Hello World

```
echo "Hello World!"
```

Nim is an efficient, expressive, and elegant language created by Andreas Rumpf and released in 2008. It is a compiled and statically typed programming language. Nim combines concepts from languages such as Python, Ada, and Modula.

Basic Type

```
42,-4        # int  - Integer
42, 4        # uint - Integer (not negative)
true, false  # bool - Boolean
"42"         # string - String
'a','4'      # char - Character
42.0         # float - Floating-point
```

Comments

```
# This is a comment
echo "Hello!" # This is a comment

#[ This is a multiline
 comment
]#
```

Variable & Constant

```
var myVar1:string = "My variable" # Variable declared as string.
var myVar2 = "My variable" # Variable implicitly declared as string.
let myConstant = "Constant value" # Constant. Immutable value.
var salary:float = 9_000.00  # Float variable.
                             # _ is ignored. This is useful for long numbers.
const PI = 3.14 # Constant. Immutable value. Constants are computed at compile time.
```

Import & Include

```
import os # Imports a lib.
import strutils
import json, othermodule  # Imports a lib & a module.
include "folder/file.nim" # Includes a file. Useful to
include "file1.nimf".     # split a big file in parts.
```

Operator

```
5 + 2    # Addition
5 - 2    # Subtraction
5 * 2    # Multiplication
5 / 2    # Division
# Integer division    # Integer modulo operation (remainder)
5 div 2 # returns: 2  5 mod 2 # returns: 1
```

Advanced Type 1/2

```
# Table.
# Dictionary or Hash Structure.
import tables
var dateOfBirth: Table[string, uint]
dateOfBirth["Mike"] = 1995
dateOfBirth["John"] = 1961
echo dateOfBirth["Mike"]
1995
# Array.
# Array or matrix Structure.
var myArray: array[0 .. 2, bool] = [false,true,false]
echo myArray[1]
true
```

Advanced Type 2/2

```
# Sequence.
# List of values Structure.

let mySequence: seq[string] = @["abc", "def"]
echo mySequence[1]
def
# Tuple.
# Useful to return multiple values from a
# func or proc.
let pos: tuple[x, y, z: int] = (x: 100, y: 50, z: 30)
echo pos                    echo pos.x    echo pos[1]
(x: 100, y: 50, z: 30)      100           50

# Tuple results can be unpacked:
let (a,b,_) = pos # Variables a, b created.
                  # a = 100, b = 50.
                  # Variable z ignored.
```

String Operations 1/2

```
# Concatenation
echo "con" & "cat"
concat

# Quotation marks inside a string
echo """Nim is an "expressive" language"""
Nim is an "expressive" language

# String interpolation
import strformat
let nimVersion = $NimVersion
echo fmt"""Using Nim version: {nimVersion}"""
Using Nim version: 1.6.8

echo fmt"""Result of 4 x 2 = {4*2}"""
Result of 4 x 2 = 8

# Remove part of a string
echo "Nim-lang".strip[0 .. 2]
Nim

# Multiline string
let multiline = """line1
  line2"""
echo multiline
line1
  line2
```

String Operations 2/2

```
import strutils
echo "A_B_C".normalize()
abc

echo "A_B_C".normalize().toUpper()
ABC

echo " ABC ".strip().len()
3

echo len(" ABC ") == " ABC ".len()
true

echo "ABC".toLower()
abc

echo "a b c".split()
@["a", "b", "c"]

echo ["a", "b", "c"].join(",")
a,b,c

echo "*".repeat(20)
********************

echo "   a".replace(" ","*")
***a
```