

# *System Requirements*

- *Python runs on Windows, Linux / Unix, Mac OS X, and has been ported to the Java and .NET virtual machines also.*
- *No special hardware requirement*

## *Downloading and Installation*

- *Log on to [www.python.org](http://www.python.org)*
- *The home page looks as .....*

Python Programming Lan x

← → C www.python.org

python™

ABOUT >>  
NEWS >>  
DOCUMENTATION >>  
DOWNLOAD >>  
下載 >>  
COMMUNITY >>  
FOUNDATION >>  
CORE DEVELOPMENT >>

Help  
Package Index  
Quick Links (2.7.4)  
» Documentation  
» Windows Installer  
» Source Distribution  
Quick Links (3.3.1)  
» Documentation  
» Windows Installer  
» Source Distribution  
Python Jobs  
Python Merchandise  
Python Wiki  
Python Insider Blog

Python Programming Language – Official Website

Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use Python and see almost immediate gains in productivity and lower maintenance costs.

Python runs on Windows, Linux/Unix, Mac OS X, and has been ported to the Java and .NET virtual machines.

Python is free to use, even for commercial products, because of its OSI-approved open source license.

New to Python or choosing between Python 2 and Python 3? Read [Python 2](#) or [Python 3](#).

The [Python Software Foundation](#) holds the intellectual property rights behind Python, underwrites the [PyCon](#) conference, and funds many other projects in the Python community.

[Read more, -or- download Python now](#)

» **Python 3.2.4 and 3.3.1 have been released**  
[Python 3.2.4](#) and [Python 3.3.1](#) final have been released.  
Published: Sat, 6 April 2013, 22:30 +0200

» **Python 2.7.4 released**  
[Python 2.7.4](#) has been released.  
Published: Sat, 6 April 2013, 11:00 -0500

» **Release candidates for Python 3.2.4 and 3.3.1 have been released**  
Release candidates for [Python 3.2.4](#) and [Python 3.3.1](#) have been released. Please test!  
Published: Mon, 25 March 2013, 11:00 +0200

search

Advanced Search

Support the Python Community  
Help the Python community by becoming an associate member or making a one-time donation.

Python 3 Poll  
I wish there was Python 3 support in  
  
(enter PyPI package name)

Build documentation with Python...  
**Honeywell**  
[HOME](#) [ABOUT US](#) [INDUSTRIES & TECHNOLOGY](#) [NEWS & EVENTS](#) [INVESTOR](#)  
**Case Study 02**  
PowerShield Protects Photovoltaic Modules that Convert Sunlight into Electricity for up to 25 Years  
  
... joining users such as Rackspace, Industrial Light and Magic, AstraZeneca, Honeywell,

session 3.pptx

Show all downloads...

> Click on Download Python Now Hyperlink or Download option from the side menu..

Python Programming Lan x

www.python.org

python™

ABOUT >>

NEWS >>

DOCUMENTATION >>

**DOWNLOAD >>** (highlighted with a blue box)

下 載 >>

COMMUNITY >>

FOUNDATION >>

CORE DEVELOPMENT >>

Help

Package Index

Quick Links (2.7.4)  
» Documentation  
» Windows Installer  
» Source Distribution

Quick Links (3.3.1)  
» Documentation  
» Windows Installer  
» Source Distribution

Python Jobs

Python Merchandise

Python Wiki

Python Insider Blog

Python Programming Language – Official Website

Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use Python and see almost immediate gains in productivity and lower maintenance costs.

Python runs on Windows, Linux/Unix, Mac OS X, and has been ported to the Java and .NET virtual machines.

Python is free to use, even for commercial products, because of its OSI-approved open source license.

New to Python or choosing between Python 2 and Python 3? Read [Python 2](#) or [Python 3](#).

The Python Software Foundation holds the intellectual property rights behind Python, underwrites the PyCon conference, and funds numerous other projects in the Python community.

Read more, -or- [download Python now](#) (highlighted with a blue box)

» **Python 3.2.4 and 3.3.1 have been released**  
Python 3.2.4 and Python 3.3.1 final have been released.  
Published: Sat, 6 April 2013, 22:30 +0200

» **Python 2.7.4 released**  
Python 2.7.4 has been released.  
Published: Sat, 6 April 2013, 11:00 -0500

» **Release candidates for Python 3.2.4 and 3.3.1 have been released**  
Release candidates for Python 3.2.4 and Python 3.3.1 have been released. Please test!  
Published: Mon, 25 March 2013, 11:00 +0200

Support the Python Community

Help the Python community by becoming an associate member or making a one-time donation.

Python 3 Poll

I wish there was Python 3 support in

(enter PyPI package name)

Vote  Results

Build documentation with Python...

Honeywell

HOME ABOUT US INDUSTRIES & TECHNOLOGY NEWS & EVENTS INVESTOR

Case Study 02 Honeywell Sees Bright Future in Backup Solar Power

PowerShield Protects Photovoltaic Modules that Convert Sunlight into Electricity for up to 24 hours

... joining users such as Rackspace, Industrial Light and Magic, AstraZeneca, Honeywell,

Show all downloads...

session 3.pptx

➤ The Download Python Page appears. Click on the version of Python that you want to install (Python 2.7 or Python 3.3)

➤ We shall be working with Python 2.7.4

The screenshot shows a web browser window titled "Download Python" at the URL [www.python.org/download/](http://www.python.org/download/). The page features the Python logo and navigation links for "ABOUT", "NEWS", "DOCUMENTATION", "DOWNLOAD", "License", "Releases", "Windows", "Macintosh", "Other", "Source", "下轉", "COMMUNITY", "FOUNDATION", and "CORE DEVELOPMENT". A blue arrow points to the "Download Python" heading, which is highlighted with a blue border. Below it, a message states: "The current production versions are Python 2.7.4 and Python 3.3.1." Further down, instructions advise starting with one of these versions for learning Python or for stability. It suggests trying Python 3.3 if unsure, mentioning compatibility issues with some third-party software. It also provides links for MD5 checksums and OpenPGP signatures for both Python 3.3.1 and Python 2.7.4. At the bottom, there's a "PayPal DONATE" button and download links for various Python versions.

