

Contents

- Introduction to Python
- Applications of Python
- IDEs
- Introduction to Python Interpreter
- Indentation and Comments
- Keywords
- Variables – Identifiers
- Built-In Types
- Assigning Values to Variables
- Input and Output Statements
- Operators
- Control Structures
- Math & Random Modules
- List
- Tuple
- Strings
- Set
- Dictionary
- Functions
- Files
- Libraries in Python

Activate Windows
Go to Settings to activate W



Introduction to Python

Activate Windows

Go to Settings to activate Win



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

3

- Why Python
- Differences between C, Java & Python
- Features of Python
- History of Python

Activate Windows

Go to Settings to activate Win 10

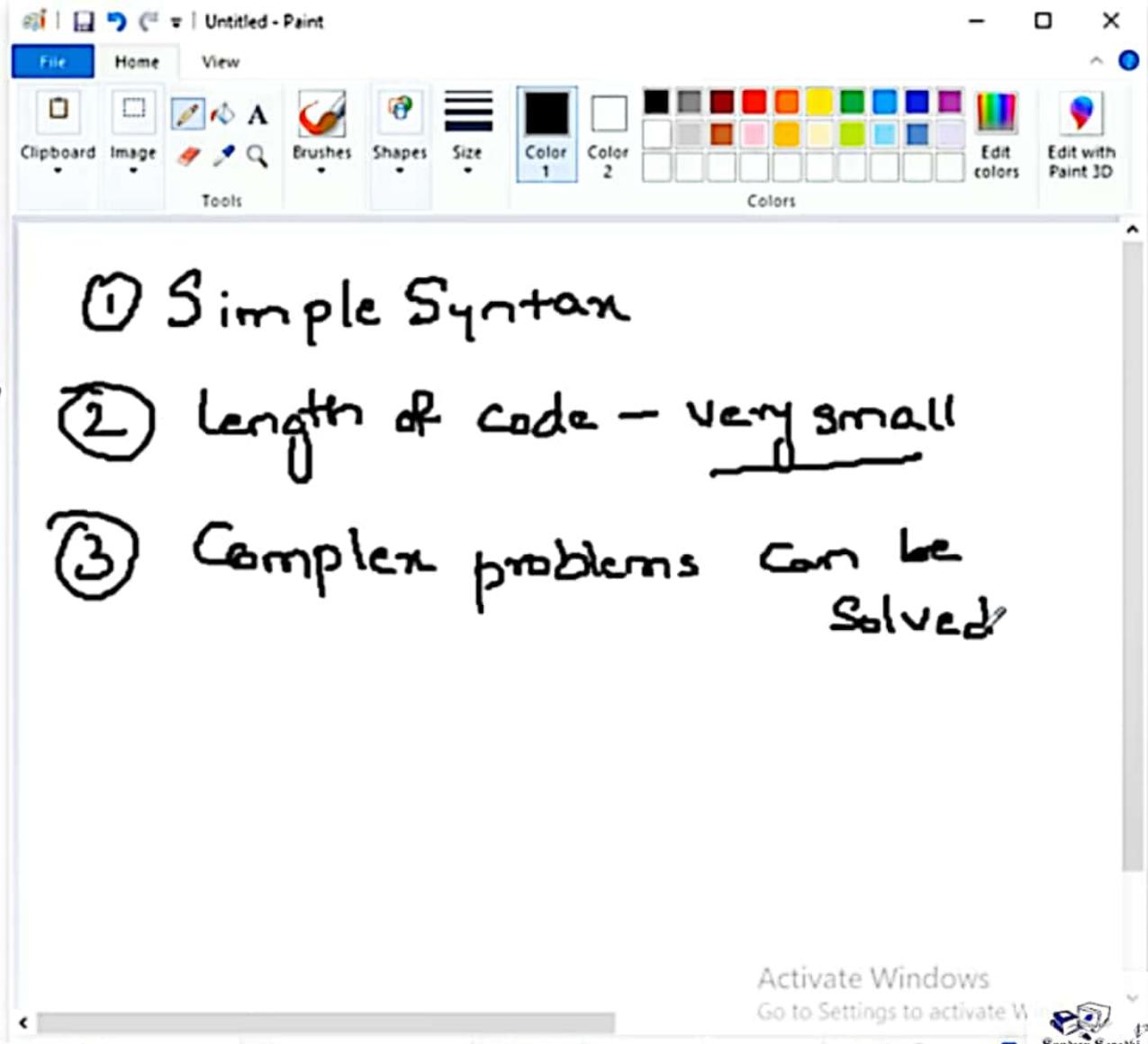


- Why Python

- Differences between C,

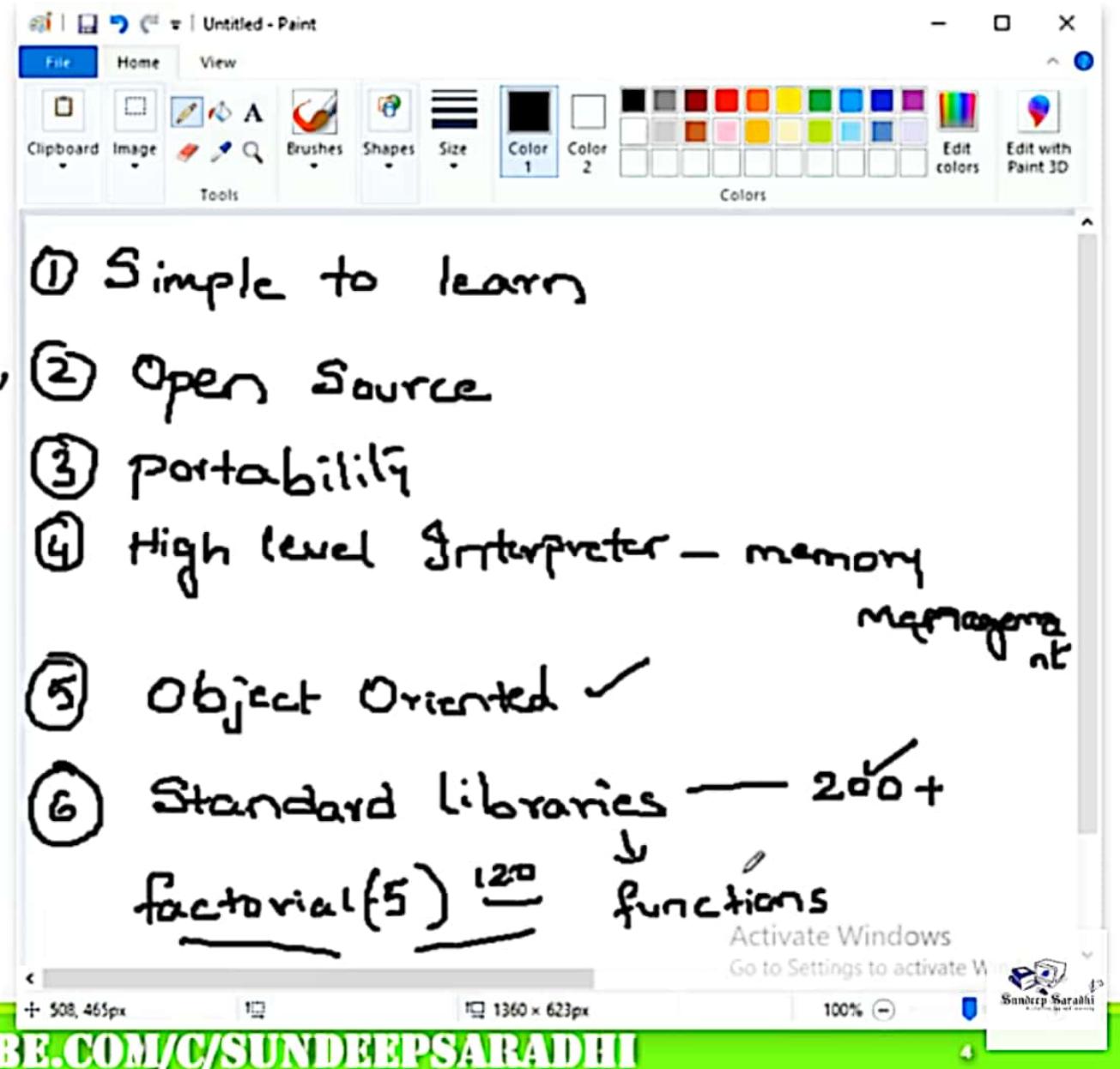
- Features of Python

- History of Python



	C	C++	JAVA	PYTHON
Language Type	Procedure Oriented Language	Object Oriented Language	Object Oriented Language	Both Procedural & Object Oriented Language
Building Block	Function Driven	Object Driven	Both Object and Class Driven	Function, Object and Class Driven
Extension	.c	.cpp	.java	.py
Platform	Dependent	Independent	Independent	Independent
Comment Style	/* */	// Single Line /* */ Multi-Line	// Single Line /* */ Multi-Line	# Single Line """ """ Multi Line
Translation Type	Compiled	Compiled	Compiled & Interpreted	Interpreted
Database Connectivity	not supported	not supported	Supported	Supported
Representing Block of Statements	{}	{}	{}	Indentation
Declaring Variables	Required	Required	Required	Not Required
Applications	Compilers , Interpreters , Embedded Programming etc.	Simple Desktop Applications, Embedded Systems etc.	Desktop GUI, Mobile, Web, Gaming etc.	Desktop, Web, Gaming, Network Programming etc.
IDE	Turbo C, Code Blocks	Turbo C++, Code Blocks	Eclipse , NetBeans etc	PyCharm , PyDev, IDLE, Jupyter Notebook, Spyder etc

- Why Python
- Differences between C, C++, Java
- Features of Python
- History of Python

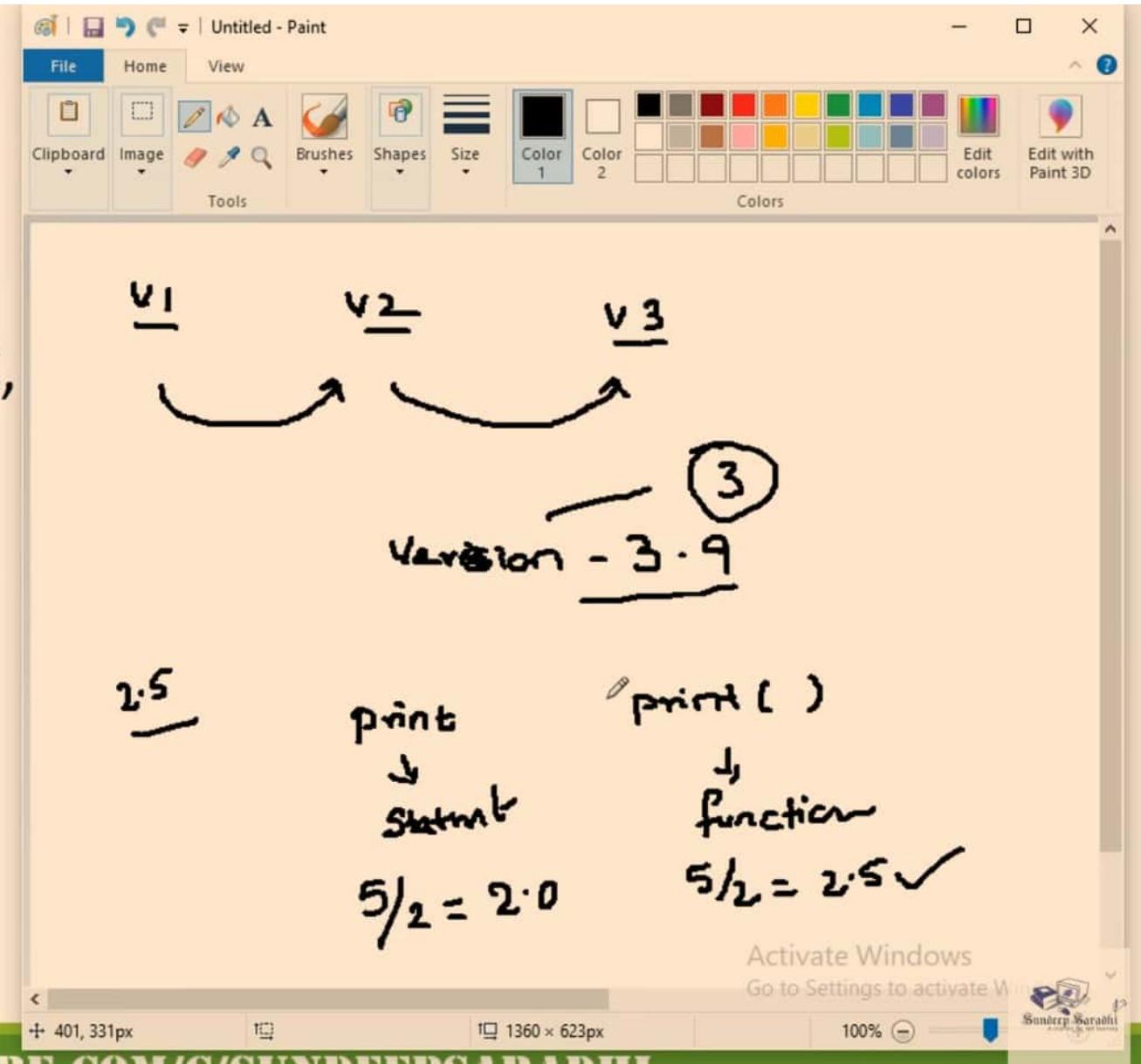


- Why Python

- Differences between C,

- Features of Python

- History of Python



Applications of Python

Activate Windows

Go to Settings to activate Win



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

6

- Web & Internet Development
- Business Applications
- Desktop GUI Applications
- Games and 3D Graphics
- Artificial Intelligence
- Network Programming
- Machine Learning
- Database Access
- Image Processing Applications

Activate Windows
Go to Settings to activate Win 10



Integrated Development Environment

IDEs

Activate Windows

Go to Settings to activate Win



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

8

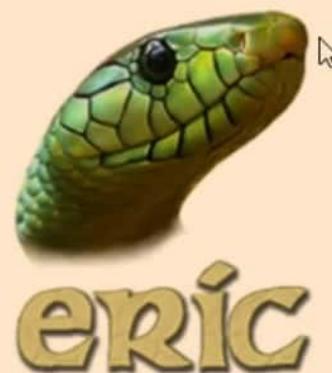


IDLE

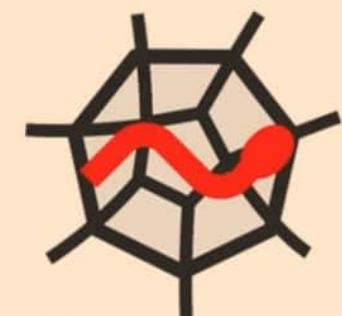
ANACONDA.



