DevOps Crash Course

by Oleksii Yakivchik

softserve

Python

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. It was created by Guido van Rossum, and released in 1991.

Python

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

Python features

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

Installation in Centos

```
~$ sudo yum install python3
~$ python3
```

softserve

Variables

- Variables are containers for storing data values.
- Python has no command for declaring a variable.
- A variable is created the moment you first assign a value to it.
- Variable names are case-sensitive.
- String variables can be declared either by using single or double quotes.
- Variables do not need to be declared with any particular type, and can even change type after they have been set.

Variables

```
x = 5
y = "John"
Y = ‘Dave’
print(x)
print(y)
print(Y)
```

softserve

Variable casting

- If you want to specify the data type of a variable, this can be done with casting.

```
x = str(3)     # x will be '3'
y = int(3)     # y will be 3
z = float(3)   # z will be 3.0
```

Built-in Data Types

Text Type: str
Numeric Types: int, float, complex
Sequence Types: list, tuple, range
Mapping Type: dict
Set Types: set, frozenset
Boolean Type: bool
Binary Types: bytes, bytearray, memoryview

You can get the data type of any object by using the type() function

Strings

Strings in python are surrounded by either single quotation marks, or double quotation marks.

'hello' is the same as "hello".

You can display a string literal with the print() function:

```
print("Hello")
```

```
print('Hello')
```

Strings

Assigning a string to a variable is done with the variable name followed by an equal sign and the string:

```
a = "Hello"
print(a)
```

Strings

Like many other popular programming languages, strings in Python are arrays of bytes representing unicode characters.

However, Python does not have a character data type, a single character is simply a string with a length of 1.

Square brackets can be used to access elements of the string.

```
a = "Hello, World!"
print(a[1])
```

Strings

To get the length of a string, use the len() function.

```
a = "Hello, World!"
print(len(a))
```

Strings

To check if a certain phrase or character is present in a string, we can use the keyword in.

```
txt = "The best things in life are free!"

print("free" in txt)
```

Strings

To check if a certain phrase or character is NOT present in a string, we can use the keyword not in

```
txt = "The best things in life are free!"

print("expensive" not in txt)
```

Strings

To concatenate, or combine, two strings you can use the + operator.

```
a = "Hello"

b = "World"

c = a + b

print(c)
```

Python

Python supports the usual logical conditions from mathematics:

- Equals: a == b
- Not Equals: a != b
- Less than: a < b
- Less than or equal to: a <= b
- Greater than: a > b
- Greater than or equal to: a >= b

Python

- If statement:

```
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

softserve

Python

- The elif keyword is pythons way of saying "if the previous conditions were not true, then try this condition".

```
a = 33
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```

Python Conditions and If statements

- The else keyword catches anything which isn't caught by the preceding conditions.

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```

Python Conditions and If statements

- The and keyword is a logical operator, and is used to combine conditional statements:

```
a = 200
b = 33
c = 500
if a > b and c > a:
  print("Both conditions are True")
```

Python Conditions and If statements

- The or keyword is a logical operator, and is used to combine conditional statements:

```
a = 200
b = 33
c = 500
if a > b or a > c:
  print("At least one of the conditions is True")
```

softserve

Python for statetement

A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

This is less like the for keyword in other programming languages, and works more like an iterator method as found in other object-orientated programming languages.

```
for x in "banana":
  print(x)
```

Further reading

https://www.w3schools.com/python/default.asp

softserve