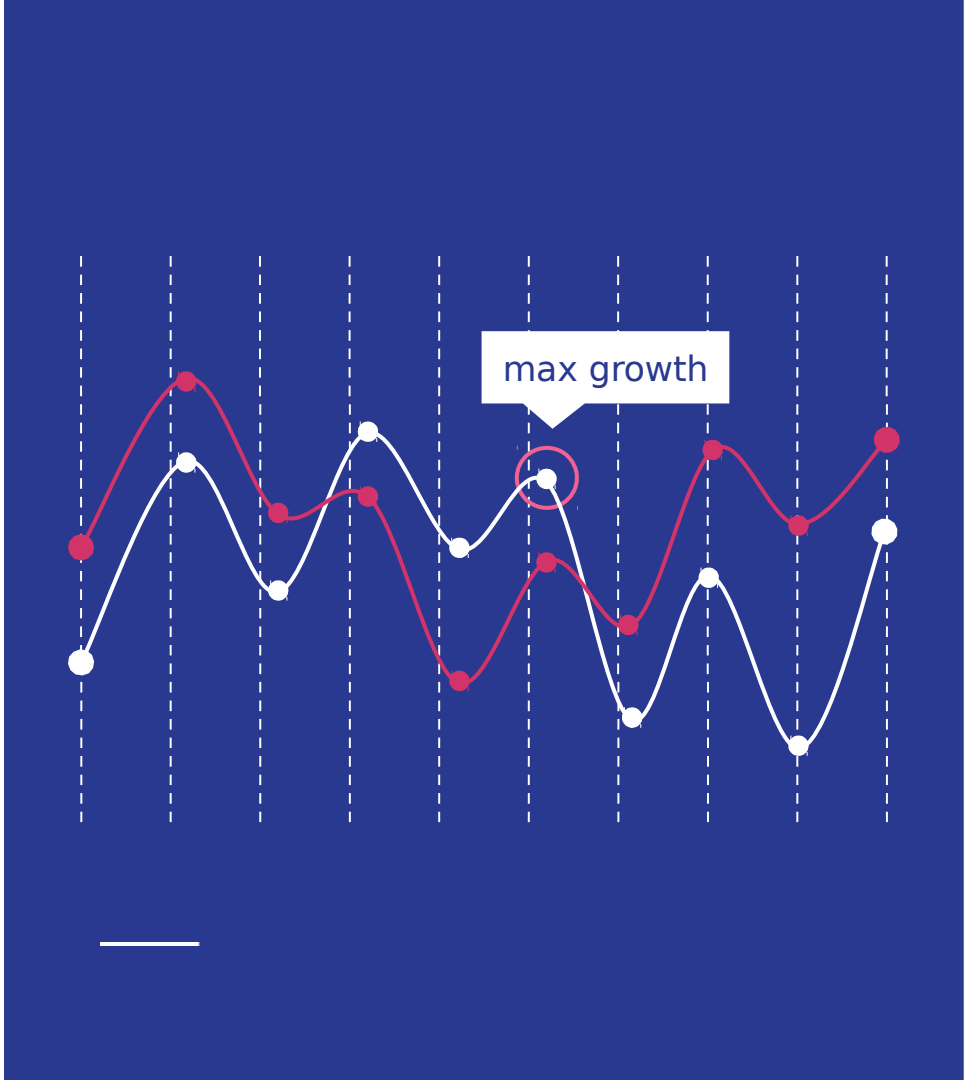
Introduction to Computer Programming

CMP 201 (2019/2020)

SORINOLU, Babafemi Gabriel



LESSON 3:
Be a Ninja Coder!

Objectives

The aim of this lesson is to build on the foundational knowledge given to the students in the first lecture.

Content(Week 1)

- **Operators**
- Decision making

Python Operators:

Operators are used to perform operations on variables and values. Python divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators
- Identity operators
- Membership operators
- Bitwise operators

Arithemetic Operator:

Arithmetic operators are used with numeric values to perform common mathematical operations

Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y

Arithemetic Operator:

Arithmetic operators are used with numeric values to perform common mathematical operations

Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y

PUZZLE:

```
x= 51 % 3
print ( x  )
```

Assignment Operator:

Assignment operators are used to assign values to variables

Operator	Example	Same As
=	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

Comparison Operator:

Comparison operators are used to compare two values

Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Logical Operator:

Logical operators are used to combine conditional statements

Operator	Description	Example
and	Returns True if both statements are true	x < 5 and x < 10
or	Returns True if one of the statements is true	x < 5 or x < 4
not	Reverse the result, returns False if the result is true	not(x < 5 and x < 10)

