miTu skillologies

Python Programming - I

Short, Unique and Mysterious

Q

Why is the Programming Language named so?

Monty Python's *Flying Circus*

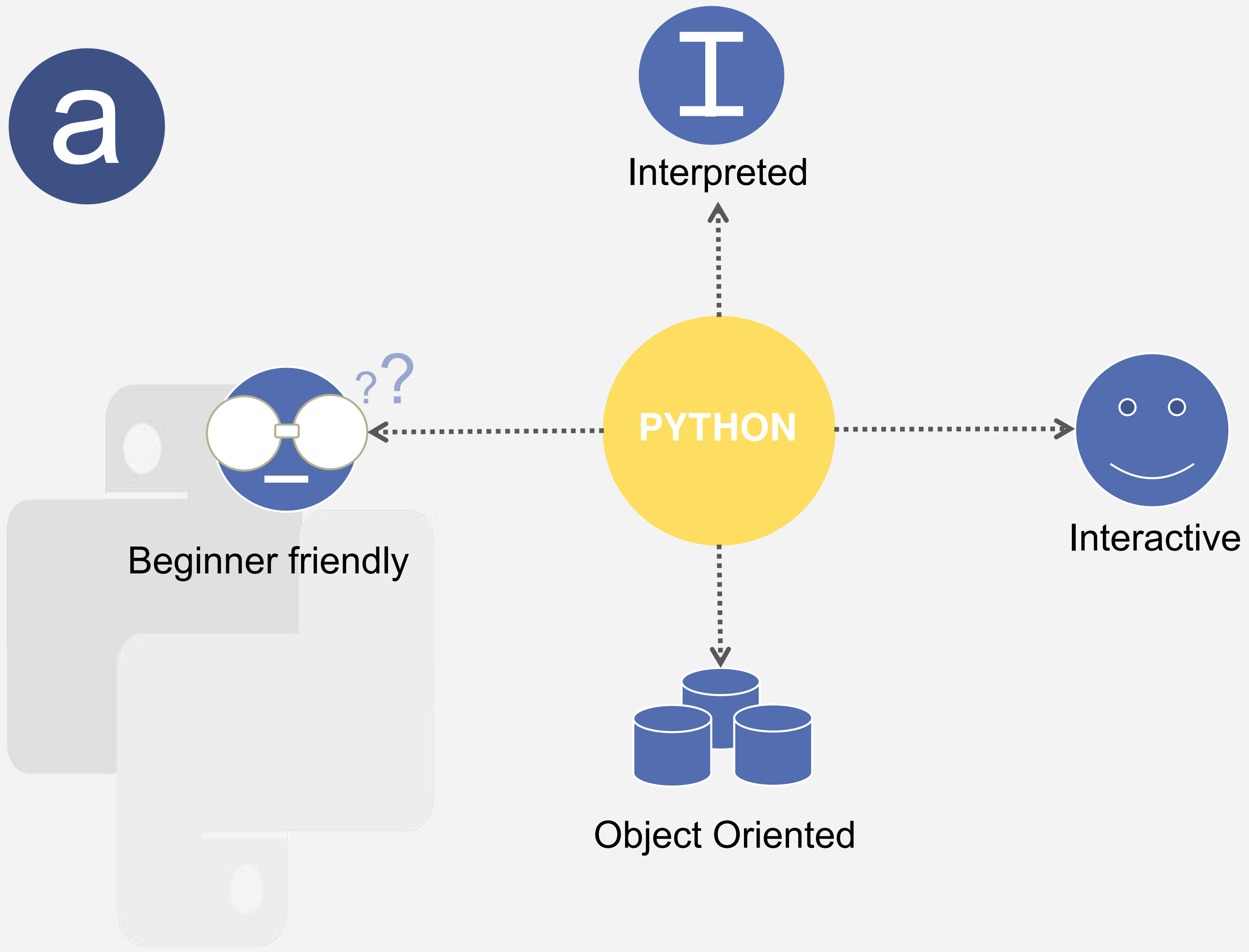
"At the time when he began implementing Python, Guido van Rossum was also reading the published scripts from "Monty Python's Flying Circus" (a BBC comedy series from the seventies, in the unlikely case you didn't know). It occurred to him that he needed a name that was **short**, **unique**, and **slightly mysterious**, so he decided to call the language Python."

