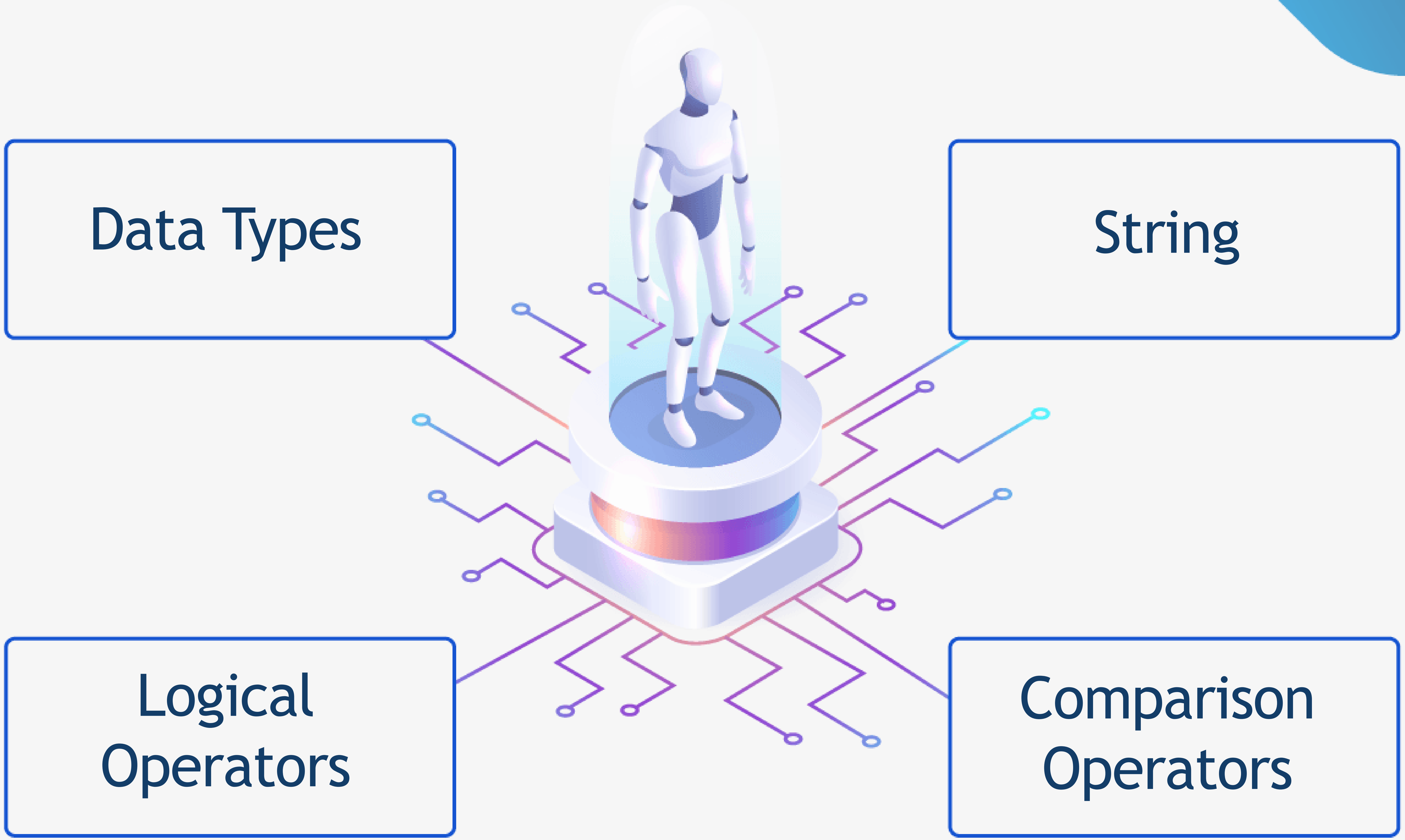
The Learning Hub

Introduction to Programming

Education and Training Solutions 2023



Data Types

String

Logical Operators

Comparison Operators

Data Types

Introduction to Data Types

- **Numbers**

Integer: 1 , 2 , 3 , 4 , etc..

Float: 1.1 , 2.2 , 3.3 , 4.4 , etc..

