



1. Introduction to Python

Subject : Programming with Python (IT3008)

Faculty: Ms. Twinkle Kosambia

Assistant Professor, Computer Science Department,

Asha M. Tarsadia Institute of Computer Science & Technology

Website: utu.ac.in

Syllabus of Programming with Python :

<https://app.utu.ac.in/utuformaccess/utusyllabus.aspx?CF=7&CM=177&SY=1>

Programming with Python (IT3008)



Programming with Python

Subject code : IT3008

Credits : 5

Theory marks : $60 + 40 = 100$

Practical marks : 100 (CIE)

Programming with Python (IT3008)



Reference books

Text book:

1. Allex Martelli, Anna Ravenscroft and Steve Holden, "Python in Nutshell", 3rd Edition, O'Reilly Publication.

Reference books:

1. Magnus Lie Hetland, "Beginning Python From Novice to Professional", Third Edition, Apress, 2017.
2. David Beazley, Brian K. Jones, "Python Cookbook", 3rd edition, O'Reilly Publication, 2016.
3. Brett Slatkin, "Effective Python: 59 Specific Ways to Write Better Python", Novatec, 2016.
4. Mark Lutz "Learning Python", 4th Edition, O'Reilly Publication, 2016.





Course outcomes



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

Unit No.	Unit Name	Course Outcomes					
		CO1	CO2	CO3	CO4	CO5	CO6
1	Introduction to Python	✓					
2	Data Structures		✓				
3	Control Structure and Functions			✓			
4	Object Oriented Programming				✓		
5	Exception Handling and Regular Expression					✓	
6	File Handling						✓

Programming with Python (IT3008)



Unit – I Introduction to Python



UKA TARSADIA
university
Imparting Knowledge. Awakening Wisdom. Transforming Lives.

- Using the Python interpreter, Variables, Identifiers and Keywords, Numbers and Expressions.

Programming with Python (IT3008)



Python : A Complete Programming Solution

- It is open source and completely free
- It runs natively on Windows, Mac OS, linux, and others, as does its standard library and the third party libraries
- Third, it fits quick scripting and large development projects equally well



UKA TARSADIA
university
Imparting Knowledge. Awakening Wisdom. Transforming Lives.



What is Python??

- Python is an open source, interpreted, high-level, general-purpose programming language.
- Python's design philosophy emphasizes code readability with its notable use of significant whitespace.
- Python is dynamically typed and garbage-collected language.
- Python was conceived in the late 1980s as a successor to the ABC language.
- Python was Created by Guido van Rossum and first released in 1991.
- Python 2.0, released in 2000,
- introduced features like list comprehensions and a garbage collection system with reference counting.
- Python 3.0 released in 2008 and current version of python is 3.8.3 (as of June-2020).
- The Python 2 language was officially discontinued in 2020



Why Python?

Python has many advantages

- Easy to learn
- Less code
- Syntax is easier to read
- Open source
- Huge amount of additional open-source libraries
- Some libraries listed below.
- **matplotlib** for plotting charts and graphs
- **BeautifulSoup** for HTML parsing and XML
- **NumPy** for scientific computing
- **pandas** for performing data analysis
- **SciPy** for engineering applications, science, and mathematics
- **Scikit** for machine learning
- **Django** for server-side web development
- And many more.

Programming with Python (IT3008)



Python's Benevolent Dictator For Life

“Python is an experiment in how much freedom programmers need. Too much freedom and nobody can read another's code; too little and expressive-ness is endangered.”

- Guido van Rossum



Programming with Python (IT3008)



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

Brief History of Python

- Invented in the Netherlands, early 90s by Guido van Rossum
- Python was conceived in late 1980s and its implementation was started in December 1989.
- Guido Van Rossum was fan of ‘Monty Python’s flying circus’ a famous TV show in Netherland.
 - Named after Monty Python
 - Based on ABC(All Basic Code) and Modula3.
- Open sourced from the beginning.
- Used by Google from the beginning



Symbol of Python



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.



Programming with Python (IT3008)



Why was Python created??

"My original motivation for creating Python was the perceived need for a higher level language in the Amoeba [Operating Systems] project.

I realized that the development of system administration utilities in C was taking too long. Moreover, doing these things in the Bourne shell wouldn't work for a variety of reasons. ...

So, there was a need for a language that would bridge the gap between C and the shell"

- Guido Van Rossum



Releases

- Created in 1989
- Python 1.0 released in 1994
- Python 2.0 released in 2000
- Python 3.0 released in 2008
- Python 2.7.9 is the recommended version
- 3.12.1 latest version

Python Programming History

A timeline of Python different versions

Python 1.0



Python 1.5



Python 1.6



Python 2.0



Python 2.1



Python 2.2



Python 3.1



Python 3.0



Python 2.6



Python 2.5



Python 2.4



Python 2.3



Python 2.7



Python 3.2



Python 3.3



Python 3.4



Python 3.5



Python 3.6





Features of Python

- Easy-to-learn
- Simple
- Easy-to-maintain
- A broad standard library
- Interactive Mode
- Portable
- Extendable
- Databases



Features of Python(Cont..)

- GUI Programming
- Open Source
- High level language
- Dynamically typed
- Platform Independent
- Procedure and Object-oriented



Applications

- **Web Development**
- **Testing**
- **Data Analysis**
- **Big Data**
- **Computer Graphics**
- **Web Scrapping**
- **Machine Learning**
- **Internet of things**



UKA TARSADIA
university

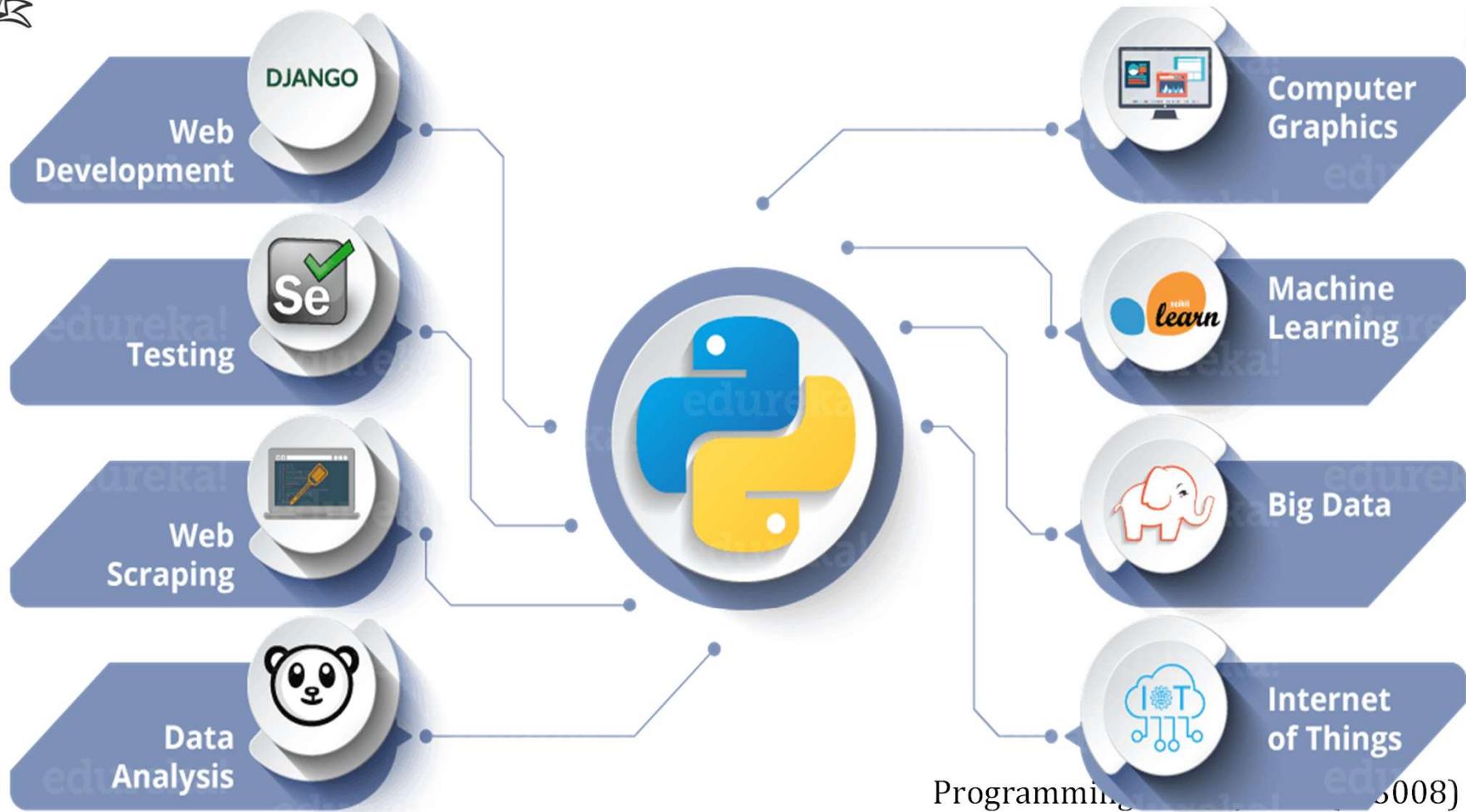
Imparting Knowledge. Awakening Wisdom. Transforming Lives.

