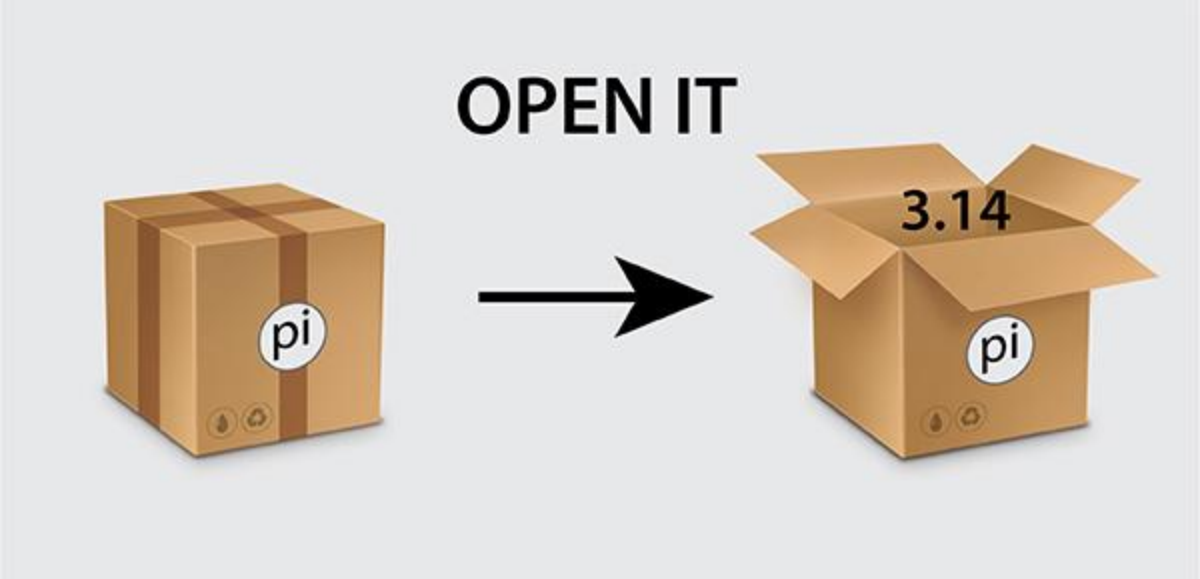
Python Identifiers

- A Python identifier is a name used to identify a variable, function, class, module or other object. An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).
- Python does not allow punctuation characters such as @, $, and % within identifiers.
- Python is a case sensitive programming language. Thus, **Manpower** and **manpower** are two different identifiers in Python.

Variable in Python

- A variable, as the name indicates is something whose value is changeable over time.
- In fact a variable is a memory location where a value can be stored. Later we can retrieve the value to use.
- But for doing it we need to give a nickname to that memory location so that we can refer to it. That's identifier, the nickname.

Variable in Python



Python Keywords

- Keywords are special words which are reserved and have a specific meaning. Python has a set of keywords that cannot be used as variables in programs.
- All keywords in Python are case sensitive.

```
help> keywords

Here is a list of the Python keywords.  Enter any keyword to get more help.

False               class               from                or
None                continue            global              pass
True                def                 if                  raise
and                 del                 import              return
as                  elif                in                  try
assert              else                is                  while
async               except              lambda              with
await               finally             nonlocal            yield
break               for                 not

help>
```

Assignment Statement

- The assignment statement gives a value to a variable:

```
>>> message = "What's up?"
>>> n = 17
>>> pi = 3.14159
```

- This example makes three assignments.
- The first assigns the string value "What's up?" to a variable named message.
- The second gives the integer 17 to n,
- and the third assigns the floating-point number 3.14159 to a variable called pi.

Assignment Statement

- The assignment statement binds a name, on the left-hand side of the operator to a value on the right-hand side.

Evaluating expressions

- An expression is a combination of values, variables, operators, and calls to functions. If you type an expression at the Python prompt, the interpreter evaluates it and displays the result:

```
>>> 1 + 1
2
>>> len("hello")
5
```

- In this example len is a built-in Python function that returns the number of characters in a string.

Operators and operands

- Operators are special tokens that represent computations like addition, multiplication and division. The values the operator uses are called operands.
- The following are all legal Python expressions

```
20+32   hour-1   hour*60+minute   minute/60   5**2   (5+9)*(15-7)
```

Operators and operands

- The tokens +, -, and *, and the use of parenthesis for grouping, mean in Python what they mean in mathematics.
- The asterisk (*) is the token for multiplication, and ** is the token for exponentiation.

Operators and operands

- In Python 3, the division operator / always yields a floating point result.
- Python gives us two different flavors of the division operator.
- The second, called floor division uses the token //.
- Its result is always a whole number — and if it has to adjust the number it always moves it to the left on the number line. So 6 // 4 yields 1