Module 3 : CSS 103 Variables and Assignment Statements, Script Mode, Functions , Modules

Notes

Python Variables

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

Rules for naming variables in python

- The name of the variable should start with either an alphabet or an underscore but it cannot start with a digit.
- More than one alpha-numeric characters or underscores may follow.
- The variable name can consist of alphabet letter(s), number(s) and underscore(s) only.
- Identifiers in Python are case sensitive.

Python Assignment Operators

Operator	Description
=	Assigns values from right side operands to left side operand
+= Add AND	It adds right operand to the left operand and assign the result to left operand
-= Subtract AND	It subtracts right operand from the left operand and assign the result to left operand
*= Multiply AND	It multiplies right operand with the left operand and assign the result to left operand
/= Divide AND	It divides left operand with the right operand and assign the result to left operand

%= Modulus AND	It takes modulus using two operands and assign the result to left operand
**= Exponent AND	Performs exponential (power) calculation on operators and assign value to the left operand
//= Floor Division	It performs floor division on operators and assign value to the left operand

Python Script Mode

- In order to write a long Python code that spans multiple files, interactive mode is not recommended instead the script mode is appropriate.
- In script mode, You write your code in a text file then save it with a .py extension which stands for "Python".
- Then by specifying commands we can run the script at the shell.

Advantages of Script Mode:

- It is easy to run large pieces of code.
- Editing your script is easier in script mode.
- Good for both beginners and experts.

Limitations of script mode:

- Can be tedious when you need to run only a single or a few lines of cod.
- You must create and save a file before executing your code.

Python - Functions

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing.

As you already know, Python gives you many built-in functions like print(), etc. but you can also create your own functions. These functions are called *user-defined functions.*

Defining a Function

You can define functions to provide the required functionality. Here are simple rules to define a function in Python.

- Function blocks begin with the keyword def followed by the function name and parentheses (( )).
- Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.
- The first statement of a function can be an optional statement - the documentation string of the function or docstring.
- The code block within every function starts with a colon (:) and is indented.
- The statement return [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

Syntax

```
def functionname( parameters ):
   "function_docstring"
   function_suite
   return [expression]
```

By default, parameters have a positional behavior and you need to inform them in the same order that they were defined.

Function Arguments

You can call a function by using the following types of formal arguments −

- Required arguments
  Required arguments are the arguments passed to a function in correct positional order.
  Here, the number of arguments in the function call should match exactly with the function definition.
- Keyword arguments
  Keyword arguments are related to the function calls. When you use keyword arguments in a function call, the caller identifies the arguments by the parameter name.
- Default arguments
  A default argument is an argument that assumes a default value if a value is not provided in the function call for that argument.
- Variable-length arguments
  You may need to process a function for more arguments than you specified while defining the function. These arguments are called *variable-length* arguments and are not named in the function definition, unlike required and default arguments.

The return Statement

- The statement return [expression] exits a function, optionally passing back an expression to the caller.
- A return statement with no arguments is the same as return None.

The Anonymous Functions

- These functions are called anonymous because they are not declared in the standard manner by using the *def* keyword.
- You can use the *lambda* keyword to create small anonymous functions.

Python Modules

A module allows you to logically organize your Python code. Grouping related code into a module makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you can bind and reference.

Simply, a module is a file consisting of Python code. A module can define functions, classes and variables. A module can also include runnable code.

- A module allows you to logically organize your Python code.
- Grouping related code into a module makes the code easier to understand and use.
- A module is a Python object with arbitrarily named attributes that you can bind and reference.

The import Statement

- You can use any Python source file as a module by executing an import statement in some other Python source file.

Locating Modules

When you import a module, the Python interpreter searches for the module in the following sequences −

1. The current directory.
2. If the module isn't found, Python then searches each directory in the shell variable PYTHONPATH.
3. If all else fails, Python checks the default path. On UNIX, this default path is normally /usr/local/lib/python/.