

CS1011

Computer

Programming

in Python

Lecture 2: Variables, Expressions, and Control Flow



Objectives



Variables, assignment, and naming conventions



Arithmetic operations and precedence



Comparison operators (==, !=, <, >, etc.)



Conditional statements (`if`, `elif`, `else`)

Variables

A variable is a named place in the memory where a programmer can store data and later retrieve the data using the variable “name”

Programmers get to choose the names of the variables

You can change the contents of a variable in a later statement

x 12.2

y 14

x 12.2

y 14

Variables

A variable is a named place in the memory where a programmer can store data and later retrieve the data using the variable “name”

Programmers get to choose the names of the variables

You can change the contents of a variable in a later statement

x 12.2

y 14

x 100

x ~~12.2~~ 100

y 14

Python Variable Name Rules

Must start with a letter or underscore _

Must consist of letters, numbers, and underscores

Case Sensitive

Mnemonic Variable Names

Since we programmers are given a choice in how we choose our variable names, there is a bit of “best practice”

We name variables to help us remember what we intend to store in them (“mnemonic” = “memory aid”)

What are these bits of code doing?

x1q3z9ocd = 35.0

x1q3z9afd = 12.50

x1q3p9afd = x1q3z9ocd * x1q3z9afd

print(x1q3p9afd)

hours = 35.0

rate = 12.50

pay = hours * rate

print(pay)

Arithmetic operations and precedence

Operator	Meaning	Example	Y
+	Addition	x= 10 y = x + 3	13
-	Subtraction	x= 6 y = x - 4	2
*	Multiplication	x= 2 y = x * 5	10
/	Division	x= 10 y = x / 7	1.4
%	Modulus (remainder)	x= 10 y = x % 7	3
**	Exponent	x= 3 y = x ** 3	27

Comparison Operators

Operator	Meaning	Example	Result
<code>></code>		<code>5 > 1</code>	<code>true</code>
<code><</code>		<code>5 < 2</code>	<code>false</code>
<code>==</code>	Equal	<code>5 == 1</code>	<code>true</code>
<code>!=</code>	Not Equal	<code>4 != 10</code>	<code>true</code>
<code>>=</code>		<code>5 >= 6</code>	<code>false</code>
<code><=</code>		<code>5 <= 6</code>	<code>true</code>

LOGIC OPERATOR S

- a and b are variable names (with Boolean values)
- not a** → True if a is False
False if a is True
- a and b** → True if both are True
- a or b** → True if either or both are True

A	B	A and B	A or B
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

Logical Operators

Operator	Meaning	Example	Result
and	True if both the operands are true	(5 == 1) and (43 >4)	false
or	True if either of the operands is true	(5 != 1) or (9 >= 10)	true
not	True if operand is false	not(5 >= 6)	true

Example

The screenshot shows a Jupyter Notebook interface with two cells. The left cell contains Python code for arithmetic operations, printing results, logical operations, and comparison. The right cell shows the output of these operations.

```
main.py    ⌂   ⏴ saved
1 x = 15
2 y = 4
3 z = x-y
4
5 # arithmetics
6 print(z)
7 print("x % y = ",x%y)
8 print("x ** y = ",x**y)
9
10 # logical
11 print ((5 == 1) and (43 >4))
12
13 # comparision
14 print (4 != 10)

11
x % y =  3
x ** y =  50625
False
True
```

Order of Operations

Order (High -> Low)	Example	Result
()	$16 - 2 * 5 / 2 + 1$	12
**	$10 * 5 + 2 / (8 + 2)$	50.2
% / *	$2 / 10 \% 45$	2.5
+ -	$10 - 4 * 2$	2
== =! =< => < >	$2 * (4 - 10)$	12
not	$(12 + 0) - 4 > 2 * 3 / 4$	true
and		
or		

Decisions

- A decision lets you run either one section of code or another, based on the results of a specific test
- **In python, we have the following decisions statements :**
 - if statement
 - else statement
 - elif statement