PyCharm



ΤΗ



Activate Windows
Go to Settings to activate Windows 10



WWW.YOUTUBE.COM/C/SUNDEEPSARADHI

9

Introduction to Python Interpreter

Activate Windows

Go to Settings to activate Win



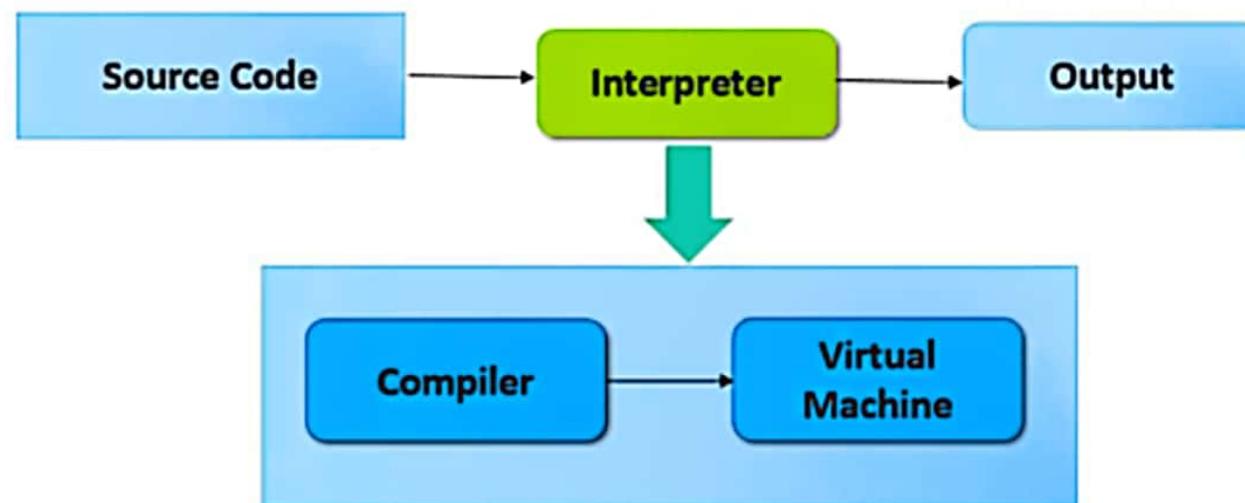
[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

10

COMPILER



INTERPRETER



Activate Windows
Go to Settings to activate Win



Indentation and Comments

Activate Windows

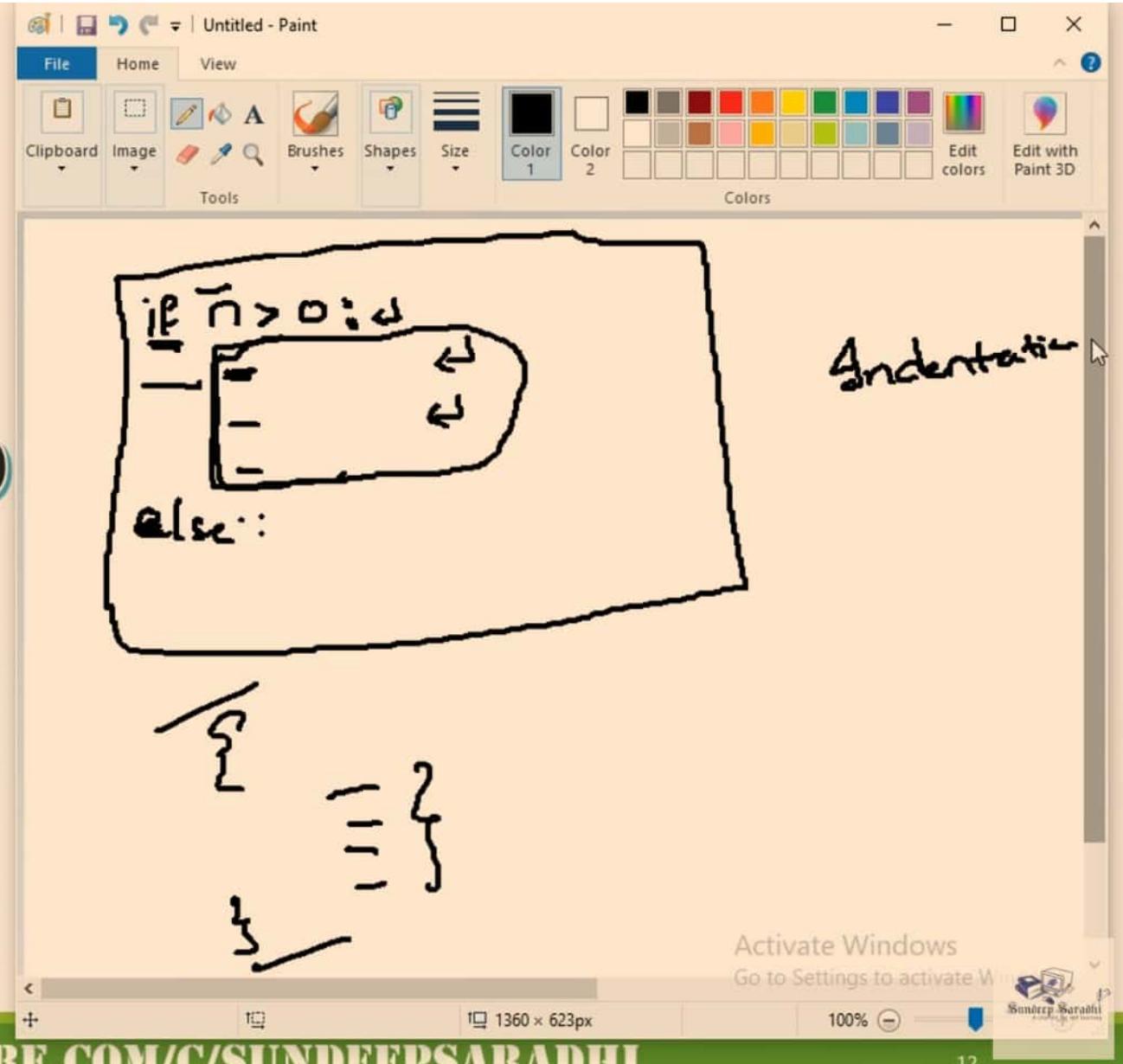
Go to Settings to activate Win



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SundeepSaradhi)

12

Indentation



WWW.YOUTUBE.COM/C/SUNDEEPSARADHI

Indentation

Single →

#

Multi-Line →

" " " "
" " " "

Home Page - Select or create a new notebook Untitled11 - Jupyter Notebook +

localhost:8888/notebooks/Untitled11.ipynb?kernel_name=python3

jupyter Untitled11 Last Checkpoint: 4 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [2]:

```
if 10>0:  
    #This is a comment  
    print("Positive Number")  
    #print("10 is greater")  
else:  
    print("10 is smaller")
```

Positive Number

In []:

In []:

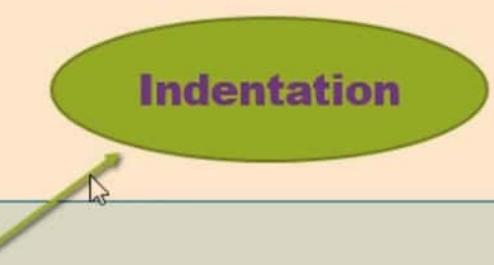
Activate Windows
Go to Settings to activate Windows



- Python requires indentation as a part of syntax.
- Indentation signifies the start and end of block of code.
- Programs will not run without correct indentation.

if n > 0 :
 print("Positive Number")
else :
 print("Negative Number")

Indentation



```
if n > 0  
{  
    print("Positive Number")  
}  
else  
{  
    print("Negative Number")  
}
```

Activate Windows
Go to Settings to activate it!



Single Line Comment :

as prefix to the line

Multi Line Comment :

""" Comment """



or

''' Comment '''

Activate Windows
Go to Settings to activate W



Keywords

Activate Windows

Go to Settings to activate Win



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

15

- **Keywords** are the reserved words which have predefined meaning and functionality.

