21 September 2024 12:06

User Interface for Web Development Part-1 | Introduction to HTML | Brain Mentors

FUNCTIONS

1. Normal functions / Function Definition style
2. Anonymous Functions
3. NFE { Named function Expression }
4. Arrow Function (Recommended)

SIF (Self Invoking function)

IIFE (Immediate Invoking Function ~~Expression~~)

general

Function → Keyword

Name

Brackets

code

return statement

{
}

LOCAL Scope V/s Global scope

within scope

→ available

→ overrid X

→ override

→ change

→ accessible → anywhere.

~~[illegible]~~ outside scope no change

1st type → Global → override, change

2nd type → Local → No change in Logic, Can not be override

1) function can Be returned inside a function.

Arrow Functions

↳ Recommended

↳ short hand syntax

↳ Pure functions ✓

this X

↳ Pure functions

↳ Lambda fun^n

↳ Very Popular

~~this~~

window → global object

this → holds the address of current calling object.

HOISTING → Placing at TOP

CMS

① Variable hoisting

```
function disply(){
    console.log('BEGINING ---z is ',z);
  var z=1000;
  console.log('START--z is ',z);
  if(10>2){
    var z =2000;
      console.log('z is ',z);
  }
  var z= 3000;
    console.log('END--z is ',z);
}
```

var z → undefined ← initialize

undefined

1000

re-assign

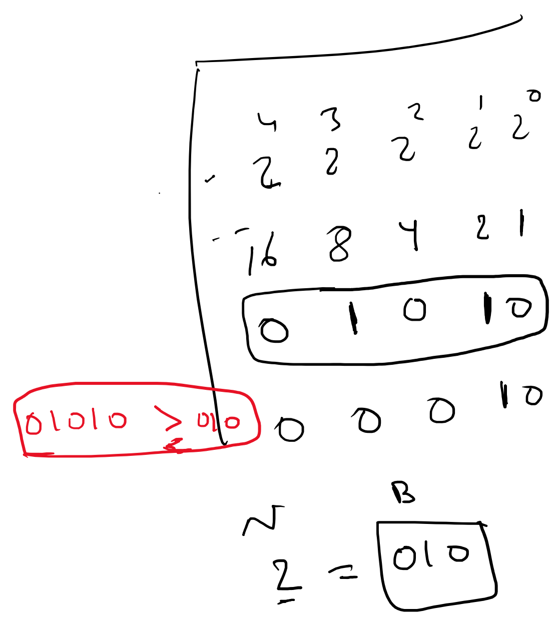
2000

3000

var → ① functional level hoisting

② initialize → undefined

② Function Hoisting



Let , const (ES6)(2015)

2 scopes

```
function disply(){
    console.log('BEGINING ---z is ',z);
  let z=1000;
  console.log('START--z is ',z);
  if(10>2){
    let z =2000;
     console.log('z is ',z);
  }
  let z= 3000;
    console.log('END--z is ',z);
}
```

Rules

Let

→ within the same scope u cannot redeclare same variable

→ you cannot access let variables before initialization

→ hoisting ✓

ES6

2015 ES6

Yes !

let z

```
function disply(){
  console.log('BEGINING ---z is ',z);
  let z=1000;
  console.log('START--z is ',z);
  if(10>2){
    let z =2000;
    console.log('z is ',z);
  }
  // let z= 3000;
  console.log('END--z is ',z);
}
```

declaration ✓

initialization ✗

initialization

hoisting ✓

initialize ✗

only declaration

{

}

TDZ

Temporal Dead Zone

→ they are only limited & available within their respective scopes.

const. (ES6-2015)

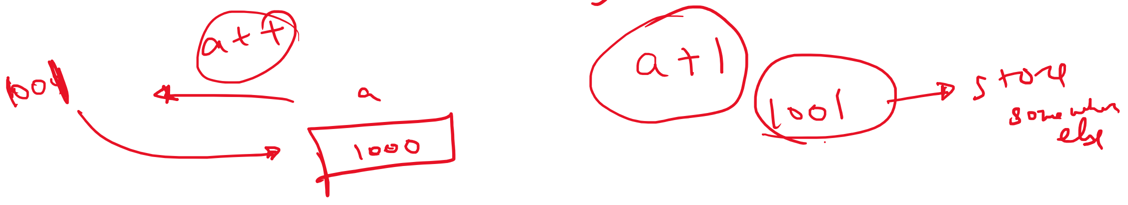
→ same Rules as of Let

const a = 1000;

a+1 ✓

a++ ✗

→ you cannot re-assign anything to a constant variable.



1001
a++
a
1000
a+1
1001
store somewhere else