

## Objective – Module 1



### Python Introduction

- Why Python and where to use it?
- What is Python and history of Python?
- Discussion about Python 2 and Python 3
- Set up Python environment for development
- Discuss about IDE's like IDLE, Pycharm and Enthought Canopy
- Demonstration on Python Installation
- Discussion about unique feature of Python

### Python Programming

- Write first Python Program
- Start programming on interactive shell.
- Using Variables, Keywords
- Interactive and Programming technique
- Comments and document interlude in Python

Slide 1

## What is Scripting Language?



A scripting language is a “wrapper” language that integrates OS functions.

The interpreter is a layer of software logic between your code and the computer hardware on your machine.

#### Wiki Says:

“Scripts” are distinct from the core code of the application, which is usually written in a different language, and are often created or at least modified by the end-user. Scripts are often interpreted from source code or bytecode.

The “program” has an executable form that the computer can use directly to execute the instructions.

The same program in its human-readable source code form, from which executable programs are derived (*e.g., compiled*)

Python is scripting language, fast and dynamic.

Slide 2

[www.ethans.co.in](http://www.ethans.co.in)

## History of Python

**Ethan's**  
Learn from experts



✓ Python was developed by **Guido van Rossum** in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science (*Centrum Wiskunde & Informatica aka CWI*) in the Netherlands.

✓ Python is inherited from ABC programming and has many features including Interactive programming, object oriented, exceptional handling and many more.

✓ Python was named on 'Monty Python's Flying Circus' a comedy series created by the comedy group Monty Python.

✓ Python Version 2.0 was released in 2000, with many major new features including a full garbage collector and support for unicode.

✓ A Java-based version of Python exists in Jython and used to work with Java code. Similarly Iron Python, a C# version exists for the .Net.

✓ Python is called 'scripting language' because of it's scalable interpreter, but actually it is much more than that.

Slide 3

[www.ethans.co.in](http://www.ethans.co.in)

## Why Python?

**Ethan's**  
Learn from experts

Easy to read	✓ Python scripts have clear syntax, simple structure and very few protocols to remember before programming.
Easy to Maintain	✓ Python code is easily to write and debug. Python's success is that its source code is fairly easy-to-maintain.
Portable	✓ Python can run on a wide variety of Operating systems and platforms and providing the similar interface on all platforms.
Broad Standard Libraries	✓ Python comes with many prebuilt libraries apx. 21K
High Level programming	✓ Python is intended to make complex programming simpler. Python deals with memory addresses, garbage collection etc internally.
Interactive	✓ Python provide an interactive shell to test the things before implementation. It provide the user the direct interface with Python.
Database Interfaces	✓ Python provides interfaces to all major commercial databases. These interfaces are pretty easy to use.
GUI programming	✓ Python supports GUI applications and has framework for Web. Interface to tkinter, WXPYthon, Django in Python make it

Slide 4

[www.ethans.co.in](http://www.ethans.co.in)

**Python Features**

Ethan's  
Learn from experts

- ✓ High Level language
- ✓ Interactive and efficient
- ✓ No compilation (Interpreter language)
- ✓ No type declarations (Hold anything)
- ✓ Automatic memory management
- ✓ High-level data types and operations
- ✓ Object-oriented programming
- ✓ Automated garbage collection
- ✓ Fewer restrictions and rules
- ✓ Wide portability
- ✓ Extendible and customizable
- ✓ Easy Debugging Techniques
- ✓ Many editors are available for programming

key  
features  
of  


Slide 5

[www.ethans.co.in](http://www.ethans.co.in)

