

Arrays in Python

What is an **Array**?

An array is a special variable, which can hold more than one value at a time.

If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:

```
car1 = "Ford"  
car2 = "Volvo"  
car3 = "BMW"
```

However, what if you want to loop through the cars and find a specific one? And what if you had not 3 cars, but 300?

The solution is an array!

An array can hold many values under a single name, and you can access the values by referring to an index number.

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

Access the Elements of an Array

Get the value of the first array item:

```
cars = ["Ford", "Volvo", "BMW"]  
  
x = cars[0]  
  
print(x)
```

The Length of an Array

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

Return the number of elements in the cars array:

```
cars = ["Ford", "Volvo", "BMW"]  
  
x = len(cars)  
  
print(x)
```

Looping Array Elements

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

Print each item in the cars array:

```
cars = ["Ford", "Volvo", "BMW"]  
  
for x in cars:  
    print(x)
```

Adding Array Elements

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

```
cars = ["Ford", "Volvo", "BMW"]  
  
cars.append("Honda")  
  
print(cars)
```

Removing Array Elements

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

Delete the second element of the cars array:

```
cars = ["Ford", "Volvo", "BMW"]  
  
cars.pop(1)  
  
print(cars)
```


Removing Array Elements

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

Delete the element that has the value "Volvo":

```
cars = ["Ford", "Volvo", "BMW"]  
  
cars.remove("Volvo")  
  
print(cars)
```

Note: `remove()` method only removes the first occurrence of the specified value.

Array methods:

Method	Description
<u>append()</u>	Adds an element at the end of the list
<u>clear()</u>	Removes all the elements from the list
<u>copy()</u>	Returns a copy of the list
<u>pop()</u>	Removes the element at the specified position
<u>remove()</u>	Removes the first item with the specified value
<u>reverse()</u>	Reverses the order of the list
<u>sort()</u>	Sorts the list

Built-in Math Functions

The `min()` and `max()` functions can be used to find the lowest or highest value in an iterable:

```
x = min(5, 10, 25)
```

```
y = max(5, 10, 25)
```

```
print(x)
```

```
print(y)
```

Built-in Math Functions

The `abs()` function returns the absolute (positive) value of the specified number:

```
x = abs(-7.25)
```

```
print(x)
```

Built-in Math Functions

The `pow(x, y)` function returns the value of `x` to the power of `y`.

```
x = pow(4, 3)
```