-Charlie Cheever
(Quora User & Founder)

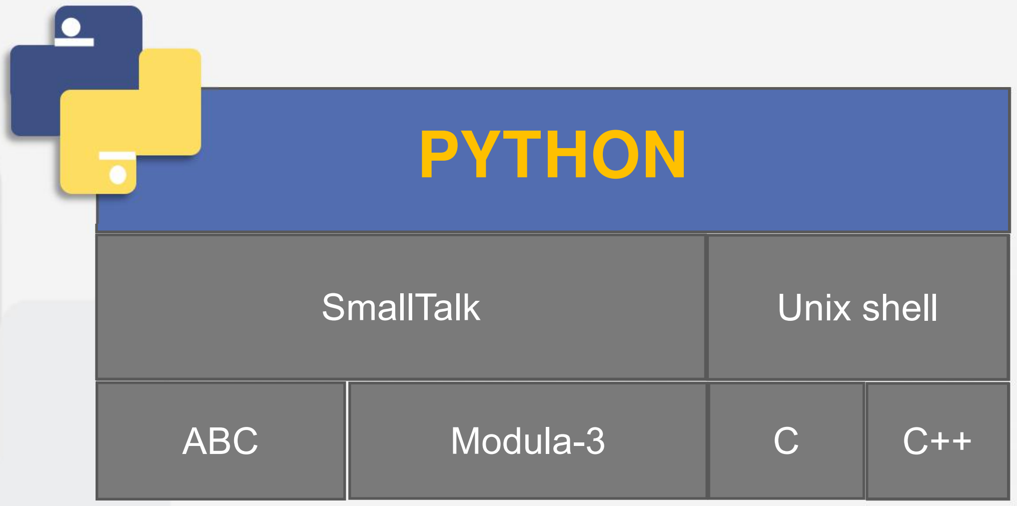
Guido van Rossum

Q

What should I know before starting?


a
Interpreted
PYTHON
Beginner friendly
Interactive
Object Oriented

BUILDING BLOCKS of PYTHON



Note - The placement of Building blocks has no relevance for dependent blocks

CONTENTS

Installation & Configuration

- In latest LTS version of Ubuntu 16.04, Python comes installed with OS. Make sure you've updated version before starting

- To check the version of Python installed using following command on terminal

```
python --version
```

- In order to update you must perform foll. steps

Open web browser and go to https://www.python.org/downloads/

Download appropriate Source file as per OS & Extract files

$

Run **./configure** script in terminal of with respect to your extracted location, then use **make** & **make install**

Installation & Configuration

SETTING UP PATH:

- Programs and other executable files can be in many directories, so operating systems provide a search path that lists the directories that the OS searches for executables.
- The path is stored in an environment variable, which is a named string maintained by the operating system. This variable contains information available to the command shell and other programs.
- The path variable is named as PATH in Unix or Path in Windows (Unix is case-sensitive; Windows is not).