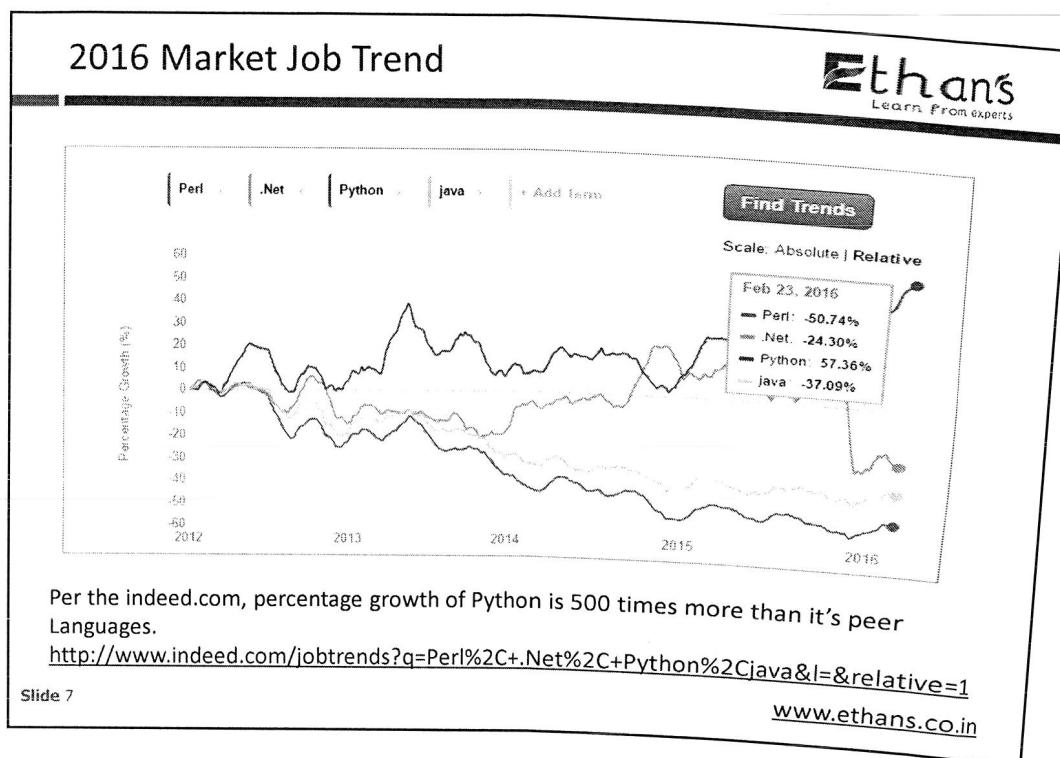
**Who uses Python?**

Ethan's  
Learn from experts

- ✓ Google search engine and many other products in Google makes extensive use of Python and it also employs Python's creator Guido van Rossum.
- ✓ The popular YouTube video sharing service is largely written in Python.
- ✓ Disney uses Python in many of their creative processes.
- ✓ Mozilla uses Python to explore their extensive code base and releases tons of open source packages built in python.
- ✓ Intel, Juniper, Cisco, Hewlett-Packard, Seagate, Qualcomm use Python for automated hardware testing.
- ✓ Dropbox file hosting service is implemented using Python, Guido van Rossum now working here.
- ✓ Morgan Stanley, BNP, JP Morgan, Citibank apply Python for financial market forecasting.
- ✓ NASA, Los Alamos, JPL, use Python for scientific programming tasks.
- ✓ The NSA uses Python for cryptography and intelligence analysis.
- ✓ Linux Weekly News is published using a Web application written in Python using the Quixote framework.
- ✓ The Red Hat Linux distribution uses Python for its installer (anaconda) and configuration utilities

Slide 6

[www.ethans.co.in](http://www.ethans.co.in)