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

Activate Windows

Go to Settings to activate Win 10



Variables - Identifiers

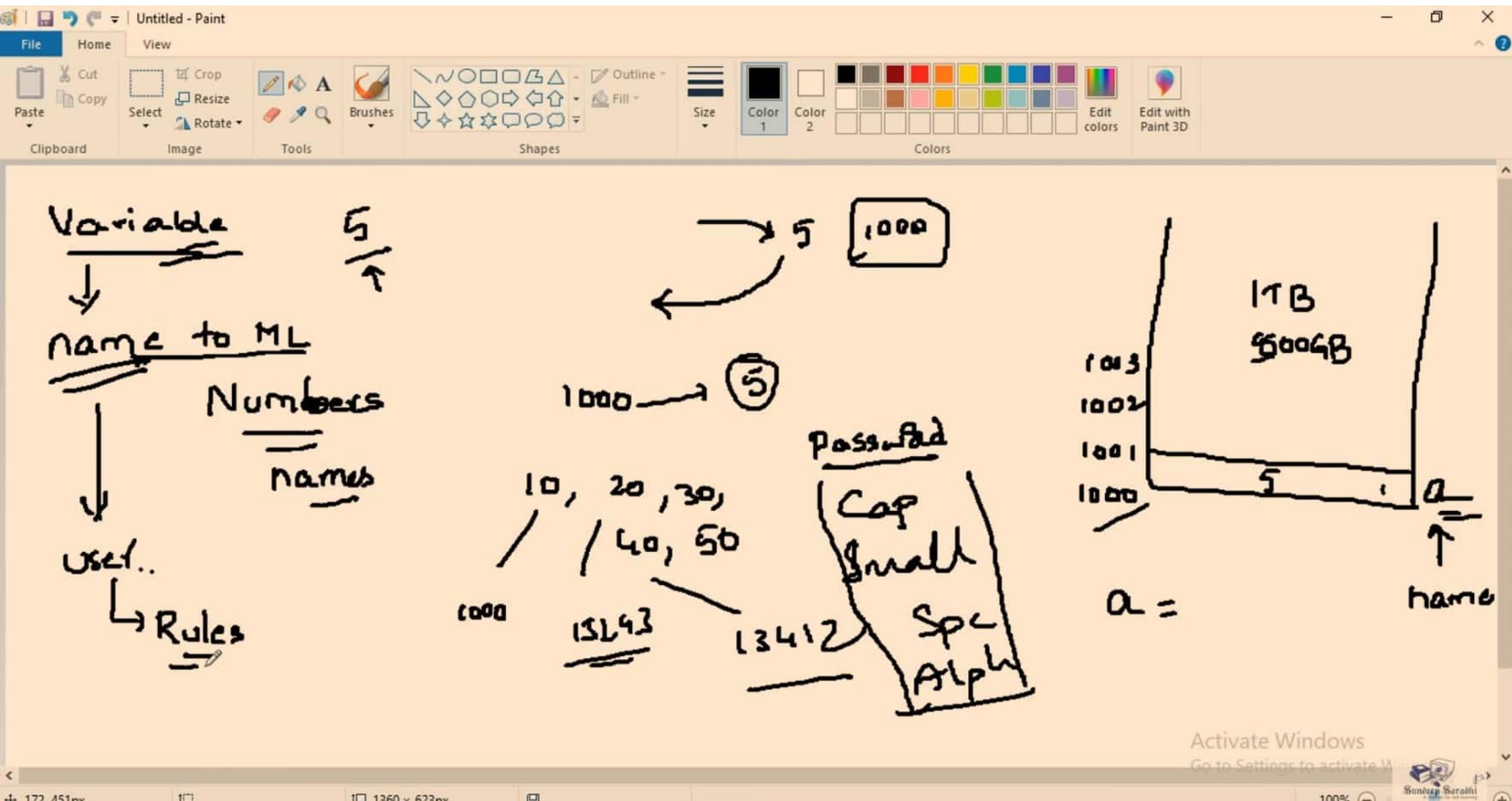
Activate Windows

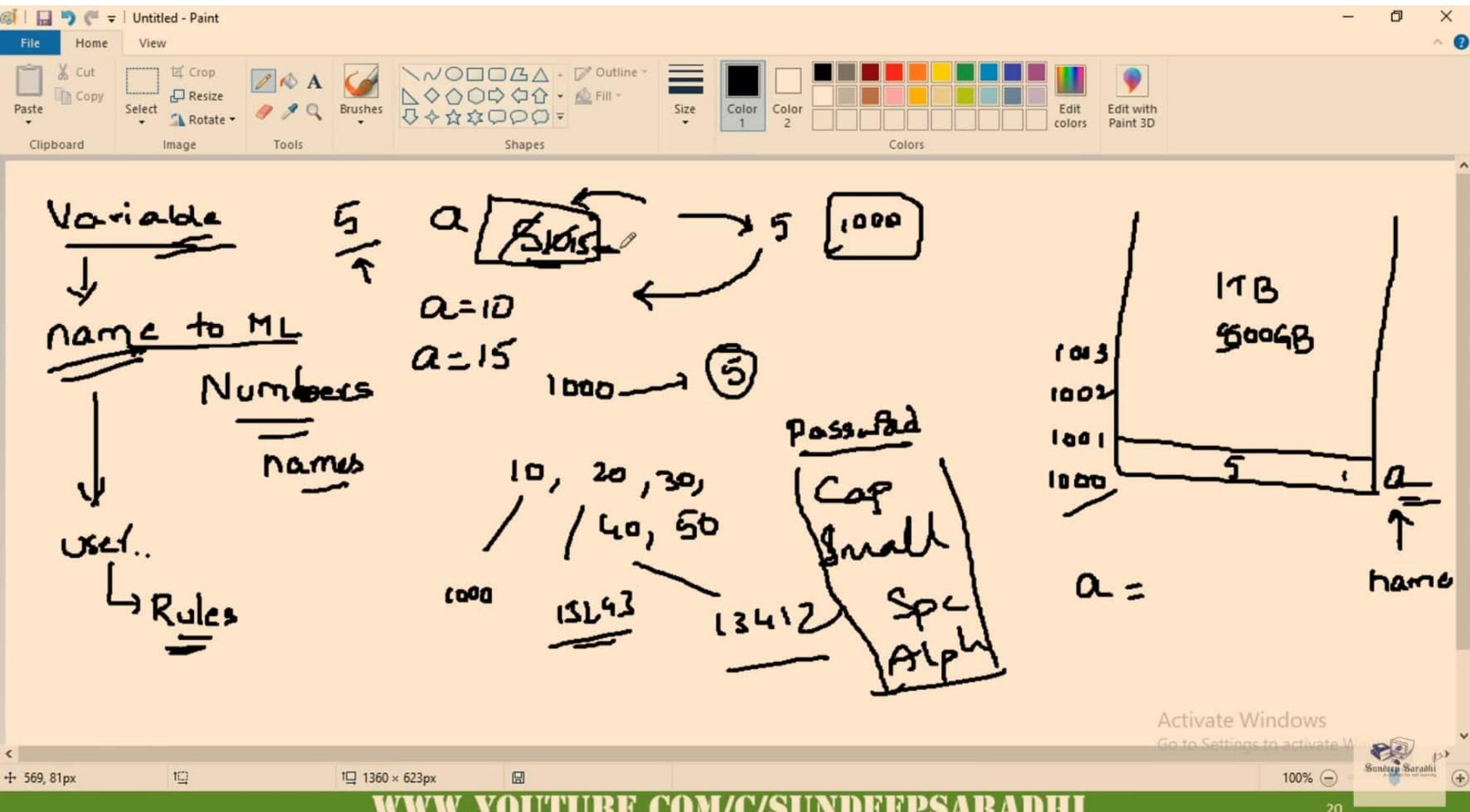
Go to Settings to activate Win 10

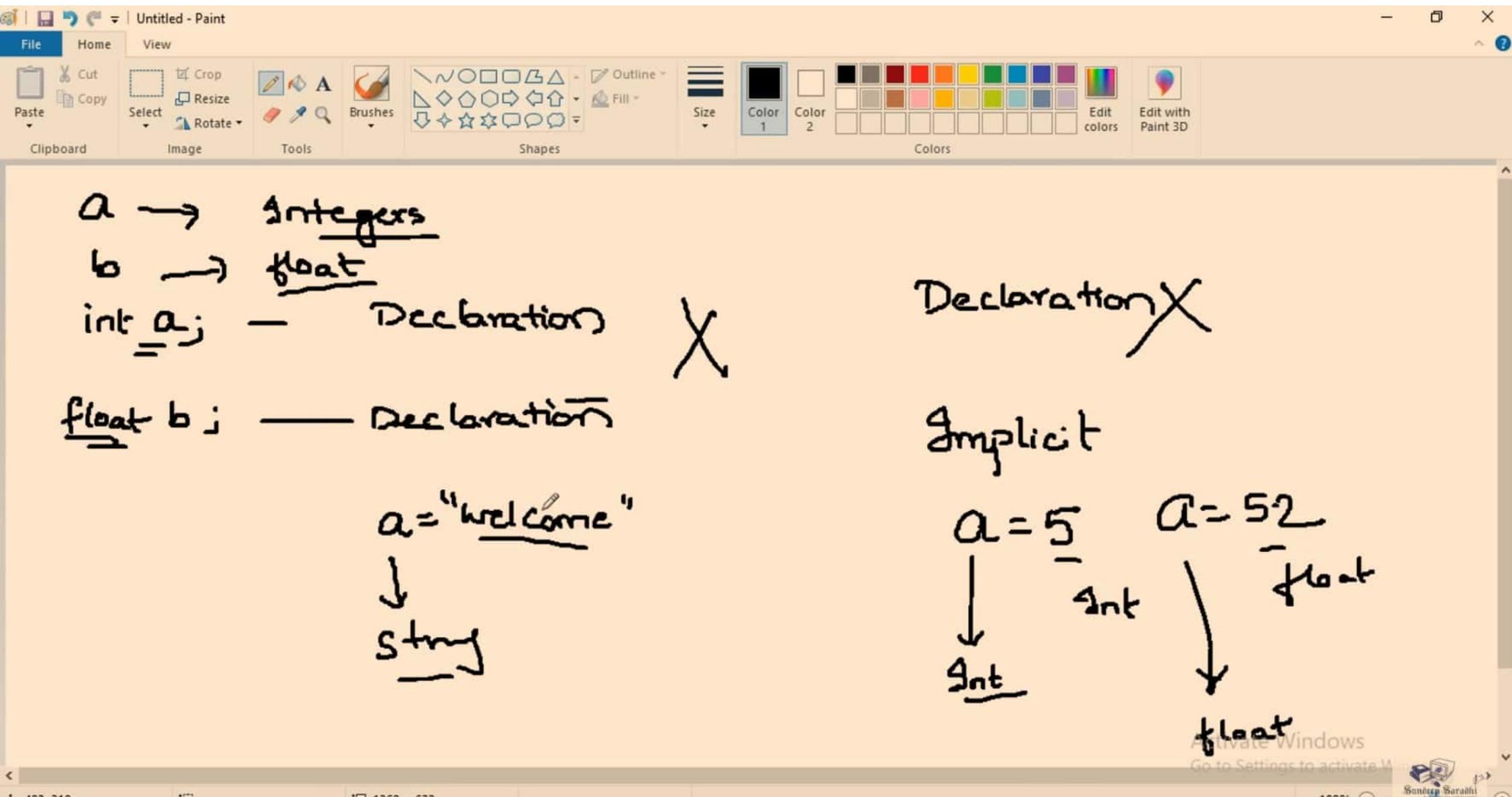


[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

17







Variable Name

- ❖ A Variable is defined as an alternate name for Memory Location and can have a short name or a more descriptive name.

Rules for Python Variables

- ❖ A Variable name must start with a letter or the underscore character

- ❖ A Variable name cannot start with a number
- ❖ A Variable name can only have alpha-numeric characters and underscores (A-z, 0-9, and _)
- ❖ Variable names are case-sensitive. (name, Name and NAME are three different variables.)
- ❖ Variable name should not match with

Activate Windows
Go to Settings to activate W



Built-In Types

Activate Windows

Go to Settings to activate Win 10



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

19

Numeric

- Integer - Whole Numbers Ex: 3, 400, 500
- Complex Number - Numbers with real and imaginary values , Ex: a+bj
- Float - Numbers with decimal point , Ex: 3.14, 4.15

Dictionary

- Unordered Key: Value pairs { key1 : value1, key2 : value2.... }

Boolean

- True or False

Set

- Unordered Collection of unique object , Ex: {10, 20, 30}

Sequence Type

- Strings - Ordered sequence of characters , Ex: "python", "sundeep"
- List - Ordered sequence of object , Ex: [10 , 20 , 30]
- Tuple - Ordered immutable sequence of objects , Ex: (10,20,30)

Go to Settings to activate W



Assigning Values to Variables

Activate Windows

Go to Settings to activate Win



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

21

1) Basic form: This form is the most commonly used form for assignment.

Example: >>> str="Hello"

```
>>> print(str)
```

Hello

2) Tuple Assignment:

```
>>> x,y=(50,100)
```

```
>>> print('x=',x)
```

x= 50

```
>>> print("y=",y)
```

y= 100

3) List Assignment: This works in the same way as the tuple assignment.

```
>>> x,y=[3,5]
```

```
>>> print('x=',x,"y=",y)
```

x= 3 y= 5

Activate Windows

Go to Settings to activate Win



4) Sequence Assignment: Any sequence of names can be assigned to any sequence of values, and Python assigns the items one at a time by position.

```
>>> a,b,c='hai'
```

```
>>> a
```

```
'h'
```

```
>>> b
```

```
'a'
```

```
>>> c
```

```
'i'
```

Activate Windows
Go to Settings to activate W



5) Extended Sequence unpacking: It allows us to be more flexible in how we select portions of a sequence to assign.

```
>>> p, *q="Hello"
```

Here, p is matched with the first character in the string on the right and q with the rest. The starred name (*q) is assigned a list, which collects all items in the sequence not assigned to other names.

```
>>> p
```

```
'H'
```

```
>>> q
```

```
['e', 'l', 'l', 'o']
```

Activate Windows
Go to Settings to activate Win 10



6) Multiple target assignment: In this form, Python assigns a reference to the same object (the object which is rightmost) to all the target on the left.

```
>>> a=b=76
```

```
>>> print(a)
```

```
76
```

```
>>> print(b)
```

```
76
```

7) Augmented assignment: The augmented assignment is a shorthand assignment that combines an expression and an assignment.

```
>>> x=8
```

```
>>> x+=1
```

```
>>> print(x)
```

```
9
```

There are several other augmented assignment forms: -=, **=, &=, etc.

Activate Windows
Go to Settings to activate W



Home Page - Select or create a new notebook Untitled11 - Jupyter Notebook +

localhost:8888/notebooks/Untitled11.ipynb?kernel_name=python3

jupyter Untitled11 Last Checkpoint: 20 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Out[8]: '10'

In [9]: a=5

In [10]: type(a)

Out[10]: int

In [11]: a=5.4

In [12]: type(a)

Out[12]: float

In [13]: a="welcome"

In [14]: type(a)

Out[14]: str

In []:

Activate Windows
Go to Settings to activate Windows



Home Page - Select or create a new notebook Untitled11 - Jupyter Notebook +

localhost:8888/notebooks/Untitled11.ipynb?kernel_name=python3

jupyter Untitled11 Last Checkpoint: 22 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [17]: `print(b)`
20

In [18]: `a,b=[10,20]`

In [19]: `a`
Out[19]: 10

In [20]: `b`
Out[20]: 20

In [21]: `a,b,c=10,20,30`

In [22]: `print(a,b,c)`
10 20 30

In []:

Activate Windows
Go to Settings to activate Windows



Home Page - Select or create a new notebook

Untitled11 - Jupyter Notebook

localhost:8888/notebooks/Untitled11.ipynb?kernel_name=python3

jupyter Untitled11 Last Checkpoint: 23 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [20]: b
Out[20]: 20

In [21]: a,b,c=10,20,30

In [22]: print(a,b,c)
10 20 30

In [23]: p,*q="hello"

In [24]: p
Out[24]: 'h'

In [25]: q
Out[25]: ['e', 'l', 'l', 'o']

In []:

Activate Windows
Go to Settings to activate Windows 10



Home Page - Select or create a new notebook Untitled11 - Jupyter Notebook +

localhost:8888/notebooks/Untitled11.ipynb?kernel_name=python3

jupyter Untitled11 Last Checkpoint: 23 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [20]: b
Out[20]: 20

In [21]: a,b,c=10,20,30

In [22]: print(a,b,c)
10 20 30

In [23]: p,*q="hello"

In [24]: p
Out[24]: 'h'

In [25]: q
Out[25]: ['e', 'l', 'l', 'o']

In []:

Activate Windows
Go to Settings to activate Windows



Home Page - Select or create a new notebook

Untitled11 - Jupyter Notebook

localhost:8888/notebooks/Untitled11.ipynb?kernel_name=python3

jupyter Untitled11 Last Checkpoint: 24 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Out[25]: ['e', 'l', 'l', 'o']

In [26]: a=b=c=d=0

In [27]: a

Out[27]: 0

In [28]: b

Out[28]: 0

In [29]: c

Out[29]: 0

In [30]: d

Out[30]: 0

In []:

Activate Windows
Go to Settings to activate Windows.

localhost:8888/notebooks/Untitled11.ipynb?kernel_name=python3#

WWW.YOUTUBE.COM/C/SUNDEEPSARADHI

25

Home Page - Select or create a new notebook

Untitled11 - Jupyter Notebook

localhost:8888/notebooks/Untitled11.ipynb?kernel_name=python3

jupyter Untitled11 Last Checkpoint: 25 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

In [30]: d

Out[30]: 0

In [31]: a=5

In [32]: a=a+5

In [33]: a

Out[33]: 10

In [34]: a+=5

In [35]: a

Out[35]: 15

In []:

Activate Windows
Go to Settings to activate Windows



Input and Output Functions

Activate Windows

Go to Settings to activate Win



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

26

Input ()

- The input() function is used to read data from keyboard.
- It reads data into string format.
- Programmer have to convert data into specific format before using them.
- To convert use the int, float and str functions
 - Int(2.3) returns 2, int("123") returns 123
 - Float(6) returns 6.0, float("3.14") returns 3.14
 - Str(123) returns "123" , str(4.5) returns "4.5"
- If the conversion isn't possible, there will be an error.

Example : int("abc"), int("two")

Activate Windows
Go to Settings to activate W



Example-1 : READING SINGLE INPUT WITHOUT REQUEST MESSAGE

```
a=input()
```

Input: 10

Example-2 : READING SINGLE INPUT WITH REQUEST MESSAGE

```
a=input("Enter the value of a:")
```

Input: Enter the value of a:

10

Example-3 : READING MULTIPLE INPUTS IN A SINGLE LINE USING split()

```
a=input().split(" ")
```

Input : 10 20 30 40 50

Activate Windows
Go to Settings to activate W



print()

The print() function prints the specified text or value to the screen.

Example-1: print("Hello World")

Output : Hello World

Example-2: ADD A NEW LINE OR VERTICAL SPACE BETWEEN TWO OUTPUTS

```
print("Hello!")
```

```
print("Welcome to Python Programming")
```

Output: Hello!

Welcome to Python Programming

Activate Windows
Go to Settings to activate W



Example-3: OPTION KEYWORD ARGUMENT “sep=“

```
print("Welcome","To","Python","Programming")
```

Output: Welcome To Python Programming

```
print ("Welcome","To","Python","Programming",sep=" \n ")
```

Output : Welcome

To

Python

Programming

```
print ("Welcome","To","Python","Programming",sep=" , ")
```

Output : Welcome,TO,Python,Programming

Activate Windows
Go to Settings to activate W



Example-4 : USING KEYWORD ARGUMENT “end=“

- “end=“ is a string appended after the last value, defaults to a new line.
- It allows the programmer to define a custom ending character for each print call other than the default newline or \n.

```
print("Hello!",end=" ")  
print("Welcome to python programming")
```

Output: Hello! Welcome to python programming

Example-5: USING FORMAT SPECIFIERS

a=10

b=2.5

```
print("%d is an integer and %f is a float."%(a,b))
```

Output: 10 is an integer and 2.5 is a float.

