Functions

Creation and hoisting:

Ex1: var a = 'Hellow Wolrd';

function b(){----------->step2

console.log('callde b');----------->step3

}

b();----------->step1

console.log(a);----------->step4

Output: called b

Hellow World

Note: First it will call b() prints called b and it will exute console.log(a);

var a is declared grobally.

Ex2: b();

console.log(a);

var a = "Hello World";

function b(){

console.log('called b!');

}

Output: called b!

undefined

Note: 1. Executes b

2. to console js does not know about b to print so it will set initially a as undefined because somewhere in js a is declared.

Ex3: b();

console.log(a);

function b(){

comsole.log('called b!');

}

Output: called b!

Uncaugth ReferenceError: a is not defined

//because a is not created

Ex4: function b(){
Console.log('called b!')
}
b();
console.log(a);
var a = "Hellow World";
Output: called b!
undefined

Ex5: function b(){
console.log('called b!');
}
var a;
b();
a = 'Hellow World';
Output: called b!
undefined

Ex6: var a = 'Hellow World';
function b(){
console.log('called b!');
}
b();
console.log(a);
Output: called b!
Hellow World;

Undefined:

Ex1: var a;
console.log(a);
Output: undefined

Ex2: console.log(a);
Output: Uncaugth ReferenceError: a is not defined

Ex3: var a;

```
console.log(a);
If(a === undefined){
        console.log('a is undefined');
}else{
        console.log('a is defined');
}
```

Output: a is undefined

Ex4: never do this(never set any value to undefined becas js will set this value initially)

```
var a  = 'Hellow World';
console.log(a);
a = undefined
If(a === undefined){
        console.log('a is undefined');
}else{
        console.log('a is defined');
}
```

Output: Hellow World
a is undefined

Creation context & Execution context:

Ex1:

```
function b(){
        console.log('Called b!') ----------1---------->creation
}
b();----------2---------->Execution
console.log('a');----------3---------->Execution
var a = 'Hellow World';----------1---------->creation
console.log(a);----------4---------->Execution
```

Output: Called b!
Undefined

Hellow World

Functions, context &variable Environments:

Ex1:
```
function b(){
        var myVar;
        console.log(myVar);
}
function a(){
        var myVar =2;
        console.log(myVar);
b();
}
var myVar =1;
console.log(myVar);
a();
```
output: 1
2
undefined

Note: If you add console.log(myVar); after a(); it will prints one more 1 because myVar declared at the global level

SCOPE: where a variable is available in your code

Ex1:
```
function b(){
        console.log(myVar);
}
function a(){
        var myVar = 2;
b();
}
var myVar =1;
a();
```
Output: 1

Ex2:
```
function a(){
```

```
        function b(){
            console.log(myVar);
        }
        var myVar = 2;
        b();
    }
    var myVar =1;
    a();
```

Output: 2

Ex3:

```
    function a(){
        function b(){
            console.log(myVar);
        }
        var myVar = 2;
        b();
    }
    var myVar =1;
    a();
    b();
```

Output: 2

Uncaugth ReferenceError: b is not defined

Because u can not call b() which is inside a()

Ex4:

```
function a(){
        function b(){
            console.log(myVar);
        }
        b();
    }
    var myVar =1;
    a();
```

Output: 1

Functions are Objects

Ex1: function greet(){
console.log('hi');
}
greet.language = 'english';
console.log(greet); // function greet(){
console.log('hi');
}
console.log(greet.language);// English

Function Statements and Function Expressions

Expression: A unit of code that results in a value. It doesn't have to save to a variable.

Ex1: a =3//3
1+2;//3
a = {greeting:'hi'}//object {greeting:'hi'}

Note: Any expression in js endsup with value.

Statement just do work.

Ex1: var a;
If(a===3){
}

Ex2: u can not assign variable to statement bcas it doesn't return a value it just does work

var a;
var b = if(a==3){
}

Function expressions:

Ex1: greet();
function greet(){
console.log('hi');---------->normal function
}

```
var anonymousGreet = function(){
        console.log('hi');----------------------------->function expression
}
anonymousGreet();
```

Output: hi

hi

Ex2:

```
greet();
function greet(){
        console.log('hi');
}
anonymousGreet();
var anonymousGreet = function(){
        console.log('hi');---------------->js it will treat as variable but value is different
}
```

Output: hi

Uncaught TyError: undefined is not a function

Note: 1. Js will set anonymousGreet to undefined

3. function expressions are not hoisted js only hoists the variables

Ex3:

```
greet();
function greet(){
        console.log('hi');
}
anonymousGreet();
var anonymousGreet = function(){
        console.log('hi');---------------------------------->function expression
}
anonymousGreet();
```

Output: hi

Uncaught TyError: undefined is not a function

Ex1: U can create anything on the fly means when you calling the function.
U can call with numbers strings objects and functions

```
function log(a){
        console.log(a);
}
log(3); //3
log('hello');  //hello
log({greeting:'hi'}); //Object  {greeting:'hi'}
log( function(){
        console.log('hi')
    }); //      function(){
                console.log('hi')
        }
```

Ex2: cool stuff

```
function log(a){
        a();
}
log(   function(){
                console.log('hi');
}   );
```

Output: hi

Note: calling the log with function expression which will assign to a variable.
Now you can call anonymous function with (); that is a();
Because functions are objects in js