Maktab Sharif

Python | Main course

Session 1

- Introduction
- Variables
- Input/Outputs
- Types
- Operators
- Conditions

by Mohammad Amin H.B. Tehrani

www.maktabsharif.ir

Chapter 0

Introduction

I'm

Mohammad amin Tehrani

Maktab Sharif

Projects

Dantia.ir
SADA (IPO)
BargNiki.com
bazmand.ir

Languages

Python
Java
Php
C/C++

Fields and Frameworks

Back-end Developer | Django, CodeIgniter
Android developer | Android studio
SQL Developer | MySql
UI/UX designer | Material Design

Education

Computer Engineering
Kharazmi University (95)

Evaluation

Maktab Sharif

Lectures

Thursday, Friday

A) 9:30 - 12:30

B) 14:30 - 17:30

Question & Answer

1. at Telegram Group
2. Q&A Classes

   Time: Will be announced later

Chapters

Variables

Input/Output

Data Types

Conditional Statements

Strings

Lists

Loops

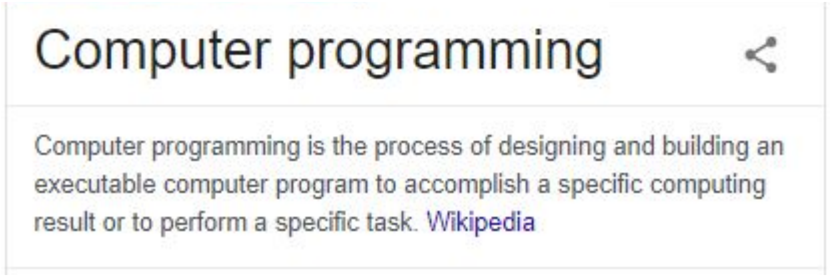
Functions

Programming

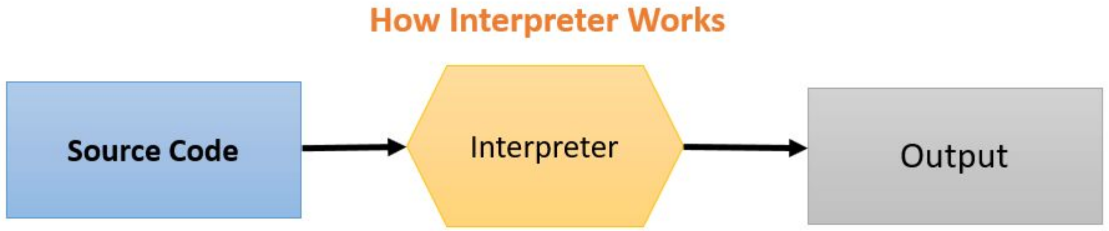
Maktab Sharif

What's Computer programming?

How does it works?
Why do we need it?
Language?
How to start?


Computer programming

Computer programming is the process of designing and building an executable computer program to accomplish a specific computing result or to perform a specific task. Wikipedia



- **High-level languages : Easy for Human**
  - Python
  - JavaScript
  - ...
- **Middle-level**
  - C
  - C++
  - Java
  - ...
- **Low-level languages : Easy for Computer (Machine)**
  - Assembly

