Date __/__/__
Page ______

↳ Hoisting in JS :-

Example ① :- (A)

```
var x = 7;

function getName(){
console.log("Namaste Javascript");
}

getName();        // invoke function
console.log(x);
```

Output :-

```
Namaste Javascript
7
```

(B)

```
getName();
console.log(x);

var x = 7;

function getName(){
console.log("Namaste Javascript");
}
```

⟹ Output :-

```
Namaste Javascript
Undefined.
```

→ If we remove var x from the figure (B) then it will show error – "x is not defined".

↳ <u>Hoisting</u> – It is phenomena in JS by which we can access these variables and functions even before we have initialized it & put some value in it with and we can access it without any error.

↳ Console.log(getName); ⟶ It prints the function itself

<u>Q.</u> But, if we write this before initializing the f^n ? (what happen)

i.e.
```
getName();
console.log(x);
console.log(getName);
var x = 7;

function getName(){

console.log("Namaste Javascript");
}
```

⟹ <u>Output:–</u>

```
Namaste Javascript
Undefined
f getName () {

Console.log("Namaste Javascript")
}
```

Date__/__/__
Page______

↳ Difference b/w not defined and undefined

When the variable is not present in the code ~~and we~~ i.e. there is no value for 'x' (be a variable). the JS will throw an error

↓

Reference x is not defined

Q. If we have getName() as arrow function then what happens?

```
    getName();
    Console.log(x);
    Console.log(getName);

    Var x = 7;

var getName = () => {

      Console.log("Namaste JS");
  }
```

→ Output :—
Error i.e
getName is not a function

↓ because

When we initialize getName as "arrowfn" then it behaves like a variable.

↓ i.e. it's value is initialized with "undefined".

Date __/__/__
Page ______

↳ If we declare function like this :—

```
var getName2 = function () {

}
```

} Then getName2 will behave like a **variable**.

Note :— Only in the case of proper function the JS copy the whole code of the function.