Complex: 1+6j , 6+3J

- **Boolean**

True -> 1

False -> 0

- **Strings**

"Tahaluf Training"

'Artificial Intelligence'

Operators

- Python divides the operators into the following groups:

Arithmetic operators (+, -, *, /, %, **, //)

Assignment operators (=, +=, -=, *=, /=, %=, //=, **=, &=, |=, ^=, >>=, <<=)

Comparison operators (==, !=, >, <, >=, <=)

Logical operators (and, or, not)

Identity operators (is, is not)

Membership operators (in, not in)

Bitwise operators (&, |, ^, ~, <<, >>)

Expressions

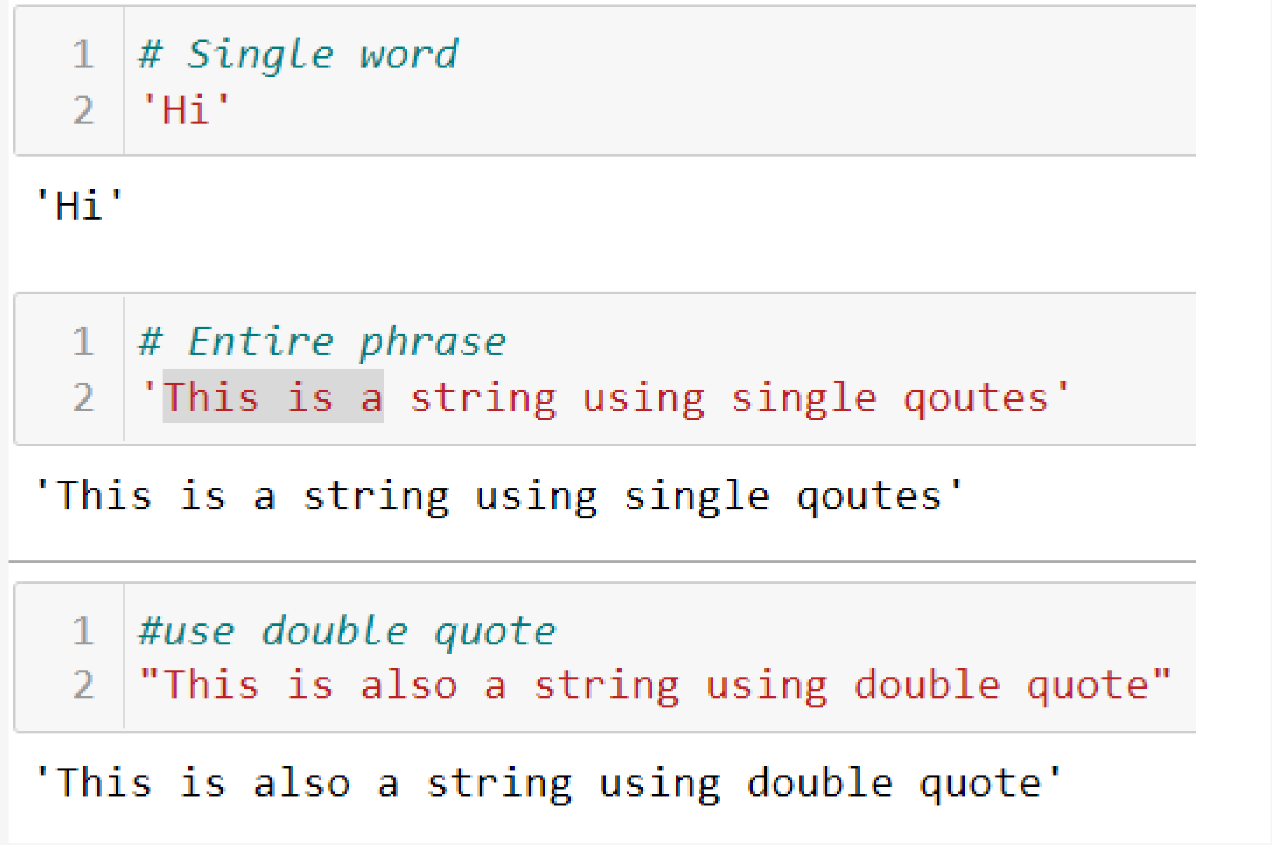
- Expressions form:
  - Identifier: Any name that refers to a class, function, or variable (X, Y).
  - Literals: Way of writing a value (1, 2, 9).
  - Operators: Tokens that represent certain operations (+, -, *).
- Examples:
  - 1+6*2
  - 4**0.5
  - 25//5
  - 5/2
  - x=5+y

The Learning Hub

Strings

Introduction to String

- **Creating Strings :** strings can be created in Python using either single quotes or double quotes



```
# Single word
'Hi'
```

```
'Hi'
```

```
# Entire phrase
'This is a string using single qoutes'
```