Programming with Python (IT3008)



UKA TARSADIA
university

Imparting Knowledge. Awakenin Wisdom. Transforming Lives





Web application

Up next

Autoplay

Python | Print first 10 natural numbers हिंदी में -mysirg.com
MySirG.com 5,548 views 6:44

हैकर क्यों पसंद करते हैं PYTHON
Programming Languages 7:46

Why Python Programming Language is Very Popular ? Better than c++ & java ??
Technology Gyan 166,205 views

IELTS Speaking Sample
IELTS Mantrahs Recommended for you 10:46

WEAR WET SOCKS TO BED
5-Minute Crafts Recommended for you 17:07

28 SIMPLE HACKS THAT WILL CHANGE YOUR LIFE
5-Minute Crafts Recommended for you

Complete Python tutorial in Hindi (2018)
Harshit vashisth



Testing



UKA TARSADIA
university
Inspiring Knowledge. Awakening Wisdom. Transforming Lives.



Programming with Python (IT3008)



Data Analysis



UKA TARSADIA
university
Imparting Knowledge. Awakening Wisdom. Transforming Lives.

python

ORACLE

Spotfire

Data Analysis Using
Python, SQL and Spotfire

A 1 day training with hands on labs & Activities

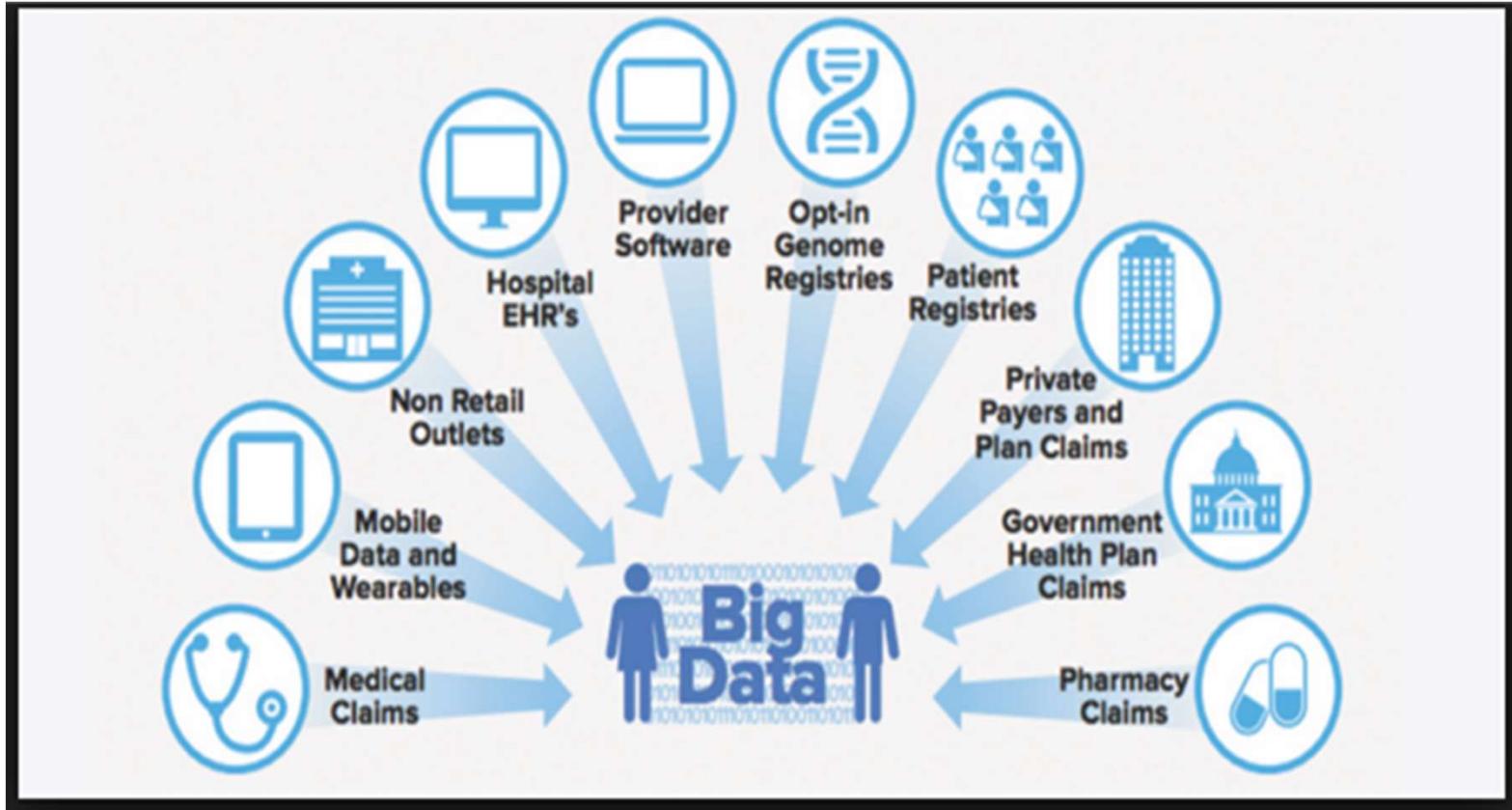
Programming with Python (IT3008)



Big Data



UKA TARSADIA
university
Imparting Knowledge. Awakening Wisdom. Transforming Lives.