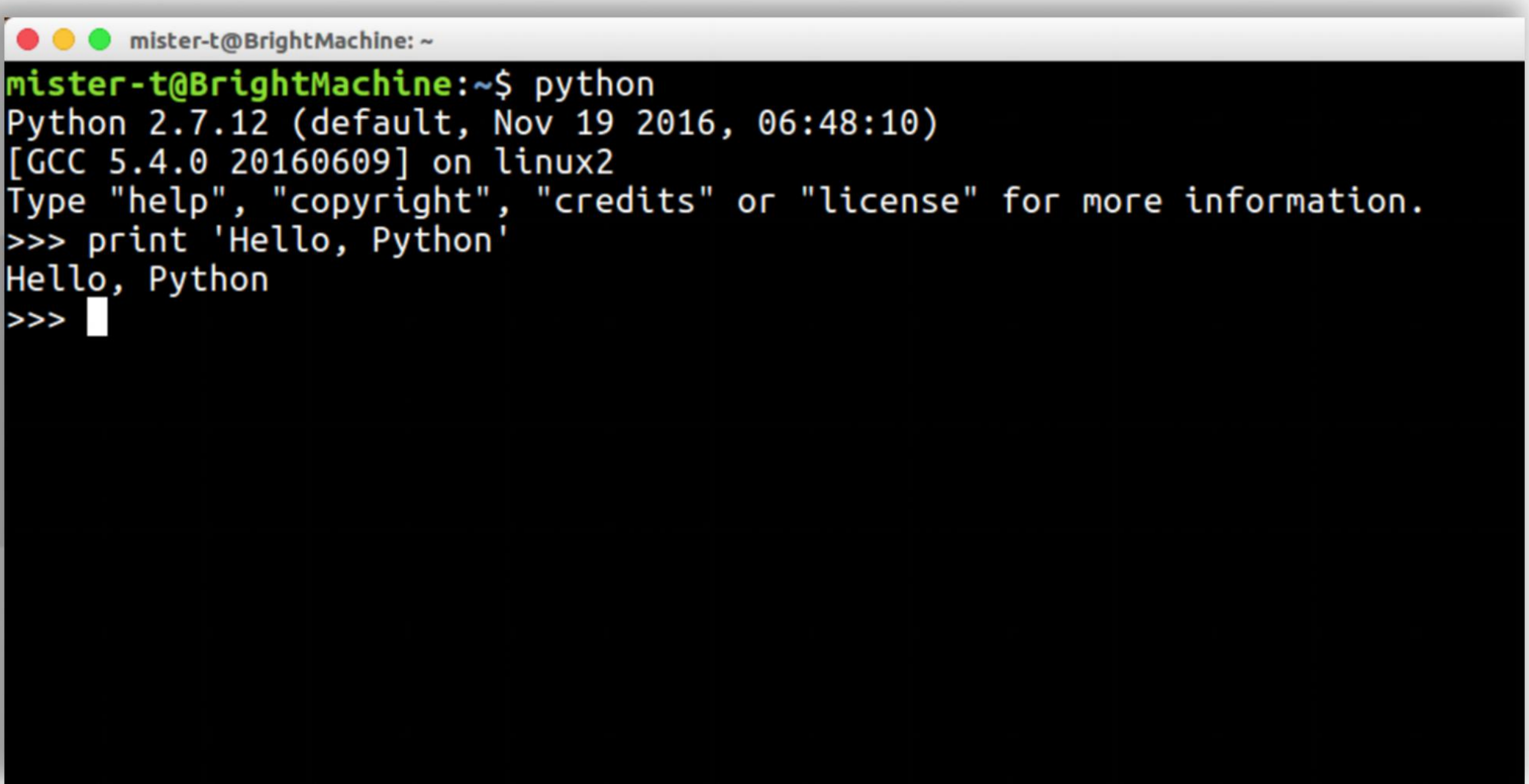
Type in Terminal: (Ctrl + Alt + T) $

`export ATH="$PATH:/usr/local/bin/python"` and press Enter.