Maktab Sharif

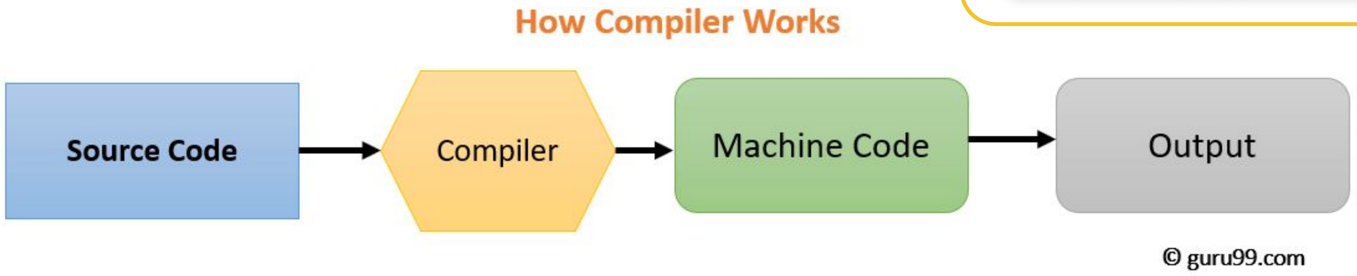
Compiler vs Interpreter

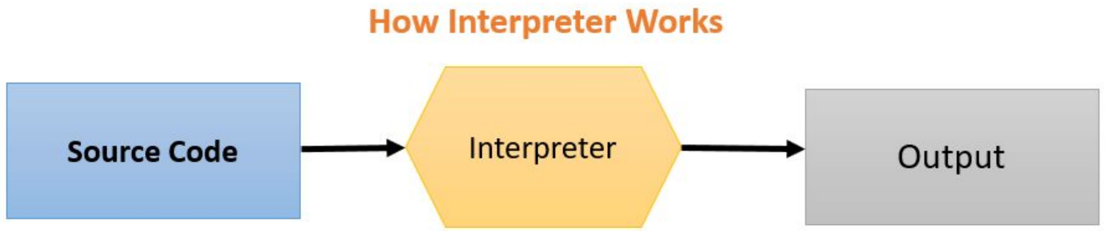
Maktab Sharif

Search It

How to compile a python code

Is Python interpreted, or compiled? **Interpreted**



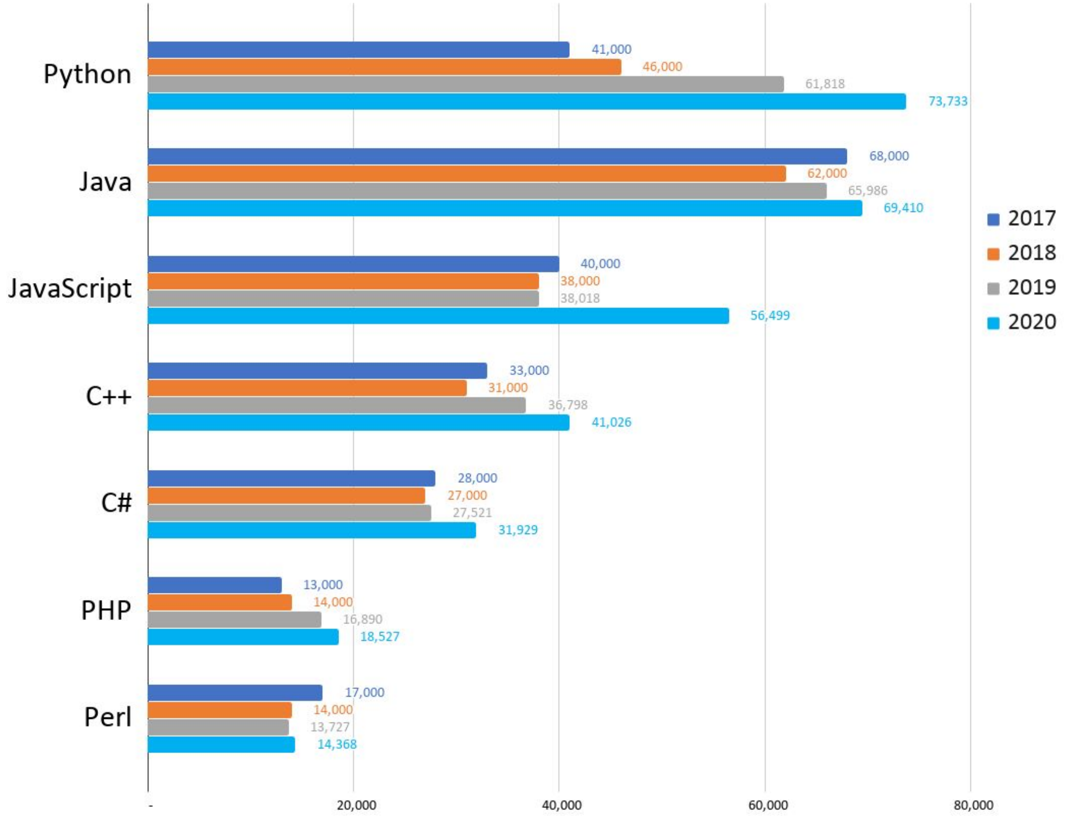


Basis of difference	**Compiler**	**Interpreter**
Programming Steps	● Create the program. ● Compile will parse or analyses all of the language statements for its correctness. If incorrect, throws an error ● If no error, the compiler will convert source code to machine code. ● It links different code files into a runnable program(know as exe) ● Run the Program	● Create the Program ● No linking of files or machine code generation ● Source statements executed line by line DURING Execution
Advantage	The program code is already translated into machine code. Thus, it code execution time is less.	Interpreters are easier to use, especially for beginners.
Disadvantage	You can't change the program without going back to the source code.	Interpreted programs can run on computers that have the corresponding interpreter.

What's Python language?

Why python?
How to start?
Applications:

- Web development
- Data science
- AI & Machine Learning
- Game development
- Desktop, Android Apps
- ...



Maktab Sharif

Programming language trends 2020

Chapter 1

Variables

Values

Maktab Sharif

A value is one of the fundamental things — like a word or a number — that a program manipulates. The values we have seen so far are 5 and "Hello, World!". We often refer to these values as objects and we will use the words value and object interchangeably.

5 is an integer, and "Hello World!" is a string, so-called because it contains a string of letters. You can identify strings because they are enclosed in quotation marks.

Values are stored in variables.