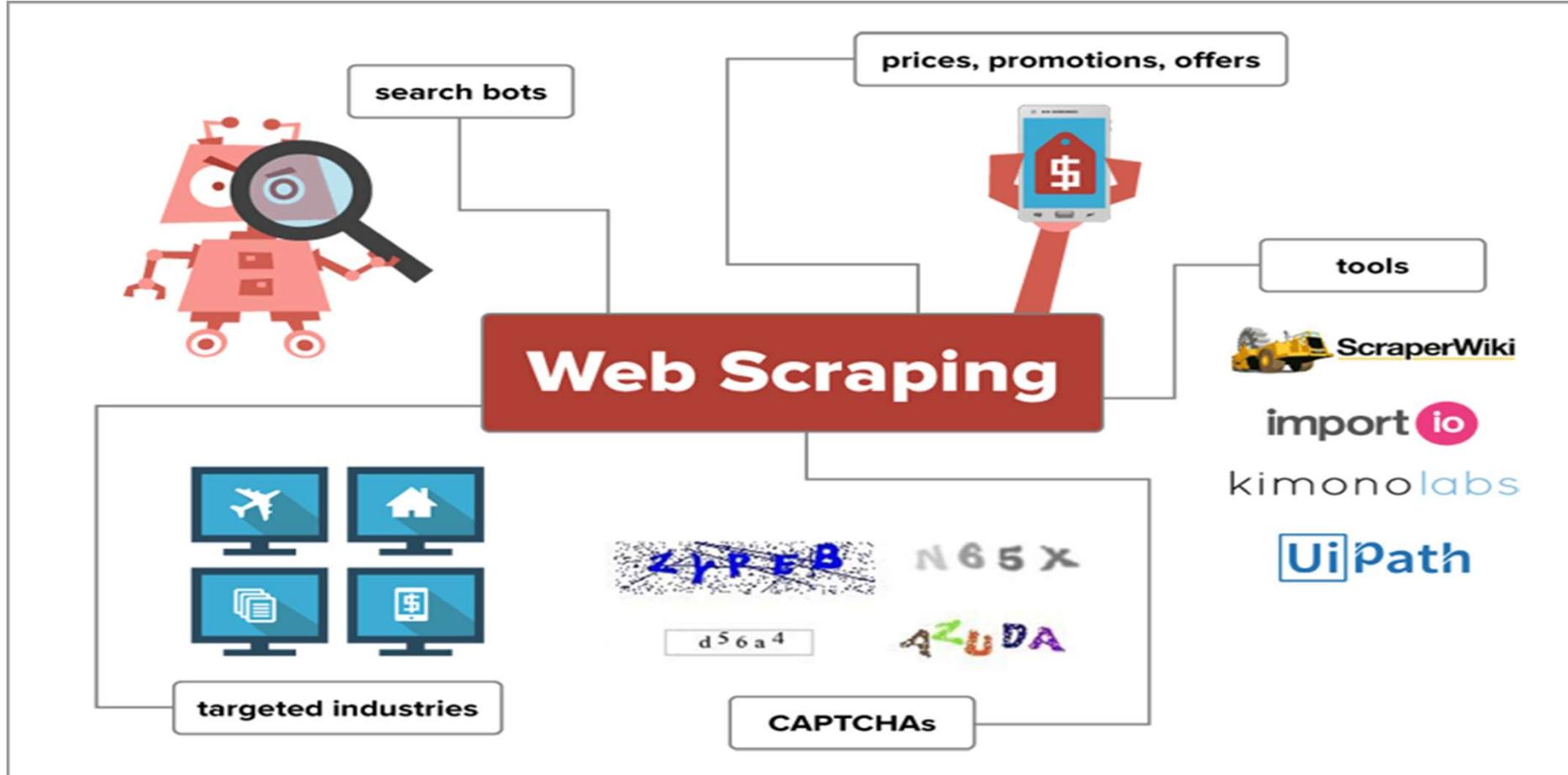
Programming with Python (IT3008)



UKA TARSADIA
university

Creating Wisdom. Transforming Lives.

Web Scrapping





Computer Graphics



Interactive Computer Graphics

Programming with Python (IT3008)



Mobile Application



UKA TARSADIA
university

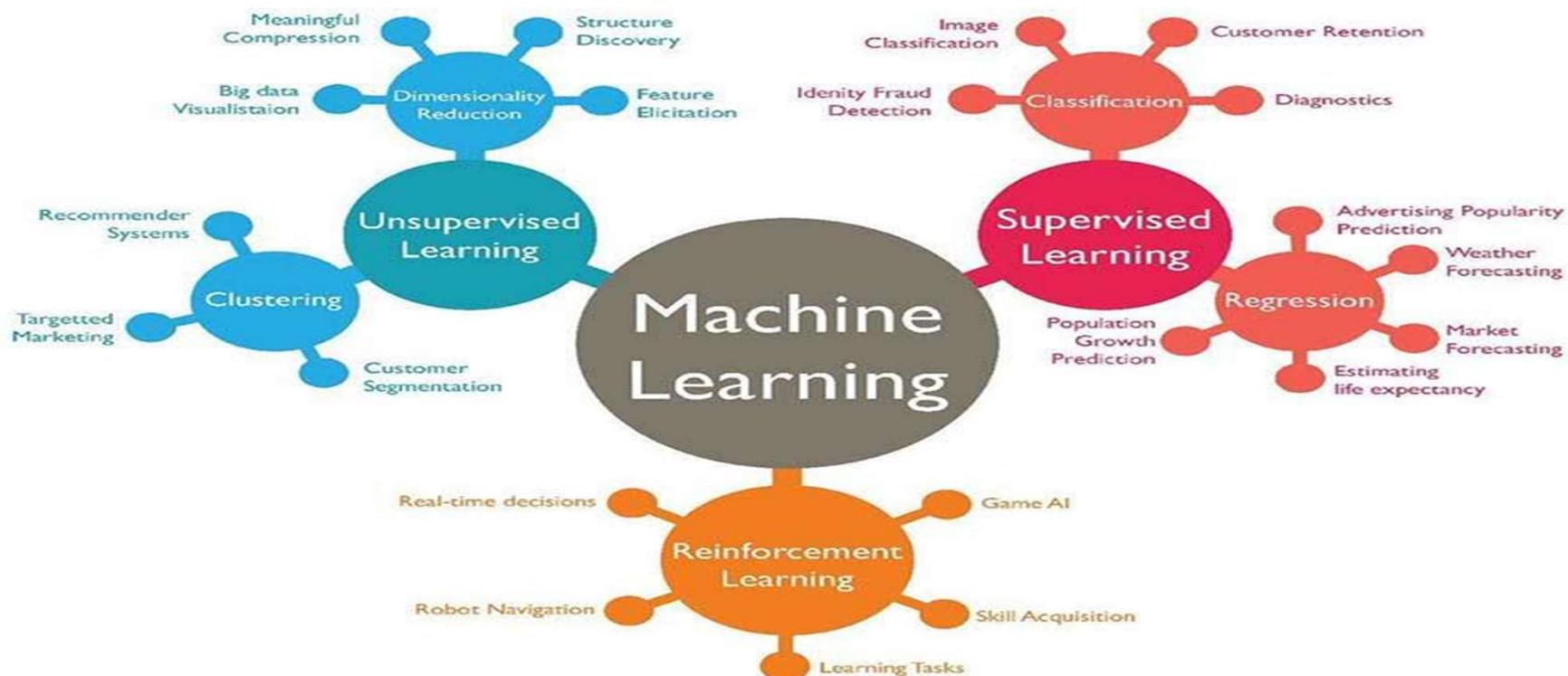
Inspiring Knowledge. Awakening Wisdom. Transforming Lives.



UKA TARSADIA
university

Inspiring Knowledge. Awakening Wisdom. Transforming Lives.

Machine Learning



Programming with Python (IT3008)



Internet of Things



UKA TARSADIA
university
Inspiring Knowledge. Awakening Wisdom. Transforming Lives.



Programming with you (T3008)



Companies using Python

Quora

edX

UBER

Pinterest

Google


Dropbox

YAHOO!

IBM

 Spotify

 reddit

 Instagram

 yelp

 NASA

You  Tube

 Eventbrite

 DISQUS



Introduction To Python

	C	Java	Python
Developed By Whom and When ??	Dennis M. Ritchie In 1972	James Gosling, June 1991.	Guido van Rossum during 1985- 1990
Open Source	No	No	Yes . GNU General Public License (GPL).
Type	Structured	Object-Oriented	Object-oriented
Complied/Interpreted	Compiled	Interpreted	<u>Interpreted</u>
Use For	Application, system, general purpose, low-level operations	Application, business, client-side, general, mobile development, server-side, web	Application, general, web, scripting, artificial intelligence, scientific computing

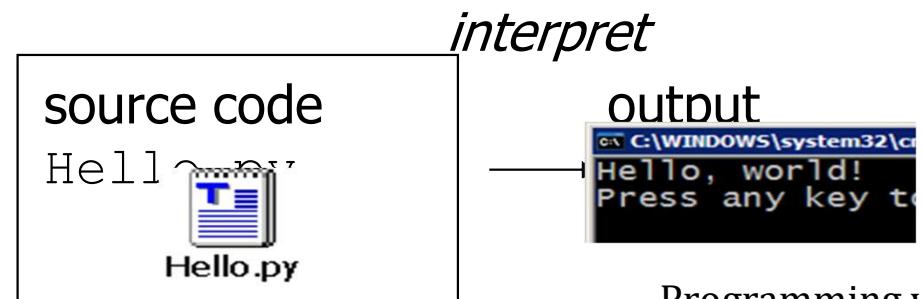
Programming with Python (IT3008)

Compiling and Interpreting

- Many languages require you to *compile* (translate) your program into a form that the machine understands.



- Python is instead directly *interpreted* into native instructions.



Programming with Python





UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

Compiler

- ▶ Create source file using text edit →
- ▶ Use compiler to syntax check and convert source file into binary →
- ▶ Use linker to turn binary files into executable format →
- ▶ Run the resulting executable format file in the operating system.