Learning Basics

RUNNING PYTHON

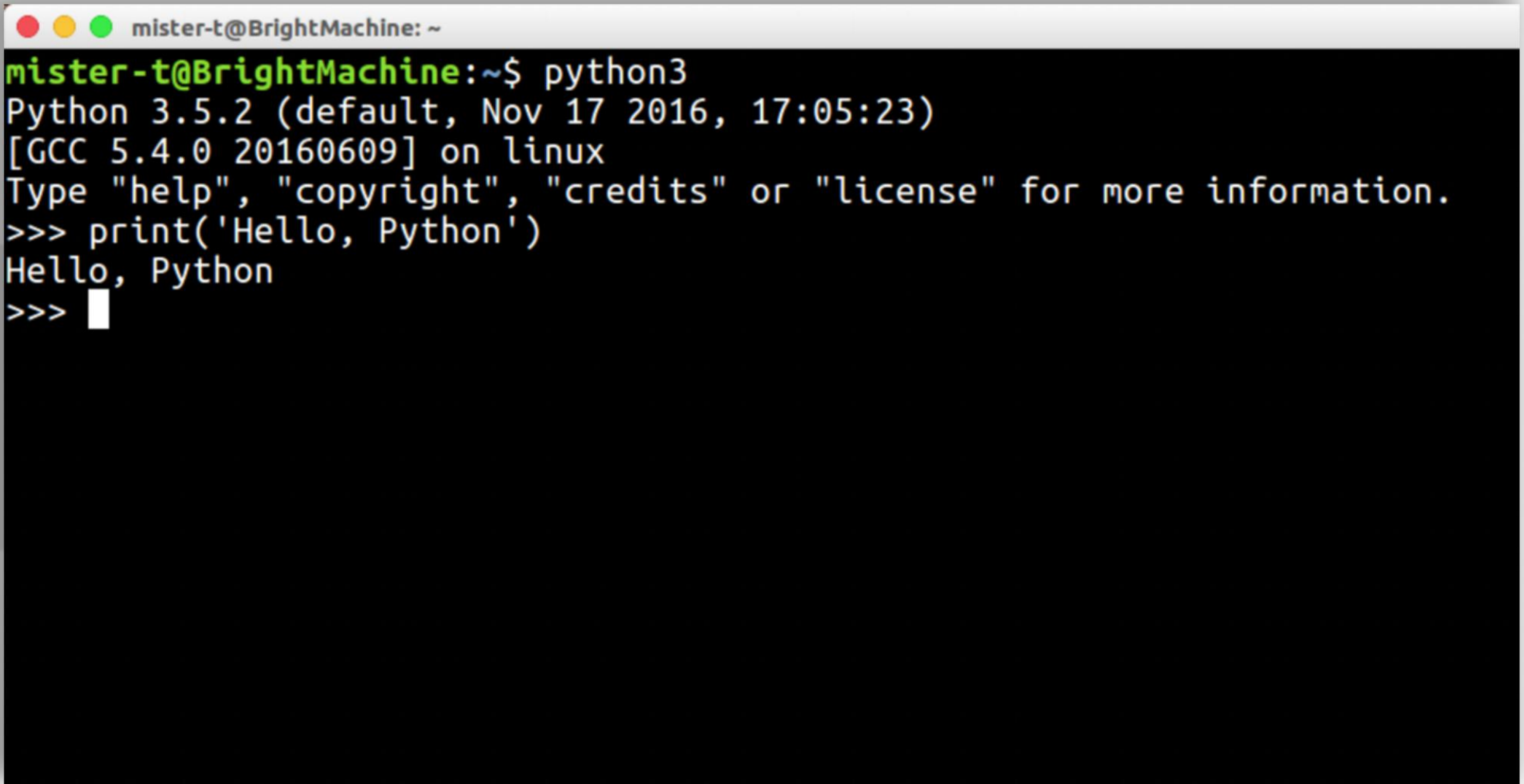
IDE: IDLE, PyCharm
Interpreted & Interactive Mode: Open $ and type in `python` or `python3` in case you've updated



```
mister-t@BrightMachine:~$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print 'Hello, Python'
Hello, Python
>>> 
```

Learning Basics

There are slight differences from version to version, and are not majorly changed with respect to syntax.
Remember - Python is a **Case-sensitive** language



```
mister-t@BrightMachine:~$ python3
Python 3.5.2 (default, Nov 17 2016, 17:05:23)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello, Python')
Hello, Python
>>> 
```

`>>>`

Python I & S

- The three closed angular brackets at the beginning of every line is the Python prompt
- It marks the beginning of the Interpreter

EXERCISE:

Type following in Interpreter mode of python

```
a = "Hello"   # Assiging of values
b = "Python"
print(a, b)
'''
Printing does not require
format specifier
'''
```

and observe the output.

