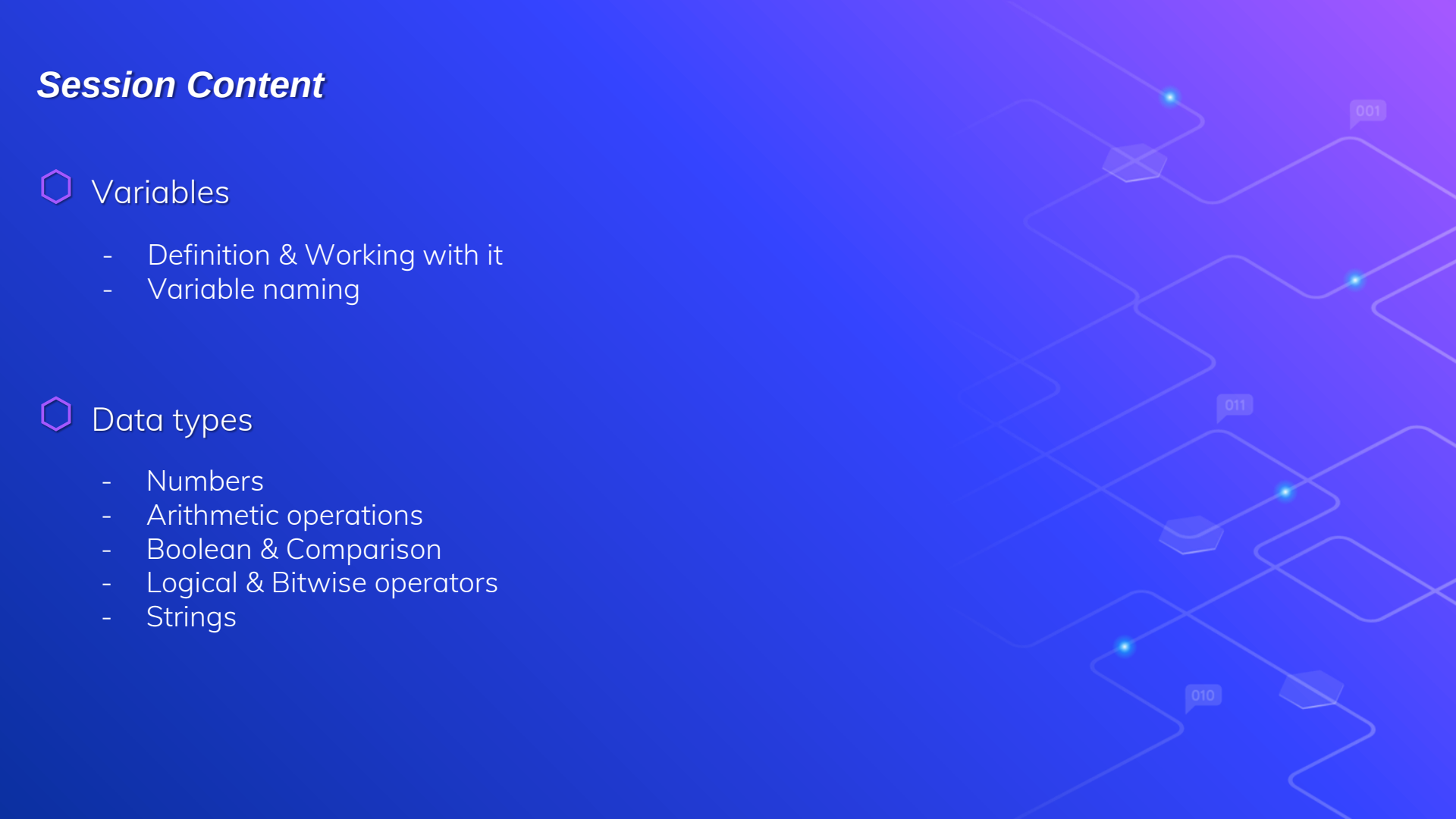
Python Programming

Session Content

- Variables
  - Definition & Working with it
  - Variable naming

- Data types
  - Numbers
  - Arithmetic operations
  - Boolean & Comparison
  - Logical & Bitwise operators
  - Strings



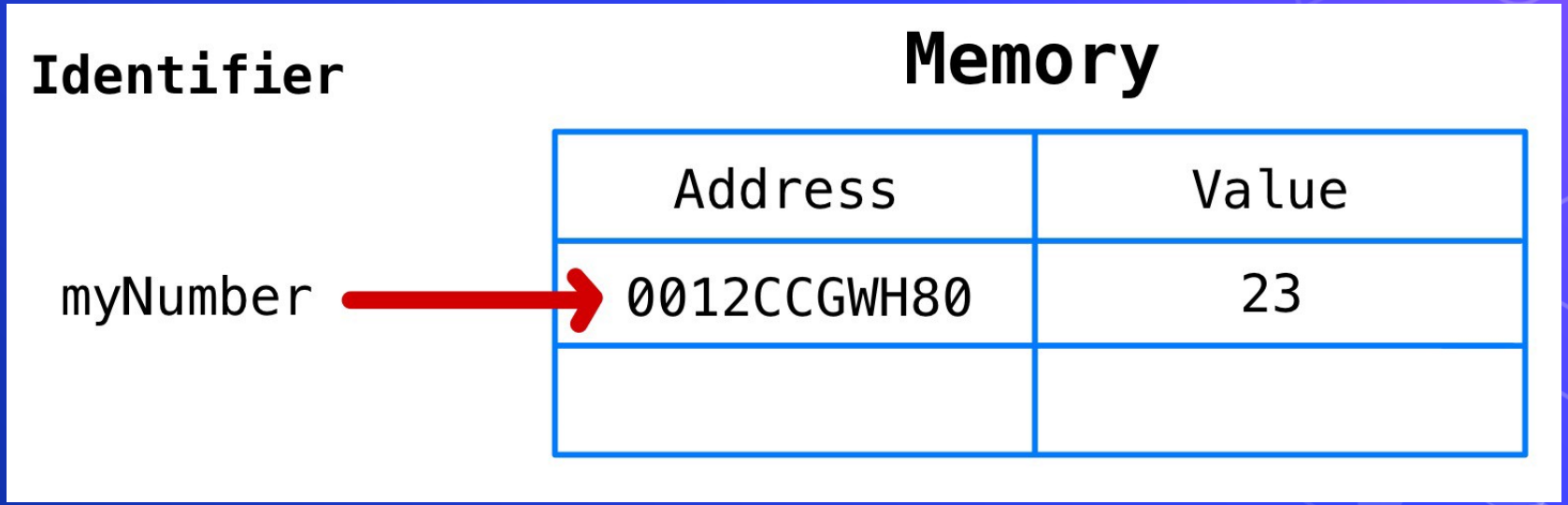
Session Content

- Variables
  - **Definition & Working with it**
  - Variable naming

- Data types
  - Numbers
  - Arithmetic operations
  - Boolean & Comparison
  - Logical & Bitwise operators
  - Strings