Identity Operator:

Identity operators are used to compare the objects, not if they are equal, but if they are actually the same object, with the same memory location

Operator	Description	Example
is	Returns true if both variables are the same object	x is y
is not	Returns true if both variables are not the same object	x is not y

Membership Operator:

Membership operators are used to test if a sequence is presented in an object

Operator	Description	Example
in	Returns True if a sequence with the specified value is present in the object	x in y
not in	Returns True if a sequence with the specified value is not present in the object	x not in y

Bitwise Operator:

Bitwise operators are used to compare (binary) numbers

Operator	Name	Description
&	AND	Sets each bit to 1 if both bits are 1
|	OR	Sets each bit to 1 if one of two bits is 1
^	XOR	Sets each bit to 1 if only one of two bits is 1
~	NOT	Inverts all the bits
<<	Zero fill left shift	Shift left by pushing zeros in from the right and let the leftmost bits fall off
>>	Signed right shift	Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off

Decision Making :

We make many if statement in our day to day activities .

Programs can also make decisions based on certain conditions.

How do you represent conditions in python ?

How can you determine the greater between two number ?

I will vote if I am 18 years or older

I will go to work if today is Tuesday

Simple If statement :

An "if statement" is written by using the if keyword.

Python relies on indentation, using whitespace, to define scope in the code.

SYNTAX:

```
if (condition) :
    //code to be executed
```

Simple If statement :

Example

```
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

Simple If statement :

Example

```
age = 33
if age >= 18:
  print(“I am eligible to vote")
```

Simple If statement :

Example

```
today= ”wednesday”
if today == “tuesday”:
  print(“I am going to work")
```

elif statement :

The elif keyword is pythons way of saying "if the previous conditions were not true, then try this condition".

SYNTAX:

```
if (condition) :
    //code to be executed
elif (condition) :
    //other code to be executed
```

elif statement :

Example

```
age = 33
if (age >= 18):
  print(“I am eligible to vote")
elif (age<18):
  print(“not eligible to vote")
```

else statement :

The else keyword catches anything which isn't caught by the preceding conditions.

SYNTAX:

```
if (condition) :
    //code to be executed
elif (condition) :
    //other code to be executed
else :
    //other code to be executed
```

else statement :

You can have an else without an elif

SYNTAX:

```
if (condition) :
    //code to be executed
else :
    //other code to be executed
```

Using the and/or operator :

The and/or keyword is a logical operator that can be used to combine conditional statements:

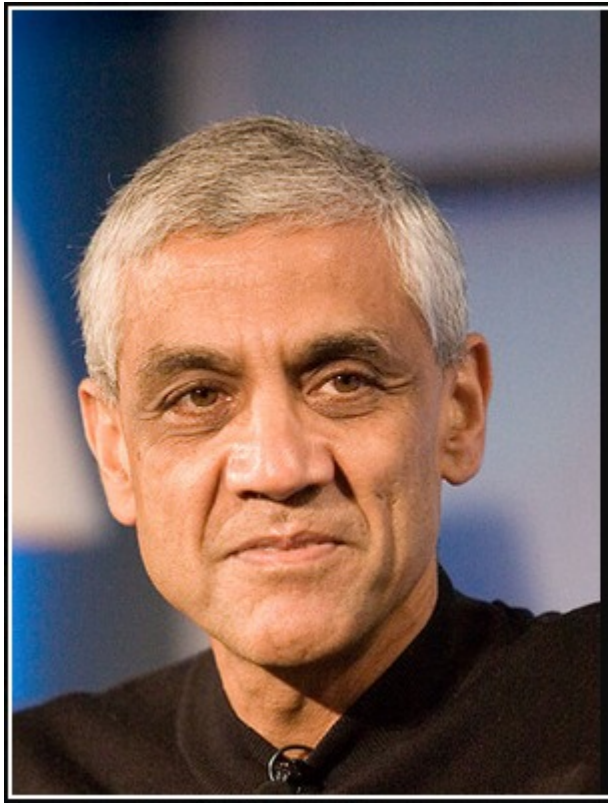
I will eat rice **and** beans

I will eat rice **or** beans

EXERCISE

(Uniqueness in code earns extra credit).

1. **Each question should be kept in a single .py file**
2. **then all zipped in a file**
3. **with your matric No. as name of the file**



Doctors can be replaced by software – 80% of them can. I'd much rather have a good machine learning system diagnose my disease than the median or average doctor.

— Vinod Khosla —

AZ QUOTES