PYTHON FOR PHYSICISTS

Dr Lucia Fonseca de la Bella

Outline

- Basics
- Think like a programmer
- `if` statement
- `elif` clause
- `else` clause
- Summary
- Quizz.py

Basics

- Data types
  - Integer `3`
  - Float `.6`
  - Boolean `True, False`
- Arithmetic operators

  `+, -, *, /, %, **, //`
- Assignment operators

  `=, +=,-=,*=`

- Comparisson operators

==	equal
!=	not equal
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
and	and – both must be true
or	or – either may be true

- Functions: `print( )`

CODE

```
x=2
x+=4
print(x)
```

```
x=2
x=x+4
print(x)
```

6

OUTPUT

Think like a programmer

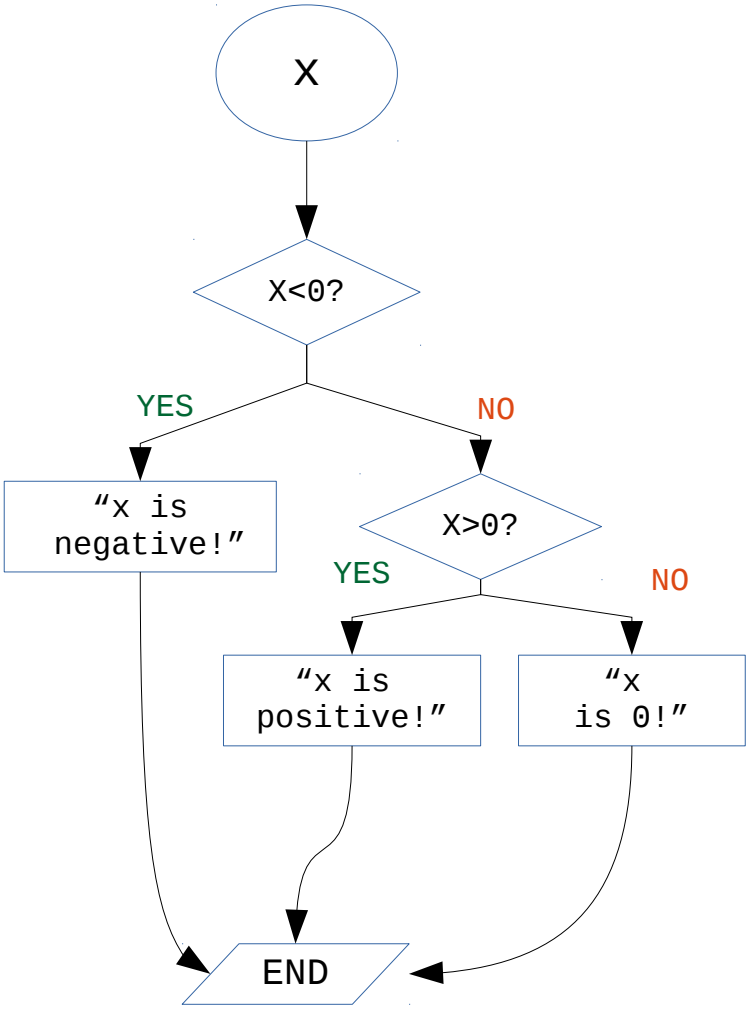
- We want to check whether a number x is negative, positive or null.
- How do we do that?

Think like a programmer

- We want to check whether a number x is negative, positive or null.
- How do we do that?

Think like a programmer

- We want to check whether a number x is negative, positive or null.
- How do we do that?



`if` statement

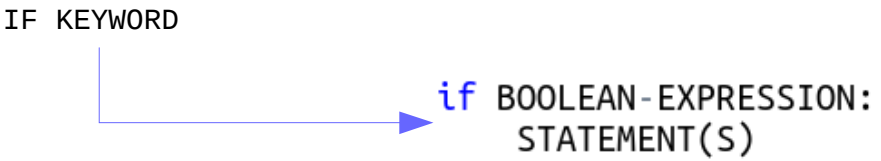
An `if` statement is a conditional statement that runs or skips code based on whether a condition is true or false.

Syntax:

`if` statement

An `if` statement is a conditional statement that runs or skips code based on whether a condition is true or false.