Download Python

The current production versions are Python 2.7.4 and Python 3.3.1.

Start with one of these versions for learning Python or if you want the most stability; they're both considered stable production releases.

If you don't know which version to use, try Python 3.3. Some existing third-party software is not yet compatible with Python 3; if you need to use such software, you can download Python 2.7.x instead.

For the MD5 checksums and OpenPGP signatures, look at the [detailed Python 3.3.1 page](#):

- [Python 3.3.1 Windows x86 MSI Installer](#) (Windows binary -- does not include source)
- [Python 3.3.1 Windows X86-64 MSI Installer](#) (Windows AMD64 / Intel 64 / X86-64 binary [1] -- does not include source)
- [Python 3.3.1 Mac OS X 64-bit/32-bit x86-64/i386 Installer](#) (for Mac OS X 10.6 and later [2])
- [Python 3.3.1 Mac OS X 32-bit i386/PPC Installer](#) (for Mac OS X 10.5 and later [2])
- [Python 3.3.1 bzipped source tarball](#) (for Linux, Unix or Mac OS X)
- [Python 3.3.1 xzipped source tarball](#) (for Linux, Unix or Mac OS X, better compression)

For the MD5 checksums and OpenPGP signatures, look at the [detailed Python 2.7.4 page](#):

- [Python 2.7.4 Windows Installer](#) (Windows binary -- does not include source)
- [Python 2.7.4 Windows X86-64 Installer](#) (Windows AMD64 / Intel 64 / X86-64 binary [1] -- does not include source)
- [Python 2.7.4 Mac OS X 64-bit/32-bit x86-64/i386 Installer](#) (for Mac OS X 10.6 and later [2])
- [Python 2.7.4 Mac OS X 32-bit i386/PPC Installer](#) (for Mac OS X 10.3 and later [2])
- [Python 2.7.4 compressed source tarball](#) (for Linux, Unix or Mac OS X)

[source tarball](http://www.python.org/ftp/python/3.3.1/python-3.3.1-macosx10.6.dmg) (for Linux, Unix or Mac OS X, more compressed)

session 3.pptx

Show all downloads...

# *Choose the required format for download (Windows x86 MSI Installer (2.7.4))*

The screenshot shows a web browser window displaying the Python 2.7.4 Release page at [www.python.org/download/releases/2.7.4/](http://www.python.org/download/releases/2.7.4/). The page includes a sidebar with links for various Python versions and platforms, and a main content area with sections for 'Resources' and 'Download'. In the 'Download' section, a list of available formats is shown, with the 'Windows x86 MSI Installer (2.7.4) [1] (sig)' link highlighted by a large blue arrow and a blue rectangular box.

2.5.6  
2.4.6  
2.3.7  
2.2.3  
2.1.3  
2.0.1  
1.6.1  
1.5.2  
ancient

Windows  
Macintosh  
Other  
Source  
下載 >>  
COMMUNITY >>  
FOUNDATION >>  
CORE DEVELOPMENT >>

Python Wiki  
Python Insider Blog  
Python 2 or 3?  
Help Fund Python  
 or  
 Non-English Resources

**Resources**

- [What's new in 2.7?](#)
- [Complete change log for this release.](#)
- [Online Documentation](#)
- [Report bugs at <http://bugs.python.org>.](#)
- [Help fund Python and its community.](#)

**Download**

This is a production release. Please [report any bugs](#) you encounter.

We currently support these formats for download:

- [XZ compressed source tar ball \(2.7.4\) \(sig\)](#)
- [Gzipped source tar ball \(2.7.4\) \(sig\)](#)
- [Bzipped source tar ball \(2.7.4\) \(sig\)](#)
- [Windows x86 MSI Installer \(2.7.4\) \(sig\)](#)
- [Windows x86 MSI program database \(2.7.4\) \(sig\)](#)
- [Windows X86-64 MSI Installer \(2.7.4\) \[1\] \(sig\)](#)
- [Windows X86-64 program database \(2.7.4\) \[1\] \(sig\)](#)
- [Mac OS X 64-bit/32-bit x86-64/i386 Installer \(2.7.4\) for Mac OS X 10.6 and later \[2\] \(sig\)](#). [You may need an updated Tcl/Tk install to run IDLE or use instructions 1]