Variables

Putting values into the variables can be realized with **assignments**. The way you assign values to variables is nearly the same in all programming languages. In most cases the equal "=" sign is used. The value on the right side will be saved in the variable name on the left side.

```
message = "What's up, Doc?"
n = 17
pi = 3.14159
```

The assignment statement links a name, on the left hand side of the operator, with a value, on the right hand side. This is why you will get an error if you enter:

```
17 = n
```

Variable Names and Keywords

Maktab Sharif

A variable name and an identifier can consist of the uppercase letters "A" through "Z", the lowercase letters "a" through "z" , the underscore _ and, except for the first character, the digits 0 through 9

Variable names can never contain **spaces**.

The variable is **sensitive** to the capitalization of letters. For example, “Message" and "message" are two different words because the “M" is uppercase in the first example and lowercase in the second example.

Example

Maktab Sharif

Choose suitable names for examples below:

A. Student on Registration form:
   1. First name
   2. Last name
   3. User name
   4. Phone number
   5. Email
   6. Class
   7. Password

B. Books on Bookstore:
   1. Name
   2. Publisher
   3. Price (dollar)

Example

Which one is illegal?

```
1.  EmAIl = “your_email@gmail.com”
2.  __phone = “09379880665”
3.  First-name = “Mohammad Amin”
4.  warn! = “Stay easy!”
5.  MsgToAdmin = “Hi babe”
6.  16to2BitConvert = 010110
7.  _ = “Hello World!”
8.  float = 1.22
9.  str = ‘Akbar’
10. class = ‘A’
```

Some Example

Maktab Sharif

Which one is illegal?

```
1.   EmAIl = “your_email@gmail.com” ->  OK
2.   __phone = “09379880665”            ->  OK
3.   First-name = “Mohammad Amin”       ->  Invalid ( - )
4.   warn! = “Stay easy!”               ->  Invalid ( ! )
5.   MsgToAdmin = “Hi babe”             ->  Invalid ( - )
6.   16to2BitConvert = 010110           ->  Invalid ( Starts With Num)
7.   _ = “Hello World!”                 ->  OK
8.   float = 1.22                       ->  Not Recommended
9.   str = ‘Akbar’                      ->  Not Recommended
10.  class = ‘A’                        ->  Invalid ( reserved keyword)
```

Variable Names and Keywords

Maktab Sharif

It turns out that class is one of the Python keywords. Keywords define the language's syntax rules and structure, and they cannot be used as variable names. Python has thirty-something keywords (and every now and again improvements to Python introduce or eliminate one or two)

and	as	assert	break	class	continue
def	del	elif	else	except	exec
finally	for	from	global	if	import
in	is	lambda	nonlocal	not	or
pass	raise	return	try	while	with
yield	True	False	None		

Summary

Maktab Sharif

1. It's **Case Sensitive**
   Exp: `phone`, `Phone` are different names
2. **Cannot** use **Space** between variable names
   Exp: `first name`, `book name` are illegal names
3. **Cannot starts** with **Numbers**
   Exp: `2key`, `7seg`, `16to2Converter` are illegal names
4. **Only** can use **'_' (Under line)**s in names
   Exp: `first-name`, `price$`, `id@` are illegal names
5. **Cannot** use **python built-in keywords** as a name
   Exp: `if`, `class`, `pass` are illegal names

Chapter 2

Input/Output

Output (print)

Maktab Sharif

`print(...)` function -> use for stream output to console

```
print("Hello World!")

print(12.2 + 5)

print("Age:", 12)

print('W'+ (5 * 'o') + 'w!')

print('A: 10 \nB: 20 \nC: 30 \nD: 40')
```

Output (print)

Maktab Sharif

Print variables:

```
a = 'Hello World!'

print(a)
```

```
r = 5

pi = 3.1415

print('Circle Area:\t', pi*r*r)
```

Output (print)

Maktab Sharif

Escape Sequence	Meaning Notes
\\	Backslash (\)
\'	Single quote (')
\"	Double quote (")
\n	ASCII Linefeed (LF)
\t	ASCII Horizontal Tab (TAB)

Input

Maktab Sharif

In order to do this, we need a way to get input from the user. Luckily, in Python there is a built-in function to accomplish this task. As you might expect, it is called input.

```
n = input("Please enter your name: ")
print("Hello", n)

a = input()
print(a * a)   # ok?
```

Example

Maktab Sharif

Area of rectangle

Write a program that get width and height of the rectangle (x and y),
Then print area of that.

Input:

```
>> 12.5
>> 3
```

output:

```
>> 37.5
```

Example

Maktab Sharif

Area of rectangle

Write a program that gets width and height of the rectangle (x and y), Then print area of that.

code:

```
x = input("Enter x:")
y = input("Enter y:")

print(x*y)
```

output:

```
Traceback (most recent call last):
  File "test.py", line 12, in <module>
    print(x*y)
TypeError: can't multiply sequence by
non-int of type 'str'
```

Chapter 3

Types

Built-in Types

Maktab Sharif

Python built-in data Types

- **Int** : An integer type that represents an integer.
- **float** : A float numeric type that contains a decimal number.
- **str** : String or str, which is a sequence of characters (such as letters and digits). Strings are specified by a number of characters between two " or ' symbols.
- **bool** : The bool type takes both **True** value and **False** value.
- ...

```
i = 12
f = 12.523
s = 'It is a String'
b = True
```