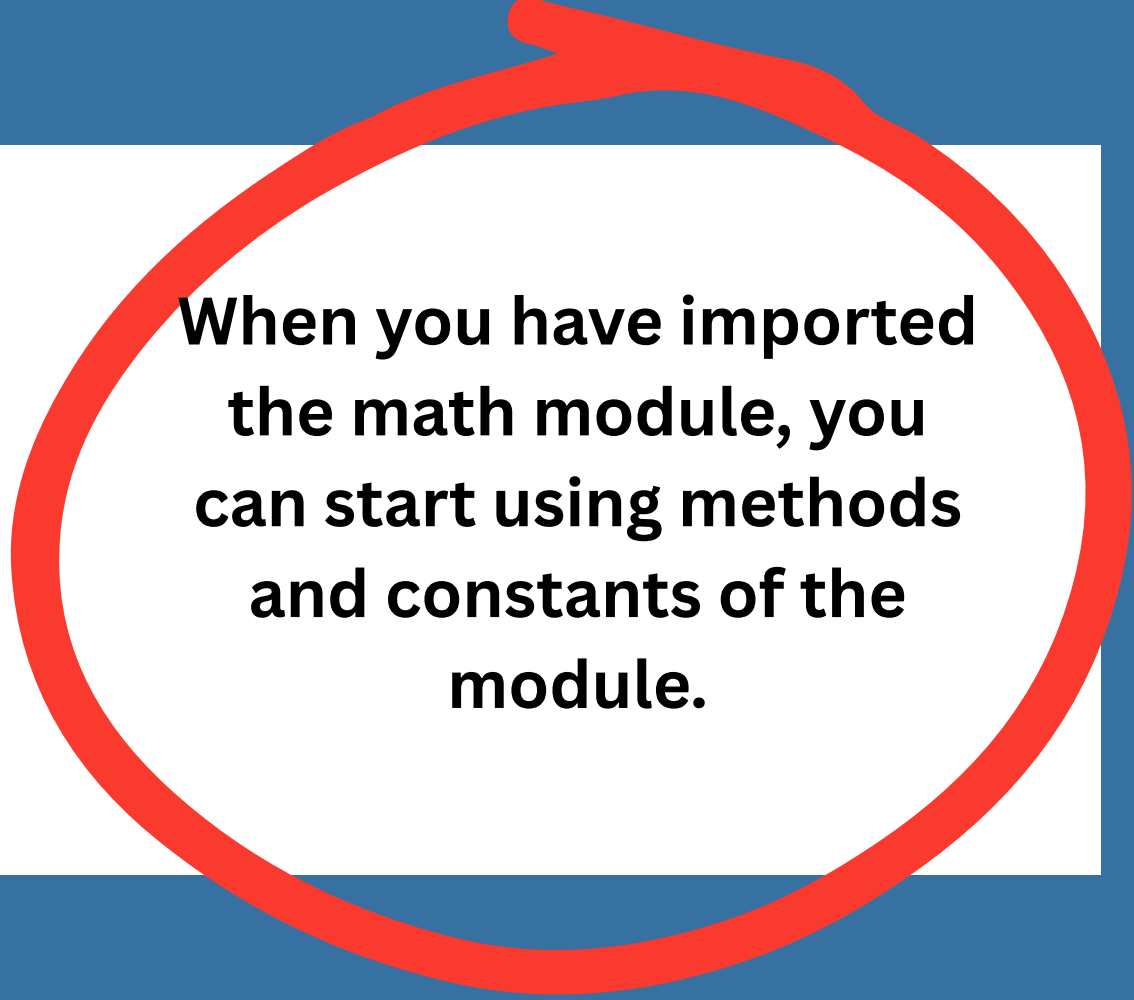
```
print(x)
```

The Math Module

Python has also a built-in module called math, which extends the list of mathematical functions.

To use it, you must import the math module:

```
import math
```



When you have imported the math module, you can start using methods and constants of the module.

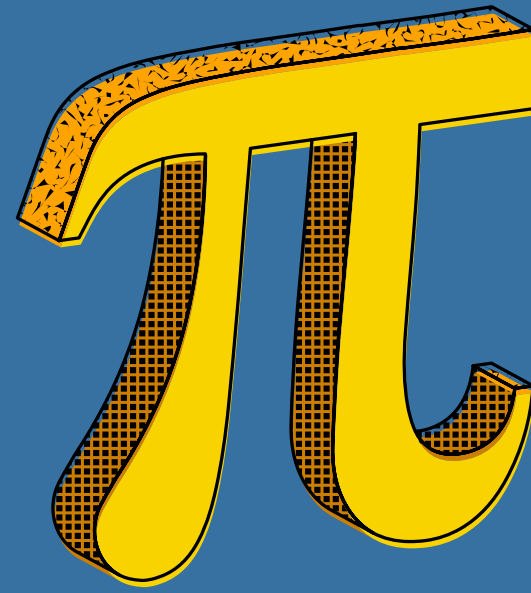
The `math.sqrt()` method for example, returns the square root of a number:

very important to add

```
import math
```

```
x = math.sqrt(64)
```

```
print(x)
```

The `math.pi` constant, returns the value of PI
(3.14...):

```
import math
```

```
x = math.pi
```

```
print(x)
```

File Handling

The key function for working with files in Python is the `open()` function.

The `open()` function takes two parameters; **filename**, and **mode**.

There are four different methods (modes) for opening a file:

"r" - Read - Default value. Opens a file for reading, error if the file does not exist

"a" - Append - Opens a file for appending, creates the file if it does not exist

"w" - Write - Opens a file for writing, creates the file if it does not exist

"x" - Create - Creates the specified file, returns an error if the file exists

File Handling

In addition you can specify if the file should be handled as binary or text mode

"t" - Text - Default value. Text mode

"b" - Binary - Binary mode (e.g. images)

To open the file, use the built-in **open()** function.

The **open()** function returns a file object, which has a **read()** method for reading the content of the file:

Example:

```
f = open("demofile.txt")  
print(f.read())
```

This code opens file in read mode
and then prints it

