AN INTRODUCTION TO PYTHON

Mr.Saurabh Jadhav

WHAT IS PYTHON?

Python is a scripting programming language known for both its simplicity and wide breadth of applications. For this reason it is considered one of the best languages for beginners.

WHETTING YOUR APPETITE

If you do much work on computers, eventually you find that there's some task you'd like to automate. For example, you may wish to perform a search-and-replace over a large number of text files, or rename and rearrange a bunch of photo files in a complicated way. Perhaps you'd like to write a small custom database, or a specialized GUI application, or a simple game.

DON’T WORRY!

- Beginners
- Intermediate
- Expert

Workshop Will be 100% Practical and Simple

- All program will be practically explained.
- Don’t worry about forgetting the code, It will be given to you.

LETS GET STARTED !

1. Install Python
2. Now you are King !
3. Live Your life and Create Awesome Programs !

HOW TO PRINT ?

The print() Function

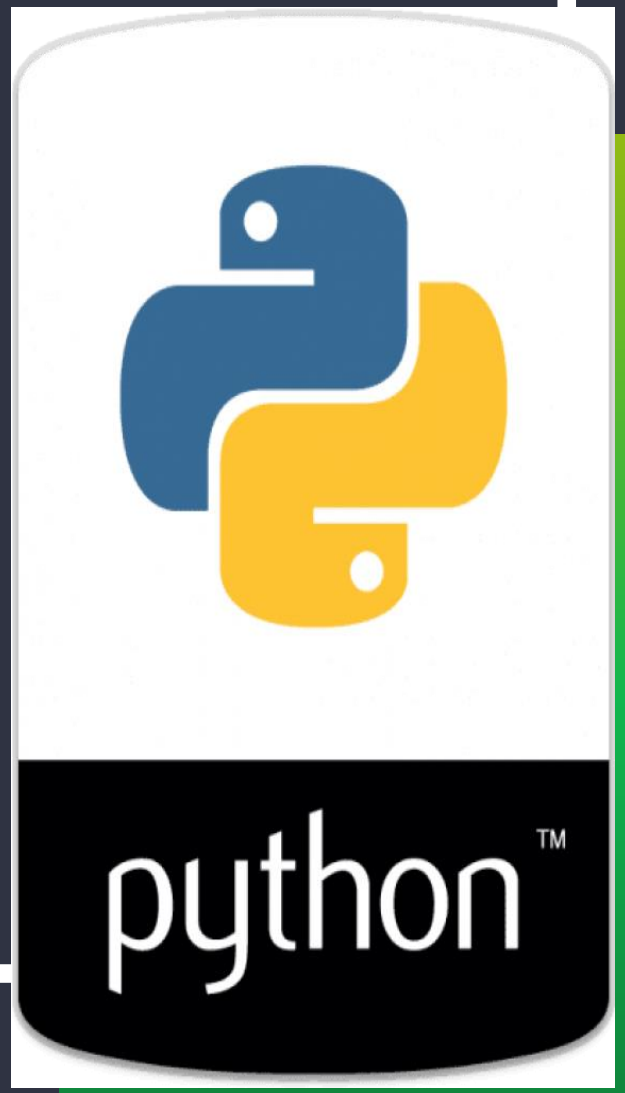
```
>>> print('Hello world!')
Hello world!
```

```
>>> a = 1
>>> print('Hello world!', a)
Hello world! 1
```

Play with Numbers !

- Perform all the Arithmetic Operations here.