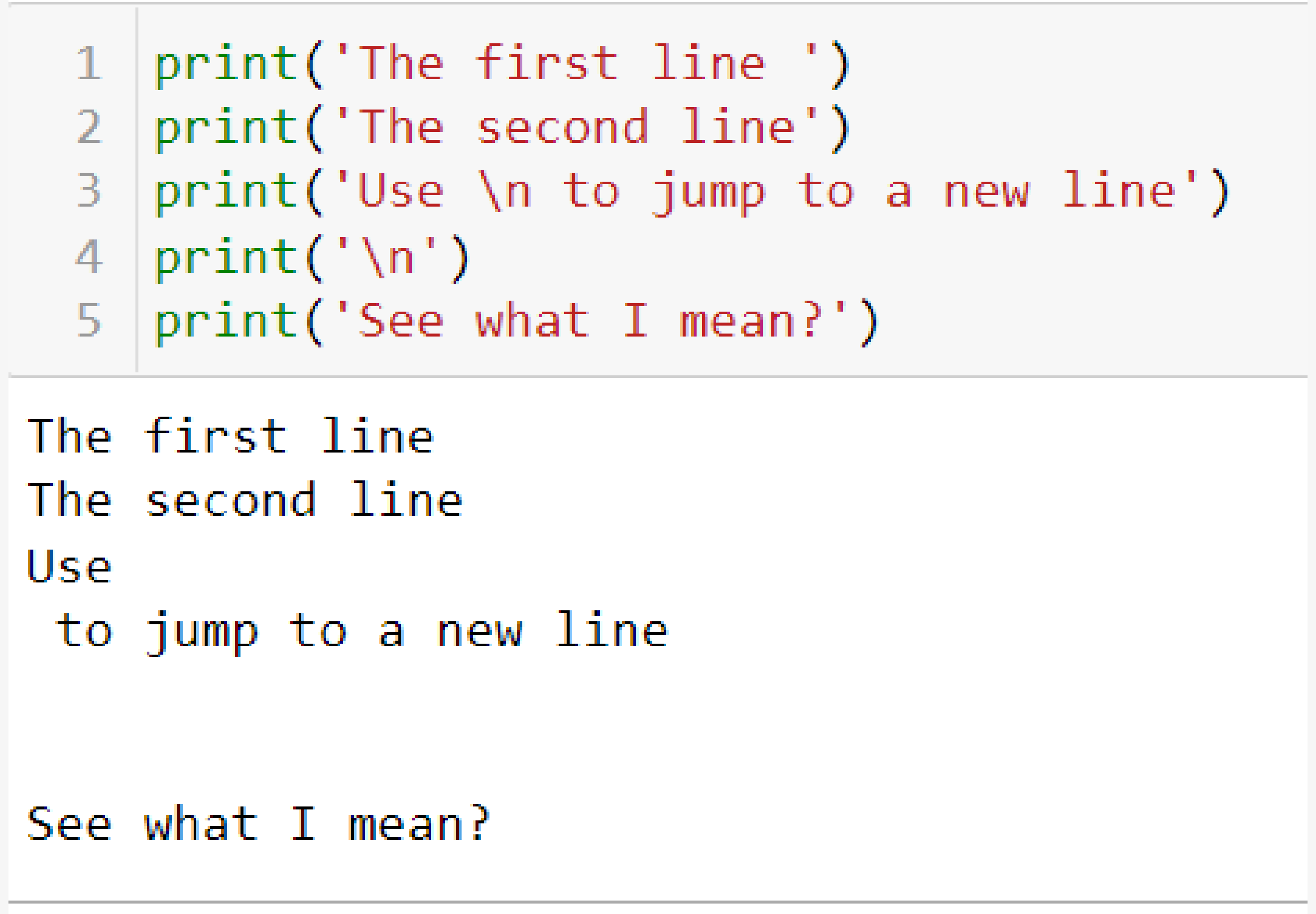
```
'This is a string using single qoutes'
```

```
#use double quote
"This is also a string using double quote"
```

```
'This is also a string using double quote'
```

Introduction to String

- **Print Srting:** using a print statement.



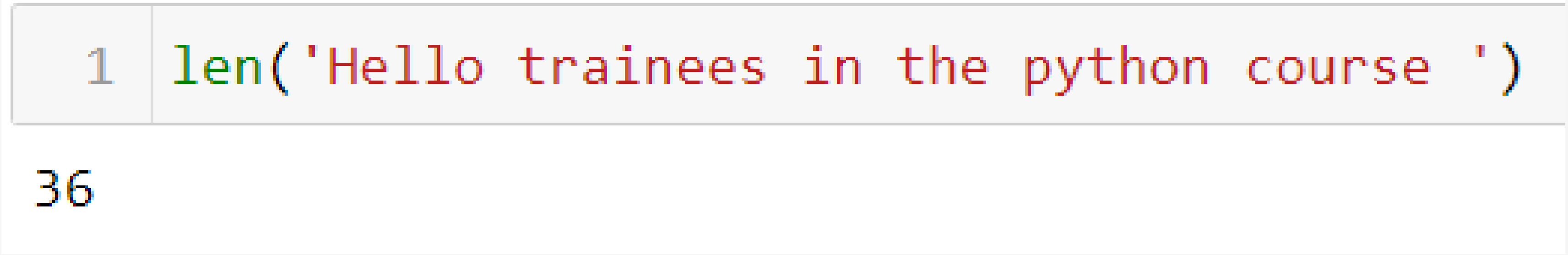
```python
print('The first line ')
print('The second line')
print('Use \n to jump to a new line')
print('\n')
print('See what I mean?')
```

```
The first line
The second line
Use
 to jump to a new line


See what I mean?
```

Introduction to String

- **String length:** To get the length of a string, use the built-in function len().