Programming with Python (IT3008)

The Python Interpreter

- Python is an interpreted language
- The interpreter provides an interactive environment to play with the language
- Results of expressions are printed on the screen

```
>>> 3 + 7  
10  
>>> 3 < 15  
True  
>>> 'print me'  
'print me'  
>>> print 'print me'  
print me  
>>>
```

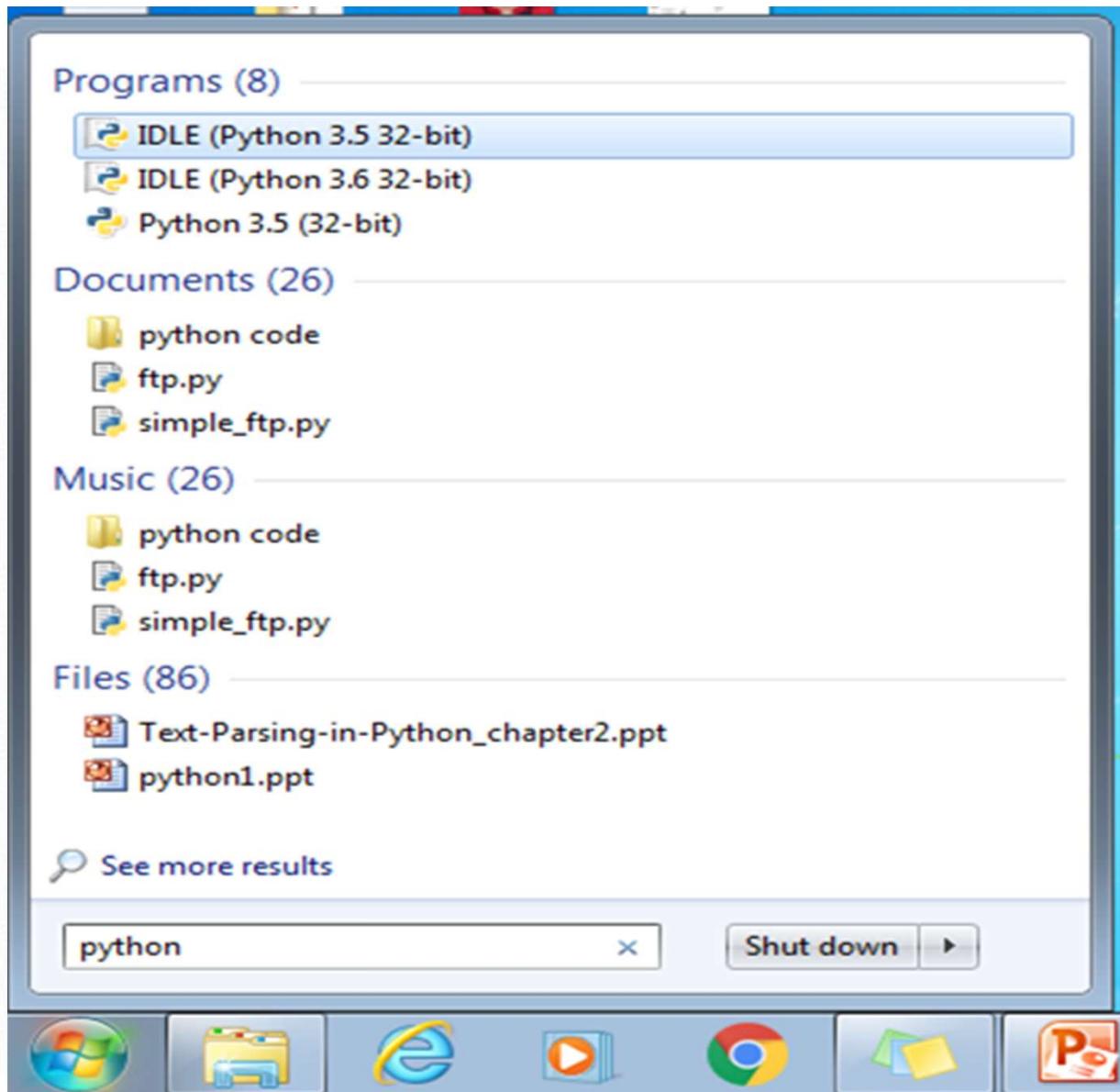


Installing Python

- For Windows & Mac:
 - To install python in windows you need to download installable file from <https://www.python.org/downloads/>
 - After downloading the installable file you need to execute the file.
- For Linux :
 - For ubuntu 16.10 or newer
 - sudo apt-get update
 - sudo apt-get install python3.8
- To verify the installation
 - Windows :
 - python --version
 - Linux :
 - python3 --version (linux might have python2 already installed, you can check python 2 using python --version)
- Alternatively we can use anaconda distribution for the python installation
 - <http://anaconda.com/downloads>
 - Anaconda comes with many useful inbuilt libraries.



After installing the Python Ver#2.7.7, go to start menu then click on python 2.7 in that one you can select python (command line) it is prompt with >>>





IDLE Development Environment

- IDLE is an Integrated DeveLopment Environ-ment for Python, typically used on Windows
- Multi-window text editor with syntax highlighting, auto-completion, smart indent and other.
- Python shell with syntax highlighting.
- Integrated debugger with stepping, persistent breakpoints, and call stack visibility.



IDLE Development Environment



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

74 *Python Shell*

File Edit Shell Debug Options Windows Help

Python 2.3.4 (#53, May 25 2004, 21:17:02) [MSC v.1200 32 bit (Intel)] on win32

Type "copyright", "credits" or "license()" for more information.

Personal firewall software may warn about the connection IDLE makes to its subprocess using this computer's internal loopback interface. This connection is not visible on any external interface and no data is sent to or received from the Internet.

IDLE 1.0.3

```
>>> for i in [x**2 for x in range(5)]:  
     print i  
  
0  
1  
4  
9  
16  
>>> for i in [x**2 for x in range(  
           range([start,] stop[, step]) -> list of integers
```

Ln: 20 Col: 34



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

C:\Users\Your Name>python

Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> print("Hello, World!")

C:\Users\Your Name>python

Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> print("Hello, World!")

Hello, World!



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

```
C:\Users\Your Name>python --version
```

```
C:\Users\Your Name>python helloworld.py
```

```
exit()
```

Programming with Python (IT3008)



Running Interactively on UNIX

On Unix...

```
% python
```

```
>>> 3+3
```

```
6
```

- Python prompts with '>>>'.
- To exit Python (not Idle):
 - In Unix, type CONTROL-D
 - In Windows, type CONTROL-Z + <Enter>
 - Evaluate exit()



http://docs.python.org/

Overview — Python v2.6.1 documentation

modules | index

Python v2.6.1 documentation

Welcome! This is the documentation for Python 2.6.1, last updated Jan 29, 2009.

Parts of the documentation:

- What's new in Python 2.6?**
or all "What's new" documents since 2.0
- Tutorial**
start here
- Using Python**
how to use Python on different platforms
- Language Reference**
describes syntax and language elements
- Library Reference**
keep this under your pillow
- Python HOWTOs**
in-depth documents on specific topics
- Indices and tables:**
- Global Module Index**
quick access to all modules
- General Index**
all functions, classes, terms
- Glossary**
the most important terms explained
- Extending and Embedding**
tutorial for C/C++ programmers
- Python/C API**
reference for C/C++ programmers
- Installing Python Modules**
information for installers & sys-admins
- Distributing Python Modules**
sharing modules with others
- Documenting Python**
guide for documentation authors
- Search page**
search this documentation
- Complete Table of Contents**
lists all sections and subsections



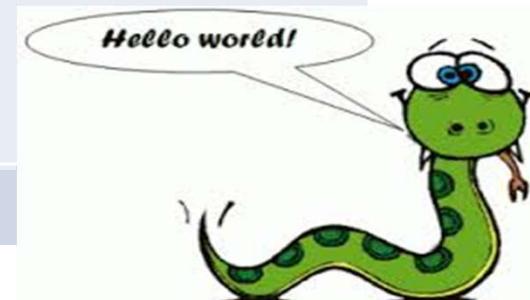
Running Programs on UNIX

- Call python program via the python interpreter
 - % python fact.py
- Make a python file directly executable by
 - Adding the appropriate path to your python interpreter as the first line of your file

```
#!/usr/bin/python
```
 - Making the file executable
 - % chmod a+x fact.py
 - Invoking file from Unix command line
 - % fact.py

Program 1 : Hello World

Language	Program
C	#include<stdio.h> void main() { printf("Hello World"); }
Java	class HelloWorldApp { public static void main(String[] args) { System.out.println("Hello World!"); } }
Python	print "Hello World"





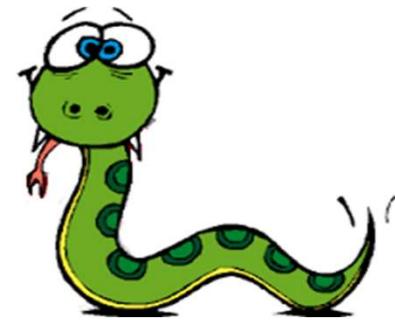
First Lesson In Python



UKA TARSADIA
university
Imparting Knowledge. Awakening Wisdom. Transforming Lives.