Is 123 Equal to '123' ?

type() function

Maktab Sharif

Type function

type(x) function returns the type of variable ‘x’

Example

```
i = 12
f = 12.523
s = 'It is a String'
b = True
x = input()
```

```
print(type(i))
print(type(f))
print(type(s))
print(type(b))
print(type(x))
```

type() function

Maktab Sharif

Type function

type(x) function returns the type of variable ‘x’

Example

```
i = 12
f = 12.523
s = 'It is a String'
b = True
x = input()
```

```
print(type(i))  # int
print(type(f))  # float
print(type(s))  # str
print(type(b))  # bool
print(type(x))  # ???
```

Type Casting (Conversion)

Maktab Sharif

Type casting (Conversion)

Sometimes it is necessary to convert values from one type to another. Python provides a few simple functions that will allow us to do that. The functions int, float and str will (attempt to) convert their arguments into types **int**, **float**, **bool** and **str** respectively. We call these type conversion functions.

Example

```
print(int(12.345))
print(float(12))
print(str(12.345))
print(bool(12.345))
```

Type Casting (Conversion)

Maktab Sharif

int examples:

```
print(int(12.999))  # 12
print(int(12)) # 12 (Effectless)
print(int('123'))  # 123
print(int(True))  # 1
print(int('123.45')) #???
```

float examples:

```
print(float(12)) # 12.0
print(float(12.34)) # 12.34
print(float('-12.45')) # -12.45
print(float(True))  # 1.0
print(float('123-45')) #???
```

str examples:

```
print(str(12.345))  # '12.345'
print(str(12))  # '12'
print(str(-12)) # '-12'
print(str('123')) # '123'
print(str(True))  # 'True'
```

bool examples:

```
print(bool(12.345))  # True
print(bool(0))  # False
print(bool('')) # False
print(bool(-2)) # True
print(bool('Hi!'))  #True
```

None type (null)

Maktab Sharif

None type

The **None** keyword is used to define a null value, or no value at all.
None is not the same as 0, False, or an empty string. None is a data type of its own (NoneType) and **only None can be None.**

Example

```
print(None)
print(type(None))
print(bool(None))
print(False == None)
print(‘’ == None)
print(bool(‘’) == bool(None))
```

Example

Maktab Sharif

Area of rectangle

Write a program that gets width and height of the rectangle (x and y),
Then print area of that.

code:

```
x = float(input("Enter x:"))
y = float(input("Enter y:"))

print(x*y)
```

output:

```
> Enter x: 12.5
> Enter y: 3
> 37.5
```

Chapter 4

Operators

Operators in python

Maktab Sharif

Operators are special tokens that represent computations like addition, multiplication and division. The values the operator works on are called **operands**.

Examples:

```
print(2 + 3)
print(2 - 3)
print(2 * 3)
print(2 ** 3)
print(3 ** 2)

minutes = 645
hours = minutes / 60
print(hours)
```

Operators in python

Maktab Sharif

Arithmetic operators

+	Addition (Sum)
-	Subtraction (Minus)
*	Multiplication
/	Division
%	Modulus
**	Exponentiation
//	Floor division

```
x = 11
y = -2

print(x + y)
print(x - y)
print(x * y)
print(x / 3)
print(x % 3)
print(x % -y)
print(y ** 3)
print(x ** y)
print(x / y)
print(x // y)
```

An Example

Maktab Sharif

```
x = 11
y = -2

x = x - 6
x = x + y * 2

print('x =',x)        # ?
print('y =',y)        # ??

y = -y
x = x ** 8

print('x =',x)        # ???
print('y =',y)        # ????
```

An Example

Maktab Sharif

```
x = 11                # x = 11
y = -2                # x = -2

x = x - 6             # x = 11 - 6 = 5
x = x + y * 2         # x = 5 + (-2)*2 = 5 - 4 = 1

print('x =',x)        # x = 1
print('y =',y)        # y = -2

y = -y                # y = -(-2)  = 2
x = x ** 8            # x = 1 ** 8 = 1

print('x =',x)        # x = 1
print('y =',y)        # y = 2
```

Operators in python

Maktab Sharif

Assignment operators

=	x = 5	->	x = 5
+=	x += 3	->	x = x + 3
-=	x -= 3	->	x = x - 3
*=	x *= 3	->	x = x * 3
/=	x /= 3	->	x = x / 3
%=	x %= 3	->	x = x % 3
//=	x //= 3	->	x = x // 3
**=	x **= 3	->	x = x ** 3

```
x = 7

x += 9
print(x)
x -= 2
print(x)
x *= 2
print(x)
x /= 4
print(x)
x %= 4
print(x)
x //= 2
print(x)
x **= 4
print(x)
```