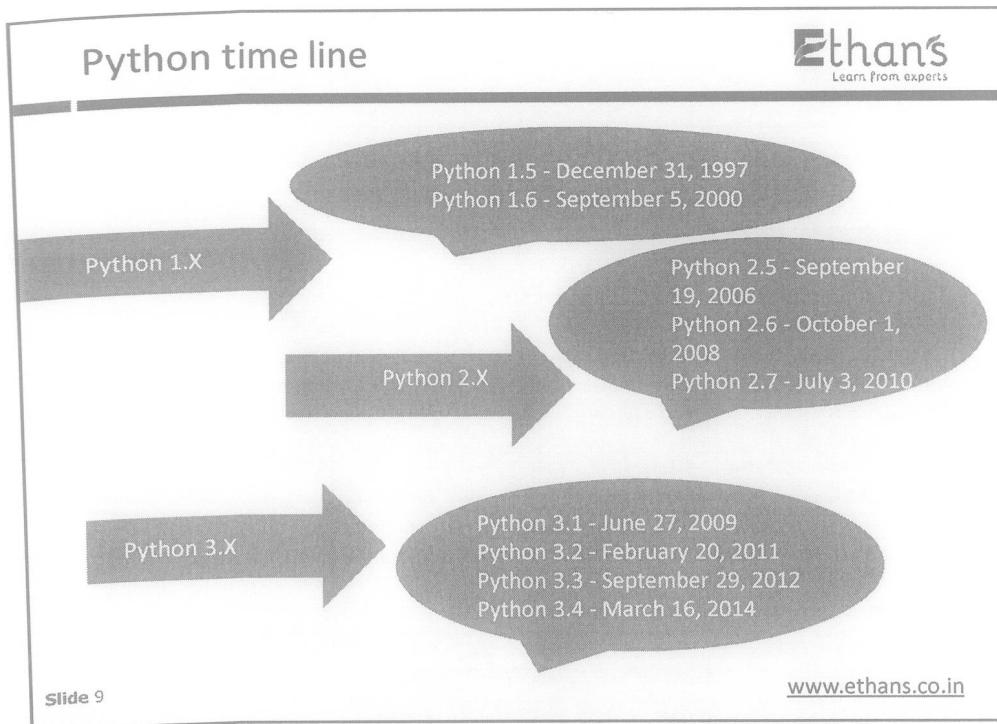
**Job In Big Data space**

Ethans Learn From experts

Skill	% of Big Data Jobs Mentioning This Skill Set (multiple responses allowed)	% Growth in Demand For This Skill Set Over the Previous Year
Java	6.62%	63.30%
Structured query language	5.86%	76.00%
Apache Hadoop	5.45%	49.10%
Software development	4.70%	60.30%
Linux	4.10%	76.60%
Python	3.99%	96.90%
NoSQL	2.74%	34.60%
Data warehousing	2.73%	68.80%
UNIX	2.43%	61.90%
Software as a Service	2.38%	54.10%

Source: <http://www.forbes.com/sites/louis columbus/2014/12/29/where-big-data-jobs-will-be-in-2015/>

Slide 8 [www.ethans.co.in](http://www.ethans.co.in)



**Python 2 or Python 3**

- ✓ There are couple of difference in Python2 and Python3, starting from syntax to added functionality.
- ✓ Syntax Difference example:
  - In Python 2, print is a function but doesn't required mandatory parenthesis.
  - In Python 3, print() is a function and has mandatory parentheses.
- ✓ Logic difference example:
  - In Python 2, division with integer returns integer, ex: 5/2 return 2 not 2.5
  - In Python 3, implicit division work accordingly, the above division return 2.5 as expected.
- ✓ In this Python course, we are going to use Python 2.7.X prefer to have version 2.7.9.
- ✓ There are couple of reasons to use Python 2 instead of Python 3, few of them are:
  - ✓ The library support for new Version 3.x are not adequate.
  - ✓ Most of the companies software's are using version 2.x and therefore it is better to say Python2 is most widely used version.

[www.ethans.co.in](http://www.ethans.co.in)

Slide 10

## How to install Python?

**Ethans**  
Learn from experts

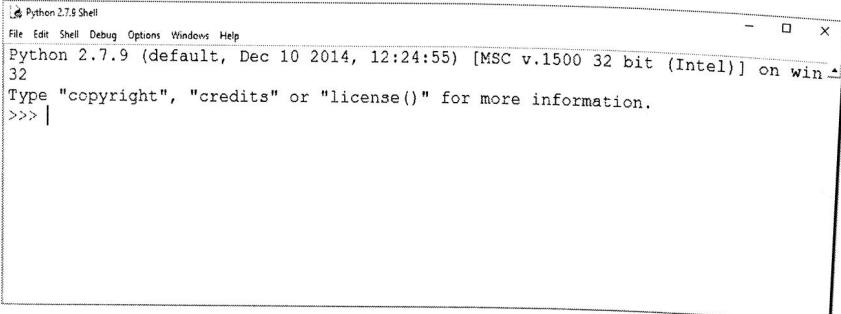
- ✓ Python is pre-installed on almost every Unix systems, including Linux and MAC.
- ✓ In case if you need latest version , you need to update the package installation of Python with the package Installation commands of the specific OS. (In case of any help ,student can take necessary assistance from Absolute classes)
- ✓ On Windows, Python version 2.7.X need to installed explicitly.
- ✓ Latest versions are available at <https://www.python.org/downloads/>
- ✓ Installation is done as per the instruction given by the installer.
- ✓ One Python is installed on Windows user need to modify the PATH variable to get it referred on DOS prompt.
- ✓ To check whether the installation is done run the command python -V. for

```
C:\Ethans>python -V
Python 2.7.9
```

