

Q. What is an Anonymous function?
Ans A function without a name is called anonymous function.

Q. Difference b/w function statement & function expression?

function statement :-

```
function a(){
  console.log("a is called");
}
```

Function Expression

```
var b = function(){
  console.log("b called");
}
```

↳ Diff. b/w fⁿ expression & fⁿ statement is hoisting.
 i.e. in function statement we can call the function before initializing it but in function expression we can not call the function before initializing because ~~it will store~~ b is treated by any other variable. It is assigned 'undefined' initially.

Function Statement (or Function Declaration):

Syntax:

```
javascript Copy code  
  
function myFunction() {  
  // code here  
}
```

Characteristics:

- Defined with the function keyword followed by the function name.
- Can be called before it's defined in the code (due to hoisting).
- Creates a named function that can be used throughout its scope.

Function Expression:

Syntax:

```
javascript Copy code  
  
const myFunction = function() {  
  // code here  
};
```

- **Characteristics:**
 - Defined as part of an expression and assigned to a variable.
 - Cannot be called before it's defined (no hoisting).
 - Creates an anonymous function (or can be named if needed) that's tied to the variable.

Key Differences:

1. Hoisting:

- Function statements are hoisted, meaning they can be used before they appear in the code.
- Function expressions are not hoisted, so they must be defined before they are called.

2. Usage Context:

- Function statements are used for defining functions that are meant to be available throughout their scope.
- Function expressions are often used for creating functions that are assigned to variables or passed as arguments.

In summary, function statements are like setting up a recipe that's available all the time, while function expressions are like writing down a recipe on a sticky note and only using it once you've put the note in place.

Overview of Topics Covered in Below pages :

Anonymous Function: A function without a name. Used when a name isn't needed.

Named Function: A function with a name. Easier to use and debug.

Parameters: Names in the function definition (e.g., a, b in function add(a, b)).

Arguments: Actual values passed to the function (e.g., 3, 4 in add(3, 4)).



Date ____/____/____

Page ____

function statement aka function declaration.

Anonymous fⁿ :-

function without name is called anonymous fⁿ.

It doesn't have its own identity.

Anonymous fⁿ are used in a place where the function are used as values.

```
function () {  
    
}
```

→ It will give error

Syntax Error: Statement require a function name.

```
var b = function () {  
  console.log("b is called");  
}
```



* Named function Expression

It is same like function expression but instead of using anonymous fⁿ we using fⁿ with a name.

eg.

```
var b = function xyz() {  
    console.log("b is called");  
}
```

↳ Corner Case :-

we can't call function as :-

xyz(); → It will give error.

↓
Reference error : xyz is not defined.

→ We can ~~call~~ ^{call} it inside the function i.e. -

```
var b = function xyz() {  
    console.log(xyz);  
}
```




* Difference b/w parameters & Arguments?

→ The values which we pass inside ^a function are ~~is~~ known as arguments.

→ These label & identifier which get those ~~labels~~ values are parameters.

* What are first class functions?

→ The ability to use functions as values, ~~is known~~ and can be passed as arguments to another function & can be ~~return~~ returned from the function. is called first class function.

E.g.

```
var b = function(param1) {  
    return function() {  
        }  
    }  
    console.log(b());
```

we can also ^{return} ~~pass~~ named function

→ returning a function

→ o/p: -

```
f() {  
    }  
}
```