Operators	Operation
**	Exponent
%	Modulus/Remaider
//	Integer division
/	Division
*	Multiplication
-	Subtraction
+	Addition

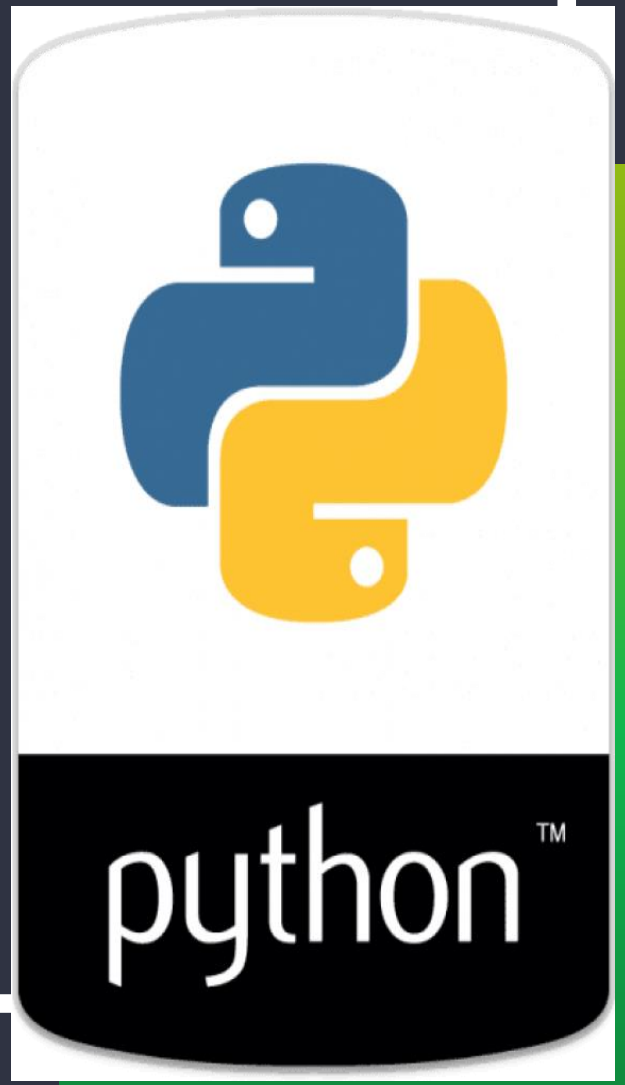


Data Types

Integers	-2, -1, 0, 1, 2, 3, 4, 5
Floating-point numbers	-1.25, -1.0, --0.5, 0.0, 0.5, 1.0, 1.25
Strings	'a', 'aa', 'aaa', 'Hello!', '11 cats'

How to define Variables?

You can name a variable anything as long as it obeys the following three rules:
1.It can be only one word.
2.It can use only letters, numbers, and the underscore (_) character.
3.It can’t begin with a number.
4.Variable name starting with an underscore (_) are considered as "unuseful”.



Comments

Inline comment:

```
# This is a comment
```

Multiline comment:

```
''' This is a  multiline
comment
'''
```

FLOW CONTROL

Comparison Operators

Operator	Meaning
==	Equal to
!=	Not equal to
<	Less than
>	Greater Than
<=	Less than or Equal to
>=	Greater than or Equal to

- **Boolean Operators**

The *and* Operator's *Truth* Table:

Expression	Evaluates to
True and True	True
True and False	False
False and True	False
False and False	False

The *or* Operator's *Truth* Table:

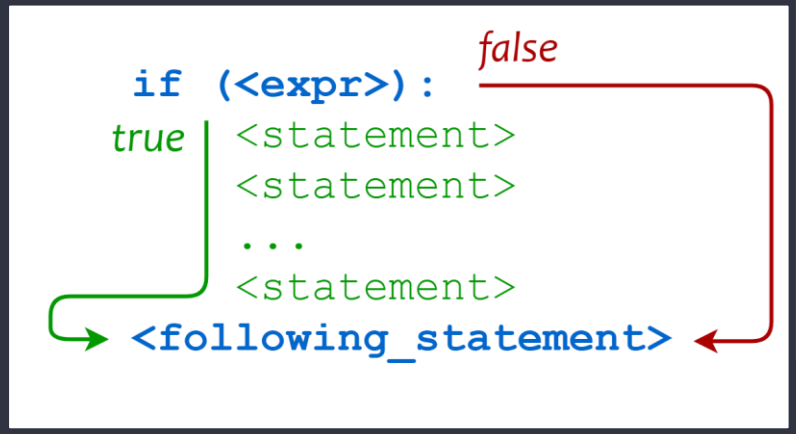
Expression	Evaluates to
True or True	True
True or False	True
False or True	True
False or False	False