Operators in python

Maktab Sharif

Assignment operators

=	x = 5	->	x = 5
+=	x += 3	->	x = x + 3
-=	x -= 3	->	x = x - 3
*=	x *= 3	->	x = x * 3
/=	x /= 3	->	x = x / 3
%=	x %= 3	->	x = x % 3
//=	x //= 3	->	x = x // 3
**=	x **= 3	->	x = x ** 3

```
x = 7

x += 9
print(x)        # x = 16
x -= 2
print(x)        # x = 14
x *= 2
print(x)        # x = 28
x /= 4
print(x)        # x = 7
x %= 4
print(x)        # x = 3
x //= 2
print(x)        # x = 1
x **= 4
print(x)        # x = 1
```

Operators in python

Maktab Sharif

Comparison operators

==	Equal
!=	Not equal
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

```
x = 7
y = 5
z = -5

print(x == y)
print(x != y)
print(x < y)
print(y > -z)
print(z <= -y)
print(x >= y)
print(-z != y)
print(x+z < y)
print(z == z)
print(-z != z)
```

Operators in python

Maktab
Sharif

Comparison operators

==	Equal
!=	Not equal
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

```
x = 7
y = 5
z = -5

print(x == y)    # False
print(x != y)    # True
print(x < y)     # False
print(y > -z)    # False
print(z <= -y)   # True
print(x >= y)    # True
print(-z != y)   # False
print(x+z < y)   # True
print(z == z)    # True
print(-z != z)   # True
```

What is the result **type?**

Precedence of Python Operators

Maktab Sharif

**	Exponentiation (raise to the power)
~ + -	Complement, unary plus and minus
* / % //	Multiply, divide, modulo and floor division
+ -	Addition and subtraction
== != <= < > >=	Comparison operators
= %= /= //= -= += *= **=	Assignment operators

Precedence of Python Operators

Maktab Sharif

```
x = 5
y = 2

z = -y ** 3 * x + 1 > 10

print(z)     # ???
```

Precedence of Python Operators

Maktab Sharif

```
x = 5
y = 2

z = -y ** 3 * x + 1 > 10           # z = ((-(2**3) * 5) + 1) > 10

print(z)                           # False
```

Use parentheses !

Chapter 5

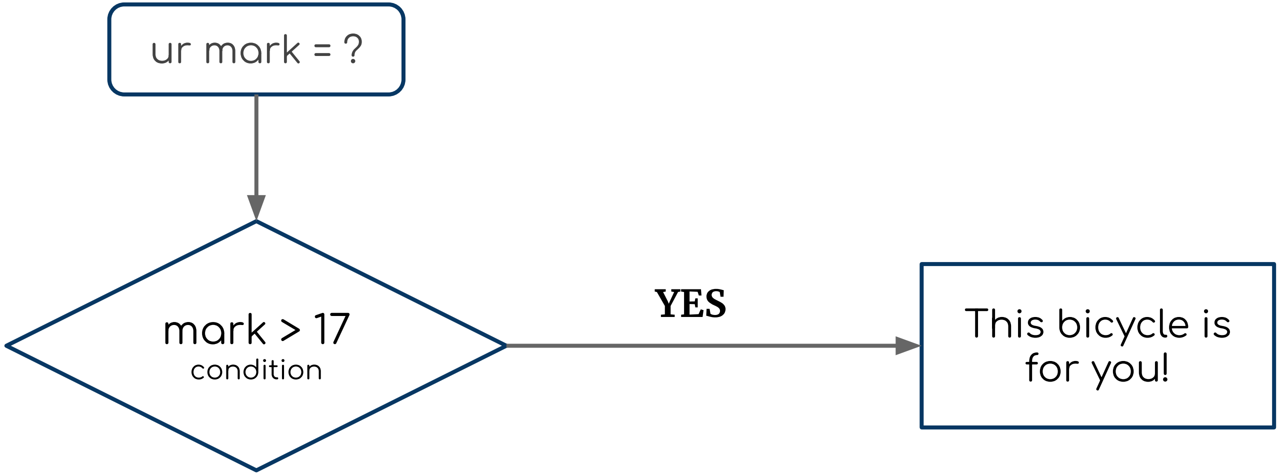
Conditional Statements

Conditions

Maktab Sharif

Final exam reward:

If you achieve mark greater than 17, I'll reward you with a bicycle, Son!



Conditions

Maktab Sharif

Final exam reward:

If you achieve mark greater than 17, then I'll reward you with a bicycle, Son!

```
mark = float(input("Enter your mark:"))

if mark > 17:
    print("This bicycle is for you!")
```

Conditions

Maktab Sharif

Final exam reward:

If you achieve mark greater than 17, then I’ll reward you with a bicycle, Son!

```
mark = float(input(“Enter your mark:”))

if mark > 17:
    print(“This bicycle is for you!”)
```

The Condition

if keyword

Conditions

Maktab Sharif

```
mark > 17
```

= ?