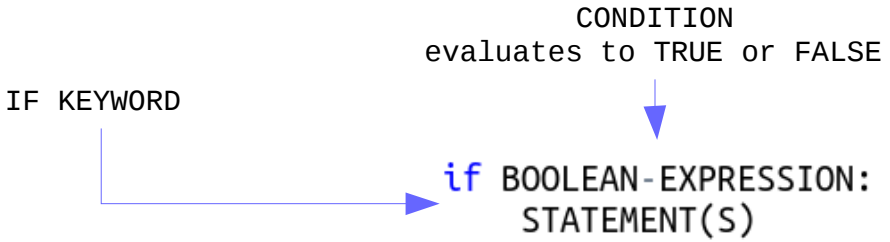
Syntax:



`if` statement

An `if` statement is a conditional statement that runs or skips code based on whether a condition is true or false.

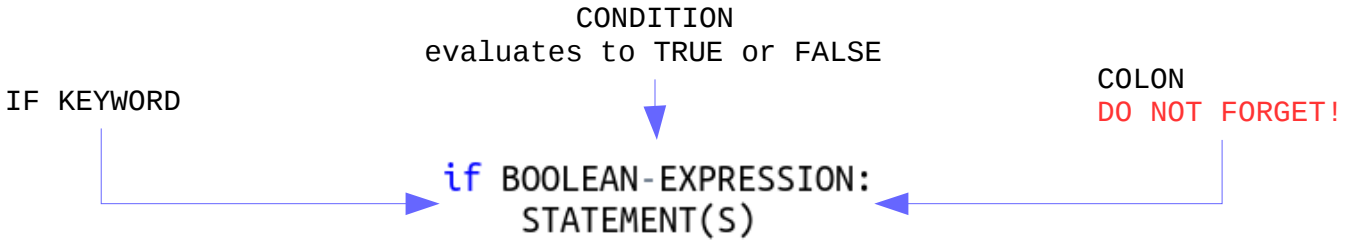
Syntax:



`if` statement

An `if` statement is a conditional statement that runs or skips code based on whether a condition is true or false.

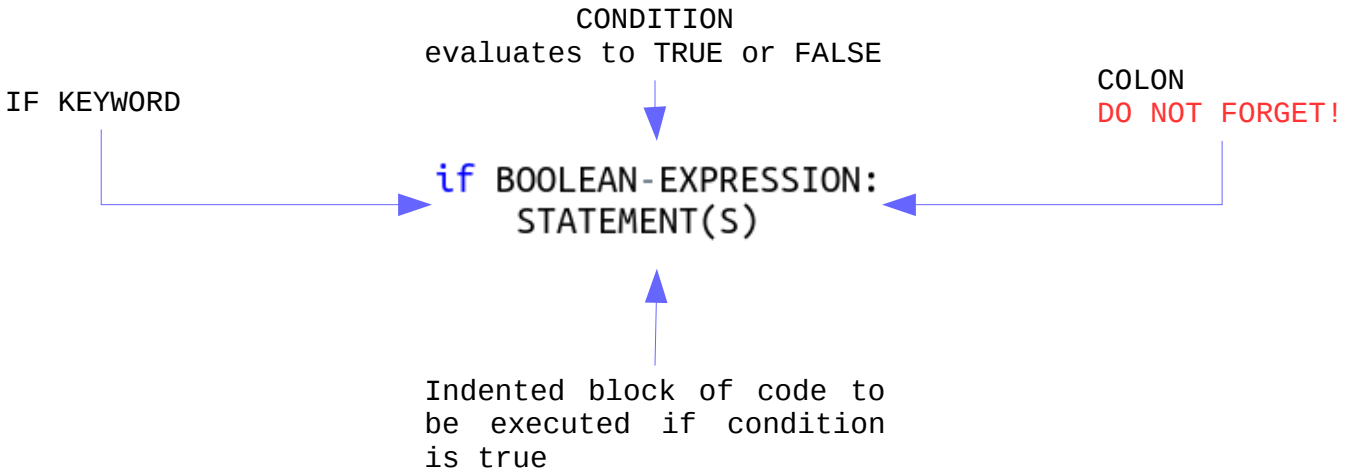
Syntax:



`if` statement

An `if` statement is a conditional statement that runs or skips code based on whether a condition is true or false.