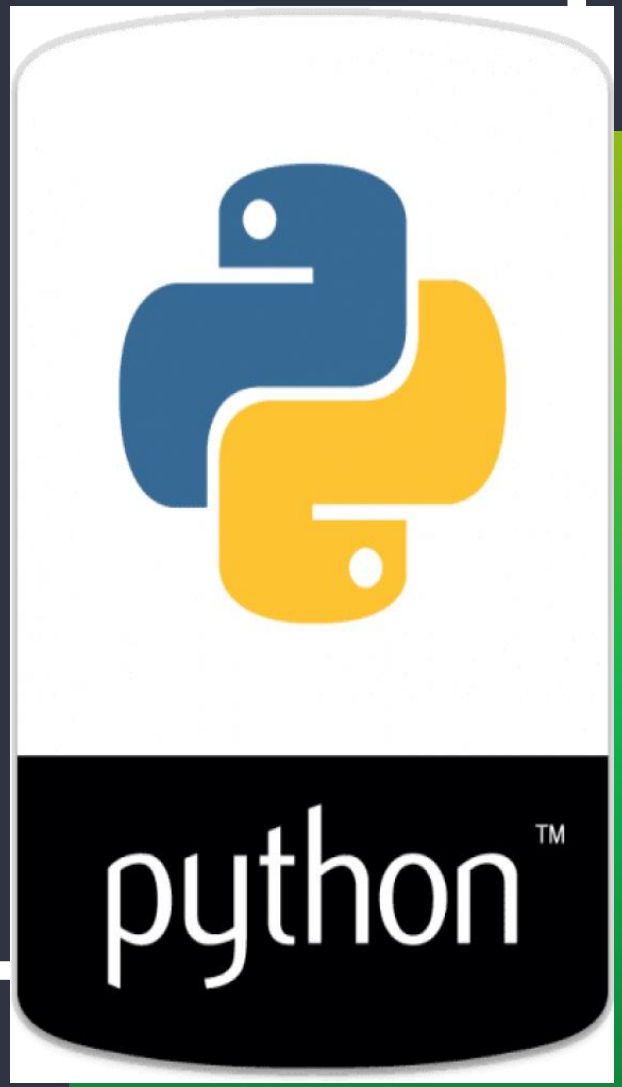
The *not* Operator's *Truth* Table:

Expression	Evaluates to
not True	False
not False	True

Conditional statements

In a Python program, the if statement is how you perform this sort of decision-making. It allows for conditional execution of a statement or group of statements based on the value of an expression.