>>>

Python I & S

- The **Interpreter** mode is easy and fast enough to execute line by line, but when it comes to one monolithic programming logic we go for writing **Scripts**.
- It is similar the way we have been writing programs for C++ & Java.

multiply.py

```
#!/usr/lib/python3.5

a = 3
b = 7

print("Multiplication of 3 * 7 = ",a * b)
```

>>> Python I & S

DATA TYPES:

- Numbers/Integers........(0, -1, 34, 2.1, 10065e, 10222.122254L)
- String..(Hello world, Sum=, Enter value)
- List...(Grocery list, to do list)
- Dictionary....................................(a:First Letter, b:Second Letter)
- Tuple..(List of Marks, Race records)

>>> Python I & S

miTu skillologies

ASSIGNING VALUES:

- A well understood paradigm in programming of assigning values to variables is much easier in Python
- Foll. the given examples-
  - ① Name = "Lord Snow"
  - ② Year = 2017
  - ③ Birth = Death = "Undecided"
- **Strings** needs to be written in either Single Quotes (' ') or Double Quotes (" ") to save it as a string
- **Number** can be assigned plainly, & can be integer or float
- **Multiple assignments** is easily possible in transitive manner

EXERCISE:

Type following in **Script** Mode

```
uname = first_name = "Emily"
pass = "password"
print("Your Credentials are UN",uname," PW",pass)
```

and execute the script

>>>

Python I & S

RESERVED WORDS:

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

>>> Python I & S

IDENTIFIERS:

1. **Class names** - All classname's must initiate with Uppercase letters.
   - Eg- Add, Round, Animate

1. **Private** - A private identifier must begin with an underscore "_".
   - Eg - _varA, _varB

1. **Strongly Private** - A Strongly private identier initiates with double underscores "__".
   - Eg - __varC, __Add

1. **Special Names** - There exists q set of names which initiate and end with doube underscores denoting it as a name with special meaning.
   - Eg - __init__

Quick Exercise

Try the following in Python Interpreter:

INTEGERS

```
>>> 5+4
9
```

```
>>> 5-4
1
```

```
>>> 5*4
20
```

```
>>> 5**4
625
```

Exponential operation: 5^4

```
>>> 5/4
1.25
```

```
>>> -5/4
-1.25
```

Integer division rounds down

```
>>> 5%4
1
```

```
>>> -5%4
3
```

remainder (mod) division returns 0 or positive

Quick Exercise

```
>>> 65536*65536
65536
```

```
>>> 4294967296*4294967296
18446744073709551616L
```

No inherent limit to Python's integer arithmetic:
can keep going until we run out of memory

Note: Long (L) is discontinued since Python 3.X

Quick Exercise

FLOATING POINT

```
>>> 1.0
1.0
```

Floating point number (& usual issues with them)

```
>>> 1/2
0.5
```

Ok

```
>>> 1/4
0.25
```

Ok

```
>>> 0.1
0.10000000000000001
```

Issues with representation in base 2 (Altough it has been covered from Python 2.7 onwards)

Quick Exercise

TESTING ACCURACY

```
>>> 65536.0*65536.0
4294967296.0

>>> 4294967296.0*4294967296.0
1.8446744073709552e+19
```

Accuracy of upto 16 places after Decimal

```
>>> 1.157920892373162e+77*1.157920892373162e+77
1.3407807929942597e+154

>>> 1.3407807929942597e+154 *
1.3407807929942597e+154
inf
```

Quick Exercise

MACHINE EPSILON

```
>>> 1.0 + 1.0e-16
1.0
```

Too small to make a difference

```
>>> 1.0 + 2.0e-16
1.0000000000000002
```

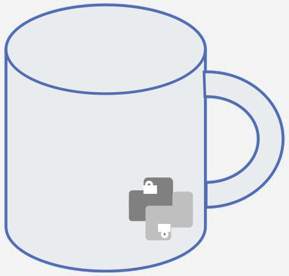
Large enough, hence more precision

```
>>> 1.0 + 1.1e-16
1.0

>>> 1.0 + 1.9e-16
1.0000000000000002
```

'e' - Means

$9.0122222900391 \times 10^{-5}$



To Be Resumed in Next Slide

@mitu_skillologies

/mITuSkillologies

@mitu_group