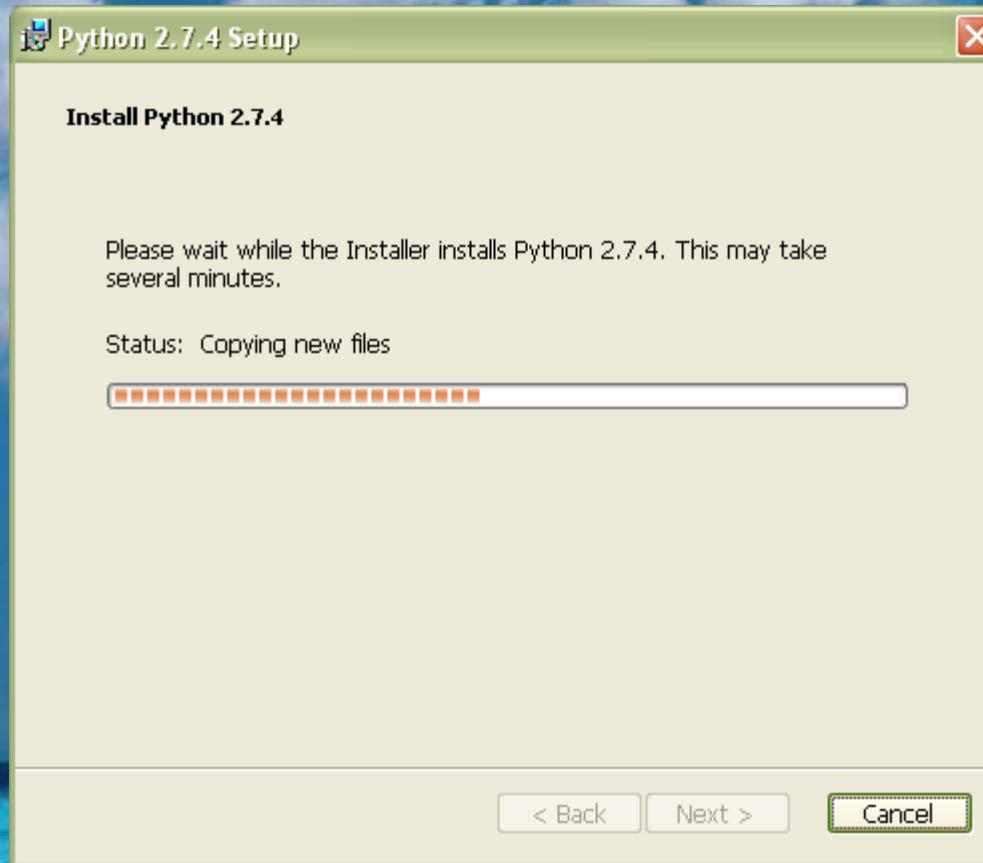
python-2.7.4.amd64-....zip  
0.1/16.6 MB, 22 mins left

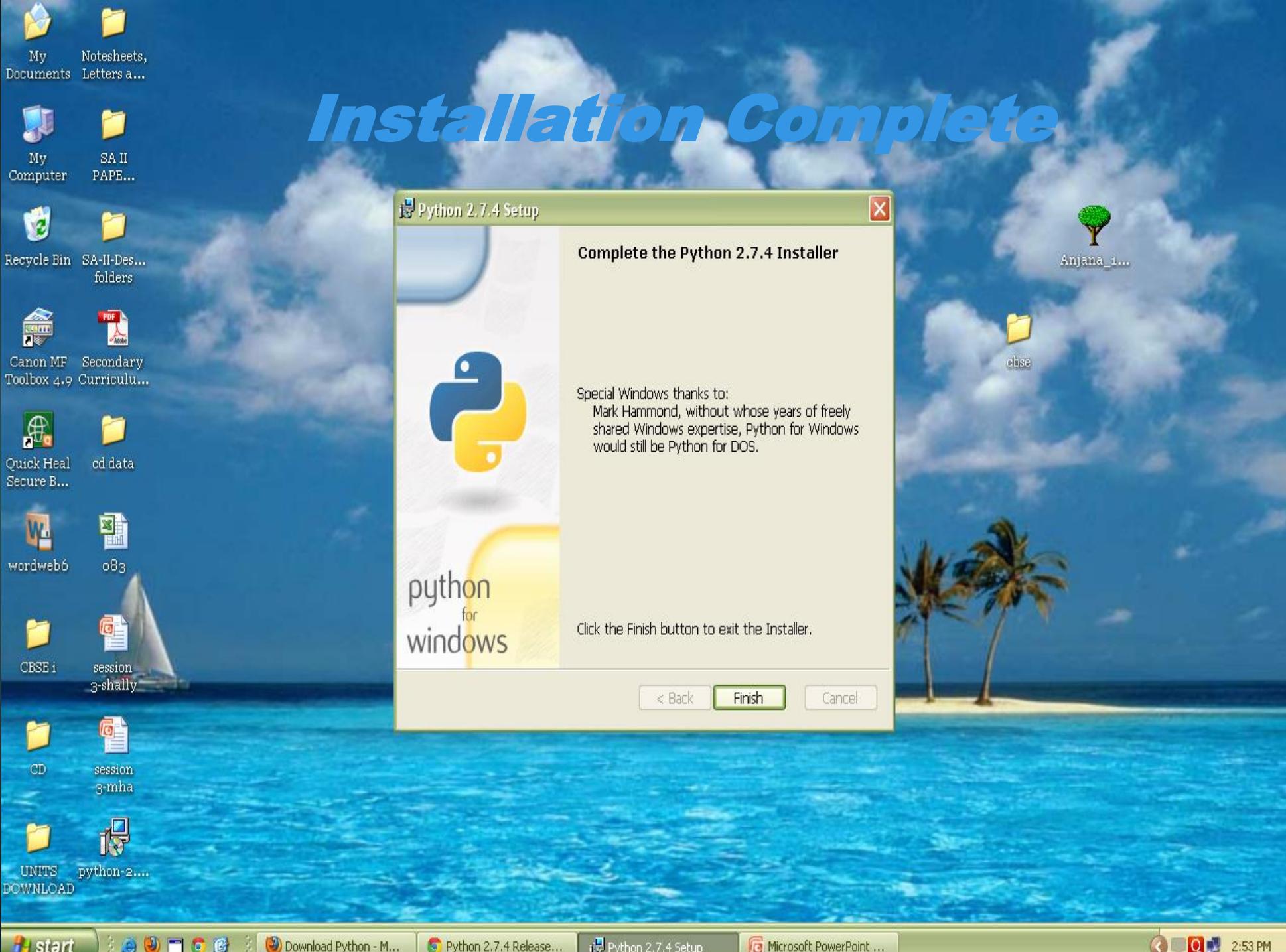
session 3.pptx

Show all downloads...