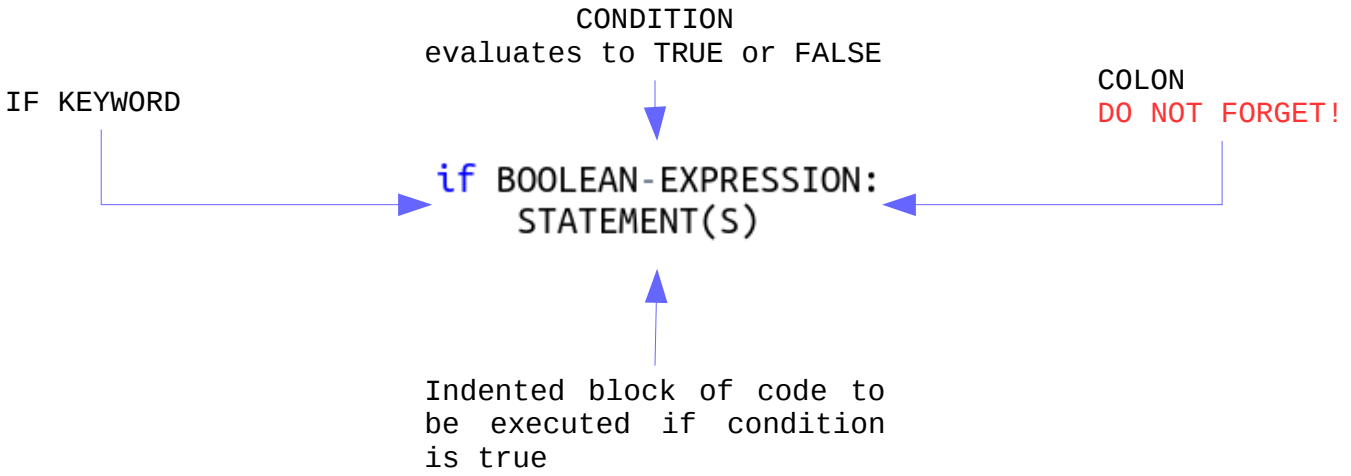
Syntax:



`if` statement

An `if` statement is a conditional statement that runs or skips code based on whether a condition is true or false.

Syntax:



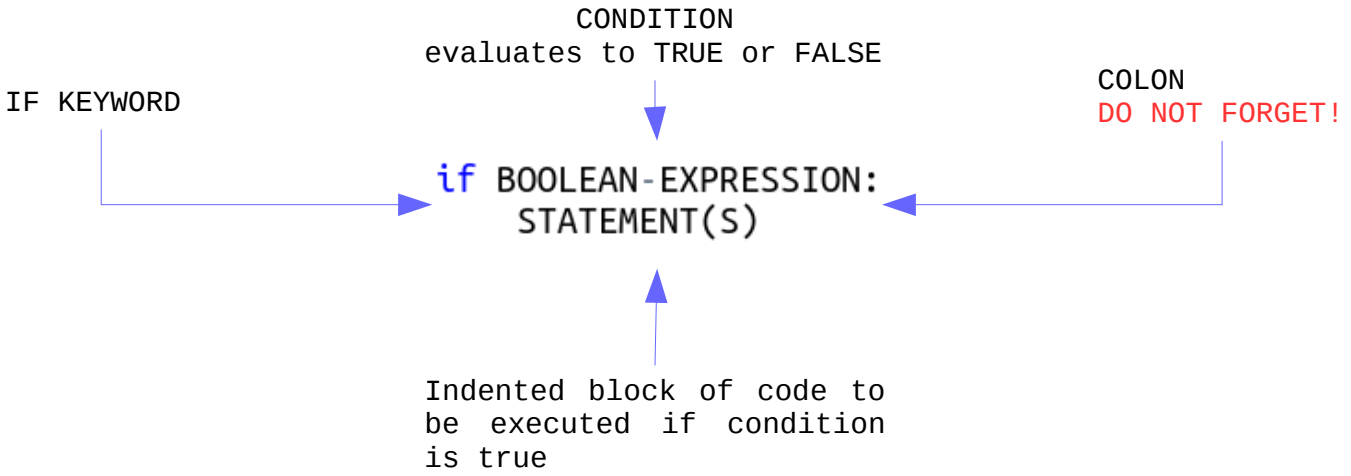
CODE

OUTPUT

`if` statement

An `if` statement is a conditional statement that runs or skips code based on whether a condition is true or false.

