



ANECO ACADEMY

JS-FUNCTION- OBJECT-ARRAY

Function in JS

- A function is a block of code that performs a specific task.

```
//defining a function
function <function-name>()
{
    // code to be executed
};
```

```
//calling a function
<function-name>();
```

```
//defining a function
function ShowMessage() {
    alert("Hello World!");
}
```

```
//calling a function
ShowMessage();
```

The diagram illustrates the syntax of a JavaScript function with the following components and annotations:

- function keyword**: Points to the word `function`.
- function name**: Points to the identifier `fname`.
- function parameter or input. Can be multiple.**: Points to the parameters `(param1,param2...)`.
- function body with the main logic or code statements.**: Points to the block of code between the curly braces `{` and `}`, specifically to `statement 1;`, `statement 2;`, and `statement 3;`.
- giving output using the return keyword**: Points to the `return` keyword in the line `return output;`.

```
function fname(param1,param2...)
{
    statement 1;
    statement 2;
    statement 3;

    return output;
}
```

Function in JS

Function with parameters:

```
//defining a function
function ShowMessage(firstName, lastName) {
    alert("Hello " + firstName + " " + lastName);
}

//calling a function
ShowMessage("Ashok", "Kumar");
```

Function with Return Value:

```
//defining a function
function Sum(val1, val2) {
    return val1 + val2;
};

//calling a function
var result = Sum(10,20); // returns 30
```

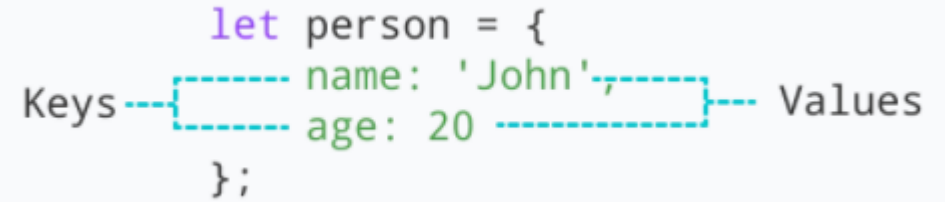
Object

JavaScript object is a non-primitive data-type that allows you to store multiple collections of data

Syntax:

```
const object_name = {  
  key1: value1,  
  key2: value2  
}
```

```
// object creation  
let person = {  
  name: 'John',  
  age: 20  
};  
console.log(typeof person); // object
```



The diagram shows the code `let person = { name: 'John', age: 20 };` with annotations. A dashed blue bracket on the left groups the property pairs, with the word "Keys" to its left. Another dashed blue bracket on the right groups the same property pairs, with the word "Values" to its right. The keywords `let` and `person` are purple, `=` is black, and the property pairs are green.

```
let person = {  
  name: 'John',  
  age: 20  
};
```

"key: value" pairs are called **properties**

Object

Accessing Object Properties:

Syntax:

objectName.key

Or

objectName["propertyName"]

```
const person = {  
  name: 'John',  
  age: 20,  
};
```

```
// accessing property  
console.log(person.name); // John
```

```
const person = {  
  name: 'John',  
  age: 20,  
};
```

```
// accessing property  
console.log(person["name"]); // John
```

Array

An array is an object that can store multiple values, create an array is by using an array literal []

Create Array:

Example:

```
// empty array
const myList = [ ];

// array of numbers
const numberArray = [ 2, 4, 6, 8];

// array of strings
const stringArray = [ 'eat', 'work', 'sleep'];

// array with mixed data types
const newData = ['work', 'exercise', 1, true];
```

Access Elements of an Array:

Example:

```
const myArray = ['h', 'e', 'l', 'l', 'o'];

// first element
console.log(myArray[0]); // "h"

// second element
console.log(myArray[1]); // "e"
```

