

Node JS Function

HCL (Higher Coding Language)

Function

Function is collections of instructions which is use for reusability.

Advantages of Function

- Code reusability
- Easy maintenance
- reduce complexity
- categorization of program
- remove boilerplate code
- fast execution

Simple Function

Simple function use when input is same.

index.js

```
function add()  
{  
  var a=10;  
  var b=20;  
  var addition=a+b;  
  console.log("Addition="+addition);  
}  
add();
```

We can also write function by anonymous (function without name) function

first.html

```
var add=function()  
{  
  var a=10;  
  var b=20;  
  var addition=a+b;  
  console.log("Addition="+addition);  
}  
add();
```

We can also write function using arrow function

HCL (Higher Coding Language)

first.html

```
var add={()=>
{
  var a=10;
  var b=20;
  var addition=a+b;
  console.log("Addition="+addition);
}
add();
```

Output

```
PS C:\Users\lenovo\Desktop\node js> node index.js
Addition=30
PS C:\Users\lenovo\Desktop\node js> █
```

Limitation of simple function is that we get same result on every input.

Parameterized or argument Function

Parameterized function use when we want to get different output on different input.

index.js

```
function add(a,b)
{
  var addition=a+b;
  console.log("Addition="+addition);
}
add(10,20)
add(30,20)
```

Output

HCL (Higher Coding Language)

```
PS C:\Users\lenovo\Desktop\node js> node index.js
Addition=30
Addition=50
PS C:\Users\lenovo\Desktop\node js> █
```

Function as expression

first.html

```
var result=function add(a,b)
{
    var addition=a+b;
    console.log("Addition="+addition);
}
```

```
result(10,20)
```

Output

```
PS C:\Users\lenovo\Desktop\node js> node index.js
Addition=30
PS C:\Users\lenovo\Desktop\node js> █
```

Anonymous Function

first.html

```
var result=function(a,b)
{
    var addition=a+b;
    console.log("Addition="+addition);
}
```

```
result(10,20)
```

Output

```
PS C:\Users\lenovo\Desktop\node js> node index.js
Addition=30
PS C:\Users\lenovo\Desktop\node js> █
```

HCL (Higher Coding Language)

Function with return type

Return type function use when we want to get inside function data outside of the function.

index.js

```
function add(a,b)
{
    var addition=a+b;
    return addition;
}
console.log("Addition="+add(10,20));
```

But there a limitation of return, that we return only one value at a time. But if we want to get multiple data we have to assign local value globally.

first.html

```
var no1,no2;
function add(a,b)
{
    no1=a;
    no2=b;
}
add(10,20)
var sum=no1+no2;
console.log("Addition="+sum);
```

Output

```
PS C:\Users\lenovo\Desktop\node js> node index.js
Addition=30
PS C:\Users\lenovo\Desktop\node js> □
```

HCL (Higher Coding Language)

Global and Local Variable

There are two types of variable exist in JS Global and local variable. Global variable scope is within the program but local variable scope within a function.

index.js

```
var a=10; //global variable
function add(b,c) // local variable
{
    var d=40; // local variable
    console.log(a)
    console.log(b)
    console.log(c)
    console.log(d)
}
add(20,30)
console.log(a)
```

Output

```
PS C:\Users\lenovo\Desktop\node js> node index.js
10
20
30
40
10
PS C:\Users\lenovo\Desktop\node js> █
```

Function Calling Function

index.js

```
function add(a,b)
{
    var addition=a+b;
    console.log("Addition="+addition);
}
```

HCL (Higher Coding Language)

```
sq(addition);
}
function sq(addition)
{
    var square=addition*addition;
    console.log("Square="+square);
}
add(10,20)
```

Output

```
PS C:\Users\lenovo\Desktop\node js> node index.js
Addition=30
Square=900
PS C:\Users\lenovo\Desktop\node js> █
```

Single Tasking Function

In ideal condition should be perform single task by a single function to increase the performance of the program.

index.js

```
function add(a,b)
{
    var addition=a+b;
    console.log("Addition="+addition);
}
add(10,20)
```

Above the program performing 3 task at a time

- 1 getting value
- 2 execute the input
- 3 showing result

So same program we can write by single tasking function.

HCL (Higher Coding Language)

index.js

```
var no1,no2,sum;
function get(a,b)
{
    no1=a;
    no2=b;
}
function add()
{
    sum=no1+no2;
}
function show()
{
    console.log("addition="+sum);
}
get(10,20);
add();
show();
```

Output

```
PS C:\Users\lenovo\Desktop\node js> node index.js
Addition=30
```

Recursion Function

If a function called itself is called recursion function.

index.js

```
var count=0;
function hello()
{
    count++;
    console.log("Hello="+count);
    hello();
}
```


HCL (Higher Coding Language)

```
}  
hello();
```

Output

```
Hello=10307  
Hello=10308  
Hello=10309  
Hello=10310  
Hello=10311  
Hello=10312  
Hello=10313  
Hello=10314  
Hello=10315  
Hello=10316  
Hello=10317  
Hello=10318  
Hello=10319
```