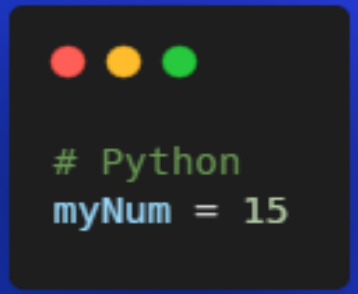
Variables

- Variable is a name that is used to refer to memory location. Python variable is also known as an identifier and used to hold value.

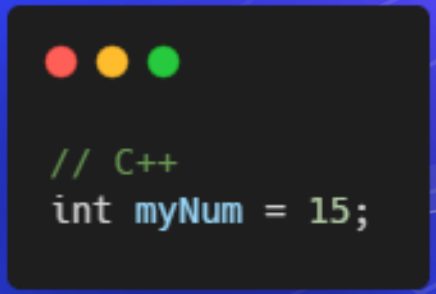


Variables

- In Python, we don't need to specify the type of variable because Python is a infer language and smart enough to get variable type.

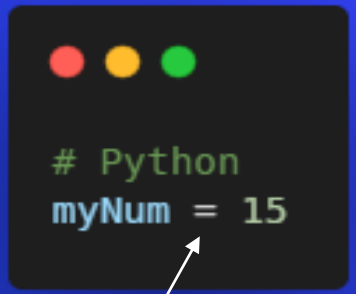


VS



Variables

- Assignment is done with a single equals sign (=).

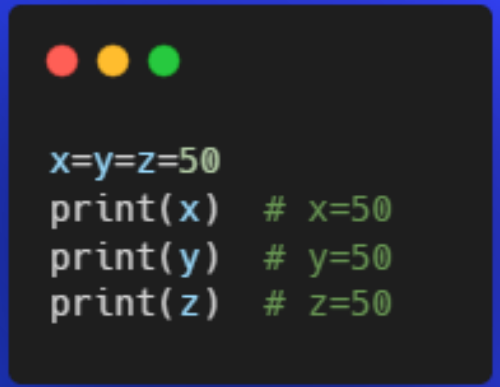


Assignment operator

Variables

- Python allows us to assign a value to multiple variables in a single statement, which is also known as multiple assignments.

1. Assigning single value to multiple variables:



```
x=y=z=50
print(x)  # x=50
print(y)  # y=50
print(z)  # z=50
```

Variables

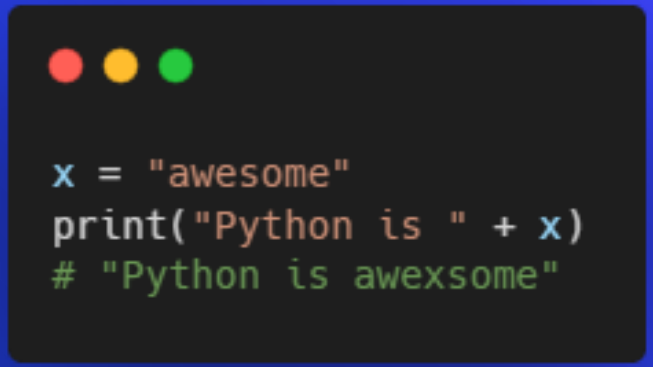
- Python allows us to assign a value to multiple variables in a single statement, which is also known as multiple assignments.

1. Assigning multiple values to multiple variables:

```
a,b,c=5,10,15
print(a) # a=5
print(b) # b=10
print(c) # c=15
```

Variables

- The Python print statement is often used to output variables.



```
x = "awesome"
print("Python is " + x)
# "Python is awexsome"
```

Variables

- We can delete the variable using the del keyword. The syntax is given below.

```
# Assigning a value to x
x = 6
print(x)
# deleting a variable.
del x
print(x)
```

```
# Output

'''
6
NameError: name 'x' is not defined
'''
```

Quiz

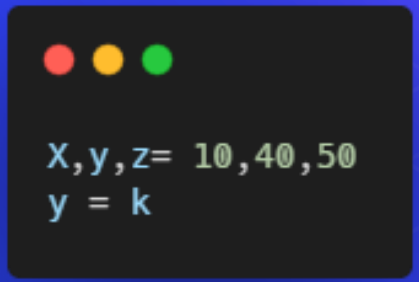
Q. What is the value of the variable **K** ?

A - 10

B - 100

C - 40

D - 50



Multiple Choice

Session Content

- Variables
  - Definition & Working with it
  - **Variable naming**

- Data types
  - Numbers
  - Arithmetic operations
  - Boolean & Comparison
  - Logical & Bitwise operators
  - Strings

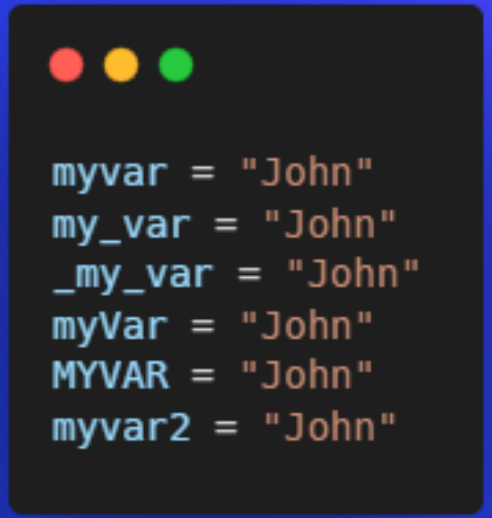
Variable naming

- The rules to name an identifier are given below.