Decisions: if statement

- It executes some code only if a specified condition is true
- **Syntax:**

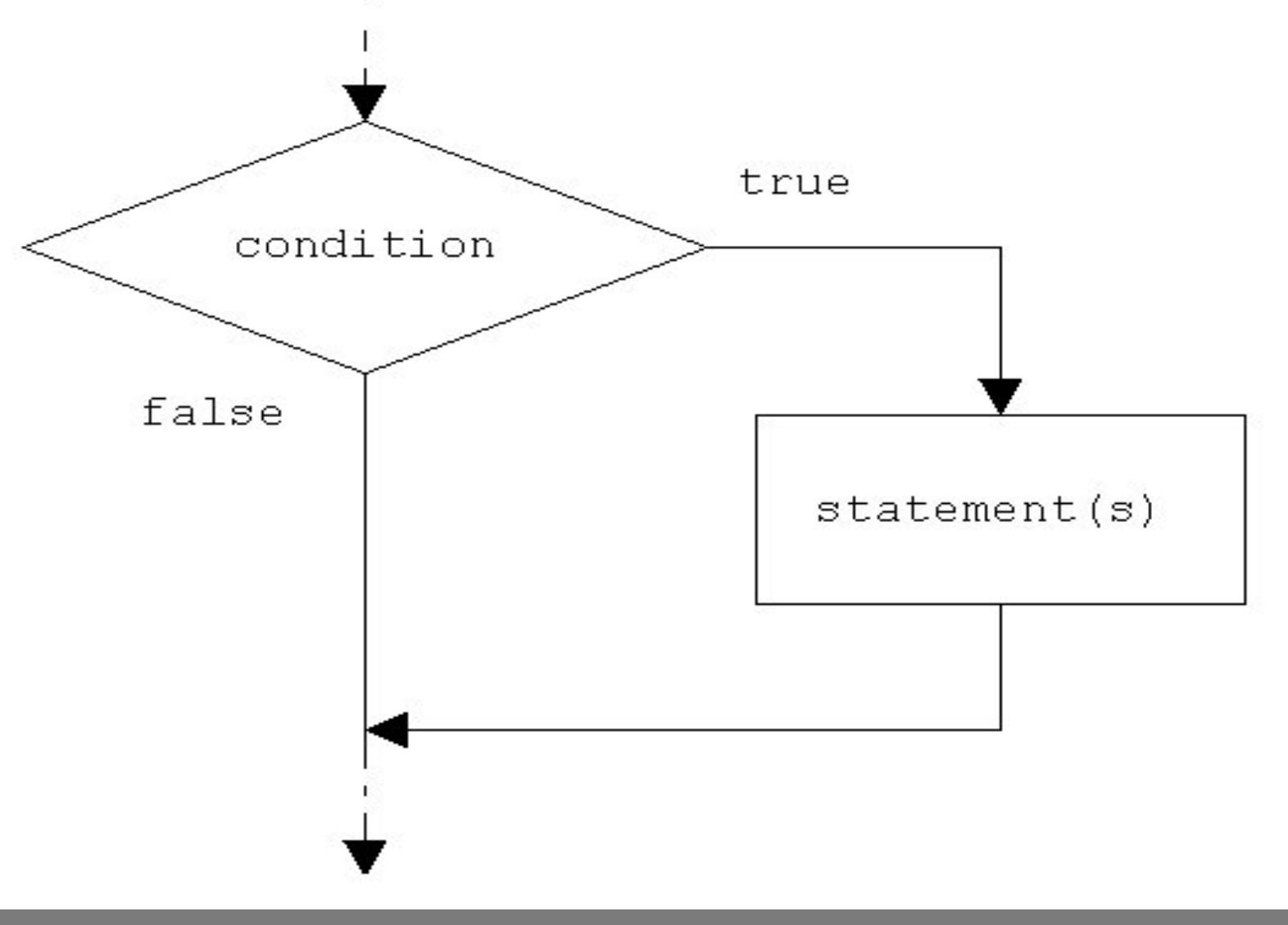
if condition:

statement(s) to be executed if condition is true

- **Note**

The statements inside the if will be indented

Decisions: if statement



Decisions: if statement

- Enter the day (sun, mon, ...) then check: if the day is thu, print (have a nice weekend)

```
1 day = input("Enter the day: ")  
2  
3 ▼ if day == "thu":  
4 |   print ("have a nice weekend")
```

```
Enter the day: thu  
have a nice weekend  
▶
```

```
1 day = input("Enter the day: ")  
2  
3 ▼ if day == "thu":  
4 |   print ("have a nice weekend")
```

```
Enter the day: sun  
▶
```

```
1 day = input("Enter the day: ")  
2  
3 ▼ if day == "thu":  
4 |   print ("have a nice weekend")
```

```
Enter the day: fdtht  
▶
```

Decisions: else statement

- It execute some code if a condition is true and another code if false
- **Syntax:**

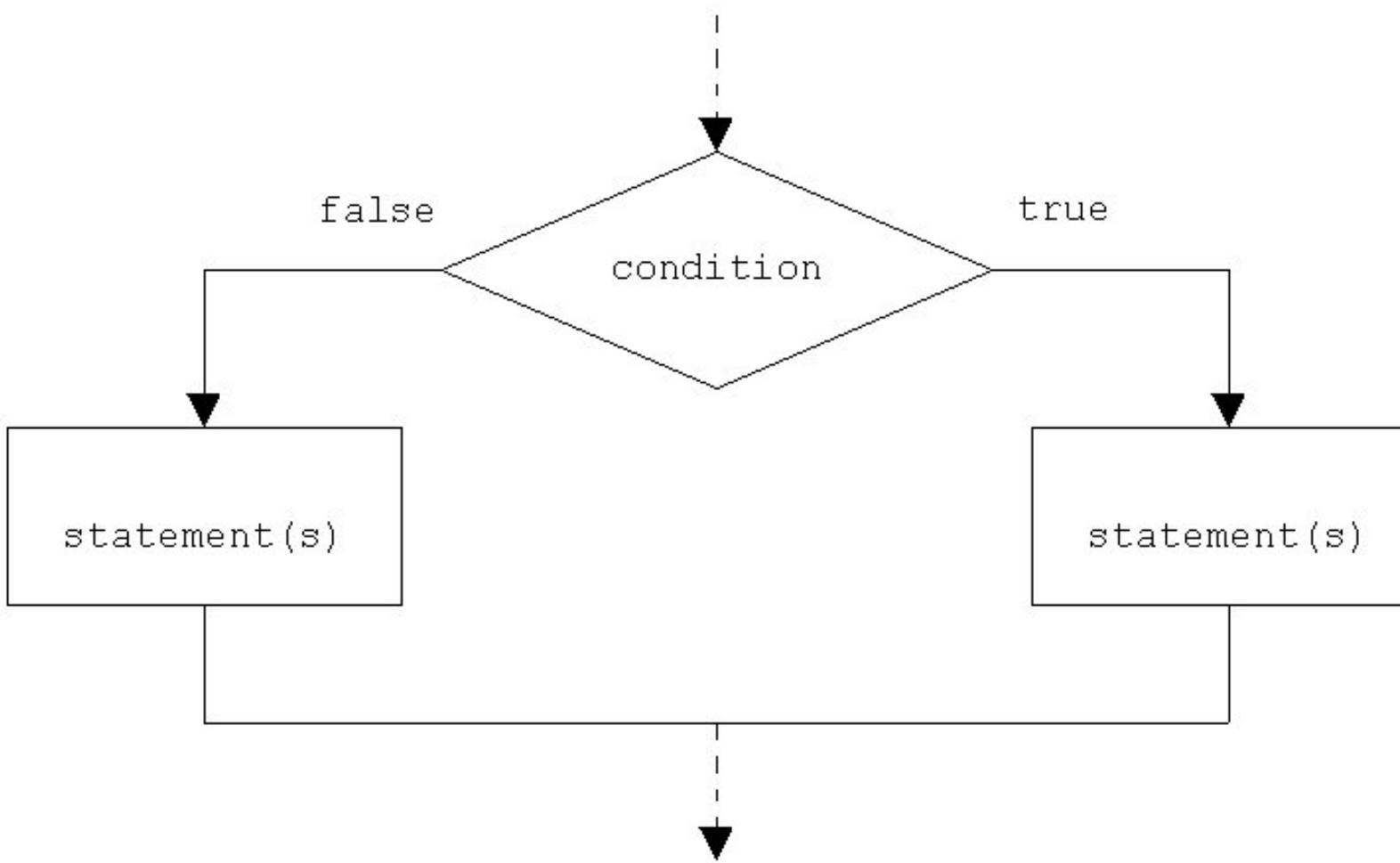
if condition:

statement(s) to be executed if condition is true

else:

statement(s) to be executed if condition is false

Decisions: else statement



Decisions: else statement

- Enter the day (sun, mon, ...) then check: if the day is thu, print (have a nice weekend), otherwise print (have a nice day)

The image shows a terminal window with three separate runs of a Python script. Each run demonstrates a different input and its corresponding output.

```
1 day = input("Enter the day: ")
2
3▼ if day == "thu":
4 |   print ("have a nice weekend")
5▼ else:
6 |   print ("have a nice day")
```

Output 1 (for 'thu'): Enter the day: thu
have a nice weekend

```
1 day = input("Enter the day: ")
2
3▼ if day == "thu":
4 |   print ("have a nice weekend")
5▼ else:
6 |   print ("have a nice day")
```

Output 2 (for 'mon'): Enter the day: mon
have a nice day

```
1 day = input("Enter the day: ")
2
3▼ if day == "thu":
4 |   print ("have a nice weekend")
5▼ else:
6 |   print ("have a nice day")
```

Output 3 (for 'jbjkbj'): Enter the day: jbjk bj
have a nice day

Decisions: else statement

- Enter the day (sun, mon, ...)
then check: if the day is
thu, fri, or sat, print (have a
nice weekend), otherwise
print (have a nice day)

```
1 day = input("Enter the day: ")  
2  
3 ▼ if (day == "thu") or (day == "fri") or (day == "sat"):  
4 |   print ("have a nice weekend")  
5 ▼ else:  
6 |   print ("have a nice day")
```

```
Enter the day: fri  
have a nice weekend  
▶
```

```
1 day = input("Enter the day: ")  
2  
3 ▼ if (day == "thu") or (day == "fri") or (day == "sat"):  
4 |   print ("have a nice weekend")  
5 ▼ else:  
6 |   print ("have a nice day")
```

```
Enter the day: tue  
have a nice day  
▶
```

```
1 day = input("Enter the day: ")  
2  
3 ▼ if (day == "thu") or (day == "fri") or (day == "sat"):  
4 |   print ("have a nice weekend")  
5 ▼ else:  
6 |   print ("have a nice day")
```

```
Enter the day: werret  
have a nice day  
▶
```

Decisions: elif statement

- It is used to perform different actions based on different conditions
- **Syntax:**

if condition1:

statement(s) to be executed if condition1 is true

elif condition2:

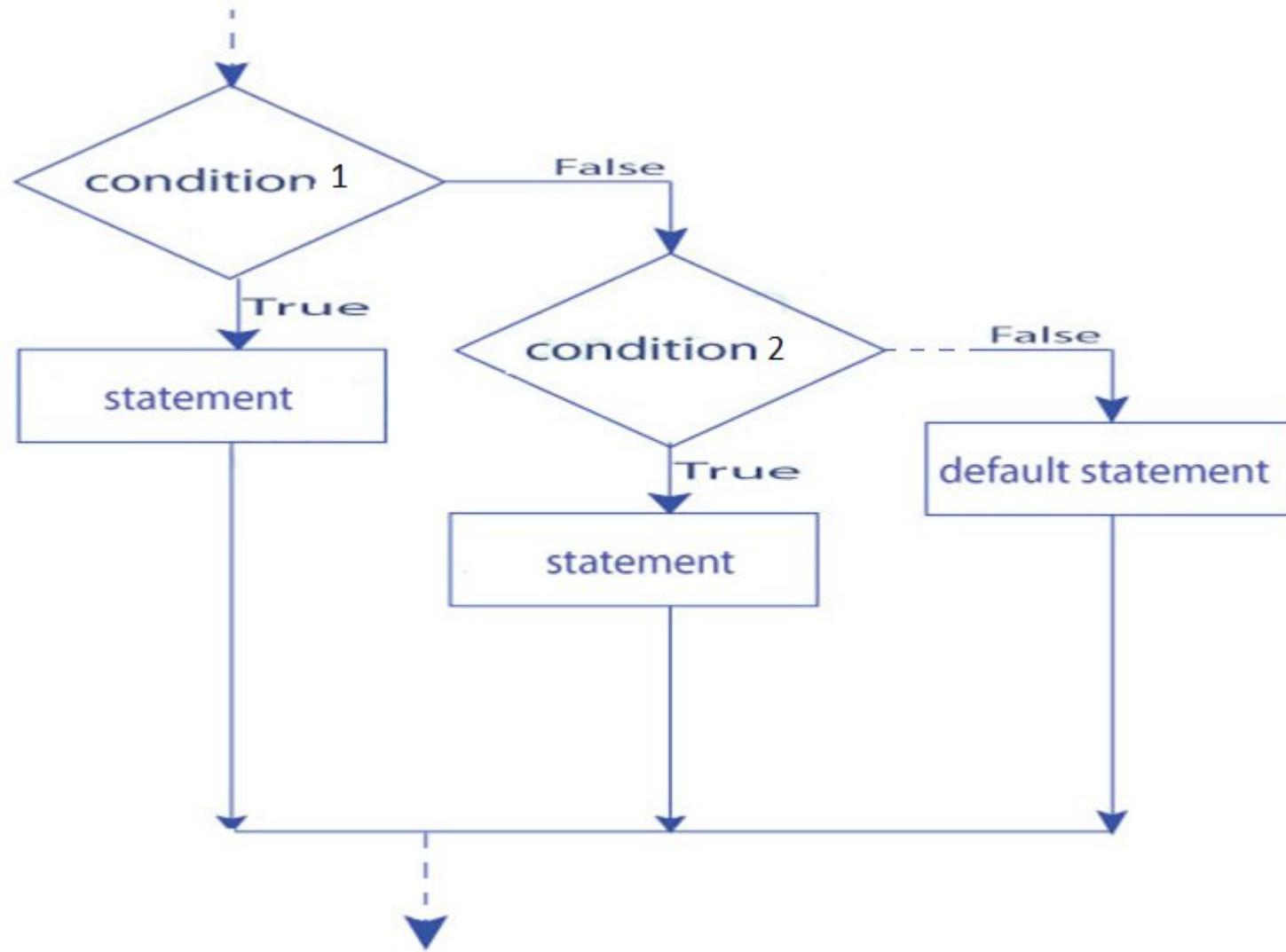
statement(s) to be executed if condition2 is true

:

else:

default statement(s) to be executed if all conditions are false

Decisions: elif statement



Decisions: elif statement

- Enter a student score, then print its letter grade (A+, A, B+,...)

```
1  score = float(input("enter the score: "))
2
3 ▼ if score >= 95:
4     letter = "A+"
5 ▼ elif score >= 90:
6     letter = "A"
7 ▼ elif score >= 85:
8     letter = "B+"
9 ▼ elif score >= 80:
10    letter = "B"
11 ▼ elif score >= 75:
12    letter = "C+"
13 ▼ elif score >= 70:
14    letter = "C"
15 ▼ elif score >= 65:
16    letter = "D+"
17 ▼ elif score >= 60:
18    letter = "D"
19 ▼ else:
20    letter = "F"
21
22  print ("the letter grade is: ", letter)
```

```
enter the score: 82
the letter grade is: B
```