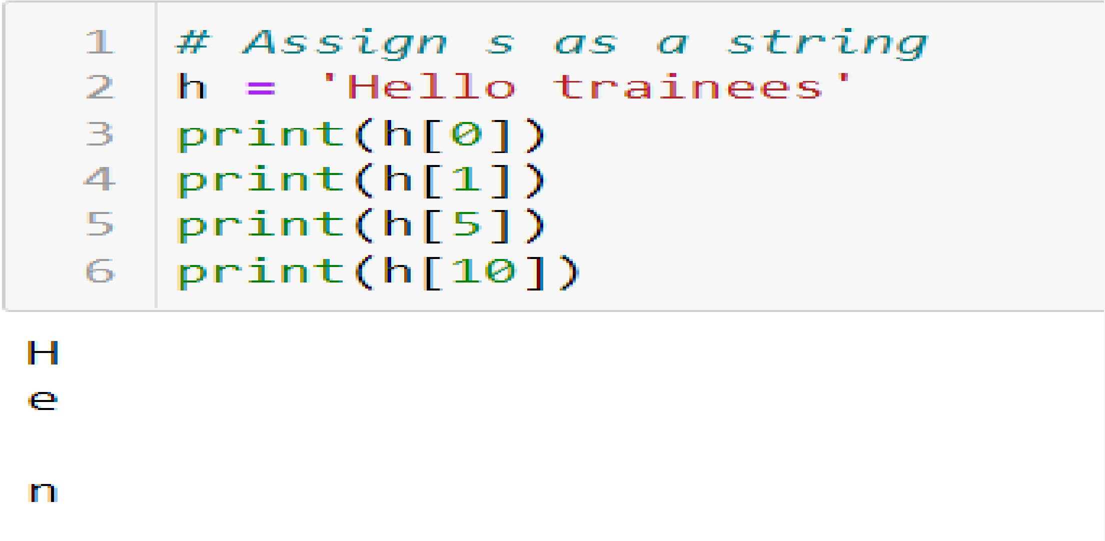


```
len('Hello trainees in the python course ')
```

```
36
```

Access String characters

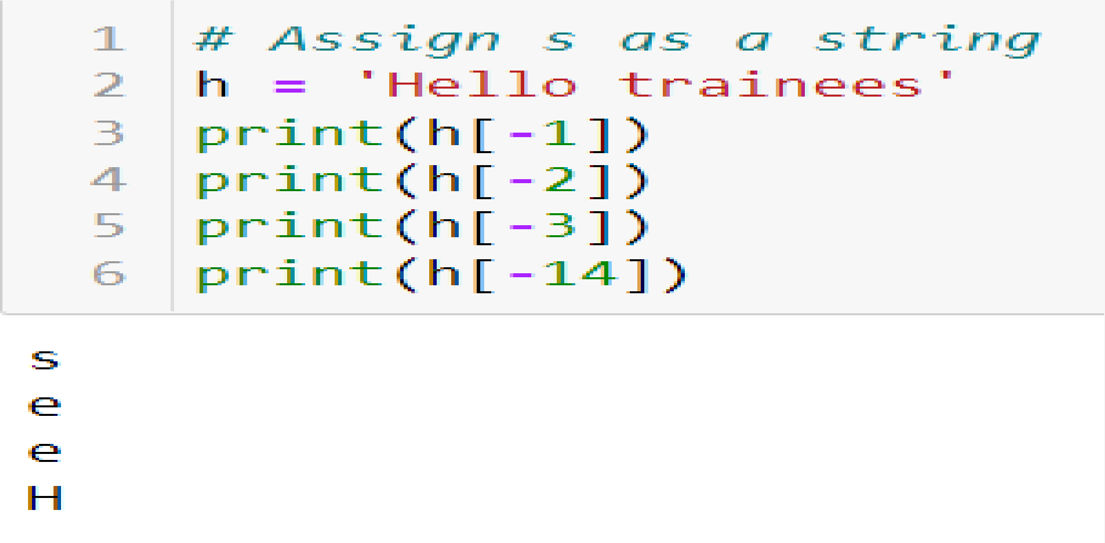
- **String Indexing**



```
# Assign s as a string
h = 'Hello trainees'
print(h[0])
print(h[1])
print(h[5])
print(h[10])
```

```
H
e
 
n
```