Activate Windows
Go to Settings to activate W



Example-6: WITHOUT USING FORMAT SPECIFIERS

a=10

b=2.5

```
print(a,"is an integer and",b,"is a float.")
```

Output : 10 is an integer and 2.5 is a float.

Activate Windows
Go to Settings to activate Win 10



Operators

Activate Windows

Go to Settings to activate Win



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsAradhi)

33

Arithmetic Operators	- + , - , * , / , // , % , **
Relational Operators	- < , > , <= , >= , == , !=
Assignment Operators	- = , += , -= , *= , /= , %= , //= etc.,
Bitwise Operators	- & , , ^ , ~ , << , >>
Logical Operators	- and , or , not
Membership Operators	- in , not in
Identity Operators	- is , is not

Activate Windows
Go to Settings to activate W



Control Structures

Activate Windows

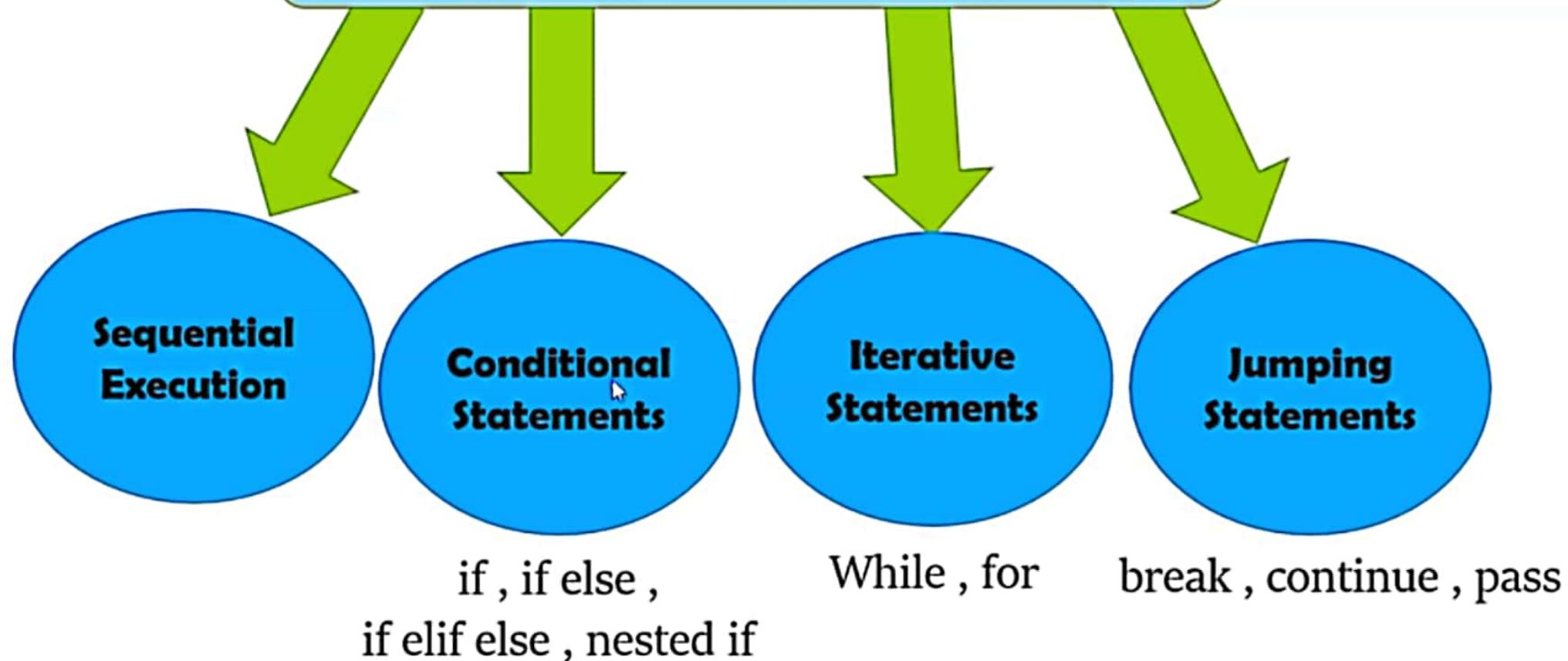
Go to Settings to activate Win



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

35

Control Structures



Activate Windows

Go to Settings to activate W



➤ Control Structures

□ Conditional Statements

- if
- if – else
- Nested If – else

□ Python Loops

- while
- for
- Nested loops

□ Jumping Statements

- continue
- break
- pass

Activate Windows
Go to Settings to activate W





Control Structures

Flow of Control

① Sequential — line by line

② Conditional — Based on Condition →
→ Boolean

③ Iterative / Loops — Group of stmt
executing repeatedly ← T/F ↓

④ Jumping ↗
Based condition



5 if $a >= 0:$ →
 print(" +ve Number")
 else:
 print(" -ve Number")

Activate Windows
Go to Settings to activate ↗



```
In [7]: a=int(input("Enter a:"))
b=int(input("Enter b:"))
if a>b:
    print("a is greater than b")
elif a<b:
    print("a is smaller than b")
else:
    print("Both are equal")

Enter a:5
Enter b:5
Both are equal
```

The screenshot shows a Jupyter Notebook interface with the following details:

- Header:** Home Page - Select or create a notebook, Untitled12 - Jupyter Notebook, Logout.
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Run, Cell Type (Code), Cell Number (1).
- Code Cell (In [10]):**

```
a=int(input("Enter a:"))
b=int(input("Enter b:"))
c=int(input("Enter c:"))
if a>b:
    if a>c:
        print("a is Big")
    else:
        print("c is Big")
else:
    if b>c:
        print("b is Big")
    else:
        print("c is Big")
```
- Output:** Enter a:4
Enter b:5
Enter c:3
b is Big
- Input Cell (In []):** An empty input field for the next cell.
- Bottom Right:** Activate Windows watermark.



Iterative

for loop
while loop

Condition - True

= } Continuously executed ←

generation

5 generations

false - stop

- ① loop variable initialize
- ② condition → Terminate
- ③ loop variable update

while condition:



Activate Windows
Go to Settings to activate



Home Page - Select or create a new notebook Untitled12 - Jupyter Notebook +

← → C ⓘ localhost:8888/notebooks/Untitled12.ipynb?kernel_name=python3

jupyter Untitled12 Last Checkpoint: 16 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [12]: `print("1")
print("2")
print("3")
print("4")
print("5")`

1
2
3
4
5

In [13]: `n=int(input("Enter n value"))
i=1 # loop variable
while i<n:
 print(i)
 i=i+1`

Enter n value5
0
1
2
3
4
5

In []:

Activate Windows
Go to Settings to activate Windows 10
Sundeep Saradhi

Home Page - Select or create a new notebook Untitled12 - Jupyter Notebook +

← → C ⓘ localhost:8888/notebooks/Untitled12.ipynb?kernel_name=python3

jupyter Untitled12 Last Checkpoint: 17 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [12]: `print("1")
print("2")
print("3")
print("4")
print("5")`

1
2
3
4
5

In [16]: `n=int(input("Enter n value"))
i=1 # loop variable
while i<=n:
 print(i,end=" ")
 i=i+1`

Enter n value100
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

In []:

Activate Windows
Go to Settings to activate Windows 10





for seq
range()

range(start , stop , step)

i in range(0, 6, 1)

$i = 0 \quad i = 1 \quad i = 2 \quad i = 3 \quad i = 4 \quad i = 5$

Loop Variable \leftarrow Start \rightarrow Starting value \leftarrow 0 \leftarrow include
Condition \leftarrow Stop \rightarrow End value \leftarrow mandatory - exclude
Updation \leftarrow Step \rightarrow difference \leftarrow 1 ✓

Activate Windows
Go to Settings to activate

1024x461px

1360x623px

100%



Home Page - Select or create a new notebook Untitled12 - Jupyter Notebook +

← → C ⓘ localhost:8888/notebooks/Untitled12.ipynb?kernel_name=python3

jupyter Untitled12 Last Checkpoint: 23 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [18]: `for i in range(0,6,1):
 print(i)`

0
1
2
3
4
5

In [20]: `for i in range(6):
 print(i)`

0
1
2
3
4
5

In []: `for i in range(1,10,2):
 print(i)`

Activate Windows
Go to Settings to activate Windows 10



Home Page - Select or create a new notebook

Untitled12 - Jupyter Notebook

localhost:8888/notebooks/Untitled12.ipynb?kernel_name=python3

jupyter Untitled12 Last Checkpoint: 23 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

In [20]: `for i in range(6):
 print(i)`

0
1
2
3
4
5

In [23]: `for i in range(1,10,3):
 print(i)`

1
4
7

In []:

Activate Windows
Go to Settings to activate Windows 10



5) Extended Sequence unpacking: It allows us to be more flexible in how we select portions of a sequence to assign.

```
>>> p, *q="Hello"
```

Here, p is matched with the first character in the string on the right and q with the rest. The starred name (*q) is assigned a list, which collects all items in the sequence not assigned to other names.

```
>>> p
```

```
'H'
```

```
>>> q
```

```
['e', 'l', 'l', 'o']
```

Activate Windows
Go to Settings to activate Win 10



Numeric

- Integer - Whole Numbers Ex: 3, 400, 500
- Complex Number - Numbers with real and imaginary values , Ex: $a+bi$
- Float - Numbers with decimal point , Ex: 3.14, 4.15

Dictionary

- Unordered Key: Value pairs { key1 : value1, key2 : value2.... }

Boolean

- True or False

Set

- Unordered Collection of unique object , Ex: {10, 20, 30}

Sequence Type

- Strings - Ordered sequence of characters , Ex: "python", "sundeep"
- List - Ordered sequence of object , Ex: [10 , 20 , 30]
- Tuple - Ordered immutable sequence of objects , Ex: (10, 20, 30)

Go to Settings to activate W



Home Page - Select or create a new notebook Untitled12 - Jupyter Notebook +

localhost:8888/notebooks/Untitled12.ipynb?kernel_name=python3

jupyter Untitled12 Last Checkpoint: 26 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

4
5

In [25]: `for i in range(1,100,3):
 print(i,end=" ")`

1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73 76 79 82 85 88 91 94 97

In [26]: `str="PYTHON"
for letter in str:
 print(letter)`

P
Y
T
H
O
N

In [27]: `l=[10,20,30,40,50]`

In []: `for ele in l:
 print(ele)`

Activate Windows
Go to Settings to activate Windows 10



Home Page - Select or create a new notebook Untitled12 - Jupyter Notebook +

← → C ⓘ localhost:8888/notebooks/Untitled12.ipynb?kernel_name=python3

jupyter Untitled12 Last Checkpoint: 39 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [28]: `for ele in l:
 print(ele)`

10
20
30
40
50

In [43]: `for i in range(5): # i = 0 to 4
 for j in range(5): # j = 0 to 4
 print("*",end=" ")
 print(" ")`

In []: `for i in range(10):
 if i==5:
 break
 else:
 print(i)`

Activate Windows
Go to Settings to activate Windows 10
Sundeep Saradhi

Math & Random Modules

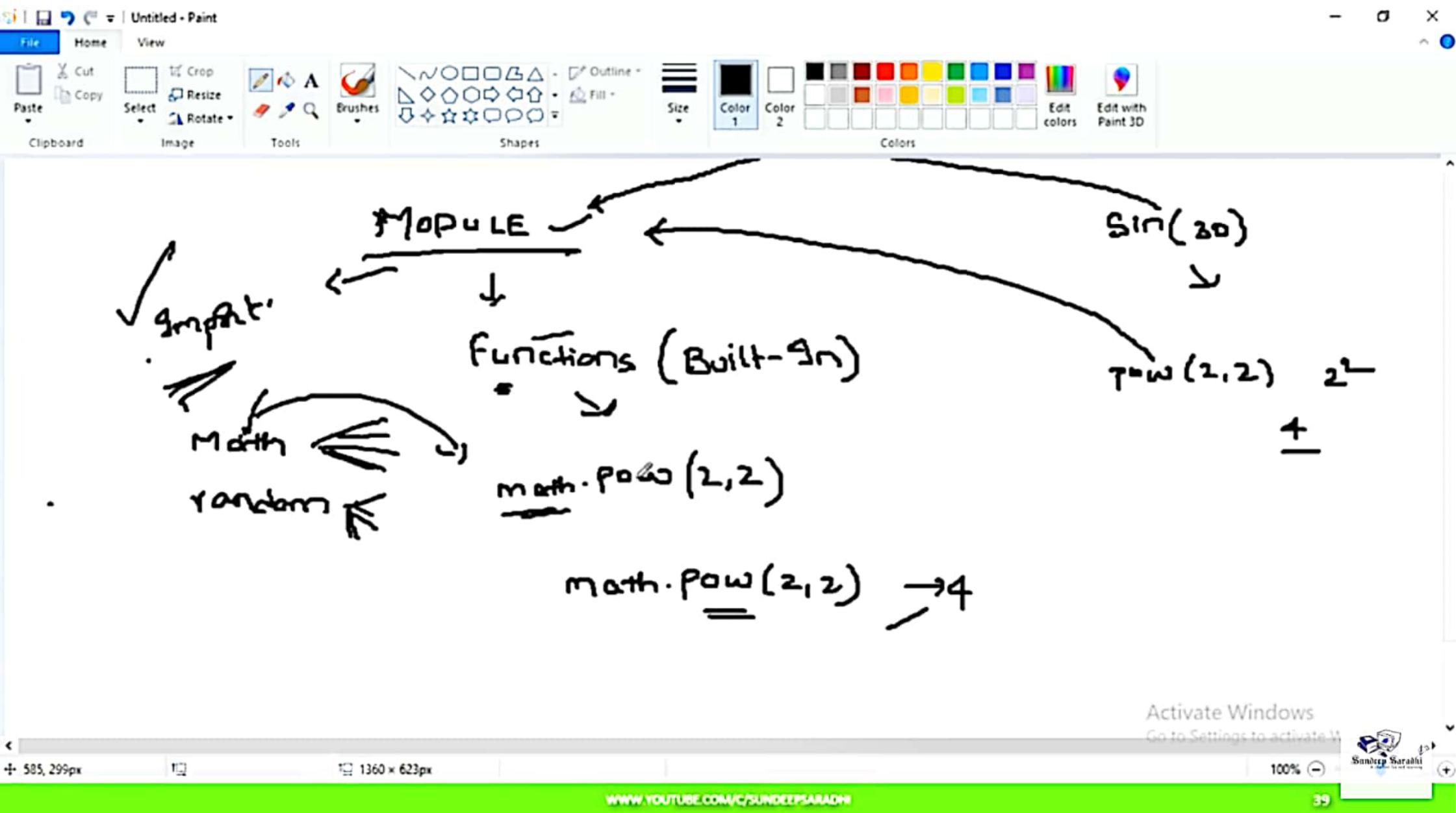
Activate Windows

Go to Settings to activate Win



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

38



Mathematical Functions

- ceil(x)
- copysign(x,y)
- fabs(x)
- factorial(x)
- floor(x)
- fsum(iterable)
- gcd(x,y)
- pow(x,y)
- sqrt(x)
- sin(x)
- cos(x)
- tan(x)
- pi
- e
- tau
- inf
- nan

Activate Windows
Go to Settings to activate W



Random Functions

- choice(seq)
- randrange(start , stop , step)
- random()
- shuffle(list)
- uniform(x , y)

Activate Windows
Go to Settings to activate W



Mathematical Functions

- ceil(x)
- copysign(x,y)
- fabs(x)
- factorial(x)
- floor(x)
- fsum(iterable)
- gcd(x,y)
- pow(x,y)
- sqrt(x)

The screenshot shows a Jupyter Notebook interface with the title "jupyter Untitled14". The notebook contains the following code cells:

- In [5]: `math.ceil(10.2)`
Out[5]: `11`
- In [6]: `math.floor(10.9)`
Out[6]: `10`
- In [7]: `a=10
b=-20`
- In [8]: `math.copysign(a,b)`
Out[8]: `-10.0`
- In [9]: `math(fabs(b))`
Out[9]: `20.0`
- In []: `math.factor`

At the bottom right, there is an "Activate Windows" watermark and a "Sundeep Saradhi" logo.

