EASYLEARNING.GURU

Data types: Numbers and Strings

Numbers

Number data types store numeric values. They are immutable data types which means that changing the value of a number data type results in a newly allocated object.

Python supports different numerical types:

- Floating point numbers, that have fractional part e.g. 3.14, 3.333
- Complex numbers, that have imaginary parts e.g. 1 + 2j
- Integers, that have no fractional part e.g. 123 , 5

Number Operations

EASYLEARNING.GURU

Operations	Interpretations
x or y	Logical OR (y is evaluated only if x is false)
x and y	Logical AND(y is evaluated only if x is true)
not x	Logical negation
x is y, x is not y	Object identity test
x<y, x<=y, x>y, x>=y	Magnitude comparison
x == y, x != y	Value equality operators
x | y	Bitwise OR
x ^ y	Bitwise XOR
x & y	Bitwise AND

Number Operations

Operations	Interpretations
x<<y , x>>y	Shift x left or right by y bits
x + y	Addition
x – y	Subtraction
x * y	Multiplication
x % y	Remainder
x / y	Division
-x , +x	Negation , identity
~x	Bitwise NOT
x ** y	Power

Hands-On

Strings

Creating strings is as simple as assigning a value to a variable.

Properties of Strings:

- Strings also have operations of their own, available as methods.

```
>>>s.find('p')
0
```

- Strings are immutable – they cannot be changed in place after they are created.

```
>>>S[0]='z'
TypeError: 'str' object does not support item assignment
```

Strings

Creating strings is as simple as assigning a value to a variable.

Properties of Strings:

- Strings are called sequences. They are iterable.

```
>>>S = 'python '   # store a string in a variable
>>>print S[0]      # indexing
P
```

- Strings are used to record textual information.

```
>>>'joe'          #single quotes are valid
>>>"joe"          #double quotes are also valid
>>>""""
python
Is easy to use """"      #When we use triple quotes,
                         strings can span several lines
                         without using the escape
                         character.
```

String operations

Operations	Interpretation
s.find('th')	Search
s.strip()	Remove whitespace
s.replace('th' , 'TH')	Replacement
s.split(' , ')	Split on delimiter
s.isdigit() , s.isaplha()	Content test
s.lower() and s.upper()	Case conversion
s.endswith('you')	End test
s.join(iterable)	Delimiter join

Hands-On