- **Negative Indexing**

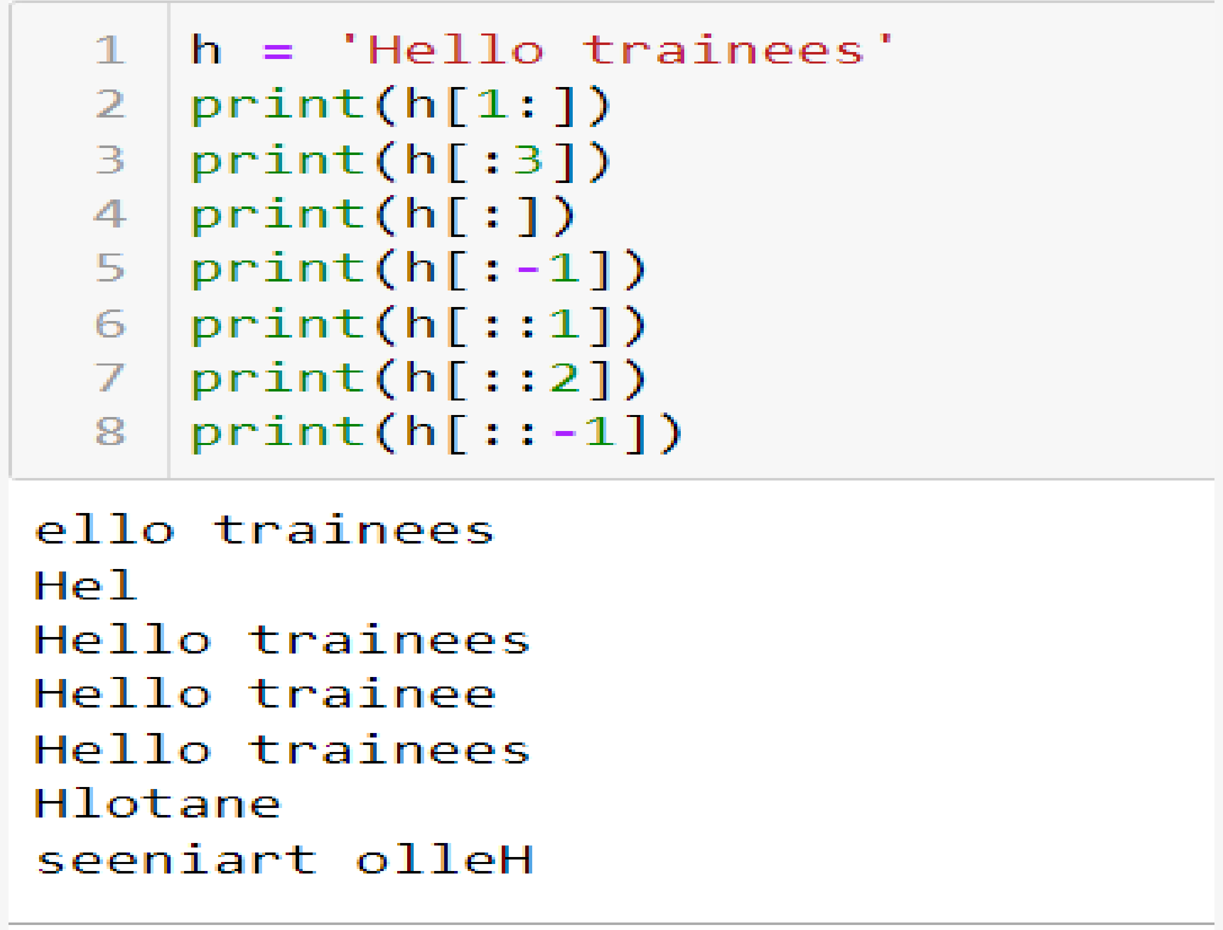


```
# Assign s as a string
h = 'Hello trainees'
print(h[-1])
print(h[-2])
print(h[-3])
print(h[-14])
```

```
s
e
e
H
```

Access String characters

- **Range of Indexes:** You can specify a range of indexes by specifying the start and end points

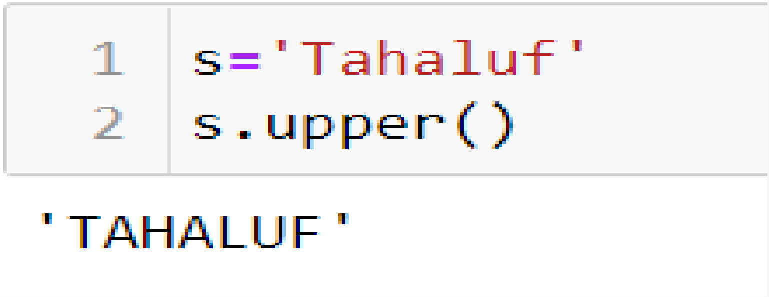


```python
h = 'Hello trainees'
print(h[1:])
print(h[:3])
print(h[:])
print(h[:-1])
print(h[::1])
print(h[::2])
print(h[::-1])
```

```
ello trainees
Hel
Hello trainees
Hello trainee
Hello trainees
Hlotane
seeniart olleH
```