- The first character of the variable must be an alphabet or underscore (_).
- All the characters except the first character may be an alphabet of lower-case(a-z), upper-case (A-Z), underscore, or digit (0-9).
- Identifier name mustn’t contain any white-space, or special character (!, @, #, %, ^, &,*).
- Identifier name must not be similar to any keyword defined in the language (print , if, for).
- Identifier names are case sensitive; for example, my name, and MyName is not the same.
- Use easy and meaningful name and related to the problem.

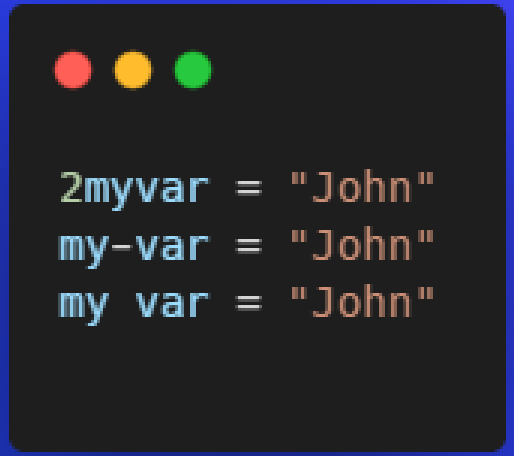
Variable naming

- Correct variable names:



Variable naming

- Wrong variable names:



Quiz

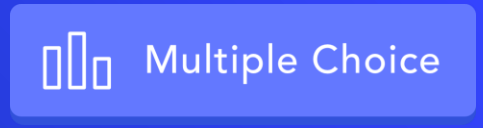
Q. Stores a piece of data, and gives it a specific name

A - variable

B - whitespace

C - interpreter

D - modulus

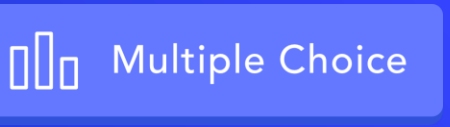


Quiz

Q. Is the name of this variable, correct? → V@riable1 = 15

A - True

B - False



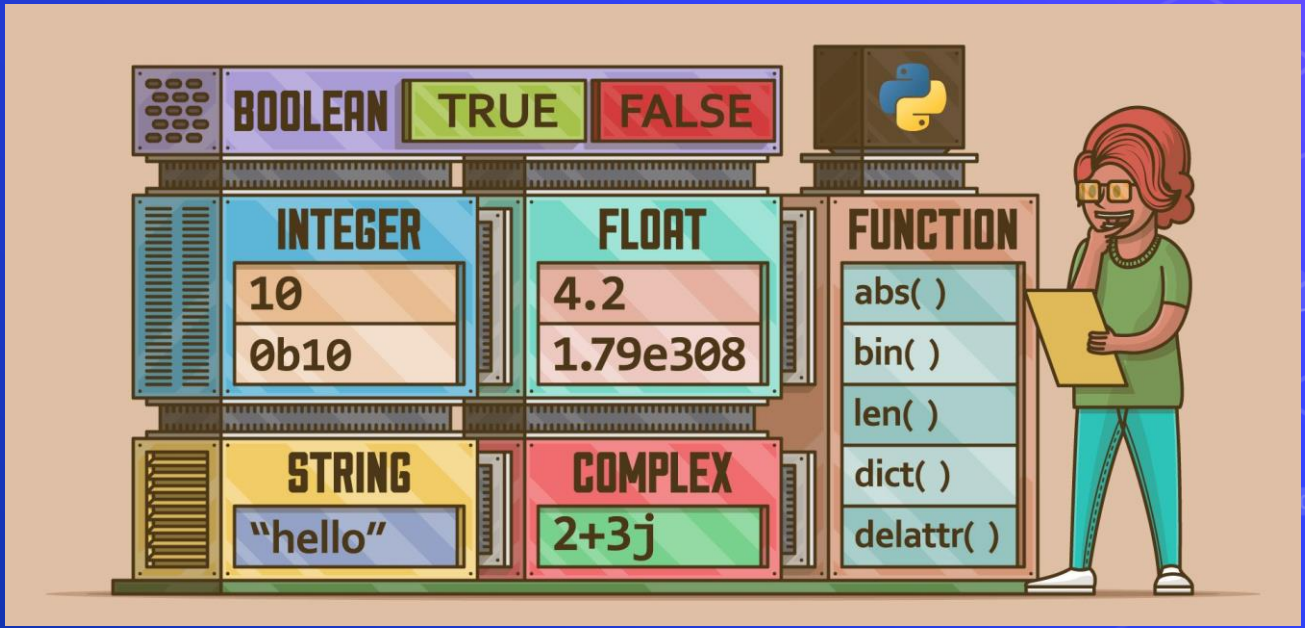
Session Content

- Variables
  - Definition & Working with it
  - Variable naming

- **Data types**
  - Numbers
  - Arithmetic operations
  - Boolean & Comparison
  - Logical & Bitwise operators
  - Strings

Data types

- Variables can hold values, and every value has a data-type.
- hence, we do not need to define the type of the variable while declaring it.
- The interpreter implicitly binds the value with its type.

