Python

PROGRAMMING

About the presenters

ROSHAN JAISWAL & RAHUL KIRPANE

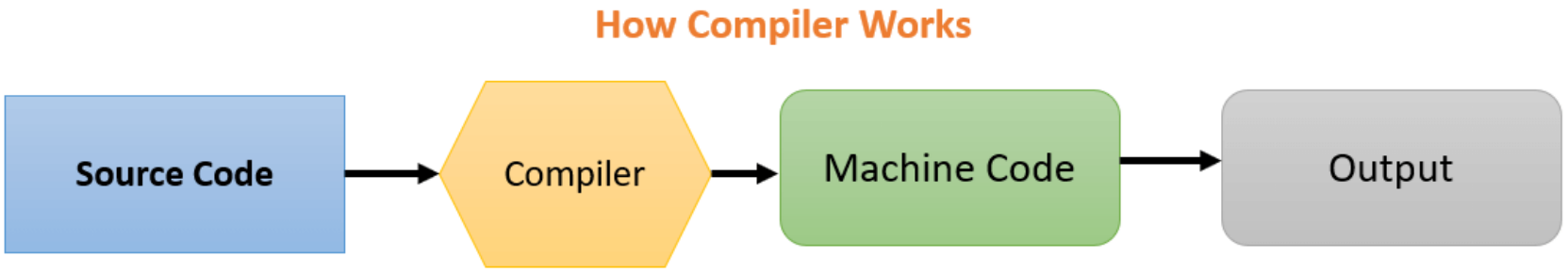
Subject Matter Expert – Applied Tech Curriculum

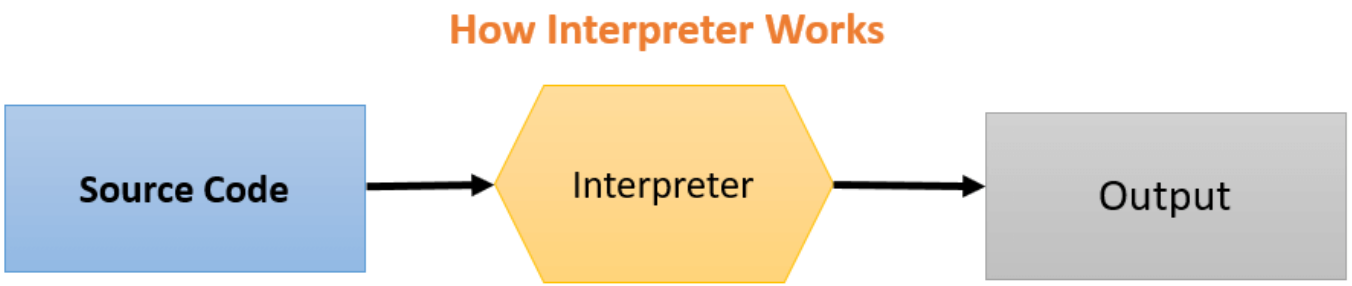
History

- Invented in the Netherlands, early 90s by Guido van Rossum Python was conceived in the late 1980s and its implementation was started in December 1989
- Guido Van Rossum is fan of ‘Monty Python’s Flying Circus’, this is a famous TV show in Netherlands
- Named after Monty Python
- Open sourced from the beginning

What is Python?

- Python is a general purpose programming language that is often applied in scripting roles
- Python is an interpreted language





Programming Language

Created in 1991 by Guido van Rossum

Cross Platform

Freely Usable Even for Commercial Use

Python is Easy

Python is easy to use, powerful, and versatile, making it a great choice for beginners and experts alike.

```
public class Main {
   public static void
   main(String[] args) {
      System.out.println("Hello
      world!");
   }
}
```


```
print("Hello world!")
```

Python is Powerful

Big names using Python

- Syntax
  - A set of rules used to determine if a certain string of words forms a valid sentence.
  - Example: I like playing cricket is a valid sentence while
  - I cricket like playing is not valid

- Semantics
  - A set of rules determining if a certain phrase makes sense
  - Example : I ate an apple -- Valid
  - Apple ate me – Not valid

Keywords

- In Python, keywords are case sensitive.
- There are 33 keywords in Python 3.7. This number can vary slightly over the course of time.