Basic Built-in String methods

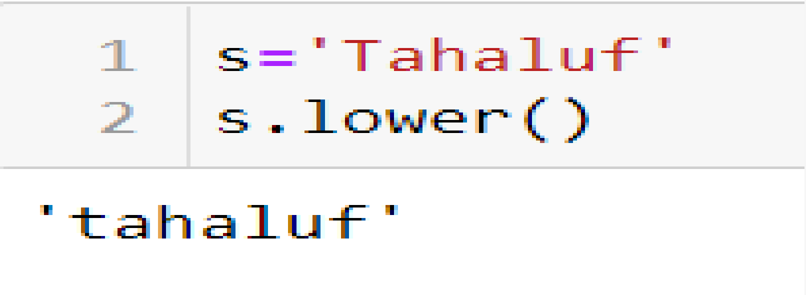
- **Upper Case:** The upper() method returns the string in upper case.



```
s='Tahaluf'
s.upper()
```

```
'TAHALUF'
```

- **Lower Case:** The lower() method returns the string in lower case.

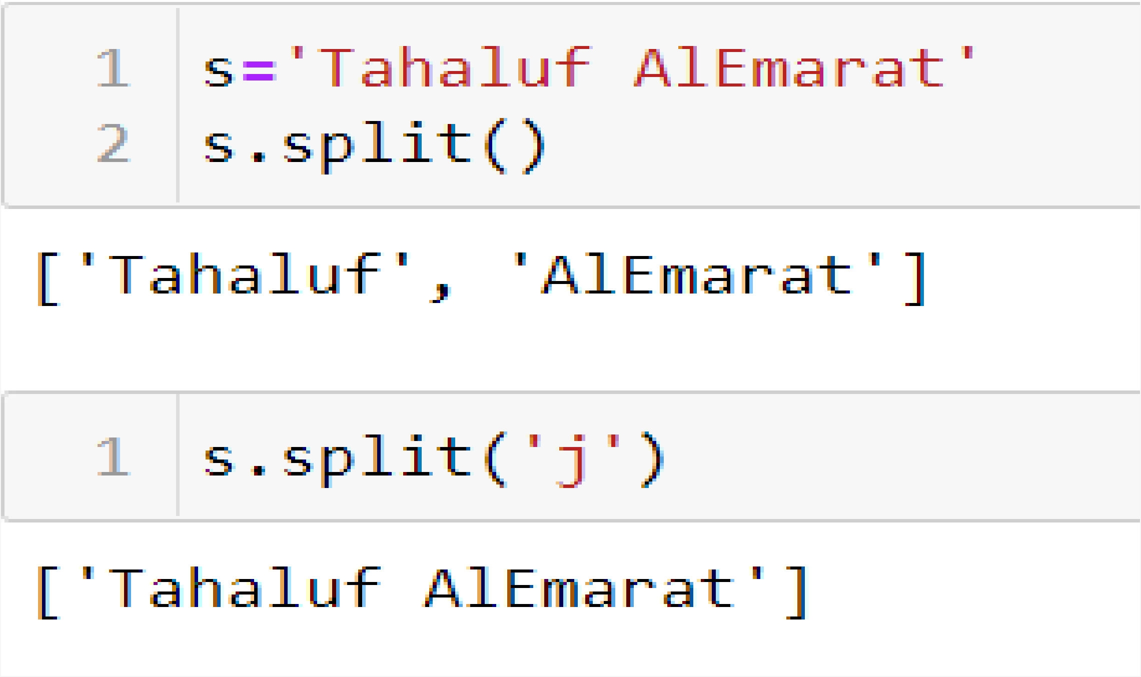


```
s='Tahaluf'
s.lower()
```

```
'tahaluf'
```

Basic Built-in String methods

- Split String: The split() method splits the string into substrings if it finds instances of the separator.



```
s='Tahaluf AlEmarat'
s.split()
```

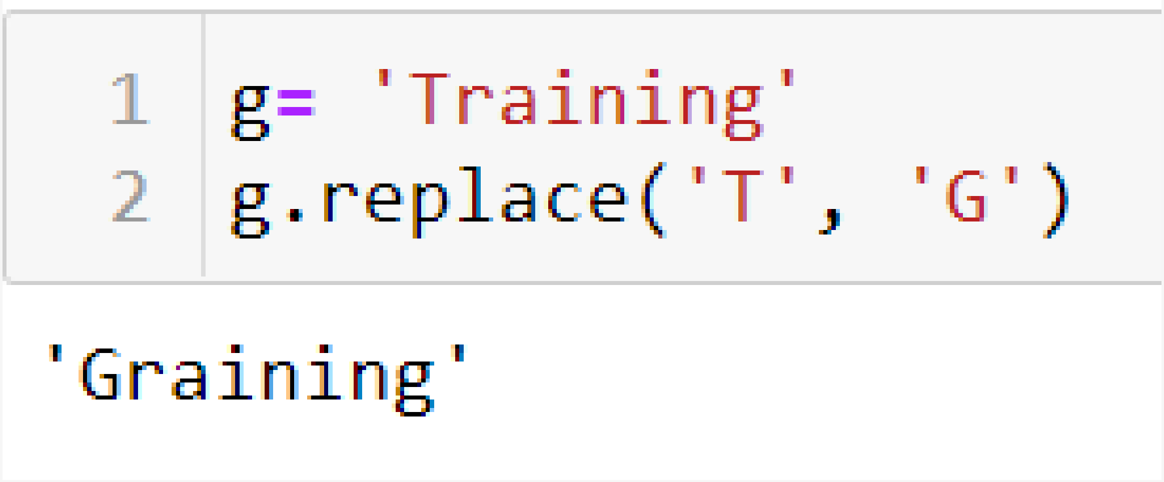
```
['Tahaluf', 'AlEmarat']
```

```
s.split('j')
```

```
['Tahaluf AlEmarat']
```