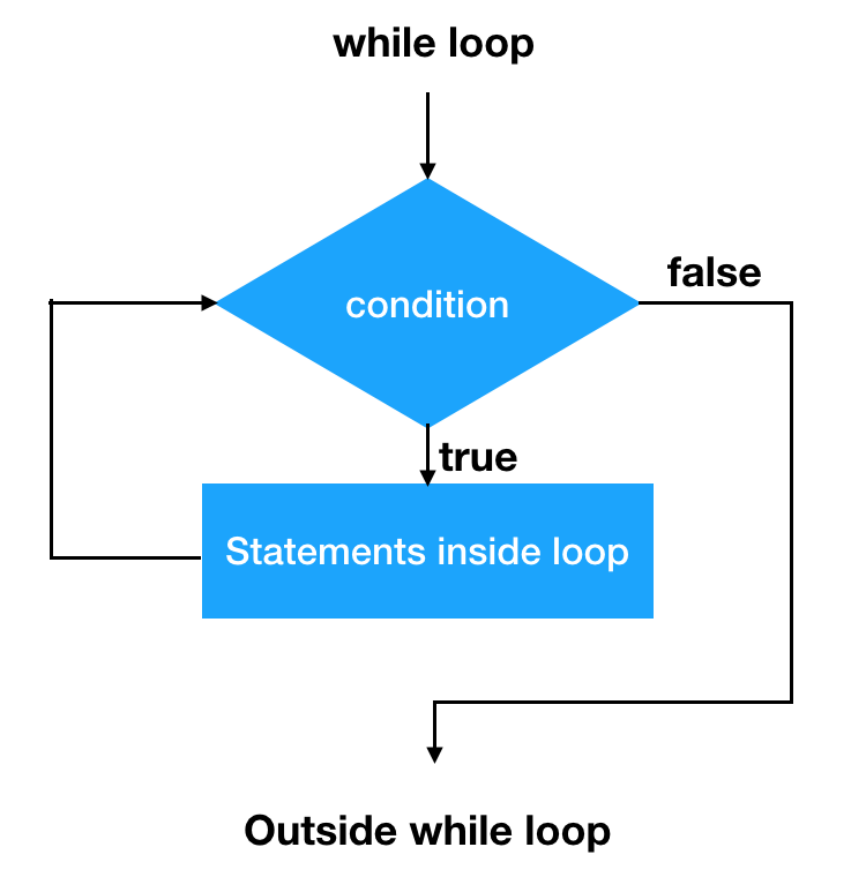


Elif clauses

```
if <expr>:
    <statement(s)>
else:
    <statement(s)>
```

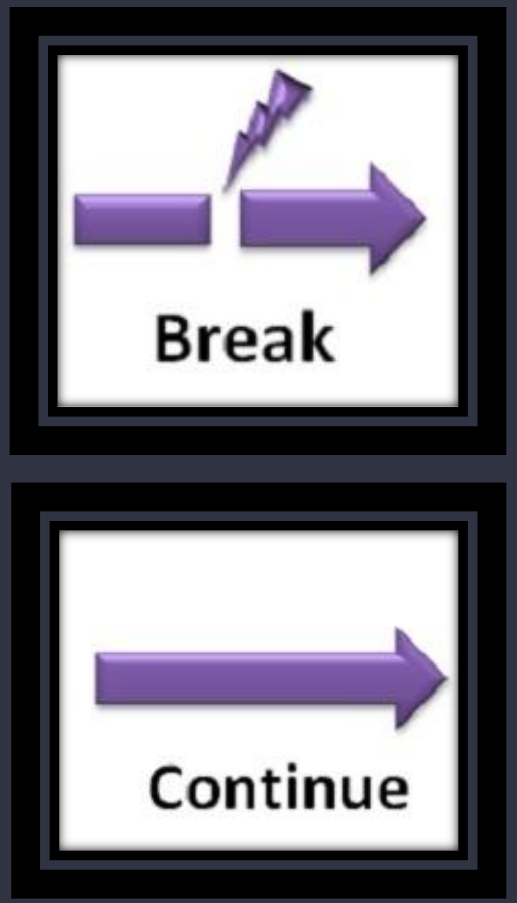
```
name = 'Bob'
age = 30
if name == 'Alice':
    print('Hi, Alice.')
elif age < 12:
    print('You are not Alice, kiddo.')
else:
    print('You are neither Alice nor a little kid.')
```

WHILE LOOP STATEMENTS




```
spam = 0
while spam < 5:
    print('Hello, world.')
    spam = spam + 1
```

BREAK & CONTINUE STATEMENT



break Statements:
If the execution reaches a break statement, it immediately exits the while loop's clause:

```
while True:
    print('Please type your name.')
    name = input()
    if name == 'your name':
        break
print('Thank you!')
```

continue Statements
When the program execution reaches a continue statement, the program execution immediately jumps back to the start of the loop

```
while True:
    print('Who are you?')
    name = input()
    if name != 'Joe':
        continue
    print('Hello, Joe. What is the password? (It is a fish.)')
    password = input()
    if password == 'swordfish':
        break
print('Access granted.')
```