Immediately invoked function expression (IIFE)

first.html

A function call automatically is called immediate invoked function expression.

```
(function add(a,b)  
{  
    var addition=a+b;  
    console.log("Addition="+addition+"<br>");  
})(10,20)
```

Output

```
PS C:\Users\lenovo\Desktop\node js> node index.js  
Addition=30
```

HCL (Higher Coding Language)

Factorial index.js

```
function fact(no)
{
    if(no==1)
    {
        return 1;
    }
    else{
        return no*fact(no-1);
    }
}
console.log("Factorial="+fact(5));
```



```
PS C:\Users\lenovo\Desktop\node js> node index.js
Factorial=120
PS C:\Users\lenovo\Desktop\node js> █
```

Arrow Function

Arrow function expressions is a new syntax to writing ordinary function expressions.

first.html

```
// wirte ordanry function
function add(no1,no2)
{
    var sum=no1+no2;
```

HCL (Higher Coding Language)

```
    return sum;
}
console.log("Sum="+add(10,20))
// wirte arrow function
var data=(no1,no2) => {
    var sum=no1+no2;
    return sum;
}
console.log("Sum="+data(10,20))
// wirte arrow function with one line
var data=(no1,no2)=>no1+no2;
console.log("Sum="+data(10,20))
```

Output

```
PS C:\Users\lenovo\Desktop\node js> node index.js
Sum=30
Sum=30
Sum=30
PS C:\Users\lenovo\Desktop\node js> █
```

Higher Coding Language

Live Project Training With 100% Placement Assistance

S no	Internship	Content	Duration
1	C,C++ with web development	C,C++,html, CSS ,JS & Project	2 Months
2	Web Development	html, CSS ,JS,Jquery,Bootstrap & Project	2 Months


HCL (Higher Coding Language)

3	C with DSA	C and DSA	2 Months
4	C++ with DSA	C++ and DSA	2 Months
5	Java with DSA	Java and DSA	2 Months
6	Java With Mysql	Core java and Mysql	2 Months
7	Front end	html, CSS ,JS, React JS & Project	2 Months
8	Back end	Node JS & Project	2 Months
9	Core java with Web Development	html, CSS ,JS,Jquery,Bootstrap,Core Java & Project	3 Months
10	Python with Web Development	html, CSS ,JS,Jquery,Bootstrap,Python & Project	3 Months
11	MERN Full Stack	html, CSS ,JS,Jquery,Bootstrap,Node JS & Live Project	6 Months
12	Java Full Stack	html, CSS ,JS,Jquery,Bootstrap,Core Java,JDBC,JSP,Servlet & Live Project	6 Months
13	Python Full Stack	html, CSS ,JS,Jquery,Bootstrap,Core Python,Advanced Python,Django & Live Project	6 Months
14	Android	Core Java and Android	4 Months
15	Fluter	Dart and Flutter	4 Months
16	Collage Minor Project	In any Technology	NA
17	Collage Major Project	In any Technology	NA
18	Python With ML	Core Python, Advanced Python,Numpy,Pandas, Mitlab, Seaborn & ML Algortithms	6 months
19	Python With AI	Core Python, Advanced Python,Numpy,Pandas, Mitlab, Seaborn & AI Algortithms	6 months
20	Python Data science	Core Python, Advanced Python,Numpy,Pandas, Mitlab, Seaborn & Data Science Algortithms	6 months
20	Python Data Analytics	Core Python, Advanced Python,Numpy,Pandas, Mitlab, Seaborn ,mysql and power BI	6 months

Our Recent Placements

Name	Photo	Company	PKG
------	-------	---------	-----

HCL (Higher Coding Language)

Khusbu Dubey		TCS	3.59 LPA
Prashant Shukla		Infosys	3.50 LPA
Tanuja Patidar		BestPeers	2.0 LPA
Sneha Gupta		Cyber Intant	NA
Rohan Sisodiya		GeeCom India	1.2 LPA
Kanchan pandey		SheThink	1.8 LPA
Sachin Choudhary		Nokia	4.0 LPA

HCL (Higher Coding Language)

Shaikhar parmar		Capgemini	3.80 LPA
Lakhan Patel		TCS	5.00 LPA
Kundan Mandloi		Maveric Systems	2.88 LPA
Yoegsh		Deloitte	3.20 LPA
Dheeraj Patel		Cognizant	4.50
Rahul		Digiprima	1.9 LPA
Jayendra		Assistant	2.4 LPA

HCL (Higher Coding Language)

Sachin		Codernaline	1.60 LPA
Sanchit		Avery Bit	1.80 LPA
Sumit Chandravanshi		Geecom India	1.20 LPA
Ajay Sailani		Avery Bit	1.20 LPA

Other Facility

- 100% Placement Assistance
- Live Coding
- Highly Experience Industrial Trainer
- Work on Live Project
- Free Printed Notes
- 2 Days Free Demo
- Mock Interview
- Certificate of Internship

HCL (Higher Coding Language)

- 1000 Rs Referral Amount
- Corporate Environment
- Lab Facility Available
- May Be Contact for Placement
- Offline Training also available

Contact No – 82368 09542, 75662 99542

Add - 109,208 Prem Plaza, Ashok Nagar,Bhwarkua, Indore – 452001 (M.P.)