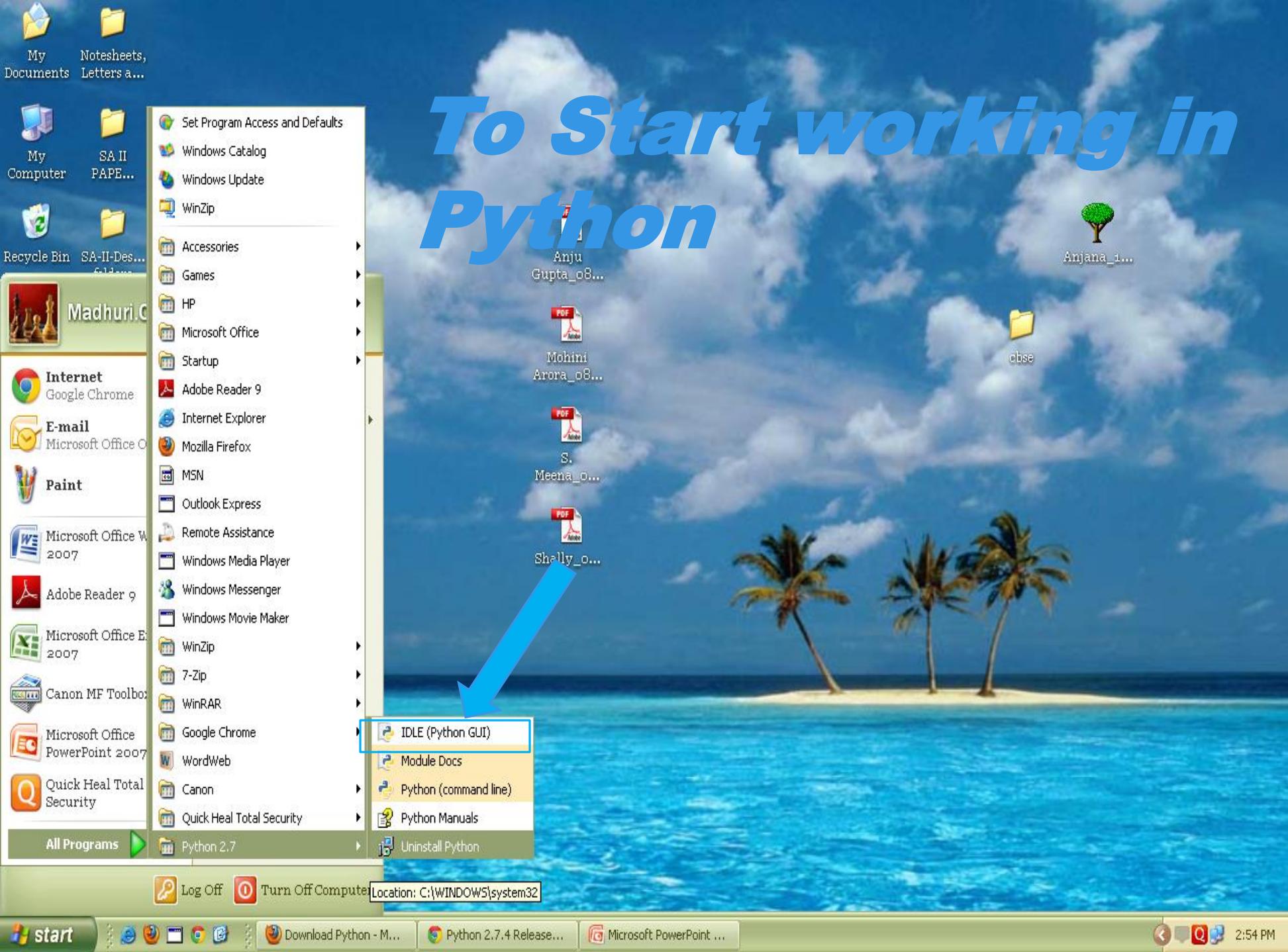


# Installation in Progress.....

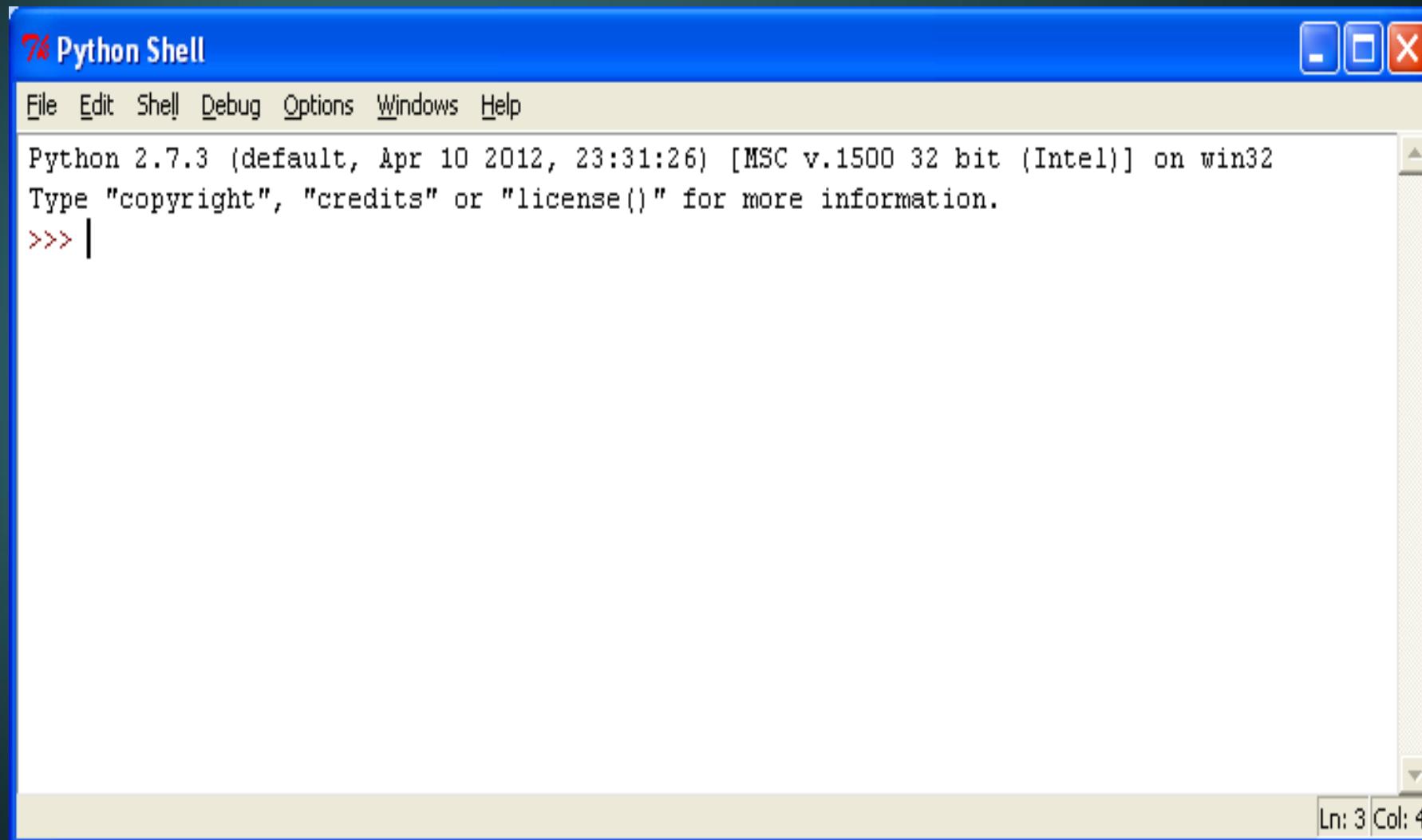




# To Start working in Python



# Welcome Screen of Python IDLE



# *Python Shell*

*Interactive Mode*



*Script Mode*



# *Interactive Mode*

- ***Commands are executed directly at the prompt***

Examples :-

```
>>> print 5+8  
13
```

```
>>>6+10  
16
```

```
>>>Print "Welcome to the world of Python"  
Welcome to the world of Python
```

```
>>> (10+5)/3  
5
```

# **Interactive Mode : Sequence of commands**

## **Example 1**

```
>>> x=10  
>>>y=40  
>>>Z=x+y  
>>>print Z  
50
```

## **Example 2**

```
>>> a=10  
>>>a+1 , a-1  
(11,9)
