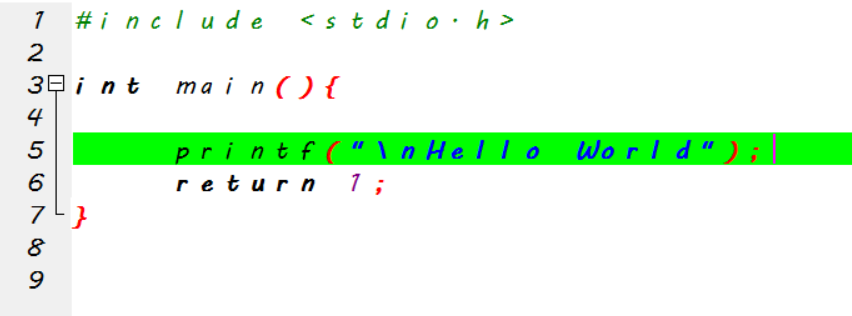
**Python Club and Workshop**

**An Introduction to Week 1 Activity and Projects**

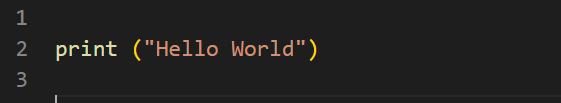
The objective of our 1st week shall be to:

* Familiarize with the Basic Python Syntax
* Learn how to do basic operations in Python
* Compare C and C++ Syntax with Python Syntax
* Learn about concepts and programming methods only possible in Python
* Learn about different Data Types and different ways to use them
* Introduction to pip and Libraries
* Project aim and general discussion for the next week

Code block in C to print Hello World



Code Block in Python to print Hello World

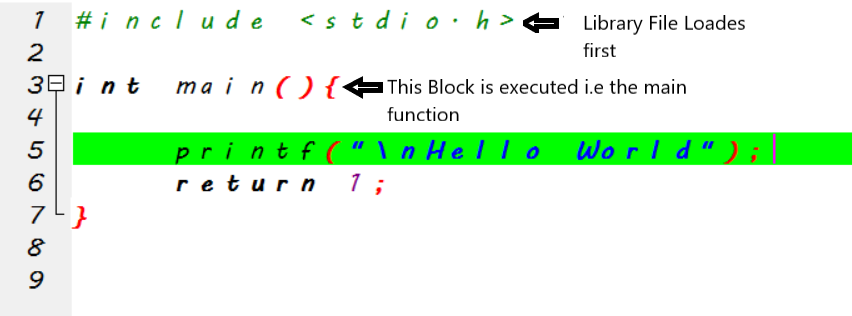


In C we are required to include a stdio library, create a function named main () and write our code to print Hello World.

But in Python we can just write print (“Hello World”) to achieve the same effect.

But how does this Exactly Work?

Let us analyze The C code and Python code to understand how the code is executed.



But in Python we neither had to load Library File to use print () nor had to define a main () function for the script to execute. That is because, in Python instead of executing the main () function, the entire file is executed line by line.

Key points to note:

