

ways of creating an object① Object literal

```
var fees = { }  
fees['Rahul'] = 200  
fees['Ram'] = 400
```

② object literal declare & initialize

```
var fees = {  
    'Rahul': 200;  
    'Ram': 300;  
}
```

③ Object constructorSyntax

```
var object-name = new Object( );
```

or

```
var object-name = { }
```

both same

ex

```
var fees = new Object( )
```

or

```
var fees = { }
```

Note -> If we want to create a single object  
we use above three

\* if we want to create multi objects we will use factory function or constructor function #2

#### ④ factory function →

```
function fees () {  
  return {
```

```
    'rahal' : 200,
```

```
    'Ram' : 500,
```

```
    total : function () {
```

```
        return (400+500+600);  
    }  
  }  
}
```

```
var f1 = new fees();
```

now f1 is object of fees we can now use or access any property of 'fees' object'

#### ⑤ → By using This keyword ← Constructor function or class in python

```
function Mobile (model-no) {
```

```
  this.model = model-no;
```

```
  this.prcu = function () {
```

```
    return (this.model  
            + " Price is 300")  
  }  
}
```

```
var samsung = new Mobile ('8810')
```

```
var holoia = new Mobile ('1110')
```



## Prototype

- Why we use prototype : - let assume there is a class name Mobile it has property like model & color
  - we create two objects of the Mobile class Samsung & Nokia.
  - now we add a property name "price" in by Samsung object in Mobile class
- Samsung.price = 1200;
- Now this property will only be available for Samsung object, it will be not available for another object like Nokia
  - This will be repetitive if we want to add same property for all the object.
  - That is why we use prototype this will add all the property for all.

Syntax classname.prototype.key = value;

Mobile.prototype.price = 1200;

ex var Mobile = function (model) {  
     this.model = model;  
     this.price = 3000;  
     ~~this.color~~ };

var Samsung = new Mobile('Galaxy');  
 var Nokia = new Mobile('3310');  
 Mobile.prototype.key = 'white';

The prototype property will not be visible in console.

#4

## Class in javascript

- Class is basically a constructor function
- When we assign the constructor function to a variable it is known as class.
- There is nothing like class in javascript it is constructor function that are used as class.

ex var Mobile = function ( ) {

    this.model = 'Galaxy';

    this.name = 'Mokia';

    this.full-name = function ( ) {

        return (this.name + this.model);

    }

};

var m1 = new Mobile ( )

var m2 = new Mobile ( )