Programming with Python





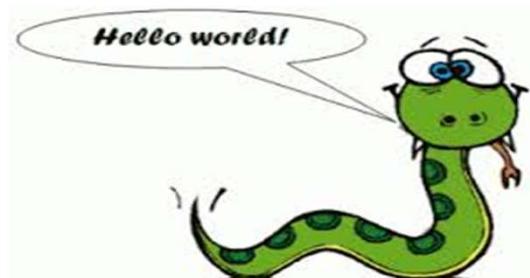
Run /Execute First Python Program

Steps :

1. Open Python Shell
2. Type your code
3. Enter

OR

1. Open Python GUI
2. Click File New –Type program –save with extension .py
3. Run Module





UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

Program 2a: Addition of two numbers

C	Java	Python
#include <stdio.h> void main() { int a,b,sum; a=10; b=10; sum=0; sum=a+b; printf("The sum is %d" ,sum); }	class AddNumbers { public static void main (String args[]) { int x, y, z; x=10; y=10; z = x + y; System.out.println("Sum of entered integers = "+z); } }	a=10 b=10 c=a+b print ("Sum is" ,c)



Lesson 3: Whitespace & comments

Programming with Python (IT3008)



Whitespace(Indentation)

- Whitespace is very important in Python .It is used to structure code.
- Python uses **indentation** instead of braces to determine the scope of expressions
- All lines must be indented the same amount to be part of the scope (or indented more if part of an inner scope)
- This **forces** the programmer to use proper indentation since the indenting is part of the program!

Programming with Python





Indentation

```
# Python code
if foo:
    if bar:
        baz(foo, bar)
else:
    qux()
```

Python embraces indentation

Programming with Python (IT3008)



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.



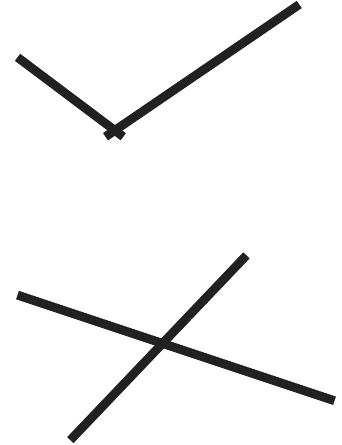
UKA TARSADIA
university

Inspiring Knowledge. Awakening Wisdom. Transforming Lives.

Indentation

```
if 5 > 2:  
    print("Five is greater than two!")
```

```
if 5 > 2:  
    print("Five is greater than two!")
```





UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

Comments

- Single Line Comment use #.
- Entire line till the end should be treated as comment

The screenshot shows a Python code editor window titled "single and multiple line comments in python.py - D:/". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The code area contains the following:

```
# This is a single line comment
print('Hello')

# This is a single line comment
n=10
print(n)
```



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

Comments

- Multi Line Comment use """ """ OR ''' '''
- Enclose a block of line as comments.

"""This is also a perfect example of multi-line comments"""



Comments

- Docstring is short for documentation string.
 - It is a string that occurs as the first statement in a module, function, class, or method definition.
 - We must write what a function/class does in the docstring.
 - Triple quotes are used while writing docstrings.
- For example:

```
script.py  IPython Shell
1 def double(num):
2     """Function to double the value"""
3     return 2*num
```

'3008)



Comments

```
# A traditional one line comment  
"""  
Any string not assigned to a variable is  
considered a comment.  
This is an example of a multi-line comment.  
"""  
  
"This is a single line comment"
```



UKA TARSADIA
university
Imparting Knowledge. Awakening Wisdom. Transforming Lives.



Lesson 2: No Explicit Variables Declaration

- Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.
- Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.
- Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable. The equal sign (=) is used to assign values to variables.



Lesson 2: No Explicit Variables Declaration

Examples :

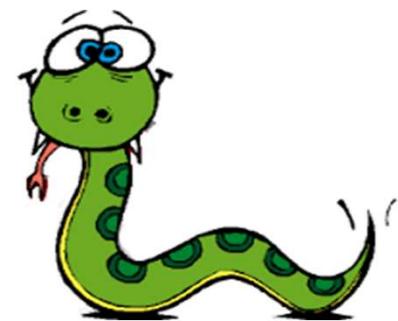
my_int=7

my_float=1.23

my_bool=True



Programming with Python





Python Variables

```
>>> count = 1          #an integer assignment  
>>> mileage = 12345.12 # a float assignment  
>>> name ='kate'      # a string assignment
```

Use single or double quotation marks to indicate that you are declaring a string variable

```
>>> print name, count, mileage  
kate 1 12345.12
```

Use a print statement to write the results of operations.

A value can be assigned to several variables simultaneously:

```
>>>a=b=c=d=1  
  
>>> print a,b,c,d  
1 1 1 1
```

Multiple Assignment

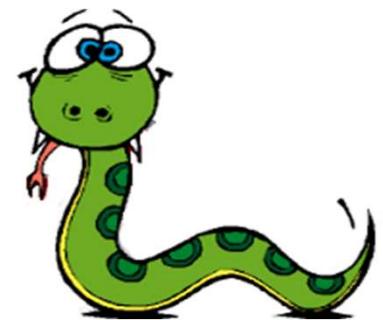
- Python allows you to assign a **single** value to **several** variables **simultaneously**. For example:
`a = b = c = 1`
- Here, an **integer** object is created with the **value 1**, and all three variables are assigned to the same memory location. You can also assign multiple objects to multiple variables. For example:
`a, b, c = 1, 2, "john"`
- Here, two integer objects with values 1 and 2 are assigned to variables a and b, and one string object with the value "john" is assigned to the variable c.



Operators in Python

- Python language supports the following types of operators.
 - Arithmetic Operators
 - Comparison (Relational) Operators
 - Assignment Operators
 - Logical Operators
 - Bitwise Operators
 - Membership Operators
 - Identity Operators

Programming with Python





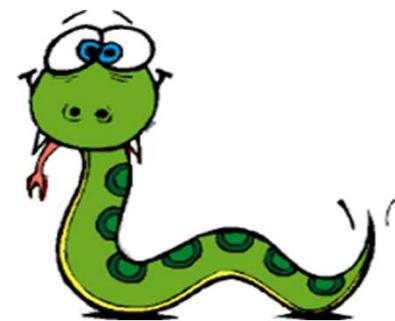
Operators in Python (Arithmetic Operator)

- Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication etc.