```

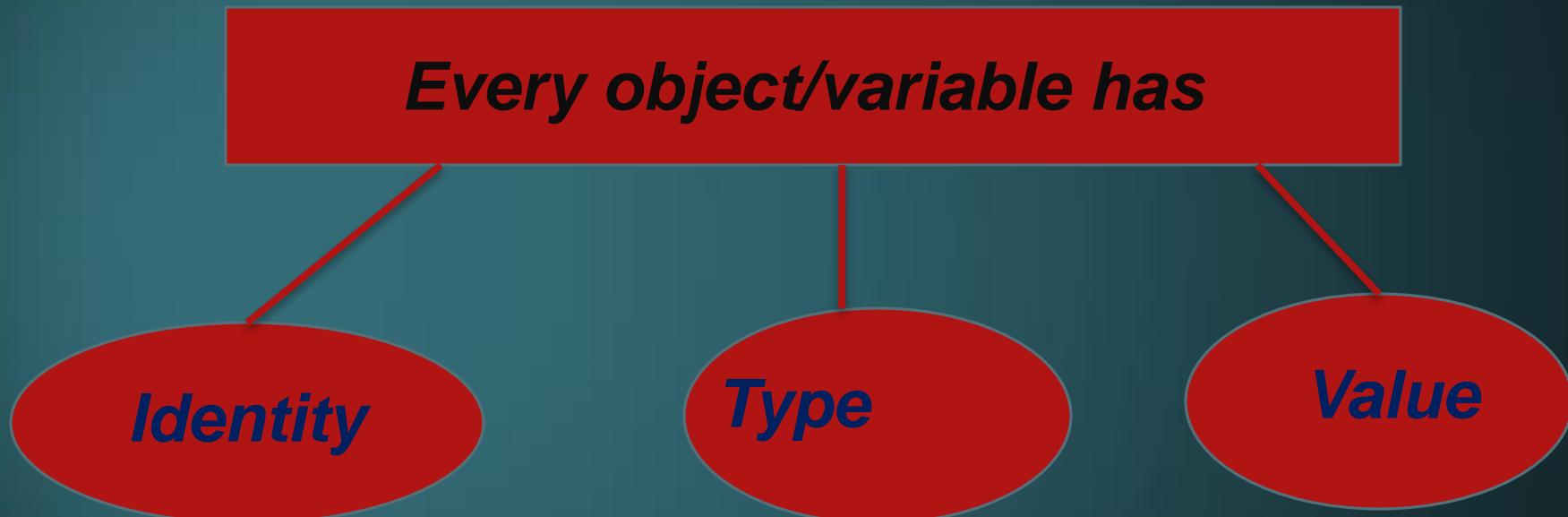
- ***to repeat the prior command in the interactive window , use ↑ to scroll backward through the command history and ↓ key to scroll forward.***



# *Script Mode*

- *File→ New*
- *Type Python program in a file*
- *Save it . Default extension is .py*
- *then use interpreter to execute the contents from the file by*
  - *Can use Run option or press F5*
  - *type the function name at command prompt*

# *Variables / Objects in Python*



# *Identity*

- *Variable's address in memory*
- *does not change once it has been created.*

*The id() command gives the identity number of a variable, it's unique for each variable*

