

# Programming

Lab Sessions

# Boolean

- A Boolean is a type of variable that can assume only two distinct values:
  - True
  - False

# Boolean Expressions

- $>$  larger than
  - $<$  smaller than
  - $==$  is equal to
  - $!=$  is not equal to
  - $>=$  larger or equal than
  - $<=$  smaller or equal than
- CAREFUL**
- $X == Y$ : is a Boolean expression; X is equal to Y
  - $X = Y$ : is a variable assignment, assigns to X the value of Y

# Boolean Expressions

- Let us try some

# Questions?

# Boolean Operators

In order of precedence

- not (negation)
  - not  $x$ ; True if  $x$  is False, and False if  $x$  is True
- and
  - $x$  and  $y$ ; is only True if both  $x$  and  $y$  are True
- or
  - $x$  or  $y$ ; is True if either  $x$  or  $y$  are True
- Note that  $x$  and  $y$  can be either Boolean variables or Boolean Expressions

# Boolean Operators

Exp 1	Exp 2	Exp 1 and Exp 2	Exp 1 or Exp 2	not Exp 1
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

# Boolean Operators

- Let us try some



# Questions?

# Decision Structures

- **if**
  - The simplest form of decision statement
  - Will execute a section of code if a given Boolean expression is True
- Structure of an if statement

**if** a>n :

**print**(a, 'is larger than', n)

# Decision Structures

- if
  - The simplest form of decision statement
  - Will execute a section of code if a given Boolean expression is True
- Structure of an if statement

```
if a > n :
```

```
    print(a, 'is larger than', n)
```

# Decision Structures

- if
  - The simplest form of decision statement
  - Will execute a section of code if a given Boolean expression is True
- Structure of an if statement

```
if a > n :
```

```
    print(a, 'is larger than', n)
```

# Decision Structures

- if
  - The simplest form of decision statement
  - Will execute a section of code if a given Boolean expression is True
- Structure of an if statement

```
if a > n :
```

```
    print(a, 'is larger than', n)
```

# Decision Structures

- if
  - The simplest form of decision statement
  - Will execute a section of code if a given Boolean expression is True
- Structure of an if statement

```
if a > n :
```

```
    print(a, 'is larger than', n)
```

# Example

- Let's write a program that will tell us if this a good day to go to the beach. It will only be a good day if it is **sunny** and the **temperature is greater than 24°C**

# Example

```
#This program evaluates if this is a good day to go to the beach
```

```
#Get user input
```

```
sunny = int(input('Is today sunny? 0 for no and 1 for yes: '))  
temperature = int(input('What is the current temperature? '))
```

```
#Evaluate if this is a good day to go to the beach
```

```
if sunny and temperature > 24:
```

```
    print('This is a good day to go to the beach')
```

## Note

- In python, 0 (integer, float or even complex) will have a Boolean value of False
- Any nonzero number will have a Boolean value of True



# Example

- Let us get rid of the 1 and 0 for the answer about the sunniness of the day and allow the user to answer yes or no.

# Example

```
#This program evaluates if this is a good day to go to the beach
```

```
#Get user input
```

```
sunny = input('Is today sunny? yes or no: ')
```

```
temperature = int(input('What is the current temperature? '))
```

```
#Convert sunny to boolean
```

```
if sunny == 'yes':
```

```
    sunny_boolean = True
```

```
if sunny == 'no':
```

```
    sunny_boolean = False
```

```
#Evaluate if this is a good day to go to the beach
```

```
if sunny_boolean and temperature > 24:
```

```
    print('This is a good day to go to the beach')
```

# Decision Structures

- else
  - Complements the if statement, executes a block of code if the condition for the if statement is false
  - Needs an if to exist
  - One of the code blocks will always be executed
- Structure of an if-else statement

**if** a>n :

**print**(a, 'is larger than', n)

**else:**

**print**(a, 'is not larger than', n)

# Example

- Back to our beach decider, let's print a message also when it is not a good day to go to the beach

# Example

```
#This program evaluates if this is a good day to go to the beach
```

```
#Get user input
```

```
sunny = input('Is today sunny? yes or no: ')
```

```
temperature = int(input('What is the current temperature? '))
```

```
#Convert sunny to boolean
```

```
if sunny == 'yes':
```

```
    sunny_boolean = True
```

```
if sunny == 'no':
```

```
    sunny_boolean = False
```

```
#Evaluate if this is a good day to go to the beach
```

```
if sunny_boolean and temperature > 24:
```

```
    print('This is a good day to go to the beach')
```

```
else:
```

```
    print('This is not a good day to go to the beach')
```

