python

Variables, Keywords, Expressions, Statements

Program

- **Sequence of instructions that specify how to perform a computation.**
- **Basic instructions in almost every program:**
  - Input: Get data from keyboard, or file or some other device.
  - Output: Display data on the screen or send data to a file or other device.
  - Math: Perform basic mathematical operations like addition and multiplication.
  - Conditional Execution: Check for certain conditions and execute the appropriate sequence of statements.
  - Repetition: Perform some action repeatedly , usually with some variation

Debugging

- Programming errors are called bugs.
- Process of tracking bugs and correct them is called debugging.

Types of Errors

- Syntax Errors : invalid code the compiler doesn't understand

```
printf(%d,s)
```

- Run Time Errors (Exceptions)

```
a= 10/0;
```

- Semantic Errors : valid code the compiler understands

```
Dangling Else
```

Dangling Else

```
if (relative)
if (my friend)
        cout << “both”;
else
cout << “neither”;
```

```
if (relative)
 {
if (my friend)
cout << “both”;
 }
 else
 cout << “neither”;
```

Formal and Natural Languages

- **Natural Languages** are the languages that people speak, such as English, Spanish, French. They were not designed by people; they evolved naturally.
- **Formal Languages** are languages that are designed by people for specific applications.
- **Programming languages are formal languages that have been designed to express computations.**
- **Parsing** is a process to figure out what the structure of the sentence is.

Difference between Natural and Formal Languages

Ambiguity: Natural languages are full of ambiguity.

Formal languages are designed to be nearly or completely unambiguous.

Redundancy: Natural languages are more redundant as compared to formal languages.

Literalness: Natural languages are full of idiom and metaphor. If I say, “The other shoe fell," there is probably no shoe and nothing falling. Formal languages mean exactly what they say.

Variable Expressions and Statements

- Value is a one of the fundamental thing, that a program manipulates.
- Variable is a name that refers to a value that maybe changed in the program.
- The assignment statement creates new variables and gives them values:
- >>> message = "What's up, Doc?"
- >>> n = 17
- >>> pi = 3.1415

Variable

- Variable names must be meaningful.
- It contains both numbers and letters, but they have to begin with a letter.
- Case sensitive.
- The underscore (_) character can appear in a name.
- Eg.
  - >>>76trombones = "big parade“
  - SyntaxError: invalid syntax
  - >>> class = "Computer Science 101“
  - SyntaxError: invalid syntax

>>> more$ = 1000000

SyntaxError: invalid syntax

Keywords

- Keywords define the language's rules and structure and they can't be used as variable names.
- Python has twenty-nine keywords:

and	def	exec	if	not	return
assert	del	finally	import	or	try
break	elif	for	in	pass	while
class	else	from	is	print	yield
continue	except	global	lambda		raise

Python in its language defines an inbuilt module “**keyword**” which handles certain operations related to keywords. A function “**iskeyword()**” checks if a string is keyword or not. Returns **true if a string is keyword, else returns false**.

```
>>> import keyword
>>> keyword.iskeyword("true")
False
```

Statements

- A statement is an instruction that the Python interpreter can execute.
- When you type a statement on the command line, Python executes it and displays the result, if there is one.
- A script usually contains a sequence of statements. If there is more than one statement, the results appear one at a time as the statements execute.

Evaluating Expressions

- An expression is a combination of values, variables, and operators.
- If you type an expression on the command line, the interpreter evaluates it and displays the result:
  - >>> 1 + 1
  - 2
- A value all by itself is considered an expression, and so is a variable.

```
>>> 17
    17
```

```
>>> x
    2
```

Simultaneous Assignments

Python also supports simultaneous assignments like this:

var1,var2……,varn=exp1,exp2……,expn

```
>>> a,b,c=4,5,6
>>> a
4
>>> b
5
>>> c
6
```

Write code to swap two numbers.

```
>>>X=1
>>>Y=3
>>>Temp=X
>>>X=Y
>>>Y=Temp
```

Can be written as:

```
>>>X=1
>>>Y=3
>>>X,Y=Y,X
```

What is the maximum possible value of an integer in Python ?