UKA TARSADIA
university
Inspiring Knowledge. Awakening Wisdom. Transforming Lives

Programming with Python





Operators in Python

(Arithmetic Operator-Overview)



Operator	Meaning	Example
+	Add two operands or unary plus	$x + y$ +2
-	Subtract right operand from the left or unary minus	$x - y$ -2
*	Multiply two operands	$x * y$
/	Divide left operand by the right one (always results into float)	x / y
%	Modulus - remainder of the division of left operand by the right	$x \% y$ (remainder of x/y)
//	Floor division - division that results into whole number adjusted to the left in the number line	$x // y$
**	Exponent - left operand raised to the power of right	$x**y$ (x to the power y)



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

- Note : consider A = 10 and B = 3

Operator	Description	Example	Output
+	Addition	A + B	13
-	Subtraction	A - B	7
/	Division	A / B	3.333333333333335
*	Multiplication	A * B	30
%	Modulus return the remainder	A % B	1
//	Floor division returns the quotient	A // B	3
**	Exponentiation	A ** B	$10 * 10 * 10 = 1000$

Python 3.6 (32-bit)



```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)]
on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a=5
>>> b=2
>>> print(a+b)
7
>>> print(a-b)
3
>>> print(a*b)
10
>>> print(a/b)
2.5
>>> print(a//b)
2
>>> print(a%b)
1
>>> print(a**b)
25
>>>
```



Program 3 : Demonstration of Arithmetic Operator

```
a=5
```

```
b=2
```

```
print(a+b)
```

```
print(a-b)
```

```
print(a*b)
```

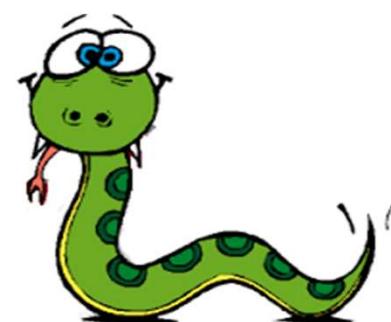
```
print(a/b)
```

```
print(a%b)
```

```
print(a//b)
```

```
print(a**b)
```

Programming with Python

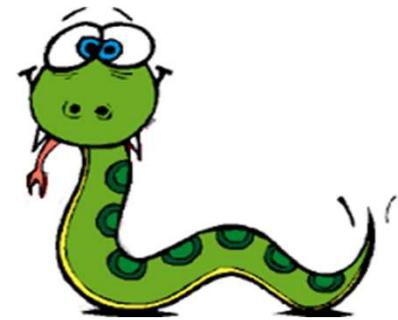




Operators in Python (Comparison Operators)

- Comparison operators are used to compare values.
- It either returns True or False according to the condition.

Programming with Python



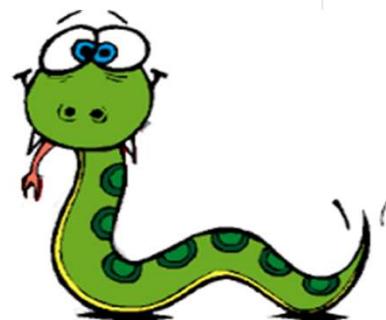


UKA TARSADIA
University

In Transforming Lives

Comparison Operator

Operator	Meaning	Example
>	Greater than - True if left operand is greater than the right	$x > y$
<	Less than - True if left operand is less than the right	$x < y$
==	Equal to - True if both operands are equal	$x == y$
!=	Not equal to - True if operands are not equal	$x != y$
>=	Greater than or equal to - True if left operand is greater than or equal to the right	$x >= y$
<=	Less than or equal to - True if left operand is less than or equal to the right	$x <= y$



Select Python 3.6 (32-bit)



Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)]
on win32

Type "help", "copyright", "credits" or "license" for more information.

```
>>> a=5
>>> b=2
>>> print("a>b",a>b)
a>b True
>>> print("a<b:",a<b)
a<b: False
>>> print("a==b",a==b)
a==b False
>>> print("a!=b",a!=b)
a!=b True
>>> print("a>=b",a>=b)
a>=b True
>>> print("a<=b",a<=b)
a<=b False
>>> ■
```



Program 4 : Demonstration of Arithmetic comparison

```
# Ordering
a > b
a >= b
a < b
a <= b

# Equality/Difference
a == b
a != b
```

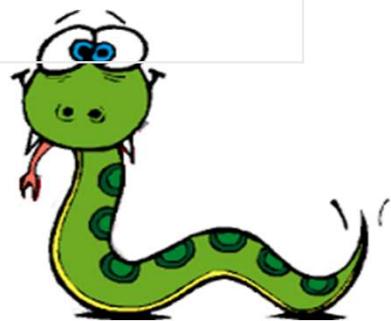


Operators in Python

(Logical Operators-Overview)

Operator	Meaning	Example
and	True if both the operands are true	x and y
or	True if either of the operands is true	x or y
not	True if operand is false (complements the operand)	not x

Programming with Python

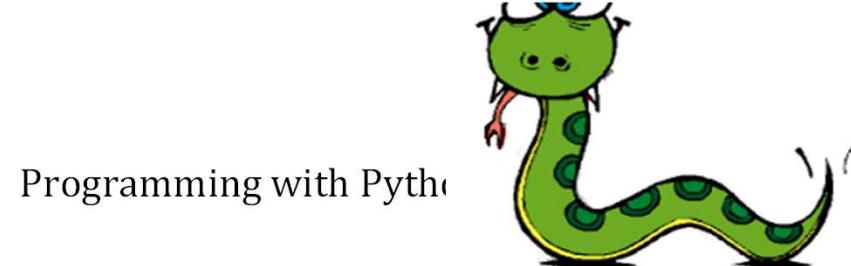




Operators in Python (Logical Operators)

- Logical operators are the and, or, not operators.
- Note : consider A = 10 and B = 3

Operator	Description	Example	Output
and	Returns True if both statements are true	A > 5 and B < 5	True
or	Returns True if one of the statements is true	A > 5 or B > 5	True
not	Negate the result, returns True if the result is False	not (A > 5)	False



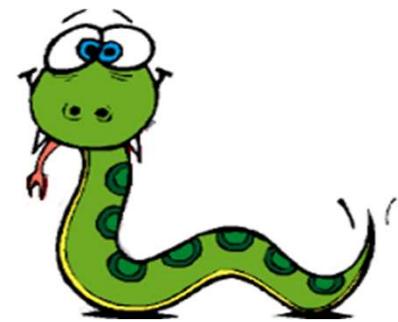
Programming with Python



Operators in Python (Bitwise Operators)

- Bitwise operators act on operands as if they were string of binary digits. It operates bit by bit, hence the name.

Programming with Python



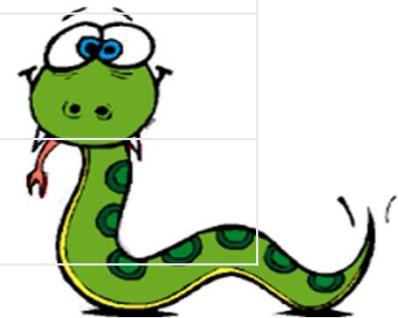


Operators in Python

(Bitwise Operator-Overview)

Operator	Meaning	Example
&	Bitwise AND	$x \& y = 0$ (0000 0000)
	Bitwise OR	$x y = 14$ (0000 1110)
~	Bitwise NOT	$\sim x = -11$ (1111 0101)
^	Bitwise XOR	$x ^ y = 14$ (0000 1110)
>>	Bitwise right shift	$x >> 2 = 2$ (0000 0010)
<<	Bitwise left shift	$x << 2 = 42$ (0010 1000)

Programming with Python





UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

- if $a = 60$; and $b = 13$;
- $a = 0011\ 1100$
- $b = 0000\ 1101$
- $a \& b = 0000\ 1100$
- $a | b = 0011\ 1101$
- $a ^ b = 0011\ 0001$
- $\sim a = 1100\ 0011$



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

Operator	Description	Example
& Binary AND	Operator copies a bit to the result if it exists in both operands	(a & b) (means 0000 1100)
 Binary OR	It copies a bit if it exists in either operand.	(a b) = 61 (means 0011 1101)
^ Binary XOR	It copies the bit if it is set in one operand but not both.	(a ^ b) = 49 (means 0011 0001)
~ Binary Ones Complement	It is unary and has the effect of 'flipping' bits.	(~a) = -61 (means 1100 0011 in 2's complement form due to a signed binary number.)
<< Binary Left Shift	The left operand's value is moved left by the number of bits specified by the right operand.	a << = 240 (means 1111 0000)
>> Binary Right Shift	The left operand's value is moved right by the number of bits specified by the right operand.	a >> = 15 (means 0000 1111)



Operators in Python

(Assignment Operators)

- Assignment operators are used in Python to assign values to variables.
- `a = 5` is a simple assignment operator that assigns the value 5 on the right to the variable `a` on the left.
- There are various compound operators in Python like `a += 5` that adds to the variable and later assigns the same. It is equivalent to `a = a + 5`.