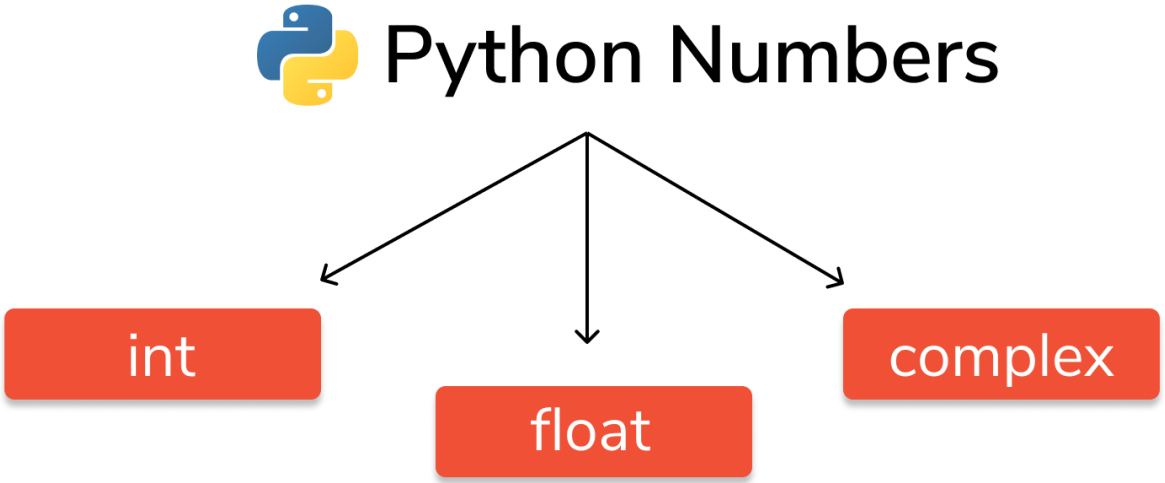


Session Content

- Variables
  - Definition & Working with it
  - Variable naming

- Data types
  - **Numbers**
  - Arithmetic operations
  - Boolean & Comparison
  - Logical & Bitwise operators
  - Strings

Numeric Types



Integer :

An integer (more commonly called an int) is a number without a decimal point.

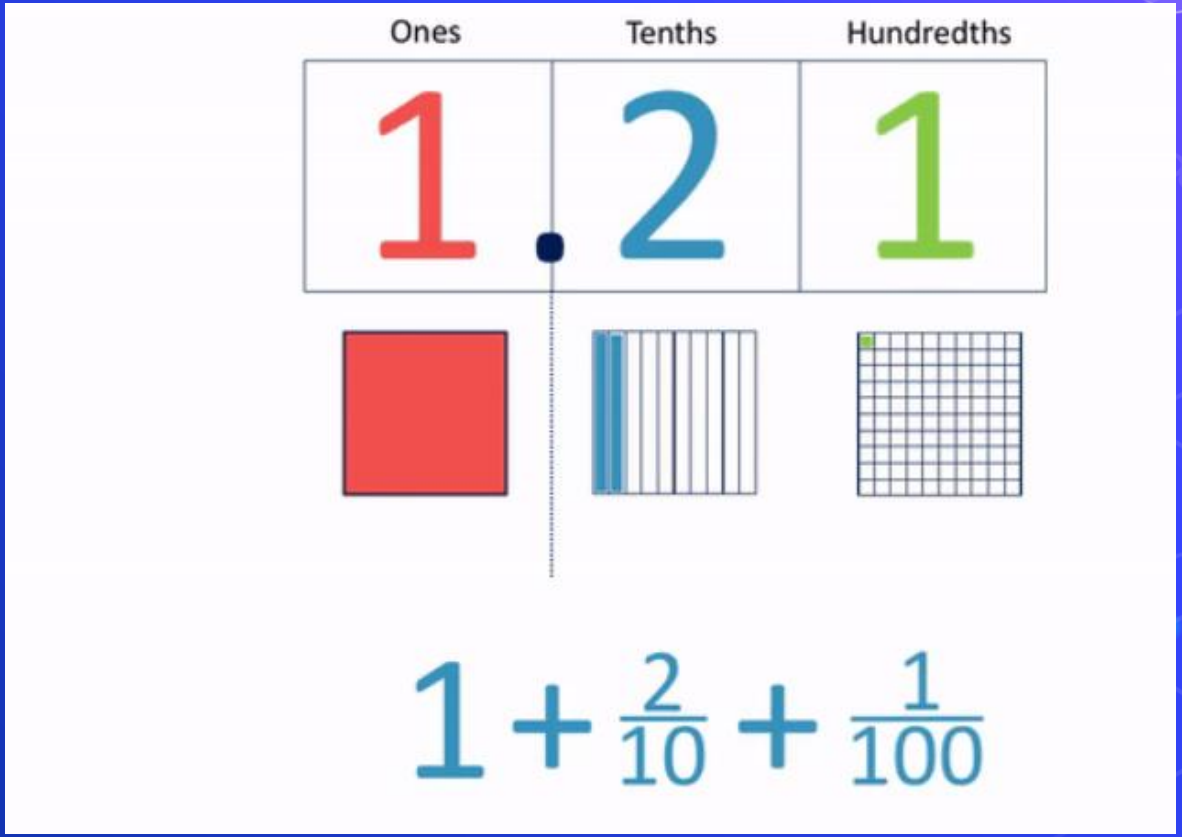
an Integer

is a whole number from the set of negative, non-negative, positive and 0 numbers.

- Negative : -1, -2, -3, -4, -5 ...
- Non-negative : 0, 6, 7, 8, 9 ...
- Positive : 1, 2, 3, 4, 5 ...
- Zero : 0 all by itself

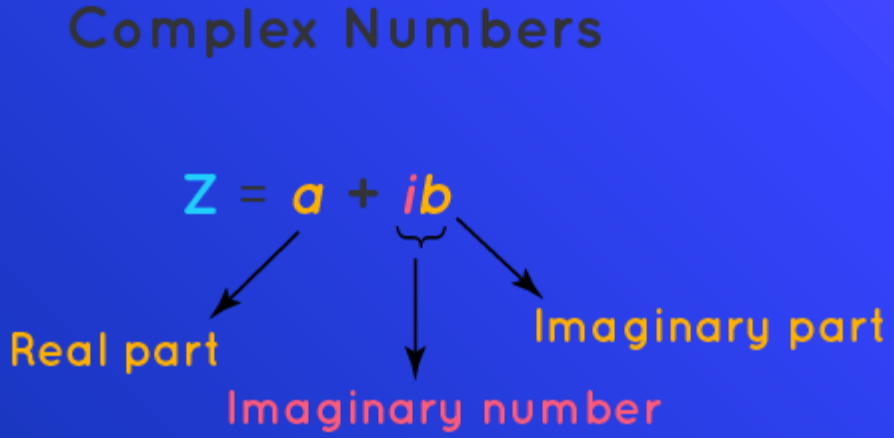
Float :

- A float is a floating-point number, which means it is **a number that has decimal place**.



complex number ：

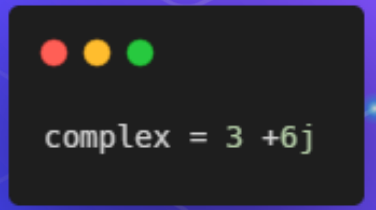
-A complex number is the sum of a real number and an imaginary number.