What's type of the result of condition?

bool

Conditional statements

Maktab Sharif

In order to write useful programs, we almost always need the ability to check conditions and change the behavior of the program accordingly. **Selection statements**, sometimes also referred to as **conditional statements**, give us this ability. The simplest form of selection is the **if statement**. This is sometimes referred to as binary selection since there are two possible paths of execution.

```
if BOOL_EXPRESSION:
    STATEMENTS_1        # executed if condition evaluates to True
    ...
    ...
    ...
```

Don't Forget TABs!

Some examples

Maktab Sharif

```
if True:
    print(“It’s True”)
```

```
if None:
    print(“It’s False”)
```

```
if 1:
    print(“It’s 1”)
```

```
if 12 >= 11:
    print(“It’s OK”)
```

```
if 12 % 2 == 1:
    print(“12 is Odd”)
```

```
if 1//2:
    print(“It’s 0.5”)
```

```
if ‘HIIII’:
    print(“It’s HIIII”)
```

```
if ‘’:
    print(“It’s Empty”)
```

Some examples

Maktab Sharif

```
if True:
    print(“It’s True”)
```
✓

```
if 12 % 2 == 1:
    print(“12 is Odd”)
```
✗

```
if False:
    print(“It’s False”)
```
✗

```
if 1//2:
    print(“It’s 0.5”)
```
✗

```
if 1:
    print(“It’s 1”)
```
✓

```
if ‘HIIII’:
    print(“It’s HIIII”)
```
✓

```
if 12 >= 11:
    print(“It’s OK”)
```
✓

```
if ‘’:
    print(“It’s Empty”)
```
✗

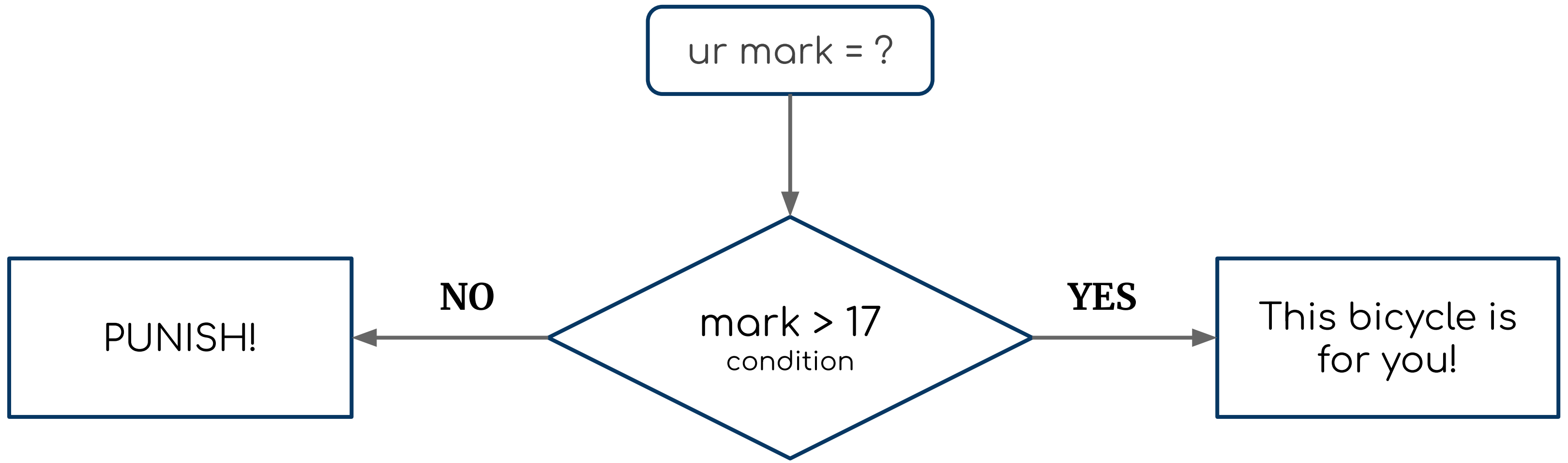
Else example

Maktab Sharif

Final exam reward:

If you achieve mark greater than 17, I'll reward you with a bicycle, Son!

Otherwise I'll punish you!



Else example

Maktab Sharif

Final exam reward:

If you achieve mark greater than 17, I'll reward you with a bicycle, Son!

Else I'll punish you!

```
mark = float(input("Enter your mark:"))

if mark > 17:
    print("This bicycle is for you!")
else:
    print("PUNISH!")
```

else keyword (otherwise)

Conditional statements w/ Else

Maktab Sharif

Another form of the if statement is one in which the else clause is omitted entirely. This creates what is sometimes called **unary selection**. In this case, when the condition evaluates to True, the statements are executed. Otherwise the flow of execution continues to the statement after the body of the if.

```
if BOOL_EXPRESSION:
    STATEMENTS_1         # executed if condition evaluates to True
    ...
else:
    STATEMENTS_2         # executed if condition evaluates to False
    ...
```

Exercise

Maktab Sharif

> Exercise
> Write a program that, gets 2 number from user,
> Then print the **greater** one.