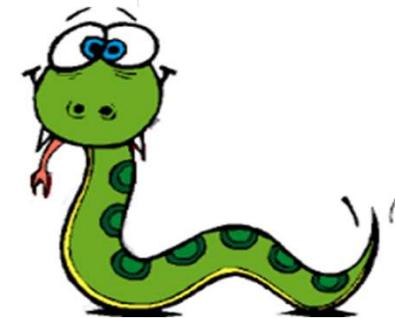
Programming with Python





Program 4 : Demonstration of Assignment Operator

Operator	Example	Equivalent to
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
//=	x //= 5	x = x // 5
**=	x **= 5	x = x ** 5
&=	x &= 5	x = x & 5
=	x = 5	x = x 5
^=	x ^= 5	x = x ^ 5
>>=	x >>= 5	x = x >> 5
<<=	x <<= 5	x = x << 5



Program 4 : Demonstration of Assignment Operator

Operator	Example	Equivalent to
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5



Program 4 : Demonstration of Assignment Operator

//=	x //= 5	x = x // 5
**=	x **= 5	x = x ** 5
&=	x &= 5	x = x & 5
=	x = 5	x = x 5
^=	x ^= 5	x = x ^ 5
>>=	x >>= 5	x = x >> 5
1/8/2019<<=	x <<= 5	x = x << 5



The screenshot shows a Python 3.6 (32-bit) terminal window. The code entered is:

```
>>> print(c)
?
>>> c+=3
>>> print(c)
10
>>> 
```

Programming with Python (IT3008)

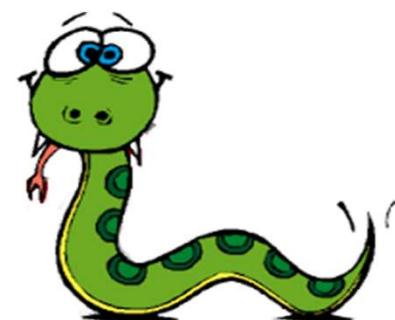


UKA TARSADIA
university
Imparting Knowledge. Awakening Wisdom. Transforming Lives.

Operators in Python (Identity Operators –Special Operators)

- is and is not are the identity operators in Python.
- They are used to check if two values (or variables) are located on the same part of the memory.
- Two variables that are equal does not imply that they are identical

Programming with Python





Operators in Python

(Identity Operators-Overview)

Operator	Meaning	Example
is	True if the operands are identical (refer to the same object)	x is True
is not	True if the operands are not identical (do not refer to the same object)	x is not True

Programming with Python





Identity & Member Operators



UKA TARSADIA
university
Imparting Knowledge. Awakening Wisdom. Transforming Lives.

- Identity Operator
- Note : consider $A = [1,2]$, $B = [1,2]$ and $C=A$

Operator	Description	Example	Output
is	Returns True if both variables are the same object	$A \text{ is } B$ $A \text{ is } C$	FALSE TRUE
is not	Returns True if both variables are different object	$A \text{ is not } B$	TRUE

- Member Operator
- Note : consider $A = 2$ and $B = [1,2,3]$

Operator	Description	Example	Output
in	Returns True if a sequence with the specified value is present in the object	$A \text{ in } B$	TRUE
not in	Returns True if a sequence with the specified value is not present in the object	$A \text{ not in } B$	FALSE



Program 5 : Demonstration of Identity Operators

```
x1 = 5
```

```
y1 = 5
```

```
x2 = 'Hello'
```

```
y2 = 'Hello'
```

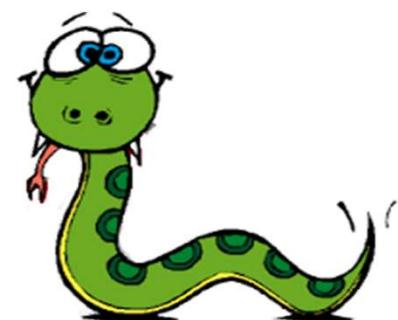
```
x3 = [1,2,3]
```

```
y3 = [1,2,3]
```

```
print(x1 is not y1)
```

```
print(x2 is y2)
```

```
print(x3 is y3)
```



Administrator: C:\Windows\system32\cmd.exe - python



C:\Users\Administrator>c:

C:\Users\Administrator>cd..

C:\>cd..

C:\>python

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> x1=5

>>> y1=5

>>> x2='Hello'

>>> y2='Hello'

>>> x3=[1,2,3]

>>> y3=[1,2,3]

>>> print(x1 is not y2)

True

>>> print(x1 is not y1)

False

>>> print(x2 is y2)

True

>>> print(x3 is y3)

False

>>>



UKA TARSADIA
university

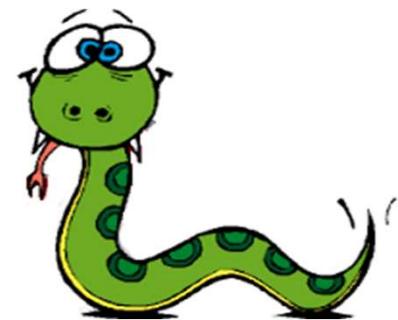
Inspiring Knowledge. Awakening Wisdom. Transforming Lives.

Operators in Python

(MemberShip Operators –Special Operators)

- **in** and **not in** are the membership operators in Python. They are used to test whether a value or variable is found in a sequence (string, list, tuple, set and dictionary).
- In a dictionary we can only test for presence of key, not the value.

Programming with Python



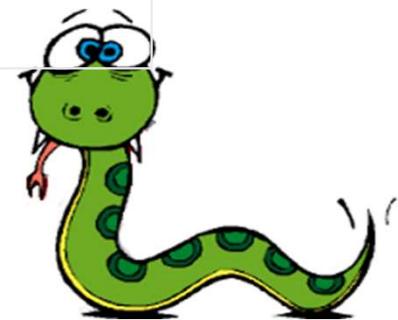


Operators in Python

(Identity Operators-Overview)

Operator	Meaning	Example
in	True if value/variable is found in the sequence	5 in x
not in	True if value/variable is not found in the sequence	5 not in x

Programming with Python





Program 5 : Demonstration of Membership Operators

```
x = 'Hello world'
```

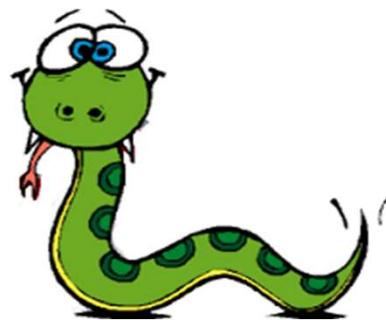
```
print('H' in x)
```

```
print('Hello' not in x)
```

```
print('hello' in x)
```



Programming with Python

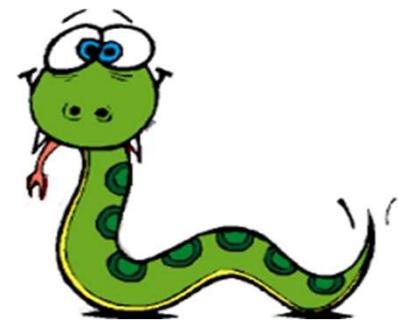




Lesson 4: New Operators and Case Sensitive

- Python has two new operators
 - Identity
 - Membership
- Python is case sensitive

Programming with Python





Operator Precedence



UKA TARSADIA
university

Inspiring Knowledge. Awakening Wisdom. Transforming Lives.