Integer :

Float :

Complex :



- We can use integer, float and complex To perform arithmetic operations

But first, We must learn about arithmetic operators.

Quiz

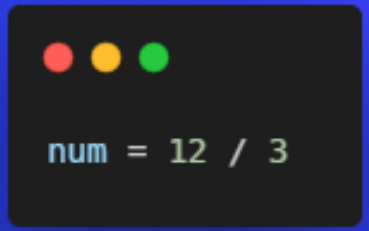
Q. What is the type of Output ?

A – int

B – float

C – complex

D – None of them



Multiple Choice

Session Content

- Variables
  - Definition & Working with it
  - Variable naming

- Data types
  - Numbers
  - **Arithmetic operations**
  - Boolean & Comparison
  - Logical & Bitwise operators
  - Strings

arithmetic operators :

- An operator that performs arithmetic operations on groups and numbers

Operators	Meaning	Example	Result
+	Addition	4 + 2	6
–	Subtraction	4 – 2	2
*	Multiplication	4 * 2	8
/	Division	4 / 2	2
%	Modulus operator to get remainder in integer division	5 % 2	1
**	Exponent	5**2 = 5^2	25
//	Integer Division/ Floor Division	5//2 -5//2	2 -3

arithmetic operators :

- An operator that performs arithmetic operations on groups and numbers

```
3 + 5        # result is 8
10 - 7       # result is 3
2 * 5        # result is 10
15 / 5       # result is 3
3 / 2        # result is 1.5
3 // 2       # result is 1
32 % 3       # result is 2
2 ** 3       # result is 8
4 ** 0.5     # result is 2
```

arithmetic operators :

- An operator that performs arithmetic operations on groups and numbers

```
x += 5   # x = x + 5
x -= 5   # x = x - 5
x *= 5   # x = x * 5
x /= 5   # x = x / 5
x %= 5   # x = x % 5
x //= 5  # x = x // 5
x **= 5  # x = x ** 5
```

A different way to write the equation

Quiz

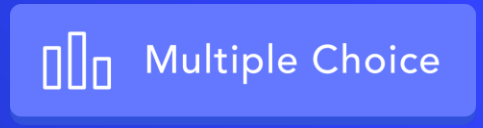
Q. Arithmetic Operators, What is a % ?

A - Multiplication

B - Modulo

C - Division

D - Increment



Quiz

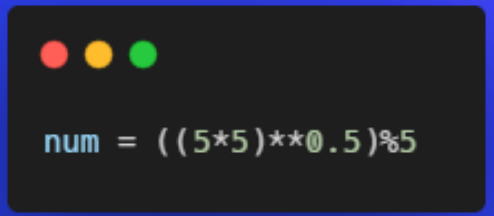
Q. What is the value of **num** ??

A - 0

B - 1

C - 5

D – 0.0



Multiple Choice

Practice # 1

Ask the user to enter the radius of a circle and then print the area of the circle .

```
r = float( input() )
print (3.14 * r * r)
(3.14 * r ** 2)
```

Practice # 2

Ask the user to enter his height in cm and his weight in kg , then print his BMI (Body Mass Index).

Practice # 3

Write an algorithm to determine the flying time in minutes between two cities given the mileage between them and the average speed of the airplane with km/h.

Practice # 4

Ask the user to enter a number in variable called “a” , then ask him to enter another number in variable called “b” , then print the result type of the Operation a/b .

Task # 1

Write an algorithm that takes the number of wins, draws and losses and calculates the number of points a football team has obtained so far.

Session Content

- Variables
  - Definition & Working with it
  - Variable naming

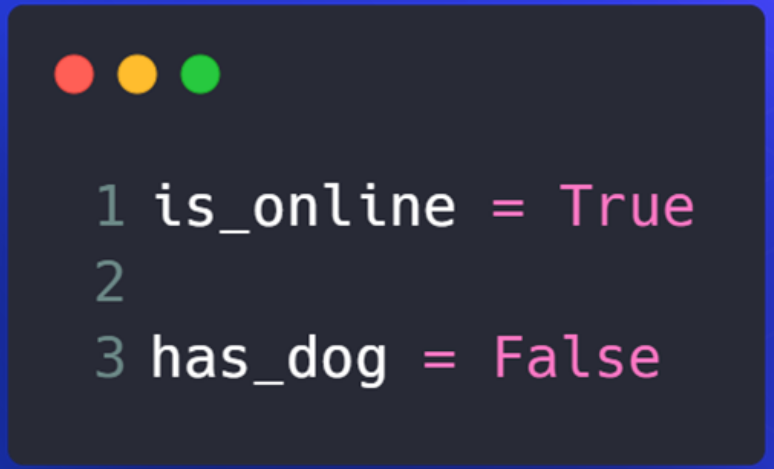
- Data types
  - Numbers
  - Arithmetic operations
  - **Boolean & Comparison**
  - Logical & Bitwise operators
  - Strings

Boolean :

- Boolean refers to a system of logical thought that is used to create true/false statements.

Boolean :

- Boolean refers to a system of logical thought that is used to create true/false statements.



Comparison operator :

- operators that compare values and return true or false .