Syntax:



CODE

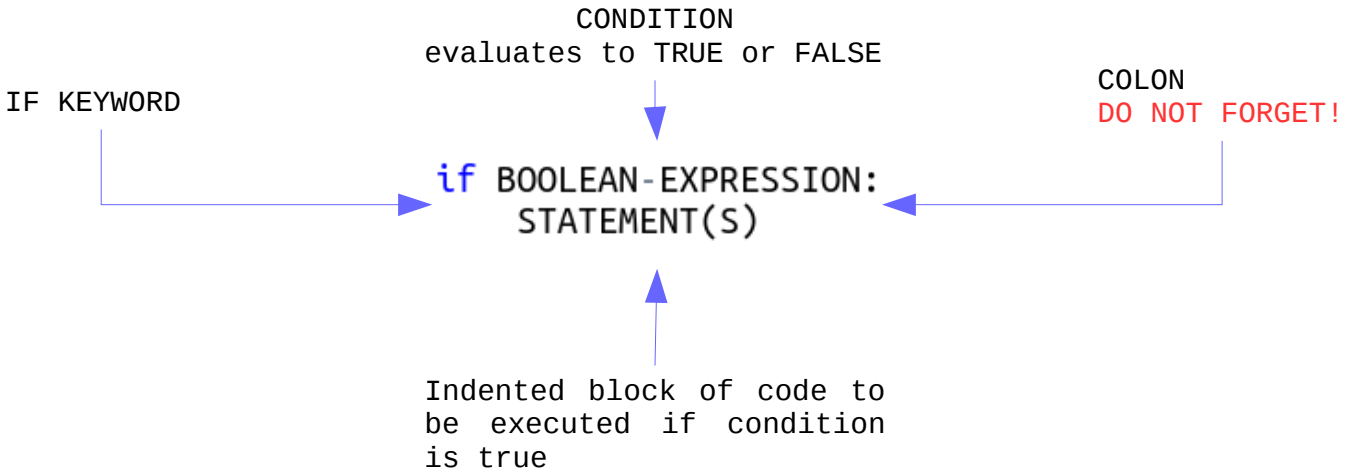
```
x = -2
if x<0:
    print("x is negative!")
```

OUTPUT

`if` statement

An `if` statement is a conditional statement that runs or skips code based on whether a condition is true or false.

Syntax:



CODE

```
x = -2
if x<0:
    print("x is negative!")
```

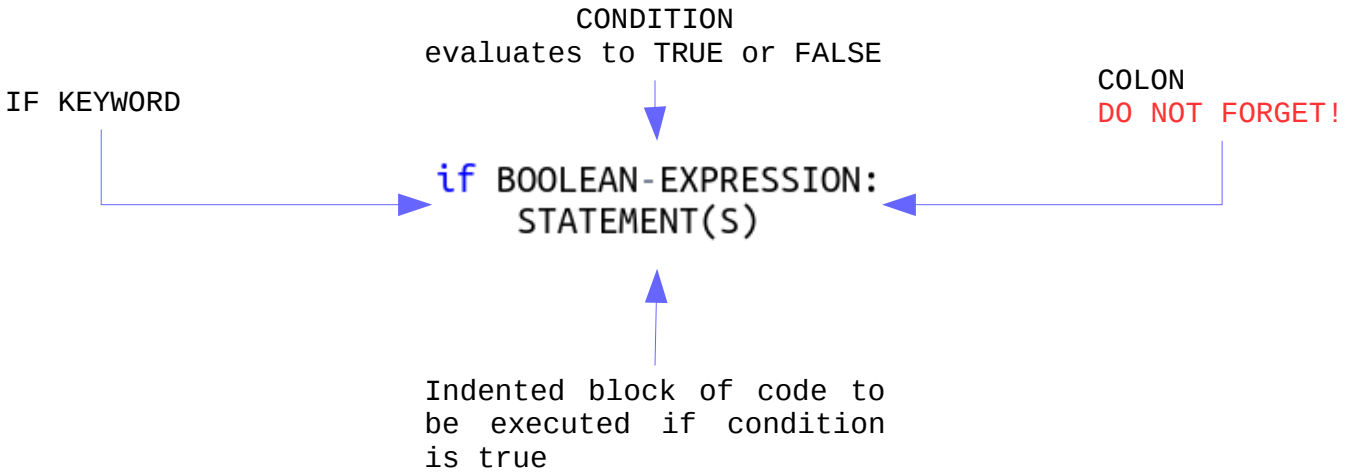
OUTPUT

```
x is negative!
```

`if` statement

An `if` statement is a conditional statement that runs or skips code based on whether a condition is true or false.

Syntax:



CODE

```
x = -2
if x<0:
    print("x is negative!")
```

OUTPUT

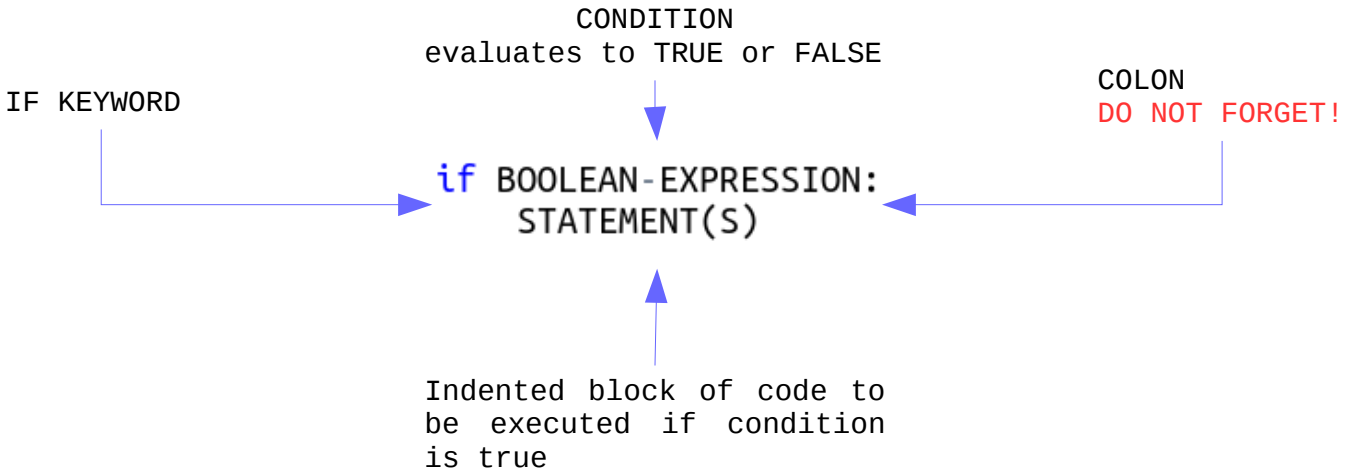
x is negative!

```
x = 3
if x<0:
    print("x is negative!")
```

`if` statement

An `if` statement is a conditional statement that runs or skips code based on whether a condition is true or false.

Syntax:



CODE

```
x = -2
if x<0:
    print("x is negative!")
```

OUTPUT

x is negative!

```
x = 3
if x<0:
    print("x is negative!")
```

`elif` clause

- `It` is short for "else if."
- Used to check for an additional condition if the conditions in the previous clauses evaluate to False.

Syntax:

```
if BOOLEAN-CONDITION_1:
    STATEMENT(S)
elif BOOLEAN-CONDITION_2:
    STATEMENT(S)
```

`elif` clause

- `It` is short for "else if."
- Used to check for an additional condition if the conditions in the previous clauses evaluate to False.

Syntax:

```
if BOOLEAN-CONDITION_1:
    STATEMENT(S)
elif BOOLEAN-CONDITION_2:
    STATEMENT(S)
```

CODE

OUTPUT

`elif` clause

- `It` is short for "else if."
- Used to check for an additional condition if the conditions in the previous clauses evaluate to False.