FOR LOOP

The *range()* function can also be called with three arguments. The first two arguments will be the start and stop values, and the third will be the step argument. The step is the amount that the variable is increased by after each iteration.

```
>>> for i in range(0, 10, 2):
>>>     print(i)
0
2
4
6
8
```

LISTS

List is a collection which is ordered and changeable. Allows duplicate members.

```
>>> spam = ['cat', 'bat', 'rat', 'elephant']

>>> spam
['cat', 'bat', 'rat', 'elephant']
```

Getting Individual Values in a List with Indexes

```
>>> spam = ['cat', 'bat', 'rat', 'elephant']
>>> spam[0]
'cat'
```

GETTING A LIST'S LENGTH WITH LEN()

```
>>> spam = ['cat', 'dog', 'moose']
>>> len(spam)
3
```

Changing Values in a List with Indexes

```
>>> spam = ['cat', 'bat', 'rat', 'elephant']
>>> spam[1] = 'aardvark'
```

Removing Values from Lists with del

```
>>> del spam[2]
>>> spam
['cat', 'bat']
```

- Getting Sublists with slices

```
>>> spam = ['cat', 'bat', 'rat', 'elephant']
>>> spam[0:4]
['cat', 'bat', 'rat', 'elephant']
```

```
>>> spam[1:3]
['bat', 'rat']
```

```
>>> spam[0:-1]
['cat', 'bat', 'rat']
```

```
>>> spam = ['cat', 'bat', 'rat', 'elephant']
>>> spam[:2]
['cat', 'bat']
```

ADDING VALUES TO LISTS WITH THE APPEND() AND INSERT() METHODS

append():

```
>>> spam = ['cat', 'dog', 'bat']

>>> spam.append('moose')

>>> spam
['cat', 'dog', 'bat', 'moose']
```

insert():

```
>>> spam = ['cat', 'dog', 'bat']

>>> spam.insert(1, 'chicken')

>>> spam
['cat', 'chicken', 'dog', 'bat']
```

- For loop in list ?
- In and Not in Operators
- Finding a Value in a List with the index() Method-eg
  - >>> spam.index('Pooka‘)
- Sorting the Values in a List with the sort() Method

  >>> spam.sort(reverse=True)

TUPLE DATA TYPE

```
>>> eggs = ('hello', 42, 0.5)
>>> eggs[0]
'hello'
```

```
>>> eggs[1:3]
(42, 0.5)
```

```
>>> len(eggs)
3
```

- Converting Types with the list() and tuple() Functions

```
>>> tuple(['cat', 'dog', 5])
('cat', 'dog', 5)
```

```
>>> list(('cat', 'dog', 5))
['cat', 'dog', 5]
```

```
>>> list('hello')
['h', 'e', 'l', 'l', 'o']
```

DICTIONARIES AND STRUCTURING DATA

values():

```
>>> spam = {'color': 'red', 'age': 42}
>>> for v in spam.values():
>>>     print(v)
red
42
```