Random

- choice(seq)
- randrange(start , stop , step)
- random()
- shuffle(list)
- uniform(x , y)

The screenshot shows a Jupyter Notebook interface with the title "jupyter Untitled14". The notebook has several cells:

- In [25]: a
Out[25]: nan
- In [26]: import random
- In [27]: l=[10,20,30,40]
- In [28]: random.choice(l)
Out[28]: 10
- In [29]: random.choice(l)
Out[29]: 20
- In [30]: random.choice(l)
Out[30]: 40
- In []: random.cho

Random

- choice(seq)
- randrange(start , stop , step)
- random()
- shuffle(list)
- uniform(x , y)

The screenshot shows a Jupyter Notebook interface with several code cells and their outputs:

- In [30]: `random.choice(1)`
Out[30]: 40
- In [34]: `random.choices(1,k=3)`
Out[34]: [40, 20, 10]
- In [33]: `random.choices(1,k=2)`
Out[33]: [10, 40]
- In [35]: `random.randrange(10,15,1)`
Out[35]: 12
- In [36]: `random.randrange(10,15,1)`
Out[36]: 12
- In []: `random.randrange(10,15,1)` (This cell is currently active, indicated by a green border.)

Random

- choice(seq)
- randrange(start , stop , step)
- random()
- shuffle(list)
- uniform(x , y)

The screenshot shows a Jupyter Notebook interface with several code cells and their outputs:

- In [40]: `random.randrange(10,15,1)`
Out[40]: 13
- In [41]: `random.randrange(10,15,1)`
Out[41]: 14
- In [42]: `random.random()`
Out[42]: 0.10662315406577016
- In [43]: `random.random()`
Out[43]: 0.39169087298519945
- In [44]: `1`
Out[44]: [10, 20, 30, 40]
- In []: `r` I

Activate Windows
Go to Settings to activate W



Random

- choice(seq)
- randrange(start , stop , step)
- random()
- shuffle(list)
- uniform(x , y)

The screenshot shows a Jupyter Notebook interface with two tabs: 'Untitled14 - Jupyter Notebook' and 'Home Page - Select or create a notebook'. The main area displays a series of code cells and their outputs:

- In [37]: `random.randrange(10,15,1)`
Out[37]: 11
- In [38]: `random.randrange(10,15,1)`
Out[38]: 12
- In [39]: `random.randrange(10,15,1)`
Out[39]: 12
- In [40]: `random.randrange(10,15,1)`
Out[40]: 13
- In [41]: `random.randrange(10,15,1)`
Out[41]: 14
- In []: `ran` █

Random

- choice(seq)
- randrange(start , stop , step)
- random()
- shuffle(list)
- uniform(x , y)

The screenshot shows a Jupyter Notebook interface with the title "Untitled14 - Jupyter Notebook". The notebook contains several code cells demonstrating the use of the `random` module:

- In [41]: `random.random()`
Out[41]: 14
- In [42]: `random.random()`
Out[42]: 0.10662315406577016
- In [43]: `random.random()`
Out[43]: 0.39169087298519945
- In [44]: `random.shuffle([1, 2, 3, 4])`
Out[44]: [1, 2, 3, 4]
- In [45]: `random.shuffle([1, 2, 3, 4])`
Out[45]: [3, 2, 4, 1]
- In [46]: `random.shuffle([1, 2, 3, 4])`
Out[46]: [3, 2, 4, 1]
- In []:

Random

- choice(seq)
- randrange(start , stop , step)
- random()
- shuffle(list)
- uniform(x , y) ↗

The screenshot shows a Jupyter Notebook interface with several code cells and their outputs:

- In [45]: `random.shuffle(l)`
- Out[45]: `[30, 20, 40, 10]`
- In [46]: `l`
- Out[46]: `[30, 20, 40, 10]`
- In [47]: `random.shuffle(l)`
- Out[47]: `[20, 10, 30, 40]`
- In [48]: `l`
- Out[48]: `[20, 10, 30, 40]`
- In [49]: `random.uniform(10,15)`
- Out[49]: `13.546007919376382`
- In [50]: `random.uniform(10,15)`
- Out[50]: `11.597144893422948`

The notebook title is "Untitled14 - Jupyter Notebook" and the URL is "localhost:8888/notebooks/Untitled14.ipynb?kernel_name=python3". The Python version is "Python 3".

