Python Arrays

Python does not have built-in support for Arrays, but Python Lists can be used instead.

Arrays are used to store multiple values in one single variable:

Example

Create an array containing car names:

```
cars = ["Ford", "Volvo", "BMW"]
```

Access the Elements of an Array

You refer to an array element by referring to the *index number*.

Example

Get the value of the first array item:

```
cars = ["Ford", "Volvo", "BMW"]

x = cars[0]

print(x)
```

```
Ford
```

Example

Modify the value of the first array item:

```
cars = ["Ford", "Volvo", "BMW"]

cars[0] = "Toyota"

print(cars)
```

```
['Toyota', 'Volvo', 'BMW']
```

The Length of an Array

Use the `len()` method to return the length of an array (the number of elements in an array).

Example

Return the number of elements in the `cars` array:

```
cars = ["Ford", "Volvo", "BMW"]

x = len(cars)

print(x)
```

```
3
```

Looping Array Elements

You can use the `for in` loop to loop through all the elements of an array.

Example

Print each item in the `cars` array:

```
cars = ["Ford", "Volvo", "BMW"]

for x in cars:
  print(x)
```

```
Ford
Volvo
BMW
```

Adding Array Elements

You can use the `append()` method to add an element to an array.

Example

Add one more element to the `cars` array:

```
cars = ["Ford", "Volvo", "BMW"]

cars.append("Honda")

print(cars)
```

```
['Ford', 'Volvo', 'BMW', 'Honda']
```

Removing Array Elements

You can use the `pop()` method to remove an element from the array.

Example

Delete the second element of the `cars` array:

```
cars = ["Ford", "Volvo", "BMW"]

cars.pop(1)

print(cars)
```

```
['Ford', 'BMW']
```

You can also use the `remove()` method to remove an element from the array.

Example

Delete the element that has the value "Volvo":

```
cars = ["Ford", "Volvo", "BMW"]

cars.remove("Volvo")

print(cars)
```

```
['Ford', 'BMW']
```

Array Methods

Python has a set of built-in methods that you can use on lists/arrays.

Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list

copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the first item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

While True Statement

In Python, the `True` keyword is a boolean expression. It's used as an alias for `1`, and the `while` keyword is used to specify a loop. The statement `while True` is used to specify an infinite `while` loop.

An infinite loop runs indefinitely until the end of time or when the program is forcefully stopped. The following code example below shows us how we can create an infinite loop with the `while True` statement.

```
while True:

    print("Hello World")
```

Output:

```
Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World
```

We created an infinite `while` loop that prints `Hello World` every time it is executed by using the `while True` statement in the code above. This approach is not recommended because it stops the code from its completion.

One workaround is the use of the `break` statement inside the infinite loop to stop the process when a particular condition is satisfied. This approach is demonstrated in the following program below.

```
i = 0

while True:

    print("Hello World")

    i+=1

    if i == 10:

        break
```

Output:

```
Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World
```

We stopped the infinite `while` loop by using the `break` statement in the code above. The execution of the infinite loop was stopped after the value of the integer variable `i` becomes equal to `10`.