Operator	Description
<code>**</code>	Exponentiation (raise to the power)
<code>~ + -</code>	Complement, unary plus and minus (method names for the last two are <code>+@</code> and <code>-@</code>)
<code>* / % //</code>	Multiply, divide, modulo and floor division
<code>+ -</code>	Addition and subtraction
<code>>> <<</code>	Right and left bitwise shift
<code>&</code>	Bitwise 'AND'
<code>^ </code>	Bitwise exclusive 'OR' and regular 'OR'
<code><= < > >=</code>	Comparison operators
<code><> == !=</code>	Equality operators
<code>= %= /= //=- += *= **=</code>	Assignment operators
<code>is is not</code>	Identity operators
<code>in not in</code>	Membership operators
<code>not or and</code>	Logical operators



Input and Output Function in Python

- Output Function

print()

- Input Function

input()



Program 2b: Addition of two numbers

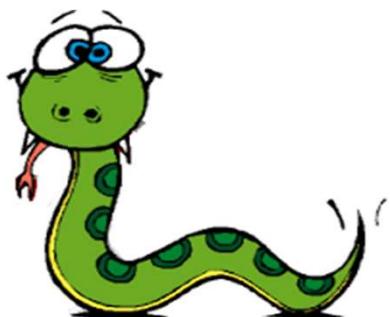
C	Java	Python
#include <stdio.h> void main() { int a,b,sum; sum=0; printf(Enter number a%d"); scanf("%d", &a); printf(Enter number b%d"); scanf("%d", &b); sum=a+b; printf("The sum is %d" ,sum); }	import java.util.Scanner; class AddNumbers { public static void main(String args[]) { int x, y, z; System.out.println("Enter two integers to calculate their sum "); Scanner in = new Scanner(System.in); x = in.nextInt(); y = in.nextInt(); z = x + y; System.out.println("Sum of entered integers = "+z); } }	a=input("Enter First no") b=input("Enter second number") sum=a+b print("Sum is",sum)



Lesson 5: By default datatype of data taken by using Input Function is string

- As by default datatype of data taken by using Input Function is string.
- Change the datatype to int to perform mathematical operations.

```
a=int(input("Enter First no"))
b=int(input("Enter second number"))
sum=a+b
print("Sum is",sum)
```

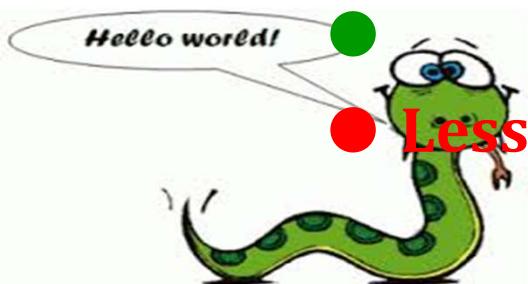




Recap 1

Recall all lessons we have learned till now

- **Lesson 1 :** No Semicolon ,No { }
- **Lesson 2 :** No need of Explicit Variable Declaration
- **Lesson 3 :** Whitespace act as delimiters and single line comments # and multi line comment with """ """
- **Lesson 4 :** New operators : Identity and Membership
- **Lesson 5 :** print() and input() are I/O functions.



Python is case sensitive language

Programming with Python (IT3008)



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

Exercise 1

1. WAP to display your name and college name
2. Modify first program add your designation with your name ,but it should be displayed on new line.
3. WAP to calculate area of circle. Define pi=3.14 and radius 2
4. Modify the program number 2 in such a way that area will be rounded number.
5. Modify the program number 4 – take radius as input from user.

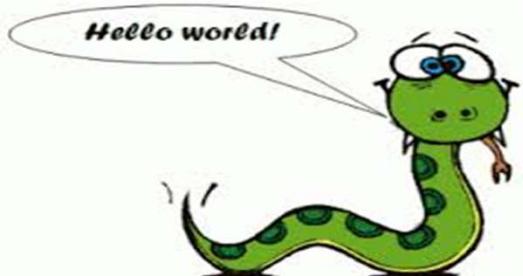


Solution of Exercise 1

- Program to display your name and college name

```
print( "Smita Dange")
```

```
print( "Fr.C.R.I.T ,Vashi")
```



Programming with Python (IT3008)



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

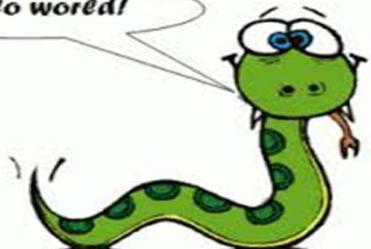
Solution of Exercise 2

- Modify first program add your designation with your name ,but it should be displayed on new line.

```
print( "Smita Dange. \n Asst. Professor")
```

```
print( "Fr.C.R.I.T ,Vashi")
```

Hello world!



Programming with Python (IT3008)



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

Solution of Exercise 3

- Program to calculate area of circle. Define pi=3.14 and radius.

```
pi=3.14      # pi ,r =3.14 ,2  Multiple Assignments
```

```
r=2
```

```
area= 3.14 *r*r  # 3.14*r**2
```

```
print("Area is ",area)
```



Programming with Python (IT3008)



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

Solution of Exercise 4

- Modify the program number 2 in such a way that area will be rounded number.

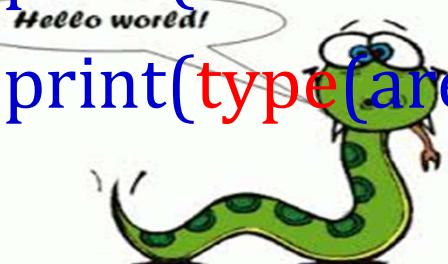
pi=3.14

r=2

```
area= 3.14 *r*r  # 3.14*r**2
```

```
print("Area is ",int(area))
```

```
print(type(area))
```



Programming with Python (IT3008)



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

Solution of Exercise 5

1. Modify the program number 4 – take radius as input from user.

pi=3.14

```
r=int(input("Enter the radius"))
```

```
area= 3.14 *r*r # 3.14*r**2
```

```
print("Area is ",int(area))
```



Programming with Python (IT3008)



New Things learn from Exercise

- \n -for new line
- Multiple Assignment
- Conversion of datatype
- x=int(x)
- y=float(y)
- Checking datatype of variable
- type()
- Operator Evaluation Sequence



Python Variables



UKA TARSADIA
university
Imparting Knowledge. Awakening Wisdom. Transforming Lives.

- A Python variable is a reserved memory location to store values.
- Unlike other programming languages, Python has no command for declaring a variable.
- A variable is created the moment you first assign a value to it.
- Python uses Dynamic Typing so,
 - We need not to specify the data types to the variable as it will internally assign the data type to the variable according to the value assigned.
 - we can also reassign the different data type to the same variable, variable data type will change to new data type automatically.
 - We can check the current data type of the variable with **type(variablename)** in-built function.
- Rules for variable name
 - Name can not start with digit
 - Space not allowed
 - Can not contain special character
 - Python keywords not allowed
 - **Should** be in lower case



Example of Python variable

demo.py

```
1 x = 10
2 print(x)
3 print(type(x))
4
5 y = 123.456
6 print(y)
7
8 x = "engineering and technology"
9 print(x)
10 print(type(x))
```

Output

```
10
int
123.456
engineering and technology
str
```



Python Keywords and Identifiers



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

- Keywords are predefined, reserved words used in Python programming that have special meanings to the compiler.
- We cannot use a keyword as a variable name, function name, or any other identifier. They are used to define the syntax and structure of the Python language.
- All the keywords except True, False and None are in lowercase and they must be written as they are. The list of all the keywords is given below.



Python Keywords List

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



Python Identifiers



UKA TARSADIA
university

Imparting Knowledge. Awakening Wisdom. Transforming Lives.

Here, language is a variable (an identifier) which holds the value 'Python'.

```
language = 'Python'
```

We cannot use keywords as variable names as they are reserved names that are built-in to Python. For example,

```
continue = 'Python'
```



Python Numbers

- Number data types store numeric values. They are immutable data types, which means that changing the value of a number data type results in a newly allocated object.
- Different types of Number data types are :
 - Int
 - Float
 - complex



Int type

- int (Integers) are the whole number, including negative numbers but not fractions. In Python, there is no limit to how long an integer value can be.
- **Example 1: Creating int and checking type**

```
1 num = -8
2 # print the data type
3 print(type(num))
```

Output:

```
<class 'int'>
```



Float type

- This is a real number with floating-point representation. It is specified by a decimal point. Some examples of numbers that are represented as floats are 0.5 and -7.823457.
- They can be created directly by entering a number with a decimal point, or by using operations such as division on integers. Extra zeros present at the number's end are ignored automatically.
- **Example 1: Creating float and checking type**

```
1 num = 3/4
2 # print the data type
3 print(type(num))
```

Output:

```
<class 'float'>
```



Complex type

- A complex number is a number that consists of the real and imaginary parts. For example, $2 + 3j$ is a complex number where 2 is the real component, and 3 multiplied by j is an imaginary part.
- **Example 1: Creating Complex and checking type**

```
1 num = 6 + 9j
2 print(type(num))
```

Output:

```
<class 'complex'>
```