# Questions?

# Decision Structures

- The elif statement
  - Short for else-if
  - An alternative block of code with an alternative condition
  - Only executed if the preceding if was not executed

```
a = 1
if a<3:
    print('First if')
if a<2:
    print('Second if')
else:
    print('Else')
```

```
a = 1
if a<3:
    print('First if')
elif a<2:
    print('Second if')
else:
    print('Else')
```

# Example

- Back to our beach decider, let's use nested elif statements to warn the user when he has entered wrong data



# Example

```
#This program evaluates if this is a good day to go to the beach

#Get user input
sunny = input('Is today sunny? yes or no: ')
temperature = int(input('What is the current temperature? '))

#Convert sunny to boolean
if sunny == 'yes':
    sunny_boolean = True
elif sunny == 'no':
    sunny_boolean = False
else:
    print('You have entered an unexpected input')

#Evaluate if this is a good day to go to the beach
if sunny_boolean:
    print('This is a good day to go to the beach')
else:
    print('This is not a good day to go to the beach')
```

# Questions?

# Decision Structures

- Nested if-else
  - if-else statements can exist inside other if-else statements
  - Be careful with the indentation
  - Should be avoided as they can get very confusing

- Nested if-else statement

```
if n>0:  
    print('n is positive')  
else:  
    if(n == 0):  
        print('n is null')  
    else:  
        print('n is negative')
```

# Example

- Back to our beach decider, let's use nested if else statements to warn the user when he has entered wrong data

# Example

```
#This program evaluates if this is a good day to go to the beach

#Get user input
sunny = input('Is today sunny? yes or no: ')
temperature = int(input('What is the current temperature? '))

#Convert sunny to boolean
if sunny == 'yes':
    sunny_boolean = True
else:
    if sunny == 'no':
        sunny_boolean = False
    else:
        print('You have entered an unexpected input')

#Evaluate if this is a good day to go to the beach
if sunny_boolean:
    print('This is a good day to go to the beach')
else:
    print('This is not a good day to go to the beach')
```

# Exercise

- Write a program where the user must input two values, the program must print the largest value, or, in case they are equal, must inform the user they are equal

# Exercise

#This program compares two values and informs the output to the user

#Getting input

```
first = int(input('Please enter the first value: '))
```

```
second = int(input('Please enter the second value: '))
```

#Comparing the values

```
if first > second:
```

```
    print(first, 'is larger than', second)
```

```
else:
```

```
    if second > first:
```

```
        print(second, 'is larger than', first)
```

```
    else:
```

```
        print('The numbers are equal')
```

# Questions?



# Repetition Structures

- while
  - Conditional repetition
  - Specifies a block of code that will be repeated **while** a condition is True
- Structure of a while statement

```
x=0
```

```
while x<10 :  
    print(x)  
    x = x + 1
```

# Repetition Structures

- while
  - Specifies a block of code that will be repeated **while** a condition is True
- Structure of a while statement

```
x=0
```

```
while x<10 :  
    print(x)  
    x = x + 1
```

# Repetition Structures

- while
  - Specifies a block of code that will be repeated **while** a condition is True
- Structure of a while statement

```
x=0
```

```
while x<10 :  
    print(x)  
    x = x + 1
```

# Repetition Structures

- while
  - Specifies a block of code that will be repeated **while** a condition is True
- Structure of a while statement

```
x=0
```

```
while x<10 :  
    print(x)  
    x = x + 1
```

# Repetition Structures

- while
  - Specifies a block of code that will be repeated **while** a condition is True
- Structure of a while statement

```
x=0
```

```
while x<10 :  
    print(x)  
    x = x + 1
```

# Example

- Back to our beach decider, let's make sure the user can only input yes or no to question about the sunniness of the day.