**Syntax : *id(variable)***

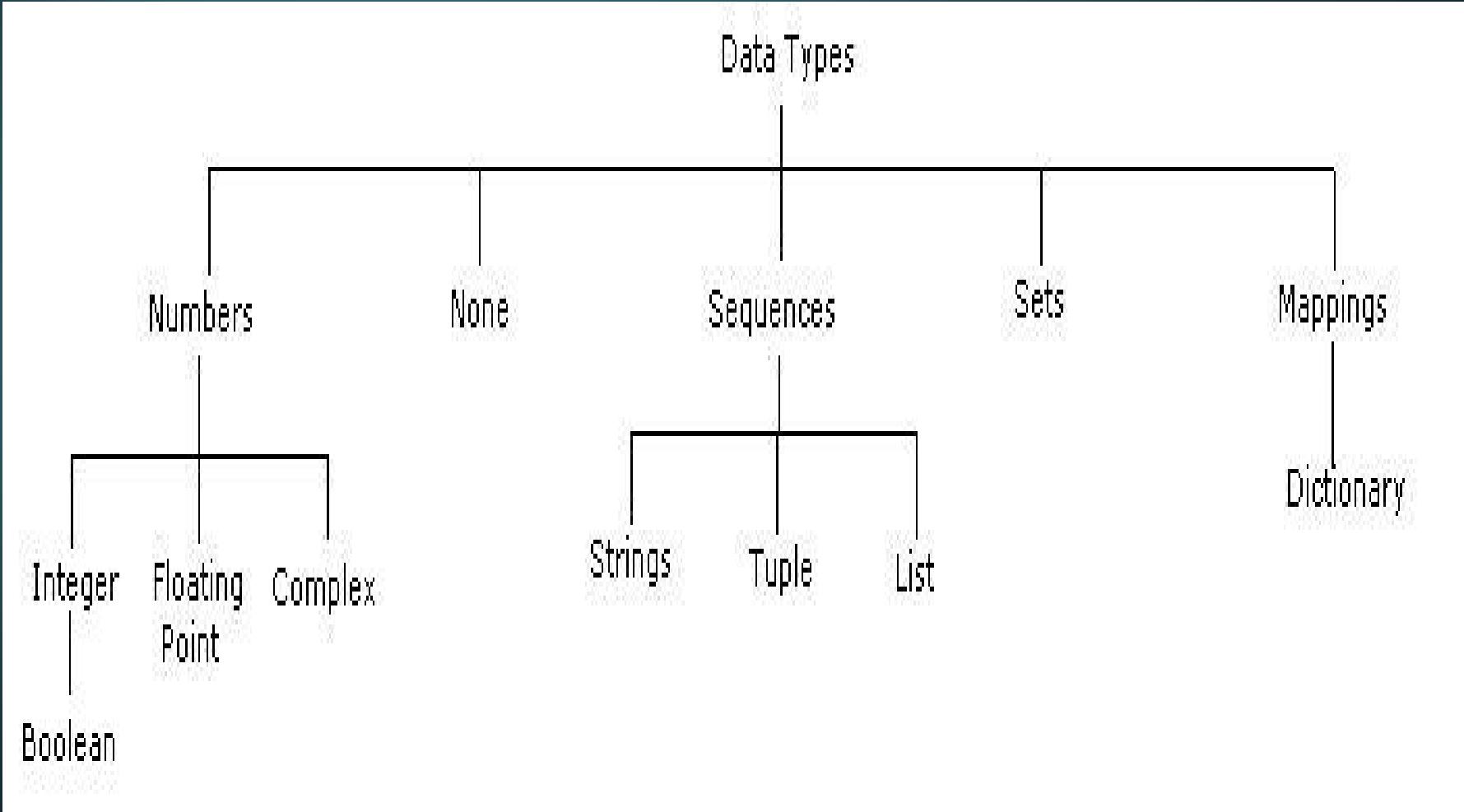
**Example :**

```
>>>id(x)  
4325671
```

# **Type (Data Type)**

*A set of values and the allowable operations on those values.*

# *Data types supported by Python*



# **Number Data Type**

## **1. Number**

### **1. Integer & Long**

*Example*

```
>>>a=10  
>>>b=1234567L  
>>>print a,b  
10, 123456L
```

### **2. Floating Point**

*Example*

```
>>>y=12.89  
>>>print y  
12.89
```

'j' is used for  
imaginary part

### **3. Complex**

*Example*

```
>>>x= 1+0j  
>>>print x.real , x.imag  
1.0, 0.0
```

# Type Conversion

- Automatic type conversion from int to float if required

```
>>> f=78.90876
```

```
>>> i=9
```

```
>>> i=i+f
```

```
>>> print i
```

```
87.90876
```

- very big integer is automatically converted to long
- To know the datatype of a variable use type()

```
>>> var=1234
```

```
>>> type(var)
```

```
<type 'int'>
```

```
>>> var=var*98765432
```

```
>>> type(var)
```

```
<type 'long'>
```

## **Integers** (contd)

### **4. Boolean**

- Boolean Type consists of a two values : True & False
- Boolean True is non zero, non null and non empty

#### **Example**

```
>>> flag=True  
>>> type(flag)  
<type 'bool'>  
>>> flag=False  
>>> type(flag)  
<type 'bool'>
```

## **2. None**

*used to signify the absence of value / false in a situation*

## **3. Sequence**

- *Ordered collection of items , indexed by integers*
- *Combination of mutable and non mutable data types*

# ***Sequence Data Type***

## ***String***

- An ordered sequence of letters and characters

- Enclosed in “ ” or ‘ ’

- Are immutable

### ***Example***

***>>>a=“Ram”***

## ***Lists***

- A sequence of values of any type

- Values in the list are called items and are indexed

- Are mutable

- Are enclosed in []