input:

```
>>  11
>>  3
```

output:

```
>>  11
```

Exercise

Maktab Sharif

Exercise

Write a program that, gets 2 number from user,
Then print the **greater** one.

```
a = float(input(“Enter first Num: “))
b = float(input(“Enter second Num: “))

if a > b:
    print(a)
else:
    print(b)
```

Boolean Expressions

Maktab Sharif

Logical (boolean) operators

and	:	both variables MUST be True -> result: True
or	:	one of variables MUST be True -> result: True
not	:	invert the result

a	not a
True	False
False	True

```
x, y = True, False

print(x and y)
print(x or y)
print((not x) or y)
print(not(x or y))
print(x and (not y))
```

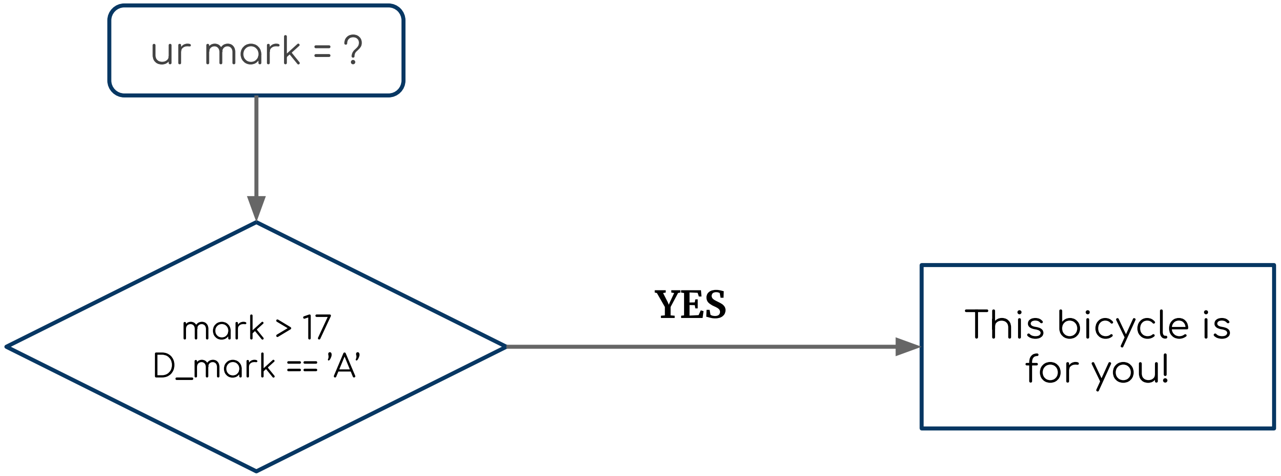
a	b	a and b	a or b
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

Example

Maktab Sharif

Final exam reward:

If you achieve total mark greater than 17 **and** discipline mark ‘A’
Then I’ll reward you with a bicycle, Son!



Example: Logical Operators

Maktab Sharif

Final exam reward:

If you achieve total mark greater than 17 **and** discipline mark ‘A’
Then I’ll reward you with a bicycle, Son!

```
mark = float(input(“Enter ur mark: “))
d_mark = input(“Enter discipline mark: “)

# Using logical Operators
ur_condition = mark > 17 and d_mark == ‘A’

if ur_condition:
    print(‘This bicycle is yours’)
```

Example: Nested-Conditions

Final exam reward:

If you achieve total mark greater than 17 **and** discipline mark 'A'
Then I'll reward you with a bicycle, Son!

