

Agenda

> OBJECTS

> Object Methods

, 'this' keywords

> Syntax

--> Set of Key-Value pairs

obj Name
↑

```
let obj = {
```

```
  key ← name : "Sumit" → Values
```

```
    age : 25
```

```
    : - - -
```

```
    - - - - -
```

```
}
```

```
let obj = {  
  brand : "Toyota",  
  model : "Supra",  
  year : 2012,  
  startEngine : function() {  
    console.log("Engine Start");  
  }  
}
```

→ object
delimitation

How to
Access → . notation
[] notation

```
let obj = {  
  brand : "Toyota",  
  model : "Supra",  
  year : 2012,  
  startEngine : function() {  
    console.log("Engine Start");  
  }  
}
```

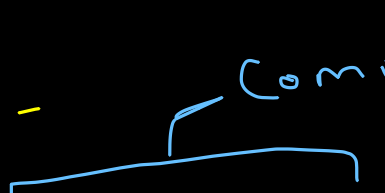
notation (exact key names & eq)

↳ obj.model → "Supra"

obj.mylbrand → undefined

↓
no key

```
let obj = {  
  brand : "Toyota",  
  model : "Supra",  
  year : 2012,  
  startEngine : function() {  
    console.log("Engine Start");  
  }  
}
```

[] notation :-  Compute the values

↳ obj[model] → X

↳ It will treat as variable

→ obj["model"] → "Supra"

→ obj.startEngine → whole function

→ obj.startEngine[()] → "Engine started"

How to delete

```
let obj = {  
  brand : "Toyota",  
  model : "Supra",  
  year : 2012,  
  startEngine : function() {  
    console.log("Engine Start");  
  }  
}
```

`delete obj.year`

`obj.brand = "Maruti"` → Replacing the value

`obj.color = "Red"`

↳ color key will get created if not present before

```
let obj = {  
  brand : "Toyota",  
  model : "Supra",  
  year : 2012,  
  startEngine : function() {  
    console.log("Engine Start");  
  }  
}
```

? How to iterate → using for...in loop

```
for (let key in obj) {  
  4 times  
  → key names  
  → obj[key]  
}
```

```
let calculator = {  
  add : function(a,b){  
    return a+b;  
  },  
  subtract : function(a,b){  
    return a-b;  
  }  
};
```

```
calculator.mul = function(a,b) {  
  return a*b;  
}
```

→ calculator.add(5, 7) → 12

→ calculator.subtract(5, 7) → -2

→ calculator.add(5) → NaN
 ↓
 5 + undefined

'this' keyword

```
let obj = {  
  name: "Rahul"
```

```
  print: function() { return
```

```
  }
```

```
    this.name }  
    }
```

obj.name

↓
"Rahul"

window()

↳ 'this' used outside
obj

```
obj = {  
  name: "Saurav"  
  print: () => { return this.name }  
}
```

→ window()

↓
undefined

Exercise 1: Create an object representing a book with properties for title, author, and year. Add a method to change the year of the book.

Exercise 2: Create an object representing a person with a method to introduce themselves.

