

## Arrow Functions

**named function** - a **function with a name** that can **invoked from anywhere in the code** accept parameters and return a value.

**named function** are used when you want to call the function multiple times from different place in your code.

Named function that takes 2 parameters and returns their sum:

```
function sum(n1 : number, n2 : number) {  
    return n1 + n2;  
}
```

Named function that takes no parameters and returns a 0:

```
function giveMeAZero() {  
    return 0;  
}
```

**Arrow Function** - a **function without a name** that can **only be invoked where it is defined** it can accept parameters and return a value (aka lambda expressions or anonymous methods/functions)

Arrow Functions are used where a function is needed in only one place in the code. (Usually as parameters to other methods)

**instead of *function* name** you code **=>** after the parameters and before the **{**

Arrow function that takes 2 parameters and returns their sum:

```
(n1 : number, n2 : number) => {  
    return n1 + n2;  
}  
  
(n1, n2) => { return n1 + n2; } // single-line processes may be code on one line  
  
(n1, n2) => return n1 + n2; // TypeScript does not {} if a single line process
```

Arrow function that takes no parameters and returns a 0:

```
() => {  
    return 0;  
}  
  
() => { return 0; }
```

Arrow functions are frequently used as arguments for other methods.

Some methods require function be passed to them that they then execute as part of their processing. The functions that are passed are referred to a **callback functions**.

When a function is used as a parameter to another function, we refer to the parameter function as a **callback function**.

The method receiving the **callback function**, calls the callback function as part of its processing.

Callback methods may either be named or Arrow Functions.