Slide 11 [www.ethans.co.in](http://www.ethans.co.in)

## IDLE Session

**Ethans**  
Learn from experts



```
Python 2.7.9 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> |
```

Best match

 Idle Run command
 IDLE (Python GUI) Desktop app

Slide 12 [www.ethans.co.in](http://www.ethans.co.in)

**Interactive shell**



C:\Ethans>python  
Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (Intel)]  
n32  
Type "help", "copyright", "credits" or "license" for more information.  
>>>

**Slide 13** [www.ethans.co.in](http://www.ethans.co.in)

**Get Started**



**Starting Python**

```
C:\Users\jatin> python
Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
```

**Hello, World!**

Interactive Mode Programming

```
>>> print "Hello, Python!"
Hello, Python!
```

Scripting Mode Programming

```
#!/usr/bin/python
print "Hello, Python!"
```

Executing program

```
$ chmod +x hello.py
$ python hello.py
```

**Slide 14** [www.ethans.co.in](http://www.ethans.co.in)

## Write first Program

**Ethan's**  
Learn from experts

```
"""
Documentation interlude
"""

# Comment 1

print 'Hello World' # comment 2

"""

Multiline comment
Multiline comment
"""

File Edit Format Run Options Windows Help
    Python Shell
    Check Module Alt+X
    Run Module F5
Documentation interlude
```

Slide 15 [www.ethans.co.in](http://www.ethans.co.in)

## Introduction Variables

**Ethan's**  
Learn from experts

### Python Variables

- Variables starts with a letter capital A to Z or small a to z or an underscore (\_) followed by zero or more letters, underscores and digits (0 to 9).
- Invalid Identifiers → special characters such as @, \$ and %
- Should not starts with number
- Should not names as pre-defined reserved keywords
- Variables are unlimited in length. Case is significant.

Slide 16 [www.ethans.co.in](http://www.ethans.co.in)

**Command and multiline statement**

**Ethans**  
Learn from experts

- Statements in Python typically end with a new line. Python does, however, allow the use of the line continuation character (\) to denote that the line should continue.
- Comments in Python is created with # sign
- Python doesn't support multiline command though an another way of doing it with documentation interlude
- Statements contained within the [], {} or () brackets do not need to use the line continuation character.

Slide 19 [www.ethans.co.in](http://www.ethans.co.in)

**Taking Help**

**Ethans**  
Learn from experts

```
>>> help('print')
The "print" statement
*****
print_stmt ::= "print" ([expression ("," expression)* [","])
               | ">>" expression [(", expression)+ [","]])
```

"print" evaluates each expression in turn and writes the resulting object to standard output (see below). If an object is not a string, it is first converted to a string using the rules for string conversions. The (resulting or original) string is then written. A space is written before each object is (converted and) written, unless the output system believes it is positioned at the beginning of a line. This is the case (1) when no characters have yet been written to standard output, (2) when the last character written to standard output is a whitespace character except "", or (3) when the last

Slide 20 [www.ethans.co.in](http://www.ethans.co.in)

## Pre-defined reserved keywords



and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

Slide 17

[www.ethans.co.in](http://www.ethans.co.in)

## Indentation language



### Python Indentations

- Python blocks of code are formed by line indentation.
- The number of spaces in the indentation in block should be same with others statements within the block

#### Correct Example

```
A = 10
if A:
    print "True"
else:
    print "False"
```

#### Incorrect Example

```
A = 10
if A:
    print "Answer"
    print "True"
else:
    print "Answer"
    print "False"
```

Slide 18

[www.ethans.co.in](http://www.ethans.co.in)

**execfile**

Ethan's  
Learn from experts

- Statements in Python typically end with a new line. Python does, however, allow the use of the line continuation character (\) to denote that the line should continue.
- Comments in Python is created with # sign
- Python doesn't support multiline command though an another way of doing it with documentation interlude
- Statements contained within the [], {} or () brackets do not need to use the line continuation character.

Slide 21 [www.ethans.co.in](http://www.ethans.co.in)

**Quiz:**

Ethan's  
Learn from experts

1 – From the following, what are the invalid variables?