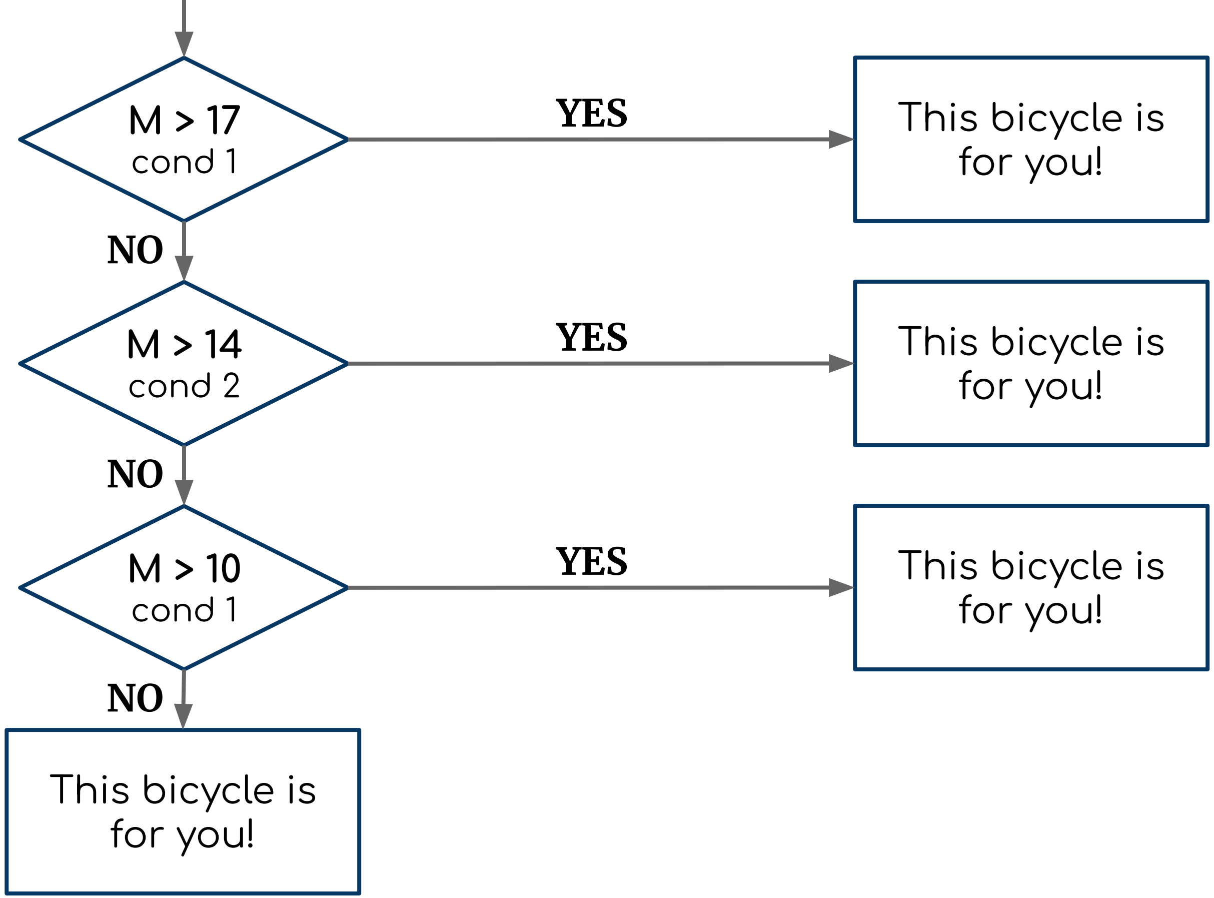
```
mark = float(input("Enter ur mark: "))
d_mark = input("Enter discipline mark: ")

if mark > 17:
    if d_mark == 'A':
        print('This bicycle is yours')
    else:
        print("You didn't Pass Discipline MARK condition")
else:
    print("You didn't Pass TOTAL MARK condition")
```

Example: Else if ...

Final exam reward:

If you achieve total mark greater than 17, I’ll reward you with a bicycle
Else if you achieve total mark between 14 and 17, I’ll reward you with a PS2,
Else if you achieve total mark greater than 10 and 14, It’s your Duty!
Else I’ll Punish you!



Maktab Sharif

Example: Else if ...

Example: Else if … (Using nested-Ifs)

Maktab Sharif

Final exam reward:

If you achieve total mark greater than 17, I'll reward you with a bicycle
Else if you achieve total mark between 14 and 17, I'll reward you with a PS2,
Else if you achieve total mark greater than 10 and 14, It's your Duty!
Else I'll Punish you!

```
mark = float(input("Enter ur mark: "))

if mark > 17:
    print('This bicycle is yours')
else:
    if mark > 14:
        print('This PS2 is yours')
    else:
        if mark >= 10:
            print("It's your Duty!")
        else:
            print("PUNISH!")
```

Chained Conditional statements (elif)

Maktab Sharif

Python provides an alternative way to write nested selection such as the one shown in the previous section. This is sometimes referred to as a chained conditional. **(else if -> elif)**

```
if BOOL_EXPRESSION_1:
    STATEMENTS_1        # executed if condition 1 evaluates to True
elif BOOL_EXPRESSION_2:
    STATEMENTS_2        # executed if condition 2 evaluates to True
elif BOOL_EXPRESSION_3:
    STATEMENTS_3        # executed if condition 3 evaluates to True
.
.
.
else:
    STATEMENTS_N        # executed if All of Them evaluates to False
```

Example: Else if … (Using elif)

Maktab Sharif

Final exam reward:

If you achieve total mark greater than 17, I'll reward you with a bicycle
Else if you achieve total mark between 14 and 17, I'll reward you with a PS2,
Else if you achieve total mark greater than 10 and 14, It's your Duty!
Else I'll Punish you!

```
mark = float(input("Enter ur mark: "))

if mark > 17:
     print('This bicycle is yours')
elif mark > 14:
     print('This PS2 is yours')
elif mark >= 10:
     print("It's your Duty!")
else:
     print("PUNISH!")
```

Pre-reading

Search about:

1. Compile python code
2. Complex type in python
3. **end** parameter in print() function

   Hint: search: *“end in python print”*
4. **sep** parameter in print() function
5. Strings in python
6. Multi-line string in python
7. Lists in python
8. Loops in python (while & for)