***Example***  
[‘spam’, 20, 13.5]

## ***Tuple***

- are sequence of values of any type

- Are indexed by integers

- Are immutable

- Are enclosed in ()

### ***Example***

***(2,4)***

## **Data Types (contd.)**

### **4. Sets :**

- *an unordered collection of values of any type*
- *No duplicate entries allowed*
- *Are immutable*

**Example**

**S=set([1,2,3,4])**

### **5. Mapping**

- *an unordered data type for eg: dictionaries*
- *A dictionary can store any number of Python objects enclosed in curly brackets*

**Example :**

**D= {1:'a' , 2:'b', 3:'c'}**

## ***Value of an object(variable)***

***to bind value to a variable assignment operator(=) can be used***

***Example :***

***>>>x=1256***

# ***Mutable and Immutable Variables***

***Mutable Variable :***

*one whose value may change in place*

***Immutable Variable :***

*# Change of value will not happen in place.*

*# Modifying an immutable variable will rebuild the same variable.*

# Example

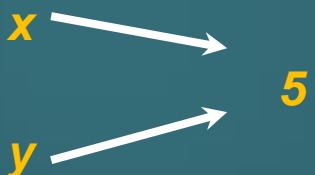
```
>>>x=5
```

*will create a value 5 referenced by x*



```
>>>y=x
```

*will make y refer to 5 of x*



```
>>>x=x+y
```

*RHS results to 10*

*value assigned to LHS (x)*

*x rebuilds to 10*



# *Example*

```
x = something      # immutable type
y = x
print x
// some statement that operates on y
print x          # prints the original value of x
x = something      # mutable type
y = x
print x
// some statement that operates on y
print x          # might print something different
```

# *Example*

## Immutable variable - x

```
x = 'foo'  
y = x  
print x      # foo  
y += 'bar'  
print x      # foo
```

## Mutable Variable - x

```
x = [1, 2, 3]  
y = x  
print x      # [1, 2, 3]  
y += [3, 2, 1]  
print x      # [1, 2, 3, 3, 2, 1]
```

## *Points to remember about variables.....*

- *Are created when they are first assigned a value*
- *Refer to an object*
- *Must be assigned a value before using them in any expression*
- *Keywords cannot be used as variable names*

# *Operators*



Operators when applied to operands form an expression

# Arithmetic Operators

Symbol	Description	Example 1	Example 2
+	Addition	>>>55+45 100	>>>'Good' + 'Morning' <i>GoodMorning</i>
-	Subtraction	>>>55-45 10	>>>30-80 -50
*	Multiplication	>>>55*45 2475	>>>'Good'* 3 <i>GoodGoodGood</i>
/	Division*	>>>17/5 3 >>>17/5.0 3.4	>>>28/3 9
%	Remainder/Modulo	>>>17%5 2	>>> 23%2 1
**	Exponentiation	>>>2**3 8	>>>2**8 256
//	Integer Division	>>>7.0//2 3.0	>>>3//2 1.0

# *Relational Operators*

<i>Symbol</i>	<i>Description</i>	<i>Example 1</i>	<i>Example 2</i>
<	<i>Less than</i>	<code>&gt;&gt;&gt;7&lt;10 True &gt;&gt;&gt; 7&lt;5 False &gt;&gt;&gt; 7&lt;10&lt;15 True</code>	<code>&gt;&gt;&gt;'Hello'&lt; 'Goodbye' False &gt;&gt;&gt;'Goodbye'&lt; 'Hello' True</code>
>	<i>Greater than</i>	<code>&gt;&gt;&gt;7&gt;5 True &gt;&gt;&gt;10&lt;10 False</code>	<code>&gt;&gt;&gt;'Hello'&gt; 'Goodbye' True &gt;&gt;&gt;'Goodbye'&gt; 'Hello' False</code>
<=	<i>less than equal to</i>	<code>&gt;&gt;&gt; 2&lt;=5 True &gt;&gt;&gt; 7&lt;=4 False</code>	<code>&gt;&gt;&gt;'Hello'&lt;= 'Goodbye' False &gt;&gt;&gt;'Goodbye' &lt;= 'Hello' True</code>
>=	<i>greater than equal to</i>	<code>&gt;&gt;&gt;10&gt;=10 True &gt;&gt;&gt;10&gt;=12 False</code>	<code>&gt;&gt;&gt;'Hello'&gt;= 'Goodbye' True &gt;&gt;&gt;'Goodbye' &gt;= 'Hello' False</code>
<i>!=, &lt;&gt;</i>	<i>not equal to</i>	<code>&gt;&gt;&gt;10!=11 True &gt;&gt;&gt;10!=10 False</code>	<code>&gt;&gt;&gt;'Hello'!= 'HELLO' True &gt;&gt;&gt; 'Hello' != 'Hello' False</code>
<i>==</i>	<i>equal to</i>	<code>&gt;&gt;&gt;10==10 True &gt;&gt;&gt;10==11 False</code>	<code>&gt;&gt;&gt;"Hello" == 'Hello' True &gt;&gt;&gt;'Hello' == 'Good Bye' False</code>

# *Logical Operators*

<i>Symbol</i>	<i>Description</i>
<i>or</i>	<i>If any one of the operand is true, then the condition becomes true.</i>
<i>and</i>	<i>If both the operands are true, then the condition becomes true.</i>
<i>not</i>	<i>Reverses the state of operand/condition.</i>