Syntax:

```
if BOOLEAN-CONDITION_1:
    STATEMENT(S)
elif BOOLEAN-CONDITION_2:
    STATEMENT(S)
```

CODE

```
x = 3
if x<0:
    print("x is negative!")
elif x>0:
    print("x is positive!")
```

OUTPUT

`elif` clause

- `It` is short for "else if."
- Used to check for an additional condition if the conditions in the previous clauses evaluate to False.

Syntax:

```
if BOOLEAN-CONDITION_1:
    STATEMENT(S)
elif BOOLEAN-CONDITION_2:
    STATEMENT(S)
```

CODE

```
x = 3
if x<0:
    print("x is negative!")
elif x>0:
    print("x is positive!")
```

OUTPUT

```
x is positive!
```

`elif` clause

- `It` is short for "else if."
- Used to check for an additional condition if the conditions in the previous clauses evaluate to False.

Syntax:

```
if BOOLEAN-CONDITION_1:
    STATEMENT(S)
elif BOOLEAN-CONDITION_2:
    STATEMENT(S)
```

CODE

```
x = 3
if x<0:
    print("x is negative!")
elif x>0:
    print("x is positive!")
```

```
x = 0
if x<0:
    print("x is negative!")
elif x>0:
    print("x is positive!")
```

OUTPUT

x is positive!

`elif` clause

- `It` is short for "else if."
- Used to check for an additional condition if the conditions in the previous clauses evaluate to False.

Syntax:

```
if BOOLEAN-CONDITION_1:
    STATEMENT(S)
elif BOOLEAN-CONDITION_2:
    STATEMENT(S)
```

CODE

```
x = 3
if x<0:
    print("x is negative!")
elif x>0:
    print("x is positive!")
```

```
x = 0
if x<0:
    print("x is negative!")
elif x>0:
    print("x is positive!")
```

OUTPUT

x is positive!

`else` clause

- It comes at the end of an if statement if used.
- This clause doesn't require a condition.
- The code in an else block is run if all conditions above that in the if statement evaluate to False.

Syntax:

```
if BOOLEAN-CONDITION_1:
    STATEMENT(S)
elif BOOLEAN-CONDITION_2:
    STATEMENT(S)
else:
    STATEMENT(S)
```

`else` clause

- It comes at the end of an if statement if used.
- This clause doesn't require a condition.
- The code in an else block is run if all conditions above that in the if statement evaluate to False.

Syntax:

```
if BOOLEAN-CONDITION_1:
    STATEMENT(S)
elif BOOLEAN-CONDITION_2:
    STATEMENT(S)
else:
    STATEMENT(S)
```

CODE

OUTPUT

else clause

- It comes at the end of an if statement if used.
- This clause doesn't require a condition.
- The code in an else block is run if all conditions above that in the if statement evaluate to False.

Syntax:

```
if BOOLEAN-CONDITION_1:
    STATEMENT(S)
elif BOOLEAN-CONDITION_2:
    STATEMENT(S)
else:
    STATEMENT(S)
```

CODE

```
x = 0
if x<0:
    print("x is negative!")
elif x>0:
    print("x is positive!")
else:
    print("x is 0!")
```

OUTPUT

else clause

- It comes at the end of an if statement if used.
- This clause doesn't require a condition.
- The code in an else block is run if all conditions above that in the if statement evaluate to False.

Syntax:

```
if BOOLEAN-CONDITION_1:
    STATEMENT(S)
elif BOOLEAN-CONDITION_2:
    STATEMENT(S)
else:
    STATEMENT(S)
```

CODE

```
x = 0
if x<0:
    print("x is negative!")
elif x>0:
    print("x is positive!")
else:
    print("x is 0!")
```

OUTPUT

```
x is 0!
```

else clause

- It comes at the end of an if statement if used.
- This clause doesn't require a condition.
- The code in an else block is run if all conditions above that in the if statement evaluate to False.

