width = 14 height = 12

width // 2

width / 2.0

1. what the data types in python? Explain.

A. Numbers: Numbers data types store numeric values. Number objects are created when you assign a value to them.

Strings: Strings in python are identified as contiguous set of characters represented in the quotation marks. Python allows either pair of single or double quotes

Lists: Lists are the most versatile of python's compound data types. A list contains item separated by commas and enclosed within square brackets ([])

Tuples: A tuple is another sequence data type that is similar to the list. A tuple consists of a number of value separated by commas. Unlike lists however tuples are enclosed within parenthesis.

Dictionary: Python's dictonaries are kind of hash-table type. They work like associative arrays or hashes found in Perl and consist of key value pairs. A dictionary key can be almost any python type, but are usually numbers or strings. Values, on the other hand, can be any arbitary python object.
Dictionaries are enclosed within curly braces.

2. Briefly Explain history of python.

Python is an interpretend, high-level, general Purpose Programming language.

Python is Created by Guido Van Rossum and first released in 1991

Python design philosophy emphasizes codes readability with its notable use of significant white space.

Guido Van Rossum was reading the published scripts from "Monty python's Flying Circus", a BBC Comedy series from 1970s. So he decided to Call the language Python.

3. Explain all the operators in python.

Operators are used to Perform operations on Variables and Values.

Python has 7 types of operators.

- Arithmetic Operators are used with numeric values to perform common mathematical operations.

| Operator | Name | Example |
|---|---|---|
| + | Addition | x + y |
| - | Subtration | x - y |
| * | Multiplication | x * y |
| / | Division | x / y |
| % | Modulus | x % y |
| ** | Exponentiation | x ** y |
| // | Floor division | x // y |

Assignment operation: Assignment operators are used to assign values to variables:

| operator | Example | Same as |
|---|---|---|
| = | x = 5 | x = 5 |
| += | x += 3 | x = x + 3 |
| -= | x -= 3 | x = x - 3 |
| *= | x *= 3 | x = x * 3 |
| /= | x /= 3 | x = x / 3 |
| %= | x %= 3 | x = x % 3 |
| //= | x //= 3 | x = x // 3 |
| **= | x **= 3 | x = x ** 3 |
| \|= | x \|= 3 | x = x \| 3 |
| ^= | x ^= 3 | x = x ^ 3 |
| >>= | x >>= 3 | x = x >> 3 |
| <<= | x <<= 3 | x = x << 3 |

Comparison operation:- Comparision operators are used to compare two value

| Operator | Name | Example |
|---|---|---|
| == | Equal | x == y |
| != | Not Equal | x != y |
| > | Greater than | x > y |
| < | less than | x < y |
| >= | Greater than or Equal to | x >= y |
| <= | less than or Equal to | x <= y |

Logical operation :- Logical operators are used to combine conditional statements.

| Operator | Description | Example |
|---|---|---|
| and | Returns True if both statements are true | x < 5 and x < 10 |
| Or | Returns True if one of the statements is true | x < 5 or x < 4 |
| not | Reverse the result, returns False if the result is true | not(x < 5 and x < 10) |

Identity Operation : Identity operators are used to compare the objects, not if they are Equal, but if they are actually the same object, with the same memory location.

| Operator | Description | Example |
|---|---|---|
| is | Returns true if both var are the same object | x is y |
| is not | Returns true if both Variables are not the same object | x is not y |

Membership operators :- Are used to test if a sequence is presented in an object.

| Operator | Description | Example |
|---|---|---|
| in | Returns true if a sequence with the specified value is present in the object | x in y |
| not in | Returns true if a sequence with the specified value is not present in the object. | x not in y |

Bitwise operation :- Bitwise operators are used to compare (binary) numbers:

| Operator | Name | Description. |
|---|---|---|
| & | AND | sets each bit to 1 if both bits are 1 |
| \| | OR | Sets each bit to 1 if one of two bits is 1 |
| ^ | XOR | Sets each bit to 1 if only one of two bits is 1 |
| ~ | NOT | inverts all the bits |
| << | zero fill left shift | Shift left by pushing zeros in from the right and let the leftmost bits fall off |
| >> | signed right shift | Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off |

4) Explain the features of Python

Some programming language features of Python are : A variety of basic data types are available : numbers, Strings.

1) Easy to learn and use
2) Expressive language
3) Interpretend language
4) Crom platform language.
5) Free and open source
6) object - Oriented language
7) Extensible
8) Large standard library.

5) Justify why python is interactive interpreted language.

~~Python~~ Unlike C/C++ etc, Python is an interpreted object - oriented Programming language. Unlike C language, which is a compiled Programming language. The compiler translates the whole code in one go rather than line by line.

This is a prompt and interact with the interpreter directly to write your programs. Supports technique of Programming that Encapsulates code within objects.