Relational Operators

Operators	Meaning	Example	Result
<	Less than	5<2	False
>	Greater than	5>2	True
<=	Less than or equal to	5<=2	False
>=	Greater than or equal to	5>=2	True
==	Equal to	5==2	False
! =	Not equal to	5! =2	True

Comparison operator :

- operators that compare values and return true or false .

```
1 5 == 5      # result is True
2 5 != 5      # result is False
3 10 > 7      # result is True
4 2 >= 5      # result is False
5 15 < 5      # result is False
6 3 <= 3      # result is True
```

Quiz

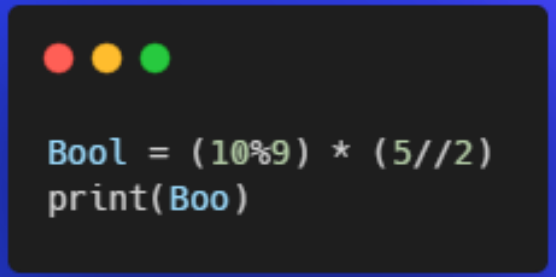
Q. What is the expected Output ?

A – 3.0

B – 2.0

C – 1.0

D – 4.0



```
Bool = (10%9) * (5//2)
print(Boo)
```

Multiple Choice

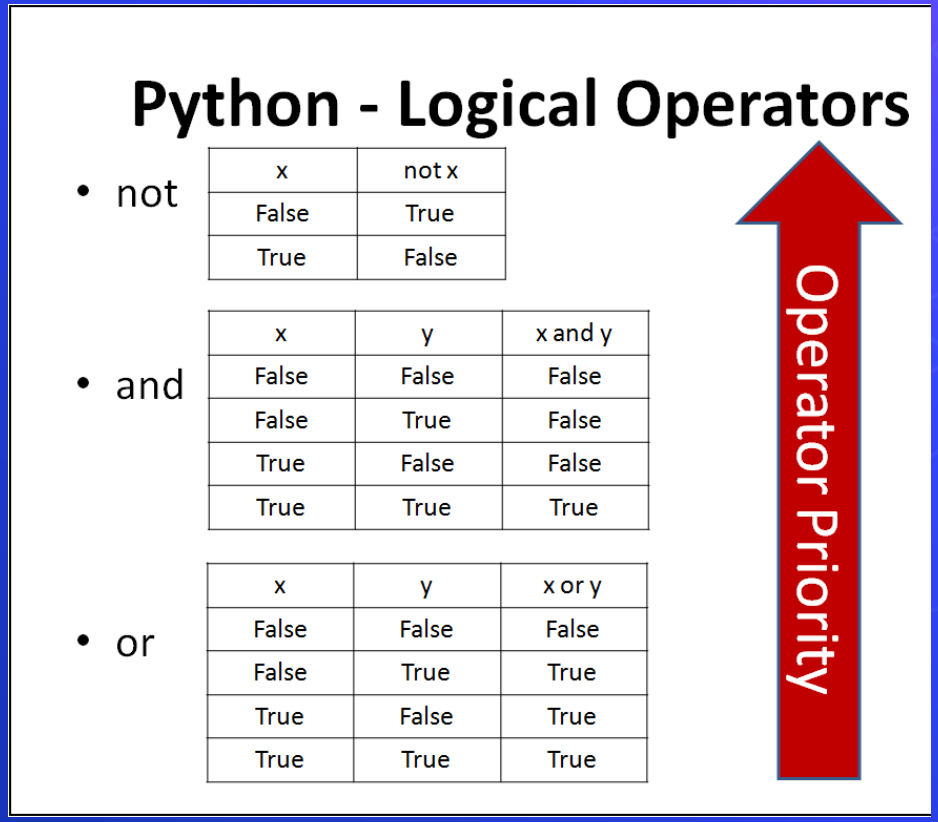
Session Content

- Variables
  - Definition & Working with it
  - Variable naming

- Data types
  - Numbers
  - Arithmetic operations
  - Boolean & Comparison
  - **Logical & Bitwise operators**
  - Strings

Logical operator :

- Logical operators are used on conditional statements (either True or False).



Logical operator :

- Logical operators are used on conditional statements (either True or False).

```
1 # AND
2 1 < 2 and 2 < 3          # Result is True
3 1 != 1 and 2 < 3         # Result is False
4 1 != 1 and 2 > 3         # Result is False
5
6
7 # OR
8 1 < 2 or 2 < 3           # Result is True
9 1 != 1 or 2 < 3          # Result is True
10 1 != 1 or 2 > 3         # Result is False
11
12
13 # NOT
14 not 1 == 1              # Result is False
15 not 1 > 10              # Result is True
```

Operators Precedence :

Python Operator Precedence

Precedence	Operator Sign	Operator Name
Highest	**	Exponentiation
	+x, -x, ~x	Unary positive, unary negative, bitwise negation
	*, /, //, %	Multiplication, division, floor, division, modulus
	+, -	Addition, subtraction
	<<, >>	Left-shift, right-shift
	&	Bitwise AND
	^	Bitwise XOR
	|	Bitwise OR
	==, !=, <, <=, >, >=, is, is not	Comparison, identity
	not	Boolean NOT
	and	Boolean AND
Lowest	or	Boolean OR

Practice

Create a program that takes a number and prints True if it's less than or equal to zero, otherwise print False.