- myVariable
- Var1
- X
- X
- \_x
- 1var
- my#Variable

2 – What will be the output of below program?

```
print 'One',
print 'Two'
# print 'Threee'
...
print 'Four'
```

Slide 22 [www.ethans.co.in](http://www.ethans.co.in)

**Fact at a glance**

**Ethans**  
Learn from experts

### Facts at a glance

- Python syntax is line oriented and space sensitive rather than free format
- Python's built-in types include numeric primitives and container types
- Integers hold arbitrary precision values
- Both floating-point and complex numbers supported in the language
- Python has expected set of operators

Slide 23 [www.ethans.co.in](http://www.ethans.co.in)

**Interesting Fact**

**Ethans**  
Learn from experts

- Monty Python's Flying Circus is a British sketch comedy series created by the comedy group Monty Python and broadcast by the BBC from 1969 to 1974.
- The first episode was recorded on 7 September and broadcast on 5 October 1969 on BBC One, with 45 episodes airing over four series from 1969 to 1974, plus two episodes for German TV.
- The members of Monty Python were highly educated.
- Terry and Michael are Oxford graduates.
- Eric Idle, John Cleese, and Graham Chapman attended Cambridge University.
- American-born member Terry Gilliam is an Occidental College graduate.

Source - Wiki

Slide 24 [www.ethans.co.in](http://www.ethans.co.in)

## Objective – Module 2



### Core Objects and built in functions

- Python Core Objects and builtin functions
- Number Object and operations
- String Object and Operations
- List Object and Operations
- Tuple Object and operations
- Dictionary Object and operations
- Set object and operations
- Boolean Object and None Object
- Different data Structures, data processing

Slide 25

## Core Objects



- Numbers
- Strings
- List
- Tuple
- Dictionary -
- Set
- Files
- None
- Boolean

- ex: 1, 1.1, 3 + 4j and 200L
- ex: 'Ethans' "Tech"
- ex: ['Ethans', "Tech"]
- ex: ('Ethans', "Tech")
- ex: {'Institute': 'Ethans'}
- ex: set([1,2,3,4])
- ex: file = open('file.txt')
- ex: None
- ex: True, False

Slide 26

[www.ethans.co.in](http://www.ethans.co.in)

## Assignment Operator

Ethans Learn from experts

```
>>> number = 10
>>> name = 'Ethan'
>>> location = 'Pune'
>>> number1 = number2 = number3 = 1
>>> # Multiple assignment
>>> number1, number2 = 10, 20
```

variables

assignment operator

expression

```
month = "May";
age = 18;
sum = 2 + 4;
average = (2 + 4 + 6) / 3;
```

Slide 27 [www.ethans.co.in](http://www.ethans.co.in)

## Numbers

Ethans Learn from experts

- ✓ Integers
- ✓ Float
- ✓ Long
- ✓ Complex

```
>>> number = 10
>>> # Number is an object of interger class
>>>
>>> number = 10.1
>>> # Number is an object of float class
>>>
>>> number = 3 + 4j
>>> # Number is an object of complex class
>>>
>>> number = 200987654323456789876543L
>>> # Number is an object of long class
```

Slide 28 [www.ethans.co.in](http://www.ethans.co.in)

## type function



```
>>> number_int = 10
>>> type(number_int)
<type 'int'>
>>>
>>> number_int = 2147483647
>>> type(number_int)
<type 'int'>
>>>
>>> number_int = 2147483648
>>> type(number_int)
<type 'long'>
>>> number_comp = 3 + 4j
>>> type(number_comp)
<type 'complex'>
>>> number_float = 3.7
>>> type(number_float)
<type 'float'>
```

Slide 29

[www.ethans.co.in](http://www.ethans.co.in)

## Builtin functions



Built-in Functions				
abs()	divmod()	input()	open()	staticmethod()
all()	enumerate()	int()	ord()	str()
any()	eval()	isinstance()	pow()	sum()
basestring()	execfile()	issubclass()	print()	super()
bin()	file()	iter()	property()	tuple()
bool()	filter()	len()	range()	type()
bytearray()	float()	list()	raw_input()	unichr()
callable()	format()	locals()	reduce()	unicode()
chr()	frozenset()	long()	reload()	vars()
classmethod()	getattr()	map()	repr()	xrange()
cmp()	globals()	max()	reversed()	zip()
compile()	hasattr()	memoryview()	round()	__import__()
complex()	hash()	min()	set()	
delattr()	help()	next()	setattr()	
dict()	hex()	object()	slice()	
dir()	id()	oct()	sorted()	