Lists

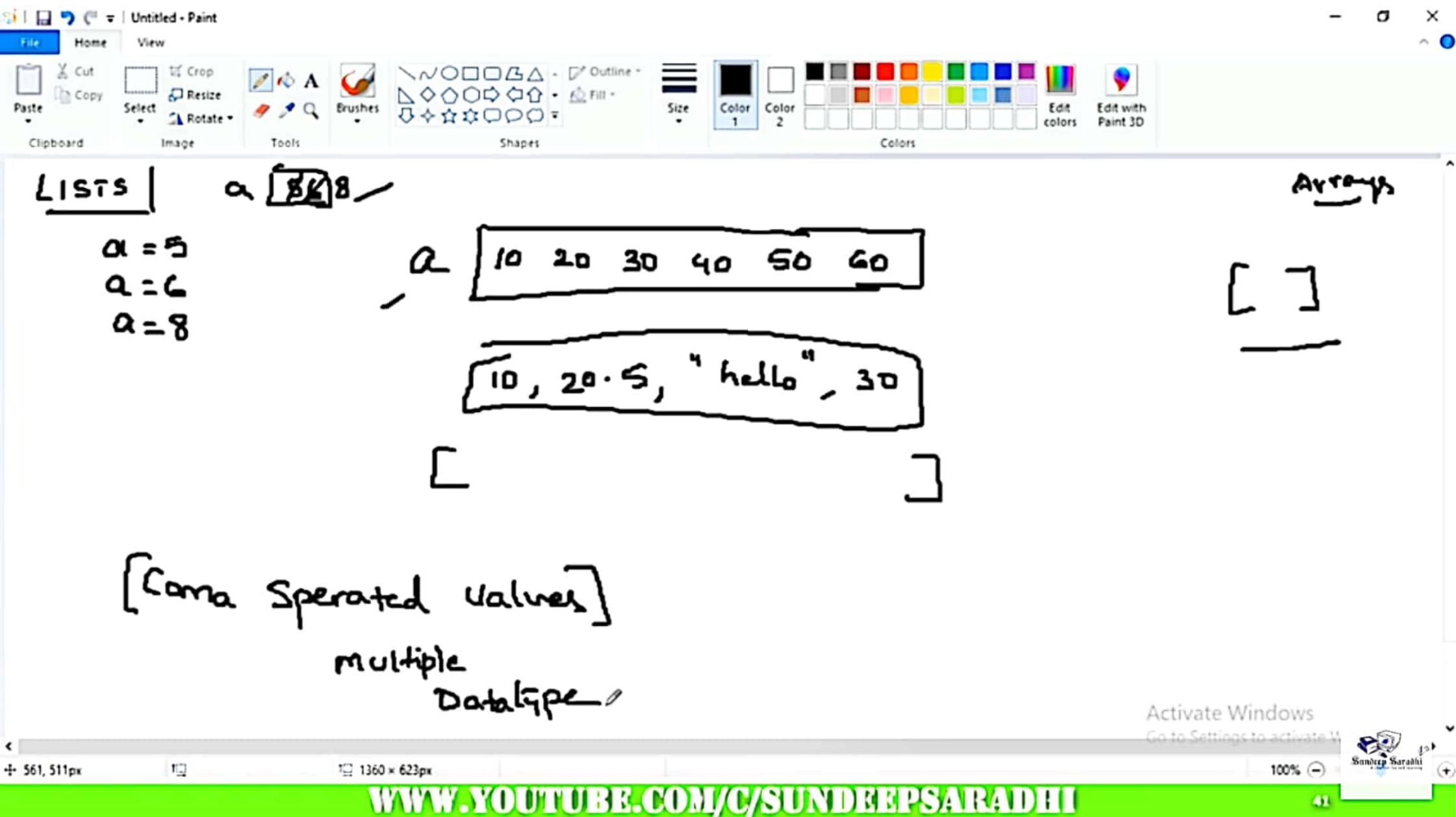
Activate Windows

Go to Settings to activate Win



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

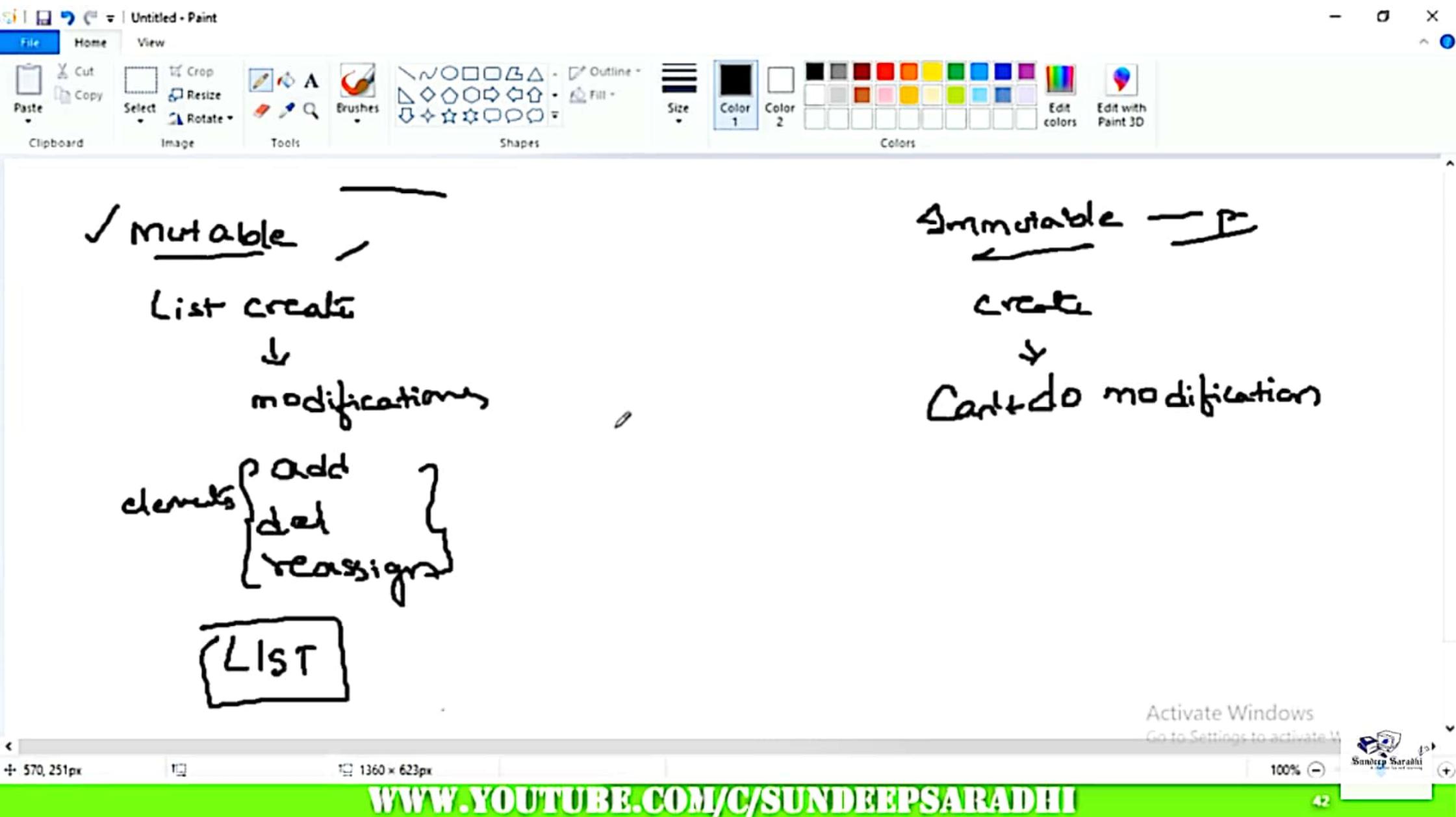
41

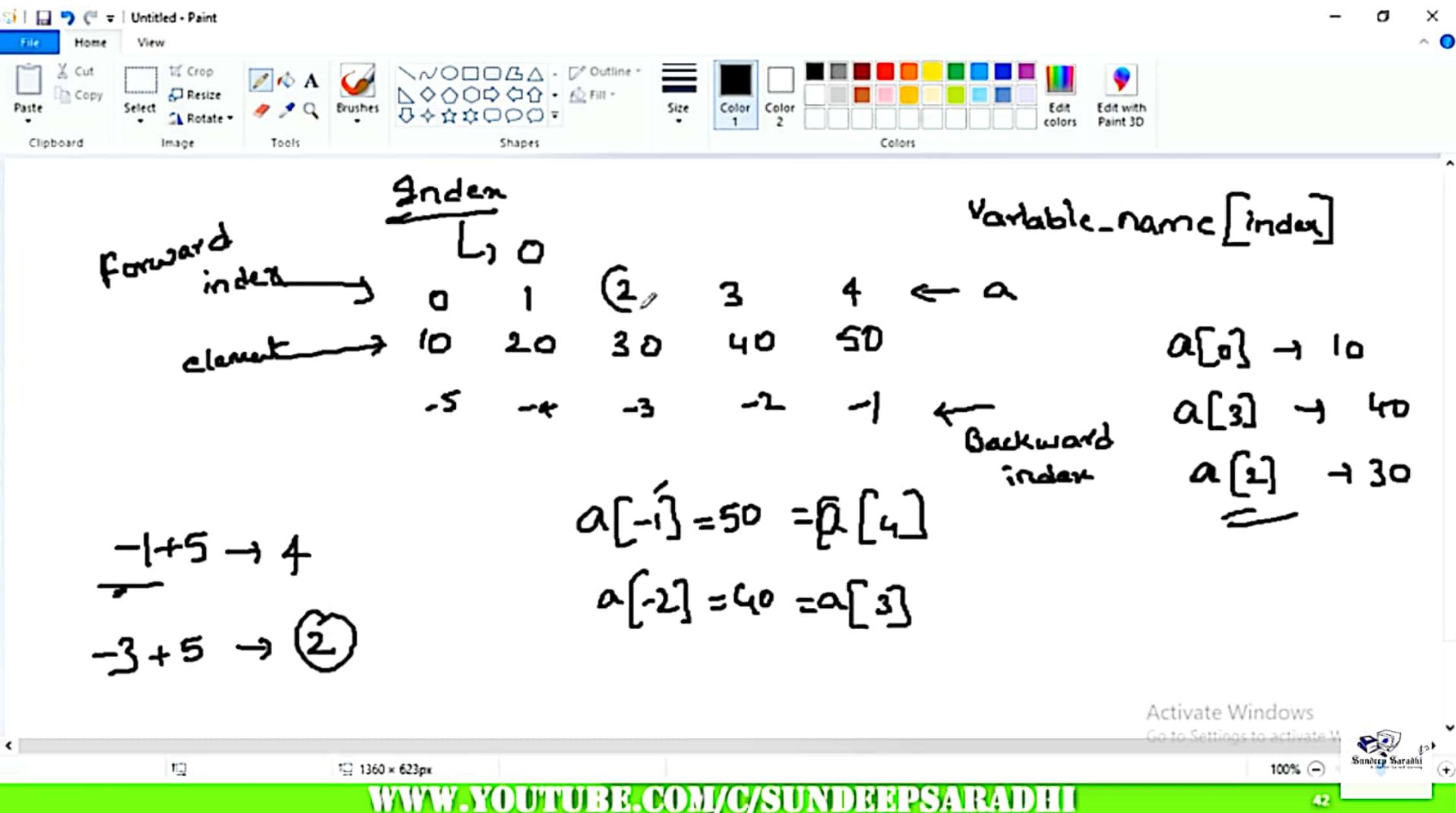


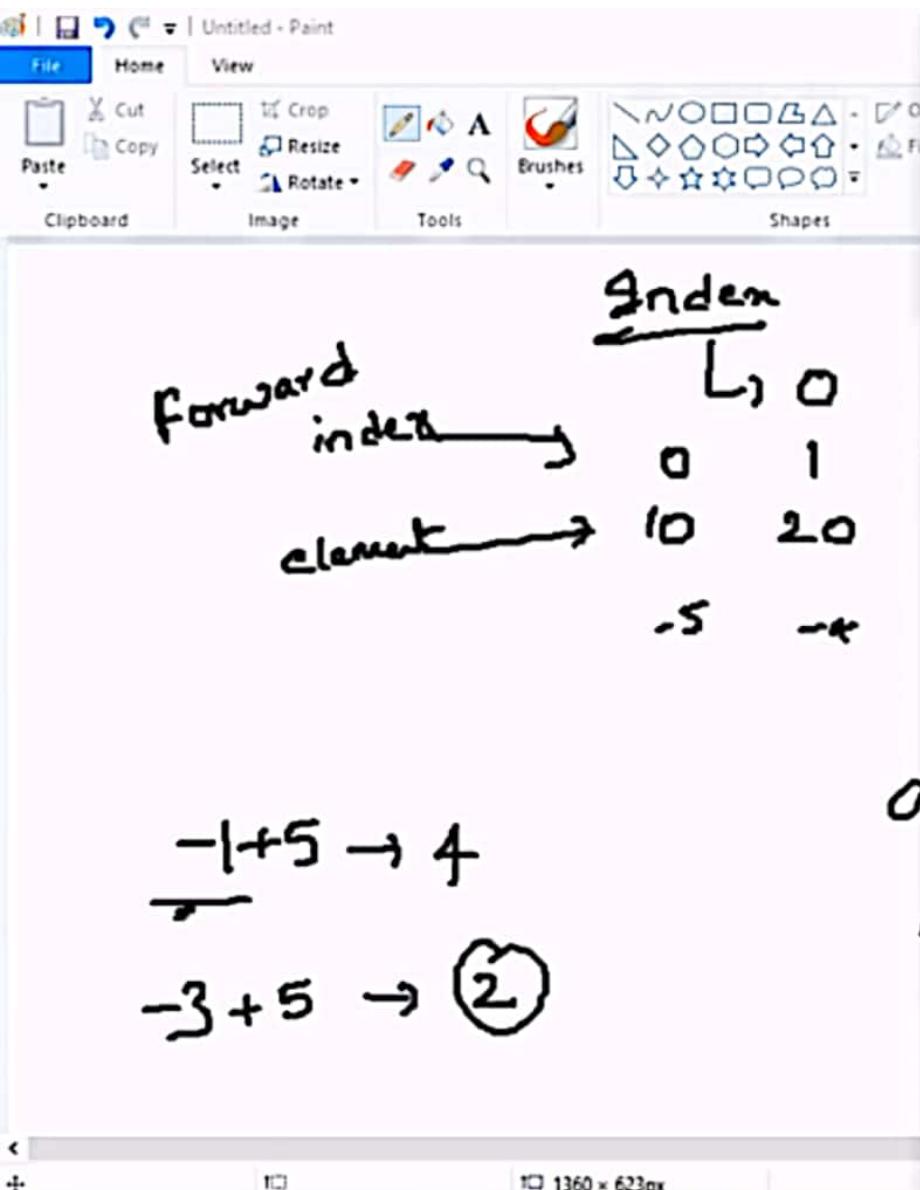
- Creating Lists
- Accessing elements from Lists
- Slicing
- Reassigning List elements
- Deleting elements
- Multidimensional Lists
- Basic Operations
 - + , * , len , min , max , sum , membership , iterations
- List Comprehension
- Built-In Methods

append , extend , insert , remove , pop , sort , count , index, reverse









Untitled14 - Jupyter Notebook | Home Page - Select or create a new notebook | +

localhost:8888/notebooks/Untitled14.ipynb?kernel_name=python3

jupyter Untitled14 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [58]: a

Out[58]: [10, 20.4, 'hello']

In [59]: a[0]

Out[59]: 10

In [60]: a[2]

Out[60]: 'hello'

In [61]: a[-1]

Out[61]: 'hello'

In [62]: a[-2]

Out[62]: 20.4

In []:

Activate Windows

Go to Settings to activate Windows 10





Activate Windows
Go to Settings to activate

- Creating Lists
- Accessing elements from Lists
- Slicing
- Reassigning List elements
- Deleting elements
- Multidimensional Lists
- Basic Operations
 - + , * , len , min , max
- List Comprehension
- Built-In Methods
 - append , extend , insert

The screenshot shows a Jupyter Notebook interface with the title "jupyter Untitled14 (unsaved changes)". The notebook contains the following code and output:

```
In [85]: a=[[10,20],[30,40]],[[50,60],[70,80]]  
Out[84]: 60  
In [86]: a[0]  
Out[86]: [[10, 20], [30, 40]]  
In [87]: a[1]  
Out[87]: [[50, 60], [70, 80]]  
In [88]: a[0][1]  
Out[88]: [30, 40]  
In [89]: a[0][0]  
Out[89]: [10, 20]  
In [ ]: a[0][0]
```

- Creating Lists
- Accessing elements from Lists
- Slicing
- Reassigning List elements
- Deleting elements
- Multidimensional Lists
- Basic Operations
 - + , * , len , min , max
- List Comprehension
- Built-In Methods
 - append , extend , insert

The screenshot shows a Jupyter Notebook interface with the title "jupyter Untitled14 (unsaved changes)". The notebook has tabs for "Untitled14 - Jupyter Notebook" and "Home Page - Select or create a new notebook". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. A toolbar below the menu contains icons for file operations like Open, Save, and Run, along with a Python logo and "Logout". The code cell area displays the following interactions:

```
In [88]: a[0][1]
Out[88]: [30, 40]

In [89]: a[0][0]
Out[89]: [10, 20]

In [90]: a[0][0][1]
Out[90]: 20

In [91]: a
Out[91]: [[[10, 20], [30, 40]], [[50, 60], [70, 80]]]

In [92]: a[1][0][1]
Out[92]: 60
```

An input cell is currently active, indicated by a green border around the "In []:" prompt.

- Creating Lists
- Accessing elements from Lists
- Slicing
- Reassigning List elements
- Deleting elements
- Multidimensional Lists
- Basic Operations
 - + , * , len , min , max
- List Comprehension
- Built-In Methods
 - append , extend , insert

The screenshot shows a Jupyter Notebook window with the title "Untitled14 - Jupyter Notebook". The URL in the address bar is "localhost:8888/notebooks/Untitled14.ipynb?kernel_name=python3". The notebook has one cell titled "jupyter Untitled14 (unsaved changes)". The cell content is as follows:

```

In [100]: b=2
           ^
           |
           +---- TypeError: can't multiply sequence by non-int of type 'list'

In [101]: a*b
           ^
           |
           +---- TypeError: can't multiply sequence by non-int of type 'list'

Out[101]: [10, 20, 30, 10, 20, 30]

In [102]: a=[10,20,30,40,50,60]
           ^
           |
           +---- SyntaxError: invalid syntax

In [103]: len(a)
           ^
           |
           +---- SyntaxError: invalid syntax

Out[103]: 6

In [104]: min(a)
           ^
           |
           +---- SyntaxError: invalid syntax

Out[104]: 10

```

The cell input area (In []:) contains the letter 'I'.

- Creating Lists
- Accessing elements from Lists
- Slicing
- Reassigning List elements
- Deleting elements
- Multidimensional Lists
- Basic Operations
 - + , * , len , min , max
- List Comprehension
- Built-In Methods
 - append , extend , insert

The screenshot shows a Jupyter Notebook window with the title "Untitled14 - Jupyter Notebook". The URL in the address bar is "localhost:8888/notebooks/Untitled14.ipynb?kernel_name=python3". The notebook contains the following code and output:

```
In [124]: a.extend([10,20,30,40])
In [125]: a
Out[125]: [10, 20, [30, 40], 10, 20, 30, 40]
In [126]: a.insert(2,50)
In [127]: a
Out[127]: [10, 20, 50, [30, 40], 10, 20, 30, 40]
In [128]: a.remove(50)
In [129]: a
Out[129]: [10, 20, [30, 40], 10, 20, 30, 40]
In [ ]: a.remove(10)
```

Tuples

Activate Windows

Go to Settings to activate Win



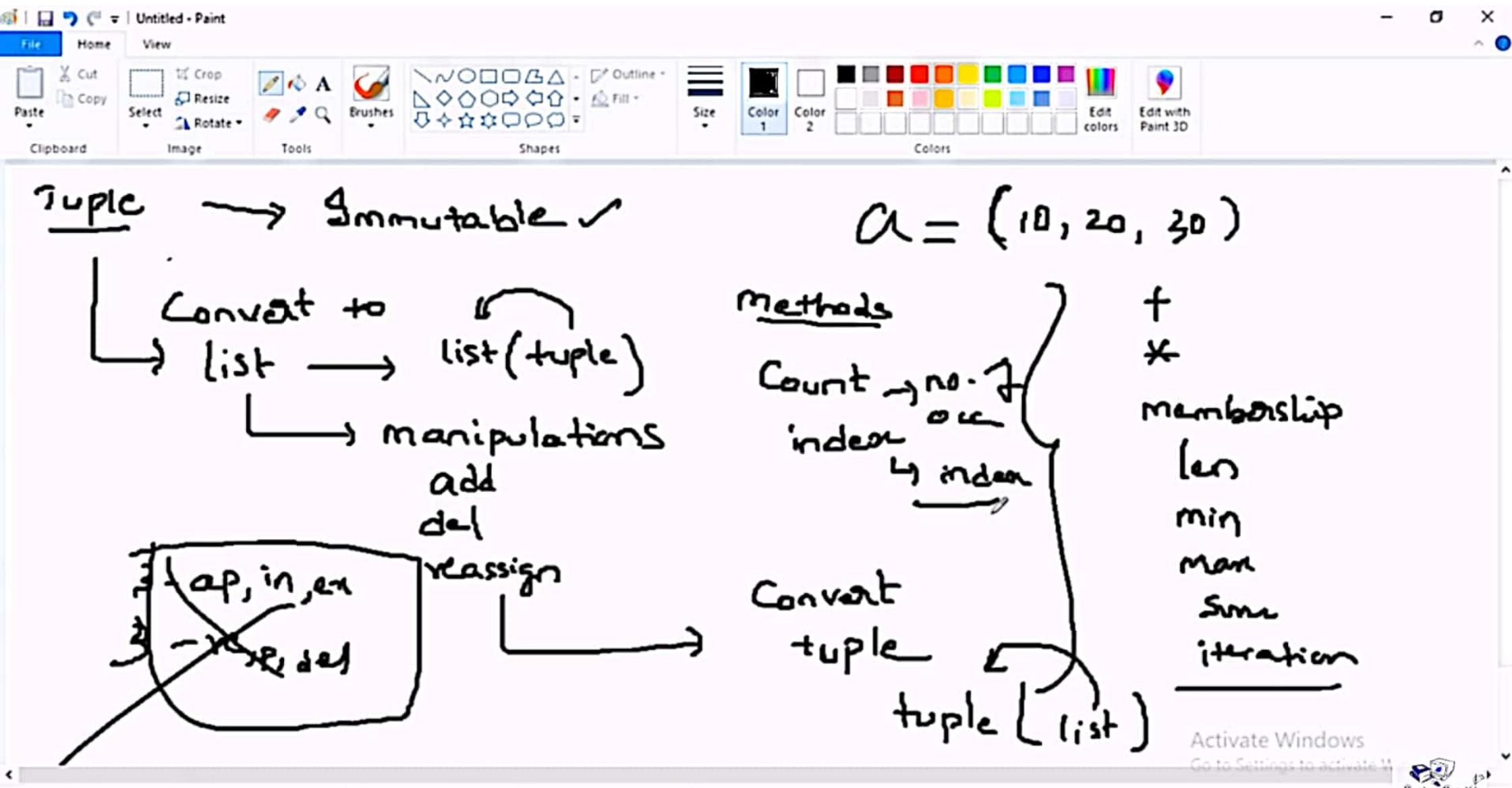
43

[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

A screenshot of the Microsoft Paint application window titled "Untitled - Paint". The interface includes a toolbar with basic drawing tools like selection, crop, resize, rotate, and various shapes and brushes. Below the toolbar is a color palette with two main colors selected, labeled "Color 1" and "Color 2". The main canvas contains handwritten notes about tuples:

- Tuple** → **Immutable**
- Accessing** ↓
- Indexing**
- Tuple create** ↓
- Can't change**
- Rev** → -1 $\circlearrowleft 1, 2$
- Slicing** $[1:3]$ \uparrow \uparrow 0 len
- Add**
del
Reassign
- $a[0] = 10$
 $a[1] = 20$
 $a[2] = 30$
- $a[2] = 30 \times$
- $a = (10, 20, 30)$
(Comma Separated Values)
multiple datatype
- $(10, 20, 30, "hello") \rightarrow$

 The notes illustrate that tuples are immutable, showing examples of slicing, assignment, and tuple creation. A circled "del" indicates that elements cannot be deleted from a tuple. A circled "a[2] = 30 ×" shows that attempting to assign to a tuple element results in an error. The tuple $(10, 20, 30, "hello")$ is shown as an example of multiple data types within a single tuple.



- Creating Tuple
- Accessing elements from Tuple
- Slicing
- Reassigning Tuple elements
- Deleting elements
- Nested Tuples
- Basic Operations
 - + , * , len , min , max , sum , membership , iterations
- Built-In Methods
 - count , index

Activate Windows
Go to Settings to activate W



Strings

Activate Windows

Go to Settings to activate Win 10



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

45

- Creating Strings
- Accessing elements from Strings
- Slicing
- Single , Double , Triple Quotes
- Format Method
- Basic Operations
 - + , * , len , min , max , sum , membership , iterations
- Built-In Methods
 - capitalize , center , count , endswith , startswith , find , index , rfind ,
rindex , isalnum , isalpha , isdigit , isspace , islower , isupper , istitle ,
ljust , rjust , lower , upper , strip , lstrip , rstrip , max , min , replace ,
split , swapcase , title , zfill

Activate Windows
Go to Settings to activate W



Sets

Activate Windows

Go to Settings to activate Win 10



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

47



Set → { comma sep values }

↓
multiple dotatypes

→ Duplicates will remove

Unique

→ Unordered

→ mutable

→ No Indenting

→ No slicing

Empty set - set() -

x { 3 → dictionary

- ❑ Creating Sets
- ❑ Set as an iterable
- ❑ Basic Operations

add , remove , discard , pop , clear len , membership ,
issubset , issuperset , union , intersection , difference , copy ,
symmetric_difference , update , intersection_update ,
difference_update , symmetric_difference_update

Activate Windows
Go to Settings to activate W



Dictionaries

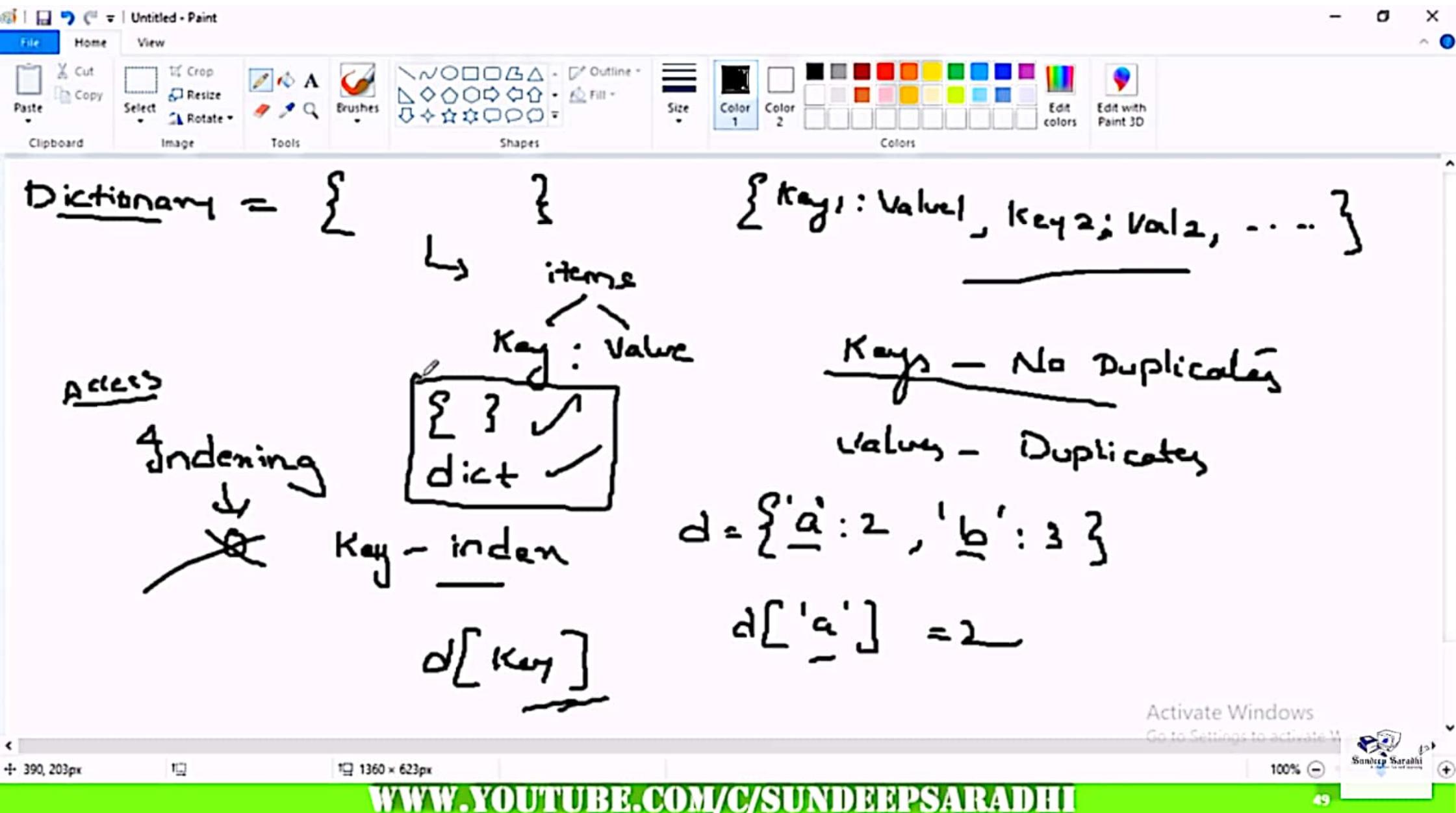
Activate Windows

Go to Settings to activate Win 10



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

49



- Creating Dictionary
- Accessing items from Dictionary
- Updating Dictionary
- Reassigning items
- Deleting items
- Dictionary Comprehension
- Built-In Methods

clear , copy , items , values , keys , update

Activate Windows
Go to Settings to activate W



Functions

Activate Windows

Go to Settings to activate Win



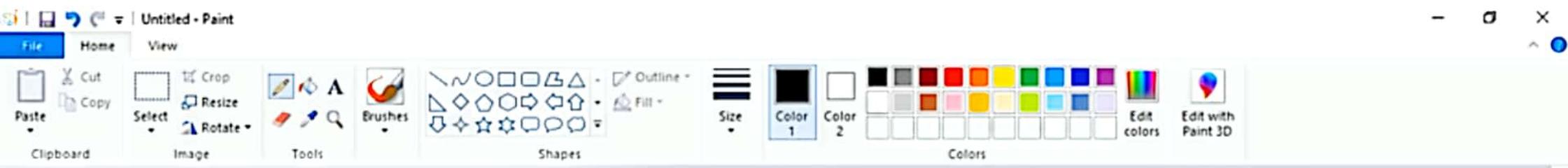
[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

51

- Types of Functions
- Function Call
- Function Definition
- Types of Arguments
 - Required Arguments , Default Arguments , Keyword Arguments , Arbitrary Arguments
- Multiple Returns
- Local and Global Variables
- Lambda Function
- Recursion

Activate Windows
Go to Settings to activate W





Types

↳ Built-in function — factorial

pow

sqrt

sin

cos

User Defined functions

Lambda functions — Anonymous

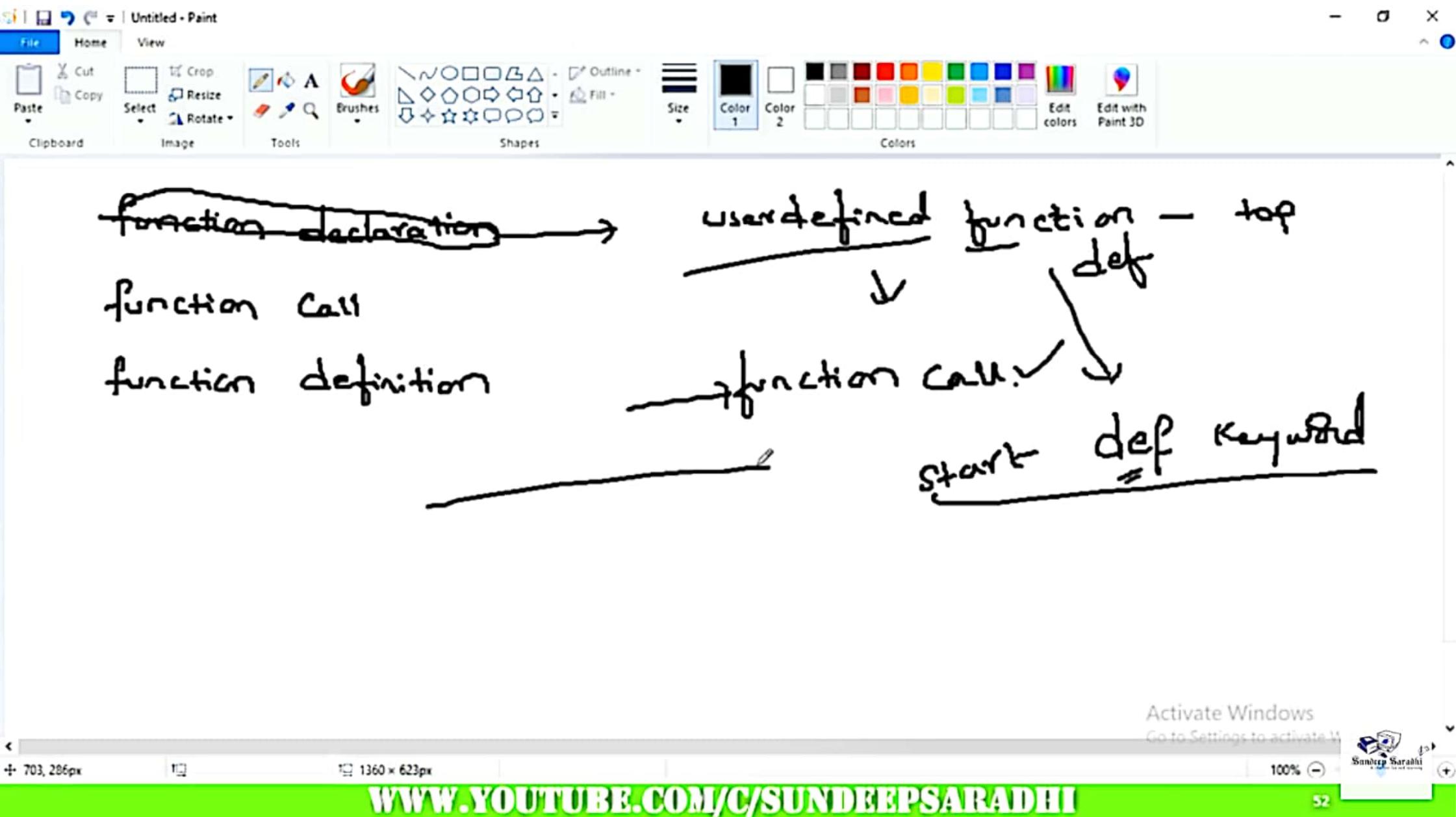
Recursion,

Activate Windows

Go to Settings to activate

100%



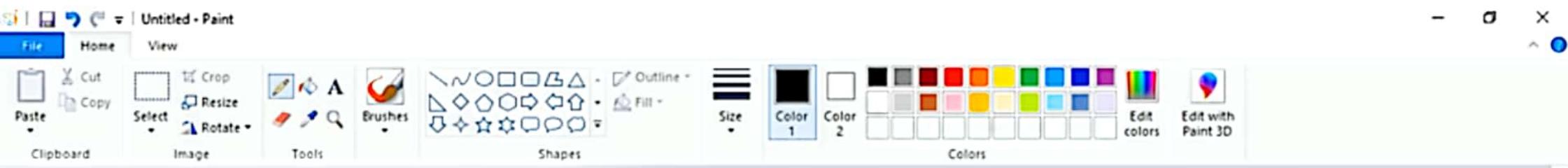


Activate Windows

Go to Settings to activate it

100%





Reg. Arg

- No. of parameters should match in both function call & def.
- Positional arguments

Default Arguments

- No. of parameters may not be equal in both
funcall & def
- Default argument

Activate Windows

Go to Settings to activate

100%





Keyword

parameters

variable name

keyword.