Basic Built-in String methods

- Replace String: The replace() method replaces a string with another string.

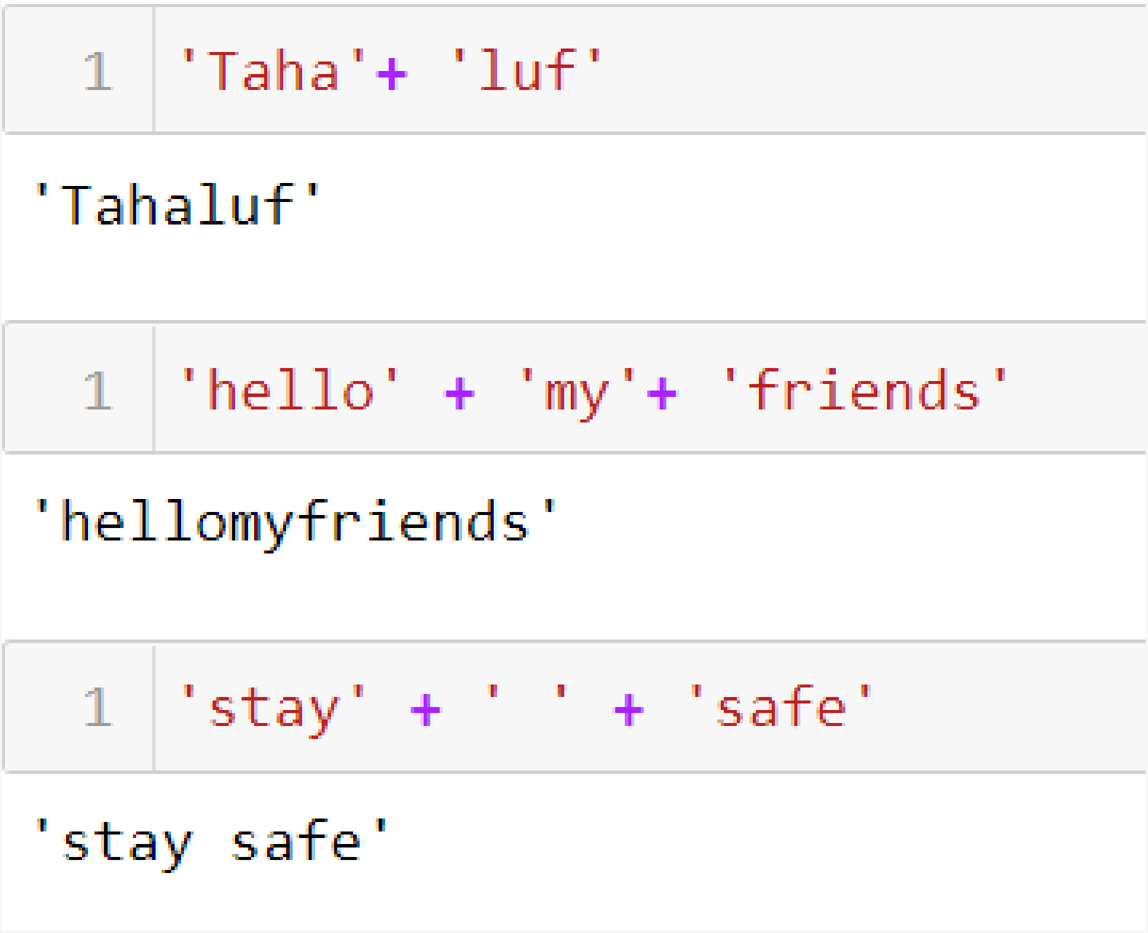


```
g= 'Training'
g.replace('T', 'G')
```

```
'Graining'
```

Strings Concatenation

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



```
'Taha'+ 'luf'
```

```
'Tahaluf'
```

```
'hello' + 'my'+ 'friends'
```