Slide 30

[www.ethans.co.in](http://www.ethans.co.in)

help

**Ethans**  
Learn from experts

```
>>> help('abs')
Help on built-in function abs in module __builtin__:

abs(...)
    abs(number) -> number

    Return the absolute value of the argument.

>>> abs(-10)
10
>>> abs(-10.1)
10.1
>>> abs(-1000L)
1000L
>>> abs(3+4j)
5.0
```

Slide 31 [www.ethans.co.in](http://www.ethans.co.in)

Builtin help

**Ethans**  
Learn from experts

```
>>> help('__builtin__')
Help on built-in module __builtin__:

NAME
    __builtin__ - Built-in functions, exceptions, and other objects.

FILE
    (built-in)

DESCRIPTION
    Noteworthy: None is the 'nil' object; Ellipsis represents '...' in slices.
```

Slide 32 [www.ethans.co.in](http://www.ethans.co.in)

## Number functions



```
>>> int('1010', 2)
10
>>> float(10)
10.0
>>> long(12)
12L
>>> complex(1, 2)
(1+2j)
```

```
>>> cmp(1,2)
-1
>>> cmp(3,2)
1
>>> cmp(2,2)
0
```

```
>>> isinstance(1, int)
True
>>> isinstance(1.1, float)
True
>>> isinstance(1L, float)
False
>>> isinstance(complex(1,2), complex)
True
```

Slide 33

[www.ethans.co.in](http://www.ethans.co.in)

## Operation on numbers



```
>>> 10 + 2
12
>>> 10 - 2
8
>>> 10 * 2
20
>>> 10 / 2
5
>>> 10 * 2 ** 2 #operator precedence
40
>>> (10 * (2 + (6 * 7)) * 2) # operator associativity
880
```

Slide 34

[www.ethans.co.in](http://www.ethans.co.in)

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

Slide 35 [www.ethans.co.in](http://www.ethans.co.in)

Operation on numbers		Ethan's Learn from experts
<pre>&gt;&gt;&gt; -5 ** 2 -25 &gt;&gt;&gt; (-5) ** 2 25 &gt;&gt;&gt; 5 -- 2 7 &gt;&gt;&gt; bin(10) '0b1010' &gt;&gt;&gt; bin(-10) '-0b1010' &gt;&gt;&gt; bin(11) '0b1011' &gt;&gt;&gt; bin(15) '0b1111'</pre>	<pre>&gt;&gt;&gt; 10 &lt;&lt; 2 40 &gt;&gt;&gt; 10 &gt;&gt; 2 2 &gt;&gt;&gt; 10 &amp; 2 2 &gt;&gt;&gt; 10   2 10 &gt;&gt;&gt; 10 ^ 2 8 &gt;&gt;&gt; 10 / 3 3 &gt;&gt;&gt; 10.0/3 3.333333333333335</pre>	

Slide 36 [www.ethans.co.in](http://www.ethans.co.in)

## String

Ethan's  
Learn from experts

```
>>> name = 'Ethans'
>>> location = "Pune" # Double quotes
>>> expertTraining = """
Python
Hadoop
Selenium
DevOps
Informatica
ETL
"""

```

Slide 37 [www.ethans.co.in](http://www.ethans.co.in)

## String Functions

Ethan's  
Learn from experts

```
>>> type(name)
<type 'str'>
>>> str(10)
'10'
>>> repr(10)
'10'
>>> `10`
'10'
>>> isinstance('Ethan', str)
True
>>> isinstance('Ethan', basestring)
True
>>> ord('a')
97
>>> chr(65)
'A'
>>> len(name)
6

```

Slide 38 [www.ethans.co.in](http://www.ethans.co.in)

## String Operations

Ethan's  
Learn From experts

```
>>> name
'Ethans'
>>> lname = 'Technologies'
>>>
>>> print name + lname
EthansTechnologies
>>> print '*' * 20, "\n" + name + lname + "\n", '*' * 20
*****
EthansTechnologies
*****
```

Slide 39 [www.ethans.co.in](http://www.ethans.co.in)

## String Indexing

Ethan's  
Learn From experts

```
>>> name[0]
'E'
>>> name[-1]
's'
>>> name[3]
'a'
>>> name[4]
'n'
>>> name[5]
's'
>>> name[len(name)]
```

Traceback (most recent call last):  
 File "<pyshell#101>", line 1, in <module>  
 name[len(name)]  
IndexError: string index out of range

Slide 40 [www.ethans.co.in](http://www.ethans.co.in)