

6. Inheritance Constructor

■ Constructor inside only parent class

```
class Father {
  constructor() {
    console.log('I am a constructor of Father Class')
  }
}

class Son extends Father {
}

const son = new Son();
```

■ Constructor inside only parent class pass parameters

```
class Father {
  constructor(msg) {
    console.log(msg)
  }
}

class Son extends Father {
}

const son = new Son("This is constructor params");
const father = new Father("This is constructor params");
```

■ Constructor inside only child class

```
class Father {
}

class Son extends Father {
  constructor() {
    super();
    console.log('I am a constructor of Son Class')
  }
}

const son = new Son();
```

■ Constructor inside only child class pass parameters

```
class Father {
}

class Son extends Father {
  constructor(msg) {
    super();
    console.log(msg)
  }
}

const son = new Son(" I am child class constructor");
```

■ Constructor inside both parent & child class

```

class Father {
  constructor() {
    console.log('I am a constructor of Father Class')
  }
}

class Son extends Father {
  constructor() {
    // Always call the parent class constructor first with `super` to ensure the parent's properties are
    // initialized.
    super();
    console.log('I am a constructor of Son Class')
  }
}

const son = new Son();

```

■ Constructor inside both parent & child class pass parameters

```

class Father {
  constructor(parentParams) {
    console.log(parentParams)
  }
}

class Son extends Father {
  constructor(childParams) {
    // Always call the parent class constructor first with `super` to ensure the parent's properties
    // are initialized.
    super();
    console.log(childParams)
  }
}

const son = new Son("I am child class constructor");
const father = new Father("I am parent class constructor");

```

#JavaScript_OOP