

APRIL 2020

03 Friday

Week 14 | 094-272

Es (Pg 172) ECMAScript 2015 / ES6  
 → Is classes → Not object  
 → Template → Is objects  
 → Keyword → class

```

  Eg..
  class classname {
    constructor() {
      // ...
    }
  }
  
```

Eg: Create object

```

  const mycar = new Car("Ford", 2014);
  
```

Auto called when object created

(Pg 174) (Com)  
Constructor Method

Special method that has to have exact same name as constructor

Execute → when new object created  
 auto

Used to initialize object properties

do not add empty define

Class Methods

```

  class classname {
    constructor() {
      // ...
    }
    method-1() {
      // ...
    }
    method-2() {
      // ...
    }
  }
  
```

APRIL 2020

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

```

  (body)
  <p id="b1"></p>
  <script>
    class Car {
      constructor(name, year) {
        this.name = name;
        this.year = year;
      }
      age(x) {
        return x - this.year;
      }
    }
    const date = new Date();
    let year = date.getFullYear();
    const mycar = new Car("Ford", 2014);
    document.getElementById("b1").innerHTML = mycar.age(year);
  </script>
  </body>
  
```

Object → mycar.age(year)  
 1 alert(this); → [object object]  
 2017  
 00 06



mp -

→ Is oops (Class is Special type of function) (gk)

↑

Declared only once → else get error

Components: Class Declaration, Class Expression

1) Class Declaration

Class is defined using ; Class keyword → declare many particular name

Naming Convention → Start with uppercase letter

JULY 2020

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

```
eg.
class Employee {
    constructor(id, name) {
        this.id = id;
        this.name = name;
    }
    details() { console.log(this.id + " " + this.name); }
}

var e1 = new Employee(1, "Aj");
e1.details();
```

Saturday

You get  
Error

Uncaught

Reference

: cannot access

'Employee' before  
initialization.

## M Class expression

Class  $\rightarrow$  Named / unnamed; Not  $\rightarrow$  assign  $\rightarrow$  class nameNot possible  $\rightarrow$  Class Declaration12  
c) Unnamed

Eg 1

var emp = class Employee {

constructor (id, name) {

this.id = id;

this.name = name;

}

var obj = new emp (1, "Mano-Santi");

console.log (obj.name); // Mano-San

Eg 2

var emp = class {

constructor (id, name, salary) {

this.id = id;

this.salary = salary;

}

console.log (emp.name);

o/p emp  
undefinedAllow  $\rightarrow$  Redefine class

12 Sunday

Eg 1

var emp = class {

constructor (id, name) {

this.id = id;

this.name = name;

details() { console.log (this.id + " " + this.name); }

var e1 = new emp (1, "Sugan");

e1.details();



JULY  
2020

13<sup>th</sup> Miguel  
Monday

```

174 var emp = class {
    constructor (id, name) { this.id = id;
    this.name = name; }
    detail () { console.log (this.id + " " + this.name) }
}

var e1 = new emp (1, "chain");
e1.detail ();
  
```

Named → Scope → class name upto Class Body  
 Class name → property to retrieve

document.write (?)

```

1 var car = class Employee {
2   constructor (name) { this.name = name; }
3   console.log (car.name);
  
```

(com) Constructor Method (pges)  
 Class → can contain only one Constructor  
 Special Type of method used to initialize/create object; called when allocated for object.

Declare com → keyword → Constructor

Is narrow use parent class through keyword → Super  
Super use → Constructor

JULY 2020

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

```

Class Company Name {
    constructor () { this.company name = "Aegis"; }
    tryout () { console.log ("Method from Company name class"); }
}
  
```

14

196-170 | Week 29

JULY  
2020

Tuesday

tryoutci;

console.log("Method from CompanyName" <sup>class</sup>);

}

class Employee extends CompanyName {

constructor(id, name) {

super();

this.id = id;

this.name = name;

var emp = new Employee(1, "John");

console.log(emp.id + " " + emp.name + " " + emp.companyName);

emp.tryoutci;

1 John Aegis

Method from CompanyName class

Macquies Prop