No need to follow position

Arbitrary

fun def — only one argu → Variable name

func call — Any no. of arguments

Activate Windows
Go to Settings to activate it

226, 311px

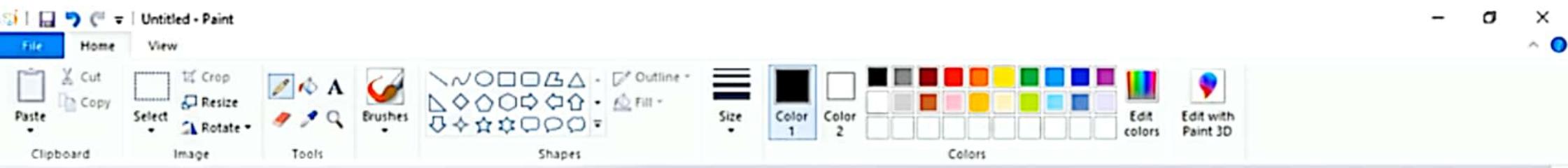
1360 x 623px

100%



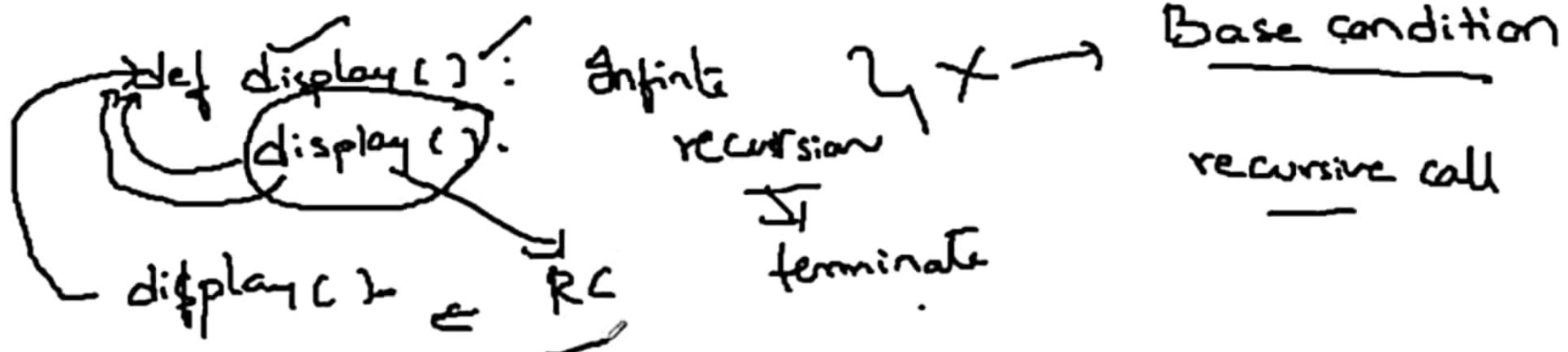
WWW.YOUTUBE.COM/C/SUNDEEPSARADHI

52



Recursion

Calling function itself.



Activate Windows

Go to Settings to activate



+ 588, 408px

1360 x 623px

100%

WWW.YOUTUBE.COM/C/SUNDEEPSARADHI

52



factorial

$$4 \rightarrow 4 \times \underline{3} \times \underline{2} \times \underline{1}$$

$$\text{fact}(4) = \underline{24}$$

$$\downarrow 4 \times \text{fact}(3)$$

Base condition

$$\downarrow 3 \times \text{fact}(2)$$

$$\downarrow 2 \times \text{fact}(1)$$

return $1.$

else

$n \times \text{fact}(n-1)$ \rightarrow Recursive
call

fact(4)

Activate Windows

Go to Settings to activate

100%



Files

Activate Windows

Go to Settings to activate Win



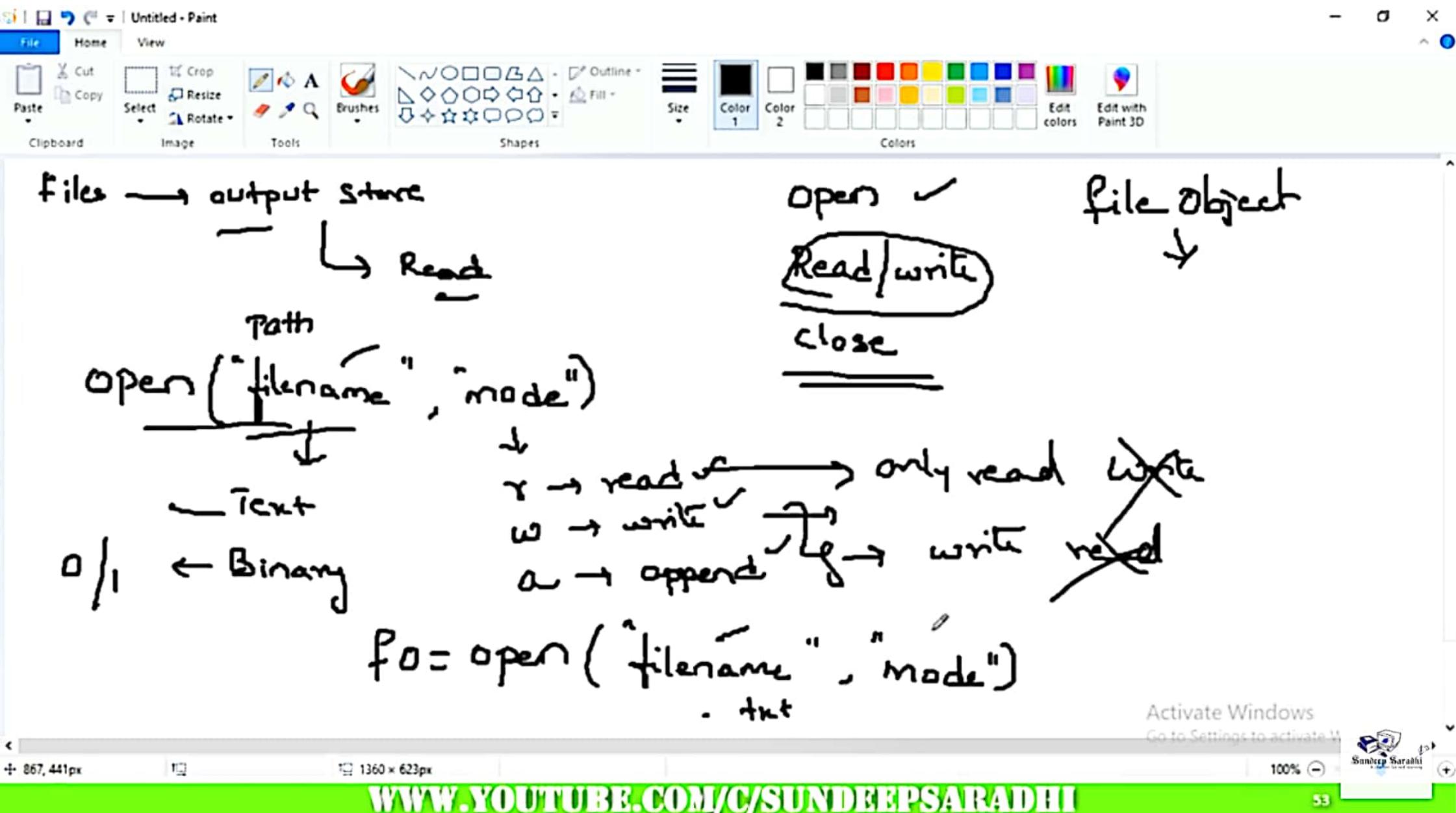
[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

53

- What is File ?
- Opening a File
- Access Modes
- Closing File
- Writing Data into File
 - write , writelines
- Reading Data from File
 - read , readline , readlines
- File Handling functions
 - tell , seek
- Usage of WITH keyword in Files

Activate Windows
Go to Settings to activate W







w } → Create

r → X

fo = open("hello.txt", "a")

fo.write()

file doesn't exist

~~Create~~

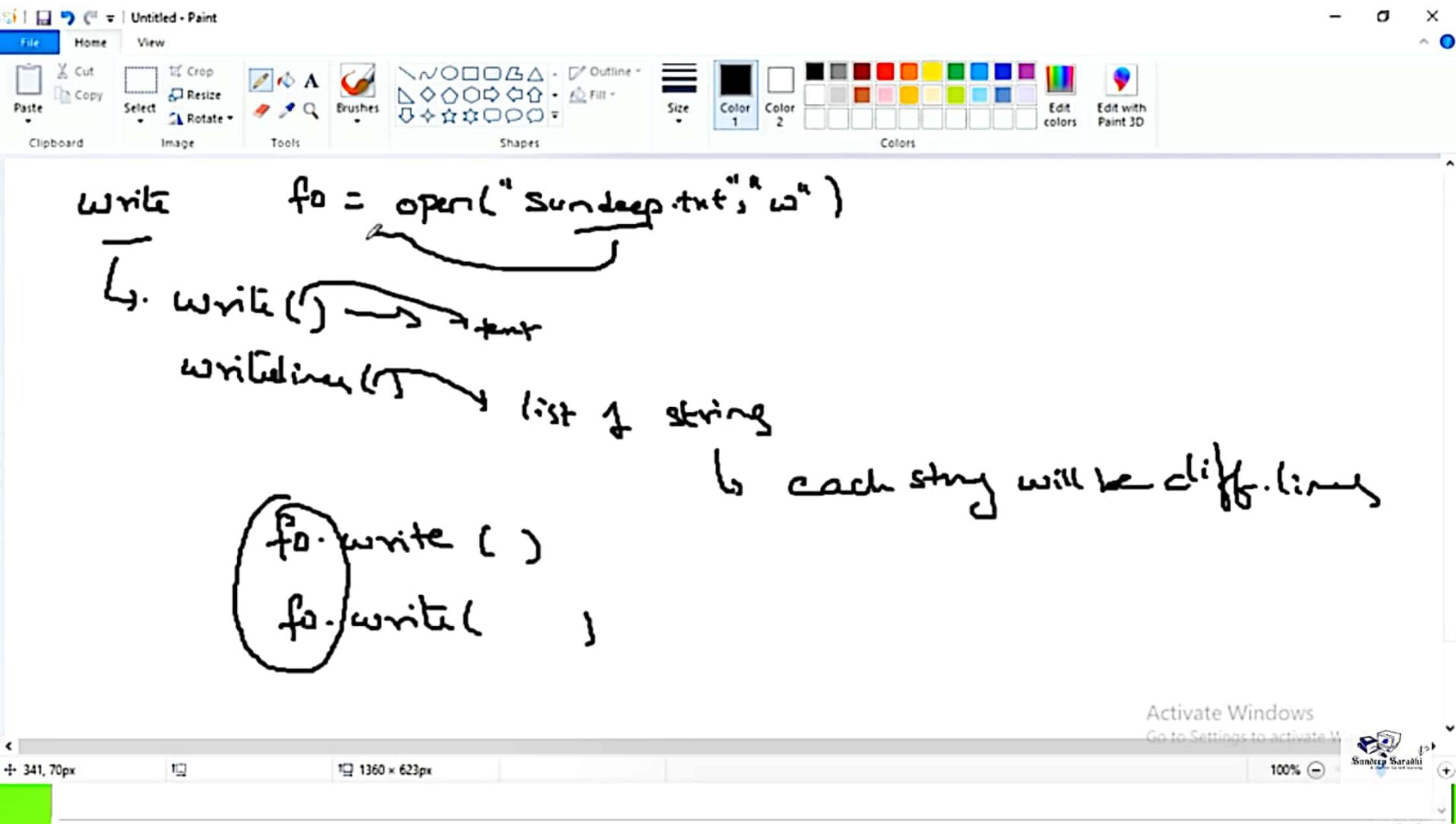


Activate Windows

Go to Settings to activate

100%





```
Python *create.py - C:/Users/Admin/AppData/Local/Programs/Python/Python36/python files/create.py (3.6.5)*
File Edit Format Run Options Window Help
>>> fo=open("Sundeep.txt","w")
RE fo.write("Hello")
.PY fo.write("Welcome")
Hel l=["\nThis\n","is\n","Files\n","concept\n"]
>>> fo.writelines(l)
RE fo.close()
.PY fo1=open("Sundeep.txt","r")
Hel print(fo1.readlines())
Wel
>>> fo1.close()
RE
.PY
Hel

>>>
RE
.PY
Hel

This
>>>
```

Activate Windows
Go to Settings to activate W



Python 3.6.5 Shell

File Edit Shell Debug Options Window Help

```
>>>
RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python36/python files/create
.py
Hello
Welco
>>>
RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python36/python files/create
.py
HelloWelcome

>>>
RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python36/python files/create
.py
HelloWelcome

This

>>>
RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python36/python files/create
.py
['HelloWelcome\n', 'This\n', 'is\n', 'Files\n', 'concept\n']
```

Activate Windows
Go to Settings to activate W



L: 49 C: 4

Libraries in Python

Activate Windows

Go to Settings to activate Win



[WWW.YOUTUBE.COM/C/SUNDEEPSARADHI](https://www.youtube.com/c/SunDeepsaradhi)

55

Data Science

- Pandas , NumPy , SciPy , Scrapy , Matplotlib , Seaborn , Scikit-Learn , TensorFlow , Scikit-Image , Librosa

Data Visualization

- Matplotlib , ggplot , Ploty , Altair , Seaborn , Bokeh , Pygal

Data Manipulation

- Pandas , NumPy

Database Access

- SQLAlchemy , Quandl

Data Modeling

- Tensorflow , PyTorch , Scikit – Learn , Keras

Activate Windows
Go to Settings to activate Win 10


Web Development

- Django , Bottle , Pyramid , Turbo Gears , Web2py , Grok , Flask , CherryPy , Hug , Falcon

Gaming

- PyGame

GUI Framework

- Tkinter , wxPython , PyQt5

Robotics

- PyRobot , DART , SOFA , Pyro

Network Programming

- NetworkX

Natural Language Processing

- NLTK , Scikit-Learn , polyglot , genism , spaCy , rasaNLU

Activate Windows
Go to Settings to activate W



Deep Learning

- TensorFlow , Torch , Caffe , Theano , Deepmat , ML.NET , Neon , Microsoft CNTK , Deeplearning 4j

Image Processing

- OpenCV , Scikit-Image , Scikit-Learn , SciPy , NumPy

AI & Machine Learning

- Scikit-Learn , Theano , TensorFlow , Keras

Activate Windows

Go to Settings to activate Win