# Assignment Operator

Symbol	Description	Example	Explanation
=	Assigned values from right side operands to left variable	>>>x=12* >>>y='greeting s'	*we will use it as initial value of x for following examples.
+=	Added and assign back the result to left operand	>>>x+=2	Will change the value of x to 14
-=	Subtracted and assign back the result to left operand	x-=2	x will become 10
*=	Multiplied and assign back the result to left operand	x*=2	x will become 24
/=	Divided and assign back the result to left operand	x/=2	x will become 6
%=	Taken modulus using two operands and assign the result to left operand	x%=2	x will become 0
**=	Performed exponential (power) calculation on operators and assign value to the left operand	x**=2	x will become 144
//=	Performed floor division on operators and assign value to the left operand	x / /= 2	x will become 6

## ***Points to remember about operators...***

✓ ***Same operator may perform a different function depending on the data type of the operands***

□ ***Division operator ( / )behaves differently with integers and floats***

□ ***Multiplication operator( \* ) and addition operator ( + ) behave differently on integers and strings***

# **Expressions and Statements**

## **Expression:**

**A consists of values i.e. constants or variables and operators**

- Examples**

$2+x$

$50.8/4+(20-4)$

## **Statement:**

**A unit of code that a python interpreter can execute**

- Examples**

`>>>print 90`      *# print statement*

`>>>x=60+40`      *# addition/assignment statement*

## ***Points to remember about Expressions.....***

- ❖ ***In case of operands with different data types , implicit typecasting occurs.***

- ***Examples***

```
>>> print 0.8/4+(20-4)  
16.2
```

***All integers are promoted to floats and the final result of the expression is a float value***

- ❖ ***In case of multiple sub expressions in a single expression , general precedence rules apply***

# ***Precedence of Operators***

<b>Operator</b>	<b>Description</b>
<code>**</code>	Exponentiation (raise to the power)
<code>+ , -</code>	unary plus and minus
<code>* , /, %, //</code>	Multiply, divide, modulo and floor division
<code>+ , -</code>	Addition and subtraction
<code>&lt;, &lt;=, &gt;, &gt;=</code>	relational operators for comparision
<code>==, !=</code>	Equality operators
<code>%=, /=, //=, -=, + =, * =, ** =</code>	Assignment operators
not; and; or	Logical operators

# ***While working in Python.....***

- ✓ *Write one python statement per line (Physical Line).*
- ✓ *Comment starts with '#' and ends at the end of a line.*
- ✓ *When entering statement(s) in interactive mode, an extra blank line is treated as the end of the indented block.*
- ✓ *White space in the beginning of line is part of indentation, elsewhere it is not significant*
- ✓ *If a statement runs into multiple physical lines , a ' \ ' is used at the end of a physical line.*

# ***Input-Output Facility***

- *Helps to interact with end user to accomplish the desired task.*
- Functions available for input are :
  - `raw_input()`
  - `input ()`

# *raw\_input()*

- **Syntax**

raw\_input ([prompt])

- **e.g. (*in interactive mode*)**

*>>>x=raw\_input ('Enter your name : ')*

*Enter your name : ABC*

## ***Concatenation of strings using raw\_input( ) method***

```
>>>Input1=raw_input("Enter the first Name: ")
```

```
Enter the first Name: KVS
```

```
>>>Input2=raw_input("Enter the second Name: ")
```

```
Enter the second input: DELHI
```

```
>>>Input1+Input2
```

```
>>>'KVS DELHI'
```

# Addition of two numbers using raw\_input( ) method

```
>>>Input1=raw_input("Enter the first  
input:")
```

```
Enter the first input: 78
```

```
>>>Input2=raw_input("Enter the second  
input:")
```

```
Enter the second input: 100
```

```
>>>Input1+Input2
```

```
>>>'78100'
```

Oops! This is not the desired  
output.

## ***Addition of two numbers using raw\_input( ) method***

```
>>>Input1=int(raw_input("Enter the first input:"))  
Enter the first input: 78
```

```
>>>Input2=int(raw_input("Enter the second input:"))  
Enter the second input: 100
```

```
>>>Input1+Input2  
>>>178
```

## ***raw\_input() -Numeric data***

- **typecast** the string data accepted from user to appropriate Numeric type.

- **e.g.**

```
>>>y=int(raw_input("enter your roll no"))
```

- E.g.
- >>>z = float(raw\_input(" enter the float value"))

# *input()*

- *Syntax*

*input ([prompt])*

- *Eg:*

*x= input ('enter data:')*

*Enter data: 2+ 1/2.0*

*Will supply 2.5 to x*

# **Output**

- *For output in Python we use ‘print’ .*
- *Syntax:*  
*print expression/constant/variable*

- *Example:*

```
>>> print "Hello"
```

*Hello*

```
>>> print 5.5
```

*5.5*

```
>>> print 4+6
```

*10*

## Basic Input Output and Process

1. WAP to add 2 numbers.[ $R=a+b$ ]
2. WAP to add 4 numbers.[ $R=a+b+c+d$ ]
3. WAP to multiply 3 numbers.[ $R=a*b*c$ ]
4. WAP to find average 5 numbers[ $R=(a+b+c+d+e)/5$ ]
5. WAP to display age after 15 years.[ $nage = age + 15$ ]
6. WAP to display  $a^3$  numbers [ $R=a*a*a$ ]
7. WAP to find the area of square. [ $A=a*a$ ]
8. WAP to find the area of rectangle [ $A=a*b$ ]
9. **WAP to find the result  $X^N$  pow()**
10. WAP to find the perimeter of rectangle[ $A=2*(l+b)$  ]
11. *WAP to find the area of circle [ $A=3.14*r*r$ ]*
12. *WAP to find the circumference of circle [ $C=2*3.14*r$ ]*
13. WAP to swap the values of two variables.[ $a = a + b; b = a - b; a = a - b;$ ]
14. WAP to input **Hours, Minutes and Seconds** and display in seconds.  
[ $TS=H*60*60+M*60+S$ ]
15. WAP to input cost and display cost after increasing 25% [ $cost+(cost*25)/100$ ]