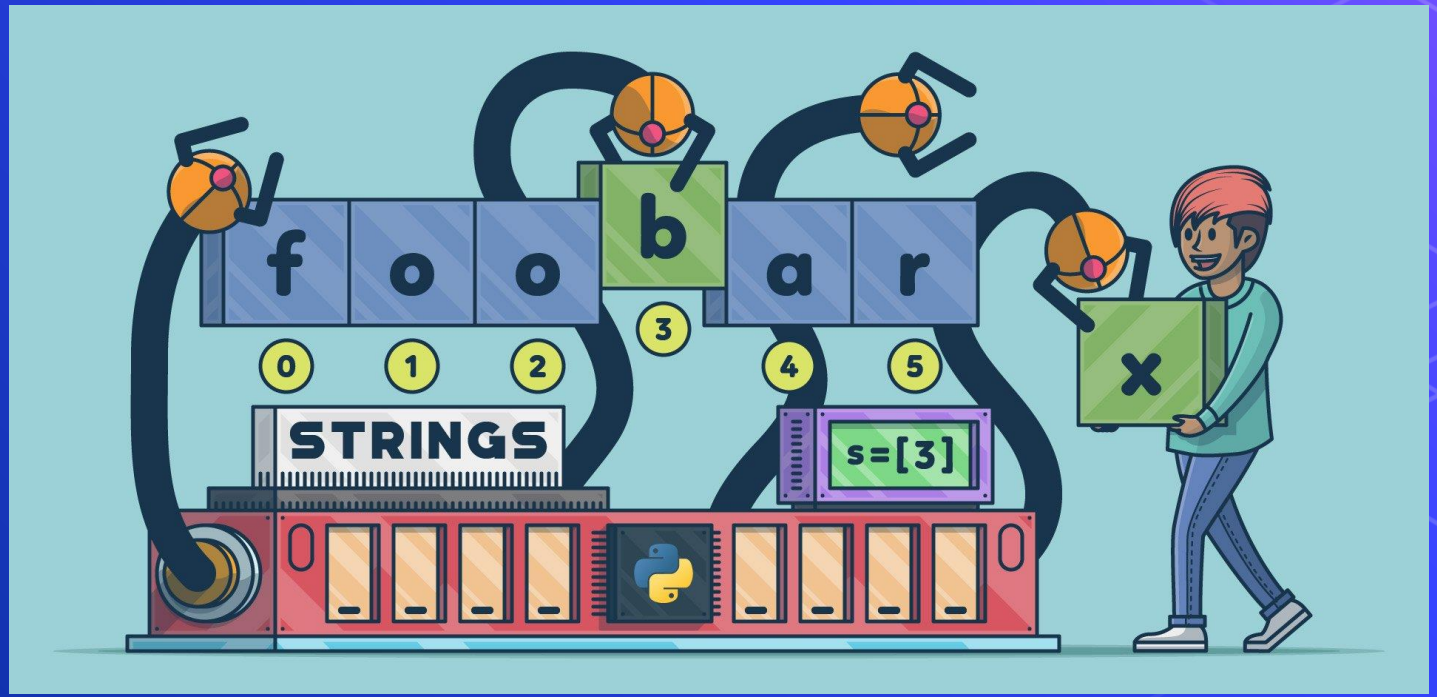
Session Content

- Variables
  - Definition & Working with it
  - Variable naming

- Data types
  - Numbers
  - Arithmetic operations
  - Boolean & Comparison
  - Logical & Bitwise operators
  - **Strings**

String :

- Python string is the collection of the characters surrounded by single quotes, double quotes, or triple quotes.
- Each character is encoded in the ASCII or Unicode character. So, we can say that Python strings are also called the collection of Unicode characters.



String :

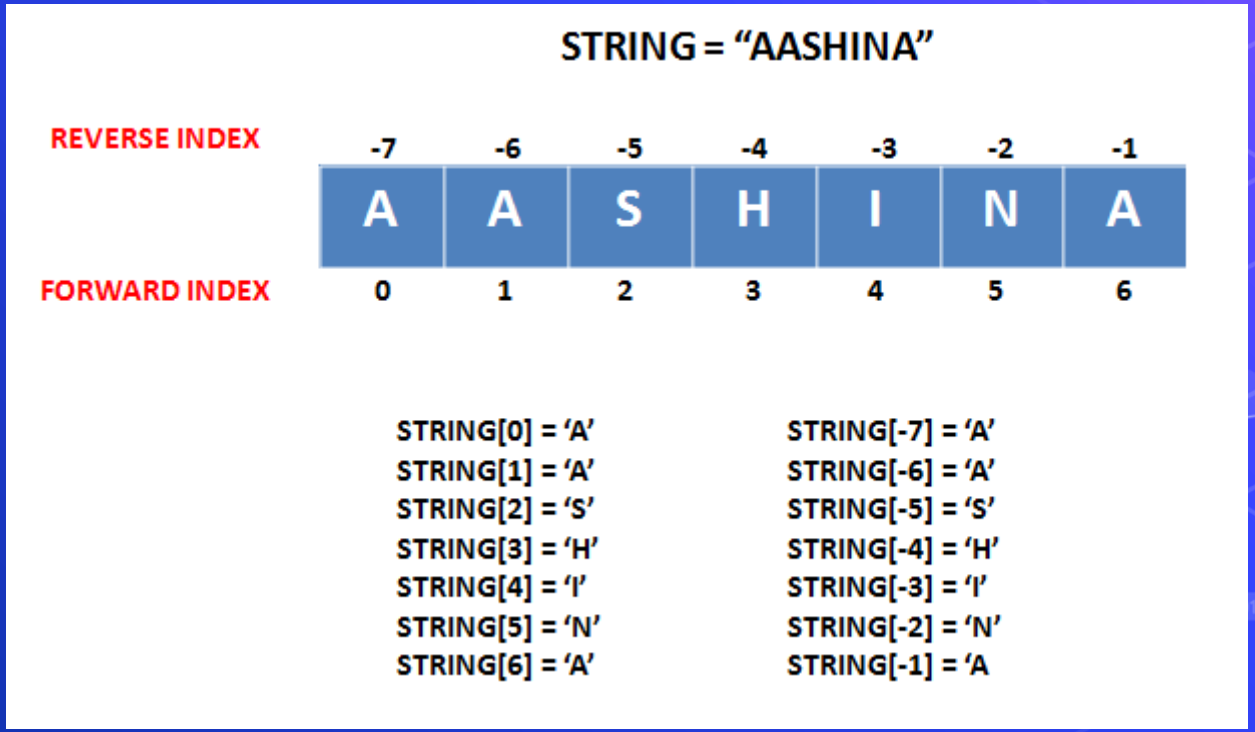
- Python string is the collection of the characters surrounded by single quotes, double quotes, or triple quotes.
- Each character is encoded in the ASCII or Unicode character. So, we can say that Python strings are also called the collection of Unicode characters.

```
#Using single quotes
str1 = 'Hello Python'
print(str1)
#Using double quotes
str2 = "Hello Python"
print(str2)

#Using triple quotes
str3 = '''''Triple quotes are generally used for
    represent the multiline or
    docstring'''
print(str3)
```

String indexing :

- Indexing allows you to access individual characters in a string directly by using a numeric value.