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 Statement
- Instructions that a Python interpreter can execute are called statements. For example, a = 1 is an assignment statement. if statement, for statement, while statement, etc.
- Multi-line statement

```
a = 1 + 2 + 3 + \
    4 + 5 + 6 + \
    7 + 8 + 9
colors = ['red',
          'blue',
          'green']
a = 1; b = 2; c = 3
```

- **Python Indentation**
- Generally, four whitespaces are used for indentation and are preferred over tabs. Here is an example.

- ```
  for i in range(1,11):
      print(i)
    if i == 5:
      break
  ```
  ```

- **Python Comments**
- #This is a comment

- **Multi-line comments**
- Use triple quotes, either ''' or """.
- """This is a
- perfect example of
- multi-line comments"""

- **Assigning multiple values to multiple variables**
- a, b, c = 5, 3.2, "Hello"
- print (a)
- print (b)
- print (c)

Python List

- List is an ordered sequence of items. It is one of the most used datatype in Python and is very flexible. All the items in a list do not need to be of the same type.
- Declaring a list is pretty straight forward. Items separated by commas are enclosed within brackets [ ].
- a = [1, 2.2, 'python']

- a = [5,10,15,20,25,30,35,40]

- # a[2] = 15
- print("a[2] = ", a[2])

- # a[0:3] = [5, 10, 15]
- print("a[0:3] = ", a[0:3])

- # a[5:] = [30, 35, 40]
- print("a[5:] = ", a[5:])

Examples

```
# Python list methods
my_list = [3, 8, 1, 6, 0, 8, 4]

# Output: 1
print(my_list.index(8))

# Output: 2
print(my_list.count(8))

my_list.sort()

# Output: [0, 1, 3, 4, 6, 8, 8]
print(my_list)

my_list.reverse()

# Output: [8, 8, 6, 4, 3, 1, 0]
print(my_list)
```

```
my_list = ['p', 'r', 'o', 'b', 'l', 'e', 'm']

# Output: True
print('p' in my_list)

# Output: False
print('a' in my_list)

# Output: True
print('c' not in my_list)
```

Python List Methods

- append() - Add an element to the end of the list
- extend() - Add all elements of a list to the another list
- insert() - Insert an item at the defined index
- remove() - Removes an item from the list
- pop() - Removes and returns an element at the given index
- clear() - Removes all items from the list
- index() - Returns the index of the first matched item
- count() - Returns the count of the number of items passed as an argument
- sort() - Sort items in a list in ascending order
- reverse() - Reverse the order of items in the list
- copy() - Returns a shallow copy of the list

Python Tuple

- Tuple is an ordered sequence of items same as a list. The only difference is that tuples are immutable. Tuples once created cannot be modified.
- Tuples are used to write-protect data and are usually faster than lists as they cannot change dynamically.
- It is defined within parentheses () where items are separated by commas.
- t = (5,'program', 1+3j)

- # t[1] = 'program'
- print("t[1] = ", t[1])

Python Dictionary

- Dictionary is an unordered collection of key-value pairs.
- It is generally used when we have a huge amount of data. Dictionaries are optimized for retrieving data. We must know the key to retrieve the value.
- In Python, dictionaries are defined within braces {} with each item being a pair in the form key:value. Key and value can be of any type.
- d = {1:'value','key':2}

- print("d[1] = ", d[1]);
  - d[1] = value

- print("d['key'] = ", d['key']);
  - d['key'] = 2

Python Strings

- String is sequence of Unicode characters. We can use single quotes or double quotes to represent strings. Multi-line strings can be denoted using triple quotes, ''' or """.
- s = "This is a string"
- s = '''A multiline
- string'''‘

- Just like a list and tuple, the slicing operator [ ] can be used with strings. Strings, however, are immutable.

Python Set

- Set is an unordered collection of unique items. Set is defined by values separated by comma inside braces { }. Items in a set are not ordered.
- a = {5,2,3,1,4}
- print("a = ", a)

Print Function

- Python 3.7.1 comes with 69 built-in functions.
- The print() function is a **built-in** function.
- The end and sep parameters can be used for formatting the output of the print() function.

Thank You