In Python, value of an integer is not restricted by the number of bits and can expand to the limit of the available memory.

```
>>> x=1000000000000000000000000000000
>>> x
1000000000000000000000000000000
```

As a side note, in Python 3, there is only one type “int” for all type of integers. In Python 2.7. there are two separate types “int” (which is 32 bit) and “long int” that is same as “int” of Python 3.x, i.e., can store arbitrarily large numbers.

- x = 10
- print(type(x))
- x = 1000000000000000000000000000000000000000
- print(type(x))

Output in Python 2.7 :

<type 'int'>

<type 'long'>

Output in Python 3 :

<type 'int'>

<type 'int'>

Operators and Operands

- Operators are special symbols that represent computations like addition and multiplication.
- The values the operator uses are called operands
- When both of the operands are integers, the result must also be an integer, and by convention, integer division always rounds down.
- +, -, *, /, %, **,//
- ** is used for exponential.
- / is also used for float division
- // is used to integer division

```
>>> 2/3
0.6666666666666666
>>> 2.0/3
0.6666666666666666
>>> 2.0/3.0
0.6666666666666666
```

```
>>> 2//3
0
>>> 2.0//3
0.0
>>> 2.0//3.0
0.0
>>> 2.0//10
0.0
>>> 2.0//10.0
0.0
```

Order of Operations

- When more than one operator appears in an expression, the order of evaluation depends on the rules of precedence.
  - Parentheses have the highest precedence and can be used to force an expression to evaluate in the order you want.
  - Exponentiation has the next highest precedence.
  - Multiplication ,Float Division, Integer Division and remainder have the same precedence, which is higher than Addition and Subtraction, which also have the same precedence.
  - Operators with the same precedence are evaluated from left to right.

Operations on Strings

- Mathematical operations can’t be performed on strings, even if the strings look like numbers.
- the + operator represents concatenation.
- For Example:
  - fruit = "banana"
  - bakedGood = " nut bread“
  - print (fruit + bakedGood)
- The * operator also works on strings; it performs repetition.
- For eg:
  - "Fun"*3 is "FunFunFun"

Composition

- One of the most useful features of programming languages is their ability to take small building blocks and compose them

Comments

• It is a good idea to add notes to your programs to explain in natural language what the program is doing. These notes are called comments.

• they are marked with the # symbol:

•For eg:

```
# compute the percentage of the hour that has elapsed
percentage = (minute * 100) / 60
```

Questions ??

Programs

1 There are 5280 feet in a mile. Write a Python statement that calculates and prints the number of feet in 13 miles

2 Write a Python statement that calculates and prints the number of seconds in 7 hours, 21 minutes and 37 seconds.

3 The perimeter of a rectangle is 2w+2h, where w and h are the lengths of its sides. Write a Python statement that calculates and prints the length in inches of the perimeter of a rectangle with sides of length 4 and 7 inches.

Programs

The circumference of a circle is $2\pi r$ where r is the radius of the circle. Write a Python statement that calculates and prints the circumference in inches of a circle whose radius is 8 inches. Assume that the constant $\pi=3.14$.

Programs

- Given p dollars, the future value of this money when compounded yearly at a rate of r percent interest for y years is p(1+0.01r)y. Write a Python statement that calculates and prints the value of 1000 dollars compounded at 7 percent interest for 10 years.
- Write a Python expression that combines the string "Joe Warren is 52 years old." from the string "Joe Warren" and the number 52 and then prints the result (Hint: Use the function str to convert the number into a string.)

Programs

Write a single Python statement that combines the three strings "My name is", "Joe" and "Warren" (plus a couple of other small strings) into one larger string "My name is Joe Warren." and prints the result.

The distance between two points (x0,y0) and (x1,y1) is $(x0-x1)^2+(y0-y1)^2$. Write a Python statement that calculates and prints the distance between the points (2,2) and (5,6).