Syntax:

```
if BOOLEAN-CONDITION_1:
    STATEMENT(S)
elif BOOLEAN-CONDITION_2:
    STATEMENT(S)
else:
    STATEMENT(S)
```

CODE

```
x = 0
if x<0:
    print("x is negative!")
elif x>0:
    print("x is positive!")
else:
    print("x is 0!")
```

```
x = 0
if x<0:
    print("x is negative!")
if x>0:
    print("x is positive!")
if x==0:
    print("x is 0!")
```

OUTPUT

x is 0!

else clause

- It comes at the end of an if statement if used.
- This clause doesn't require a condition.
- The code in an else block is run if all conditions above that in the if statement evaluate to False.

Syntax:

```
if BOOLEAN-CONDITION_1:
    STATEMENT(S)
elif BOOLEAN-CONDITION_2:
    STATEMENT(S)
else:
    STATEMENT(S)
```

CODE

```
x = 0
if x<0:
    print("x is negative!")
elif x>0:
    print("x is positive!")
else:
    print("x is 0!")
```

```
x = 0
if x<0:
    print("x is negative!")
if x>0:
    print("x is positive!")
if x==0:
    print("x is 0!")
```

OUTPUT

x is 0!

x is 0!

else clause

- It comes at the end of an if statement if used.
- This clause doesn't require a condition.
- The code in an else block is run if all conditions above that in the if statement evaluate to False.

Syntax:

```
if BOOLEAN-CONDITION_1:
    STATEMENT(S)
elif BOOLEAN-CONDITION_2:
    STATEMENT(S)
else:
    STATEMENT(S)
```

CODE

```
x = 0
if x<0:
    print("x is negative!")
elif x>0:
    print("x is positive!")
else:
    print("x is 0!")
```

OUTPUT

```
x is 0!
```

GOOD PROGRAMMING PRACTICE

```
x = 0
if x<0:
    print("x is negative!")
if x>0:
    print("x is positive!")
if x==0:
    print("x is 0!")
```

```
x is 0!
```

CPU TIME CONSUMING

Summary

Sometimes we need to perform different actions based on different conditions:

```
if  boolean_condition:
    statement
elif boolean_condition:
    statement
else:
    statement
```

- Remember
  - Indentation is important!
  - Good programming practice.

- Next lecture: complex boolean conditions.

Quizz.py

QUESTION 1. Indentation

`Error` `x is less than or equal to 10!` `` `The speed is 10m/s`

CODE

```
x = 40
if x<=10:
print("x is less than or equal to 10!")
```

```
x = 0
if x<0:
    print("x is negative!")
    if x>0:
        print("x is positive!")
        if x==0:
            print("x is 0!")
```

```
v = 10
if v:
    print("The speed is {}m/s".format(v))
```

OUTPUT

Quizz.py

QUESTION 1. Indentation

`x is less than or equal to 10!` `The speed is 10m/s`

CODE

```
x = 40
if x<=10:
print("x is less than or equal to 10!")
```

```
    x = 0
    if x<0:
        print("x is negative!")
        if x>0:
            print("x is positive!")
            if x==0:
                print("x is 0!")
```

```
v = 10
if v:
    print("The speed is {}m/s".format(v))
```

OUTPUT

`Error`

Quizz.py

QUESTION 1. Indentation

x is less than or equal to 10!

The speed is 10m/s

CODE

```
x = 40
if x<=10:
print("x is less than or equal to 10!")
```

OUTPUT

Error

```
x = 0
if x<0:
    print("x is negative!")
    if x>0:
        print("x is positive!")
        if x==0:
            print("x is 0!")
```

```
v = 10
if v:
    print("The speed is {}m/s".format(v))
```

Quizz.py

QUESTION 1. Indentation

`x is less than or equal to 10!`

CODE

```
x = 40
if x<=10:
print("x is less than or equal to 10!")
```

OUTPUT

`Error`

```
x = 0
if x<0:
    print("x is negative!")
    if x>0:
        print("x is positive!")
        if x==0:
            print("x is 0!")
```

```
v = 10
if v:
    print("The speed is {}m/s".format(v))
```