* In C, when file is executed, the file

Course to Complete in 1st week

* Data Types
  + Numbers
    - Integers
    - Floating point
    - Complex
  + Sequences
    - Iterables
    - Strings
    - Tuples
    - Lists
  + Sets
  + Frozensets
  + Dictionaries
  + None
  + Callables
    - Functions
    - Generators
    - Methods
  + Boolean Values
* Varaibles
  + Object attributes and items
  + Accessing nonexistent References
* Assignment Statements
  + PLAIN
  + Indexing
  + Slicing
  + Augmented assignments
  + Del statements
* Expression and operators
  + Comparison Chaining
  + Numeric Operations
  + Numeric Conversions
  + Comparisons
  + Sequence Operations
  + Sequence Conversions
* The Print Statement
* Control Flow Statements
* Iterators
* List Comprehensions
* Break and Continue
* Else clause on loop Statements
* The pass Statements
* The Try and raise Statements
* The with Statement
* Functions
  + Def
  + Parameters
  + Attributes of Function Objects
    - Docstring
  + Return statement
  + Calling Functions
  + Kinds of Arguments
* Namespaces
  + The global statement
  + Nested Function and Nested Scopes
* Lambda Expression
* Generators
  + Generator Expressions
* Recursion
* OOP
  + Characteristics
  + Statement
  + Class Body
  + Attributes of class object
  + Function definition in class body
  + Class private variables
  + Class documentation string
* Descriptors
  + Overriding Descriptors
  + Non Overriding Descriptors
* Instances
  + \_\_init\_\_
  + Attributes of instance objects
  + \_\_new\_\_
* Attributes Reference Basics
  + Getting an attribute from a class
  + Getting an attribute from an instance
  + Setting an attribute
* Bound and Unbound Methods
* Inheritance
  + Method resolution order
  + Overriding attributes
  + Delegation to superclass Methods
  + Cooperative superclass method calling
  + Deleting Class Attributes
* The Built-in Object type
* Class Level Methods
  + Static Methods
  + Class Methods
* Properties
  + Properties and inheritance
* \_\_dict\_\_
* \_\_slots\_\_
* \_\_getattribute\_\_
* Per-instance Methods
* Inheritance from built in types
* Special Methods
  + \_\_call\_\_
  + \_\_cmp\_\_
  + \_\_del\_\_
  + \_\_delattr\_\_
  + \_\_eq\_\_
  + \_\_ge\_\_
  + \_\_gt\_\_
  + \_\_le\_\_
  + \_\_lt\_\_
  + \_\_ne\_\_
  + \_\_getattr\_\_
  + \_\_hash\_\_
  + \_\_nonzero\_\_
  + \_\_repr\_\_
  + \_\_setattr\_\_
  + \_\_str\_\_
  + \_\_unicode\_\_
* Special Methods for Containers
  + \_\_getitem\_\_
  + \_\_setitem\_\_
  + \_\_delitem\_\_
  + \_\_len\_\_
  + \_\_contains\_\_
  + \_\_iter\_\_
* Sequences
  + \_\_add\_\_
  + \_\_mul\_\_
  + \_\_radd\_\_
  + \_\_rmul\_\_
  + \_\_iadd\_\_
  + \_\_imul\_\_
* Mappings
  + Has\_key
  + Copy
  + Get
  + Keys
  + Values
  + Iteritems
  + iterkeys
  + \_\_iter\_\_
  + Clear
  + Popitem
  + Setdefault
  + Update
* Sets
* Container Slicing
* Container Methods
* Decorators
* MetaClasses
* How python Determines a Class’s Meta Class
* How MetaClasses Creates a Class
  + Defining and using your own metaclass
* Exceptions
  + Try
  + With
  + Generator enhancements
  + Exception Propagation
  + Raise
  + Exception Objects
  + Hierarchy of Standard Exceptions
  + Standard Exception Classes
  + Custom Exception
  + Custom Exception and Multiple inheritance
  + Logging Errors
    - The logging module
  + Assert Statement
* Modules
  + Modules Object
  + Import statement
  + Module body
  + Attributes of module objects
  + Python built-ins
  + Module Documentation string
  + From statement
  + Module Loading
* The main Program
  + \_\_name\_\_=’\_\_main\_\_’
  + Reload Function
  + Circular imports
  + Sys.modules
  + Custom Importers
* Packages
  + \_\_init\_\_.py
  + Special attributes of package objects
  + Absolute versus relative imports
* String Modules and Regular Expression
* File and Text Operations
  + File Objects
  + File mode
  + Sequential and nonsequential access
  + Attributes and methods of file objects
  + Iteration on File objects
* The OS module
* Os.path module
* The time module
* Persistence and Database
  + Serialization
* Datetime module
* Random module
* Controlling Executions
  + Dynamic Execution and exec Statements
  + Avoiding exec
  + Compile and code objects
  + Garbage Collection
* Threads and Processes
  + Threads in Python
    - The thread module
    - The Queue Module
  + Threading Module
  + Thread Objects
  + Thread Synchronization Objects
  + Thread Local Storage
* Threaded Program Architecture
* Process Environment
* The subprocess Module
* The operator module
* Math and cmath module
* Factory Functions
* Client side Network Protocols Module
  + Urlparse module
  + Urllib module
  + The request class
  + Handler class
* The Socket Module
  + Socket Functions
  + Socket Class
  + Echo server and Client using TCP Sockets
  + Using UDP sockets
* The SocketServer Module
* HTTP servers
* HTMLParser Module
* BeautifulSoup Extension
* Generating HTML
  + Embedding
  + Templating
* Parsing XML with DOM
* Changing and Generating XML
* The Python Interpreter Class
  + Interp
  + Eval
  + Exec
  + Execfile
  + Get
  + Set
  + \_\_builtin\_\_ class
* PyObject Class