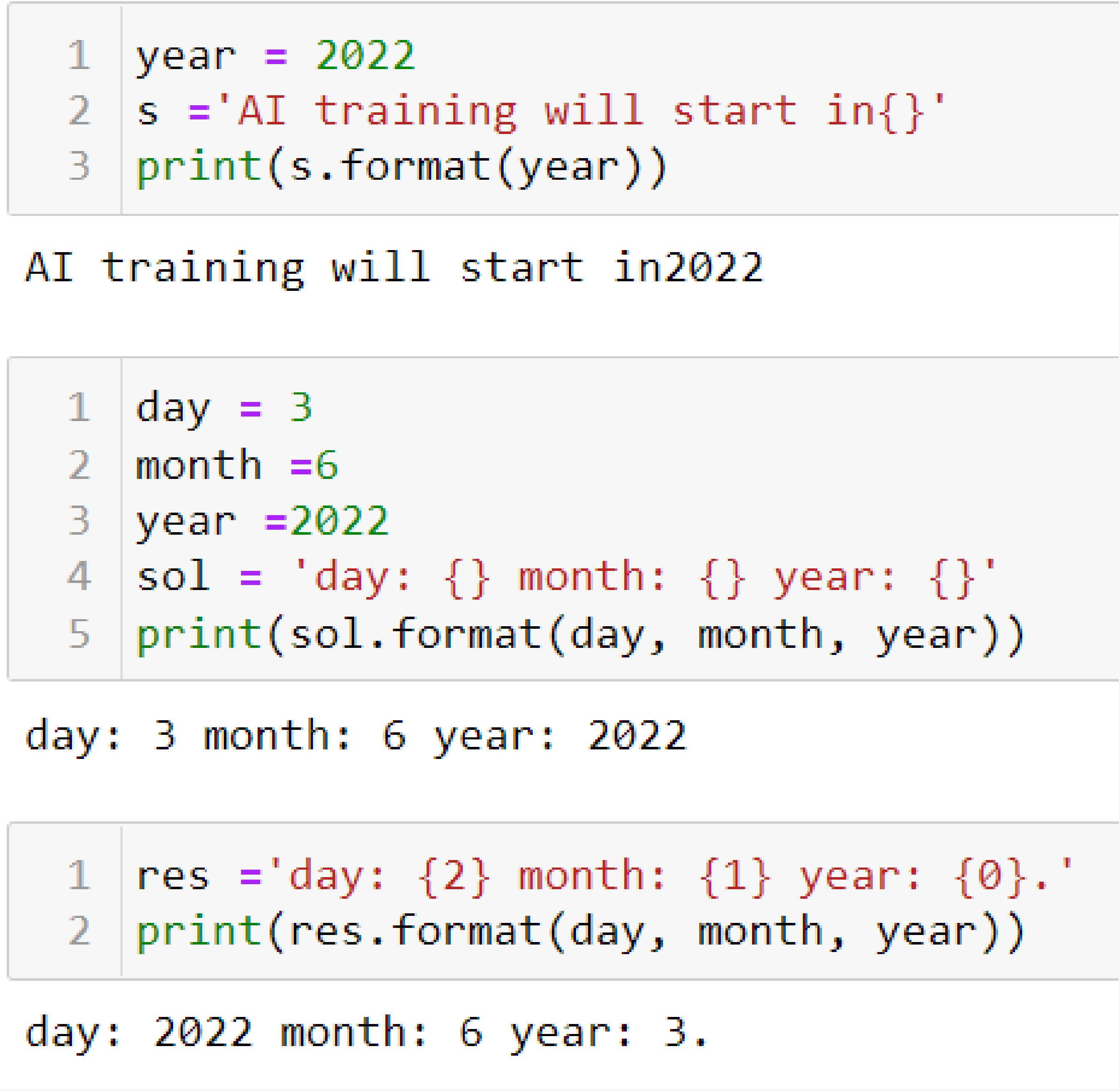
```
'hellomyfriends'
```

```
'stay' + ' ' + 'safe'
```

```
'stay safe'
```

Strings Format

- Use the format() method to insert numbers into strings.



```python
year = 2022
s ='AI training will start in{}'
print(s.format(year))
```

```
AI training will start in2022
```

```python
day = 3
month =6
year =2022
sol = 'day: {} month: {} year: {}'
print(sol.format(day, month, year))
```

```
day: 3 month: 6 year: 2022
```

```python
res ='day: {2} month: {1} year: {0}.'
print(res.format(day, month, year))
```

```
day: 2022 month: 6 year: 3.
```

Strings Escape Characters

Escape	Summary	Examples	Output
\'	Single Quote	txt = ‘It\’s fine.’	It’s fine
\\	Back Slash	txt = “hi \\ guys”	hi \ guys
\n	New Line	txt = ‘hi \n there’	hi there
\t	Tab	txt= ”alpha\tbeta”	alpha beta
\b	Back Space	txt = "Hello \bFriends!"	HelloFriends!

References

- Python Tutorial (w3schools.com)
- The Python Tutorial — Python 3.10.7 documentation
- Python Tutorial for Beginners: Learn Programming Basics [PDF] (guru99.com)

THANK YOU