`The speed is 10m/s`

Quizz.py

BONUS QUESTION!! Leap year – What would the output be?

```
# Python program to check whether
# the input year is a leap year or not

year = 2021

# To get year (integer input) from the user
# year = int(input("Enter a year: "))

if (year % 4) == 0:
   if (year % 100) == 0:
       if (year % 400) == 0:
           print("{0} is a leap year".format(year))
       else:
           print("{0} is not a leap year".format(year))
   else:
       print("{0} is a leap year".format(year))
else:
   print("{0} is not a leap year".format(year))
```

```
2021 is a leap year
```

```
2021 is not a leap year
```

Remember:
% returns the remainder

```
print(5%2)
1
```

Quizz.py

BONUS QUESTION!! Leap year – What would the output be?

```
# Python program to check whether
# the input year is a leap year or not

year = 2021

# To get year (integer input) from the user
# year = int(input("Enter a year: "))

if (year % 4) == 0:
   if (year % 100) == 0:
       if (year % 400) == 0:
           print("{0} is a leap year".format(year))
       else:
           print("{0} is not a leap year".format(year))
   else:
       print("{0} is a leap year".format(year))
else:
   print("{0} is not a leap year".format(year))
```

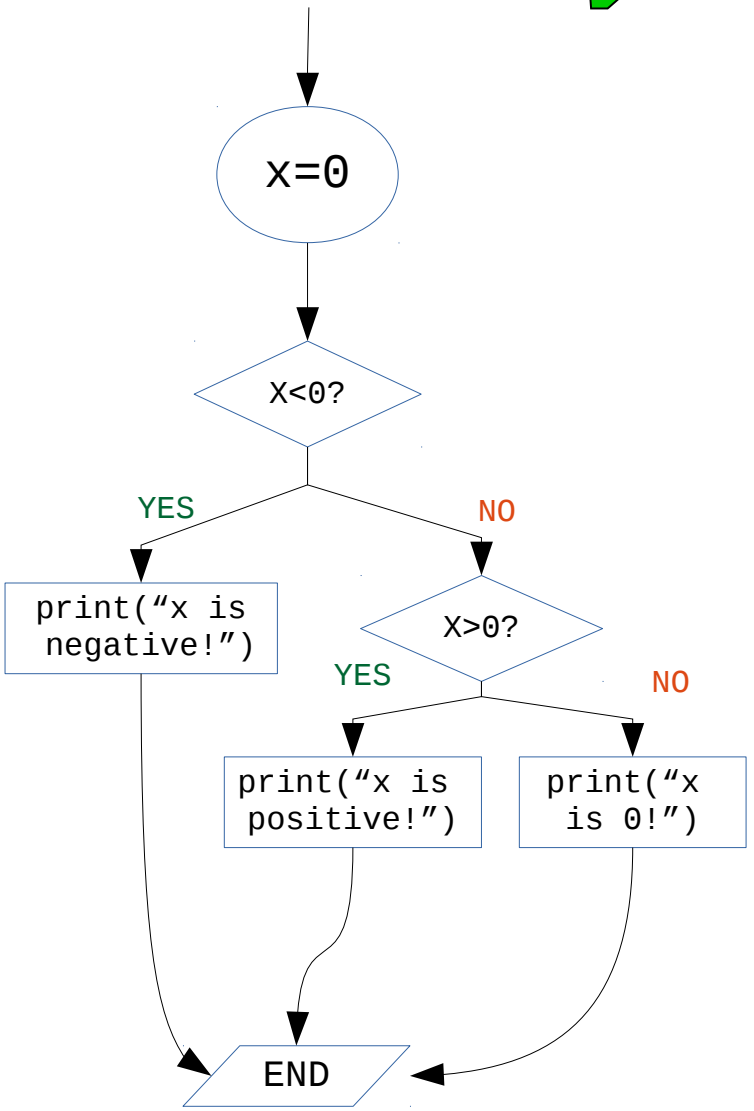
```
2021 is a leap year
```

```
2021 is not a leap year
```

Remember:
% returns the remainder

```
print(5%2)
1
```

```
if x<0:
    print("x is negative!")
elif x>0:
    print("x is positive!")
else:
    print("x is 0!")
```




```
if x<0:
    print("x is negative!")
if x>0:
    print("x is positive!")
if x==0:
    print("x is 0!")
```