

# Callback function

A callback is a function passed as an argument to another function

This technique allows a function to call another function

A callback function can run after another function has finished

```
// main function
function greet(x)
{
    console.log("hi")
    x()
}

//callback function
function varshith()
{
    console.log("Iam varshith")
}

// passing callback function as a argument
greet(varshith)
```

parameters : which are in the function declaration

ex

```
function varshith(x,y)
{
    return x*y;
}
```

here x,y are parameters

Arguments

which are used in function calling  
varshith(10,5)

10,5 are arguments

```
node /tmp/I0B7DYuwgB.js
hi
Iam varshith

=== Session Ended. Please Run the code again ===
```

# Benefits of callback functions

## Example: Program with setTimeout()

```
// program that shows the delay in execution

function greet() {
  console.log('Hello world');
}

function sayName(name) {
  console.log('Hello' + ' ' + name);
}

// calling the function
setTimeout(greet, 2000);
sayName('John');
```

Run Code >>

## Output

```
Hello John
Hello world
```

## By using callback function

```
// Callback Function Example
function greet(name, myFunction) {
  console.log('Hello world');

  // callback function
  // executed only after the greet() is executed
  myFunction(name);
}

// callback function
function sayName(name) {
  console.log('Hello' + ' ' + name);
}

// calling the function after 2 seconds
setTimeout(greet, 2000, 'John', sayName);
```

## Output

```
Hello world
Hello John
```