keys():

```
>>> for k in spam.keys():
>>>     print(k)
color
age
```

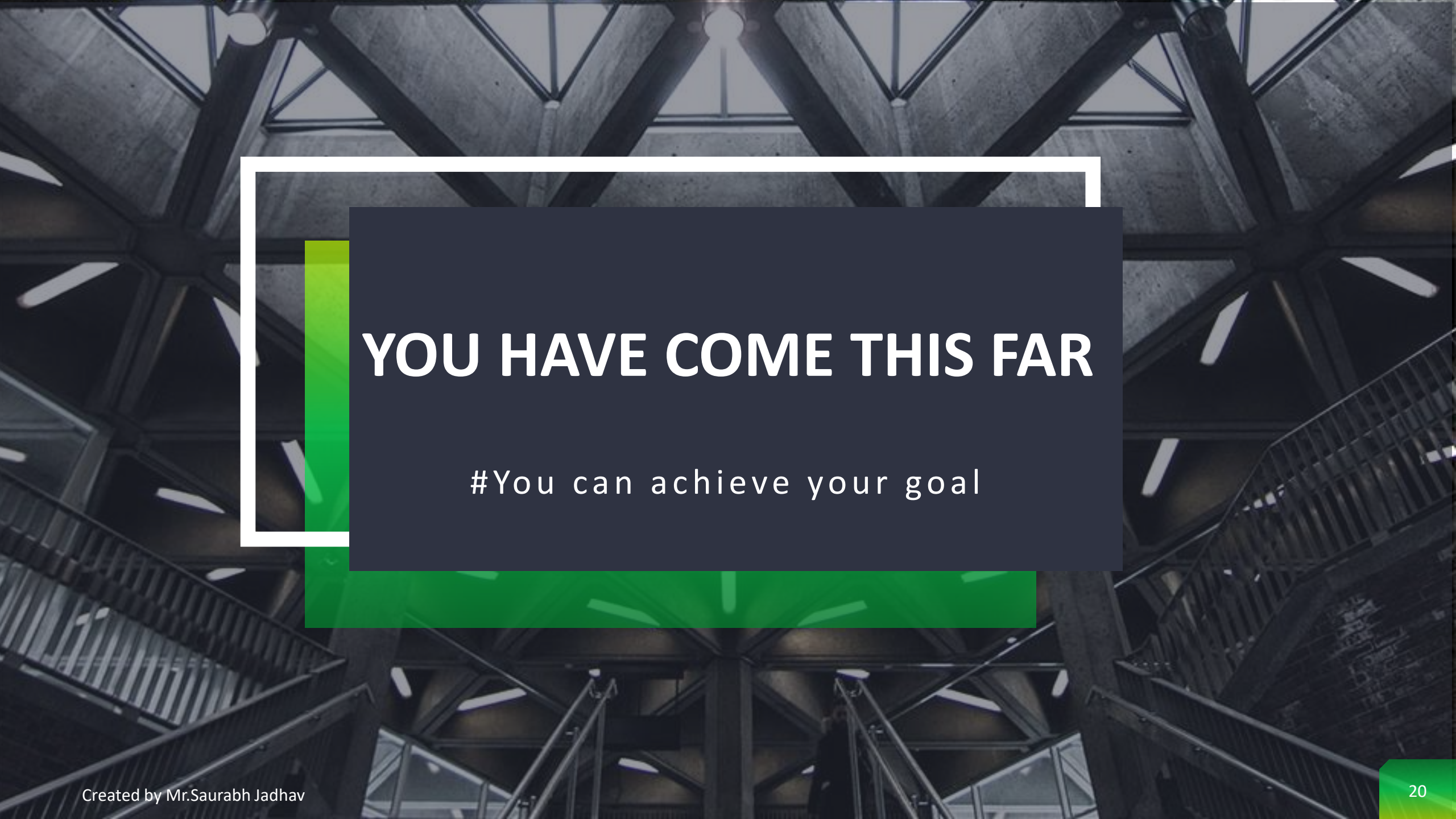
Example of Dictionary

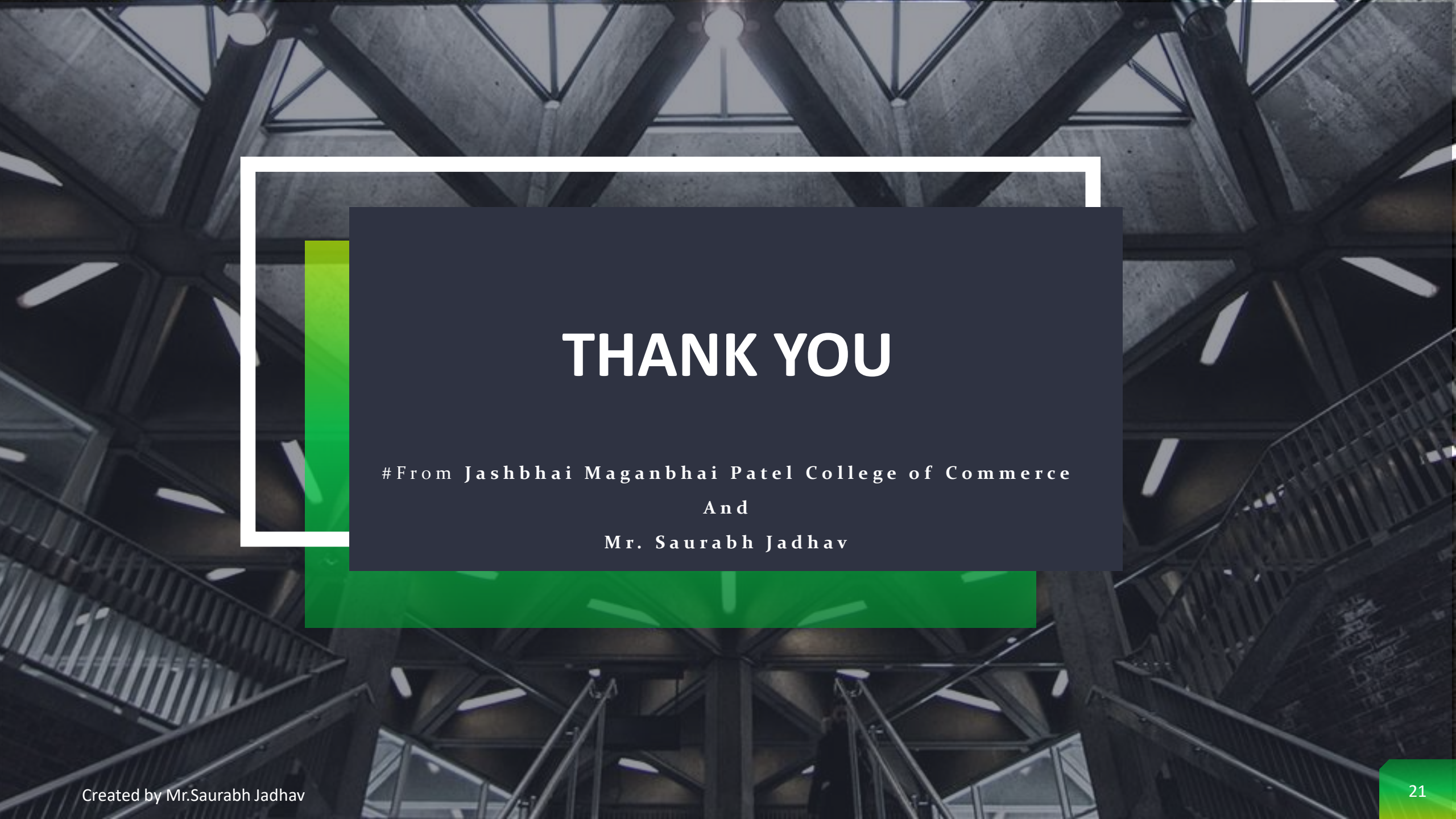
```
myCat = {'size': 'fat', 'color': 'gray', 'disposition': 'loud'}
```

items():

```
>>> for i in spam.items():
>>>     print(i)
('color', 'red')
('age', 42)
```


THERE’S NO STOPPING HERE..
#This is Just a Beginning


YOU HAVE COME THIS FAR
#You can achieve your goal



THANK YOU

#From **Jashbhai Maganbhai Patel College of Commerce**

And

Mr. Saurabh Jadhav