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
  =, +=, -=, *=

- Comparison operators
  
<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>equal</td>
</tr>
<tr>
<td>!=</td>
<td>not equal</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater than or equal to</td>
</tr>
<tr>
<td>&lt;=</td>
<td>less than or equal to</td>
</tr>
<tr>
<td>and</td>
<td>and – both must be true</td>
</tr>
<tr>
<td>or</td>
<td>or – either may be true</td>
</tr>
</tbody>
</table>

- Functions: print()
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?
An `if` statement is a conditional statement that runs or skips code based on whether a condition is true or false.

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

**Syntax:**

```plaintext
IF KEYWORD

if BOOLEAN-EXPRESSION:
  STATEMENT(S)
```
An if statement is a conditional statement that runs or skips code based on whether a condition is true or false.

**Syntax:**

```
IF KEYWORD
CONDITION
  evaluates to TRUE or FALSE

if BOOLEAN-EXPRESSION:
  STATEMENT(S)
```
An if statement is a conditional statement that runs or skips code based on whether a condition is true or false.

**Syntax:**

```plaintext
if BOOLEAN-EXPRESSION:
    STATEMENT(S)
```

- **IF** KEYWORD
- **CONDITION** evaluates to **TRUE** or **FALSE**
- **COLON**
  - **DO NOT FORGET!**
An if statement is a conditional statement that runs or skips code based on whether a condition is true or false.

Syntax:

```
if BOOLEAN-EXPRESSION:
  STATEMENT(S)

if
CONDITION evaluates to TRUE or FALSE
```

DO NOT FORGET!

Indented block of code to be executed if condition is true
An if statement is a conditional statement that runs or skips code based on whether a condition is true or false.

Syntax:

```
if BOOLEAN-EXPRESSION:
    STATEMENT(S)
```

- **IF KEYWORD**
- **CONDITION** evaluates to TRUE or FALSE
- **DO NOT FORGET!**
- **COLO**
- Indented block of code to be executed if condition is true

**CODE**

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

**Syntax:**

```python
if BOOLEAN-EXPRESSION:
    STATEMENT(S)
```

- **IF KEYWORD** evaluates to TRUE or FALSE
- **COLON** DO NOT FORGET!
- Indented block of code to be executed if condition is true

**Example Code:**

```python
x = -2
if x<0:
    print("x is negative!")
```
An if statement is a conditional statement that runs or skips code based on whether a condition is true or false.

Syntax:

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

Output:

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

**Syntax:**

```
if BOOLEAN-EXPRESSION:
  STATEMENT(S)
```

*CONDITION evaluates to TRUE or FALSE*

*DO NOT FORGET!*

*Indented block of code to be executed if condition is true*

**CODE**

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

**OUTPUT**

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

Syntax:

```
if BOOLEAN-EXPRESSION:
    STATEMENT(S)
```

- **IF KEYWORD**
- **CONDITION** evaluates to **TRUE** or **FALSE**
- **COLON** DO NOT FORGET!

**Indented block of code to be executed if condition is true**

**CODE**

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

**OUTPUT**

```
x is negative!
```
• It is short for "else if."

• Used to check for an additional condition if the conditions in the previous clauses evaluate to False.

**Syntax:**

```python
if BOOLEAN-CONDITION_1:
    STATEMENT(S)
elif BOOLEAN-CONDITION_2:
    STATEMENT(S)
```
• It is short for "else if."

• Used to check for an additional condition if the conditions in the previous clauses evaluate to False.

**Syntax:**

```python
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:**

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

**CODE**

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

**OUTPUT**

```python

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

**Syntax:**

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

**CODE**

```python
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!")
```

**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:

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

**CODE**

```python
x = 3
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:**

```python
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)
```

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

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

**CODE**

```python
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!
```
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**

```python
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:

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

CODE

```python
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!
```

```
 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:**

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

---

**Code**

```python
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**

**CPU TIME CONSUMING**
Sometimes we need to perform different actions based on different conditions:

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

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

- **Next lecture:** complex boolean conditions.
**QUESTION 1.** Indentation

**Error**  
*x is less than or equal to 10!*

**The speed is 10m/s**

```python
# 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**
QUESTION 1. Indentation

CODE

```python
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
QUESTION 1. Indentation

x is less than or equal to 10!

The speed is 10 m/s

CODE

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

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

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

OUTPUT

Error
QUESTION 1. Indentation

x is less than or equal to 10!

---

**CODE**

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

---

**OUTPUT**

```
Error
```

---

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

---

```
The speed is 10m/s
```

---

```python
v = 10
if v:
    print("The speed is {}m/s".format(v))
```
BONUS QUESTION!! Leap year – What would the output be?

```python
# 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))
```

Remember:
% returns the remainder

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

Remember: % returns the remainder

print(5%2)
1