("Invalid")
```

Δεν είναι πλήρης validation, αλλά είναι καλό πρώτο rules-based example.

From One Decision to Repeated Interaction

Σε πολλά interactive programs μία ερώτηση δεν αρκεί.
Θέλουμε:

- να ρωτάμε ξανά
- να περιμένουμε σωστή είσοδο
- να σταματάμε όταν έρθει μια συγκεκριμένη απάντηση

Εδώ μπαίνει ξανά το `while`.

while with Text Input

```
answer = input("Type yes to continue: ")
answer = answer.lower()

while answer != "yes":
    answer = input("Please type yes: ")
    answer = answer.lower()

print("Thank you")
```

Η condition είναι λεκτική, όχι αριθμητική.

Validation Pattern

```
command = input("Enter start or stop: ")
command = command.lower()

while command != "start" and command != "stop":
    command = input("Please enter start or stop: ")
    command = command.lower()
```

Pattern πολύ χρήσιμο για ανθεκτικά προγράμματα.

Accumulator with User Input

```
count = 0
word = input("Enter a word, stop to finish: ")
word = word.lower()

while word != "stop":
    count = count + 1
    word = input("Enter a word, stop to finish: ")
    word = word.lower()

print("You entered " + str(count) + " words")
```

Το accumulator pattern δεν περιορίζεται σε αριθμητικές λίστες.