# Example

```
#This program evaluates if this is a good day to go to the beach

#Get user input
sunny = input('Is today sunny? yes or no: ')

#Making sure the user answered with yes or no
while sunny != 'yes' and sunny != 'no':
    print('Please enter only yes or no')
    sunny = input('Is today sunny? yes or no: ')

temperature = int(input('What is the current temperature? '))

#Convert sunny to boolean
if sunny == 'yes':
    sunny_boolean = True
else:
    sunny_boolean = False

#Evaluate if this is a good day to go to the beach
if sunny_boolean:
    print('This is a good day to go to the beach')
else:
    print('This is not a good day to go to the beach')
```

# Repetition Structures

- while
  - If the condition is False at the start of the loop, the code block will never be executed
  - There must be a commands inside the block that will eventually make the condition become False, or you code will enter an infinite loop



# Questions?

# Example

- Let's write a program that prints the sum of all even number from 100 to 200.

# Example

```
#Definying variables
```

```
number = 100
```

```
sum = 0
```

```
#While loop summing all even numbers
```

```
while number <= 200:
```

```
    sum = sum+number
```

```
    number = number+2
```

```
#Print the result
```

```
print(sum)
```

# Questions?

# Exercise

- Write a program where the user can input an integer number, the program will print all number from 1 to the inputted number

# Exercise

```
#Get user input
final_number = int(input('Please enter a
number:'))

#The current number to be printed
current_number = 1

#Loop printing the current number and
incrementing it
while current_number <= final_number:
    print(current_number)
    current_number = current_number+1
```

# Questions?

# Repetition Structures

- for
  - Countable repetition
  - Specifies a block of code that will be repeated a given number of times
- Structure of a for statement

```
for x in (0, 1, 2, 3):  
    print(x)
```



# Repetition Structures

- for
  - Countable repetition
  - Specifies a block of code that will be repeated a given number of times
- Structure of a for statement

```
for x in (0, 1, 2, 3):  
    print(x)
```

# Repetition Structures

- for
  - Countable repetition
  - Specifies a block of code that will be repeated a given number of times
- Structure of a for statement

```
for x in (0, 1, 2, 3) :  
    print(x)
```

# Repetition Structures

- for
  - Countable repetition
  - Specifies a block of code that will be repeated a given number of times
- Structure of a for statement

```
for x in (0, 1, 2, 3) :  
    print(x)
```

# Repetition Structures

- for
  - Countable repetition
  - Specifies a block of code that will be repeated a given number of times
- Structure of a for statement

```
for x in (0, 1, 2, 3) :  
    print(x)
```

# Range of Values

- A range can be defined given the values of where it begins, ends, and what is the increment between values
- Example

```
for x in range(0, 5, 1):
```

```
    print(x)
```

start (Optional) –  
If omitted, beginning = 0

stop (mandatory) –  
stop value is NOT included

increment (Optional)-  
If omitted, increment = 1

# Example

- Let's write a program that prints the sum of all even number between two values specified by the user.

# Example

```
# Program that sums all even numbers within a range
specified by the user

# Get user input
start = int(input('Please input the start of the range: '))
end = int(input('Please input the end of the range: '))

# For loop summing all values that are even
sum = 0
for value in range(start, end+1):
    if value%2 == 0:
        sum = sum+value

# Print result
print(sum)
```

# Questions?



# Continue and Break

- `continue`
  - Tells python to treat the loop execution as complete and move to on to the next
- `break`
  - Tells python to stop the execution of the loop entirely and move on to the next part of the code
- Work on for or while loops

```
for value in range(start, end+1):  
    if value == 6:  
        continue  
    if value%2 == 0:  
        sum = sum+value
```

Will not include 6 in the sum

```
x = 0  
while x < 10:  
    if(x==6):  
        break  
    print(x)  
    x = x + 1
```

Will stop the loop when x=6

# Example

- Let's write that asks the user to input number until he decides to quit, and then prints the sum of all odd numbers he entered that were different from 9.

# Example

```
#Program that calculates the sum of odd numbers entered by user until he decides to quit
```

```
#declare the sum
```

```
sum = 0
```

```
while True:
```

```
    value = int(input('Please type a number, type 0 to quit: '))
```

```
    #if value is 0, break from the loop
```

```
    if value == 0:
```

```
        break
```

```
    #If value is 9, do nothing and return to start of loop
```

```
    if value == 9:
```

```
        continue
```

```
    #Else, If value is odd, sum it
```

```
    elif value%2 != 0:
```

```
        sum = sum + value
```

```
print(sum)
```

# Questions?

# Fin