String indexing :

- Indexing allows you to access individual characters in a string directly by using a numeric value.

```
# String slicing
String ='ASTRING'

# Using indexing sequence
print(String[0]) #A
print(String[3]) #R
```

String Slicing :

- You can return a range of characters by specify the start index and the end index, separated by a colon, to return a part of the string.

str = “HELLO”

H	E	L	L	O
-5	-4	-3	-2	-1

str[-1] = ‘O’ str[-3:-1] = ‘LL’

str[-2] = ‘L’ str[-4:-1] = ‘ELL’

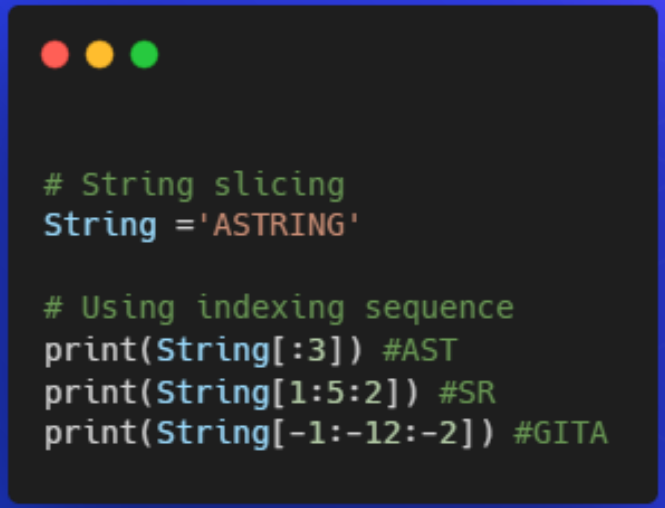
str[-3] = ‘L’ str[-5:-3] = ‘HE’

str[-4] = ‘E’ str[-4:] = ‘ELLO’

str[-5] = ‘H’ str[::-1] = ‘OLLEH’

String Slicing :

- You can return a range of characters by specify the start index and the end index, separated by a colon, to return a part of the string.



```
# String slicing
String ='ASTRING'

# Using indexing sequence
print(String[:3]) #AST
print(String[1:5:2]) #SR
print(String[-1:-12:-2]) #GITA
```

Quiz

Q. What is the expected Output ?

A - worl

B – llo,

C - world

D - Hello

Multiple Choice

Modify Strings:

- Python has a set of built-in methods that you can use on strings.

- lower(): Converts all uppercase characters in a string into lowercase.
- upper(): Converts all lowercase characters in a string into uppercase.
- title(): Convert string to title case.

- ✓ To see all string function : https://www.w3schools.com/python/python_strings_methods.asp

Modify Strings:

- Python has a set of built-in methods that you can use on strings.

```python
# Python3 program to show the
# working of upper() function
text = 'Epsilon AI'

# upper() function to convert
# string to upper case
print("\nConverted String:")
print(text.upper())

# lower() function to convert
# string to lower case
print("\nConverted String:")
print(text.lower())

# converts the first character to
# upper case and rest to lower case
print("\nConverted String:")
print(text.title())

# original string never changes
print("\nOriginal String")
print(text)
```

```
#OUTPUT
'''
Converted String:
EPSILON AI

Converted String:
epsilon ai

Converted String:
Epsilon Ai

Original String
Epsilon AI
'''
```

Quiz

Q. What is the expected Output ?

A - WELCOME TO MY WORLD

B - Welcome To My World

C - welcome to my world

D – Welcome To my world

Multiple Choice

Modify Strings:

- Python has a set of built-in methods that you can use on strings.

- replace(): returns a copy of the string where all occurrences of a substring is replaced with another substring.
- split(): return a list of the words of the string, If the optional second argument sep is absent or None
- strip(): It returns a copy of the string with both leading and trailing white spaces removed

✓ To see all string function : https://www.w3schools.com/python/python_strings_methods.asp

Modify Strings:

- Python has a set of built-in methods that you can use on strings.

```python
# Python3 program to demonstrate the
# use of replace() method

string = "I love pyhon"

# Prints the string by replacing all
print(string.replace("pyhon", "Java"))
```

```
# OUTPUT
'''

I love Java

'''
```

Modify Strings:

- Python has a set of built-in methods that you can use on strings.

```
# Python3 program to demonstrate the
# use of split() method

line = "I Love Python"
print(line.split())
```

```
# OUTPUT
'''

['I', 'Love', 'Python']

'''
```

Modify Strings:

- Python has a set of built-in methods that you can use on strings.

```python
# Python3 program to demonstrate the use of
# strip() method

string = """    I love python """

# prints the string without stripping
print(string)

# prints the string by removing leading and trailing whitespaces
print(string.strip())
```

```
# OUTPUT
'''

    I love python
I love python

'''
```

Quiz

Q. A function used to return a trimmed version of the string

A – strip( )

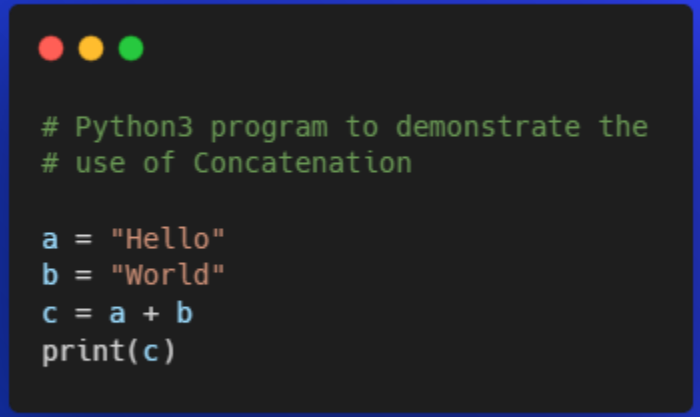
B – upper( )

C - split( )

D – replace( )

String Concatenation:

- To concatenate, or combine, two strings you can use the + operator.



```python
# Python3 program to demonstrate the
# use of Concatenation

a = "Hello"
b = "World"
c = a + b
print(c)
```

```
# OUTPUT
'''

HelloWorld

'''
```