

```
1.var dateToday = new Date('October 15, 1996 05:35:32:77 GMT+11:00');
```

How to fetch the millisecond according to universal time from a given Date object?

Ans:-getMilliseconds()

2. Write a JavaScript program to display the reading status (i.e. display book name, author name and reading status) of the following books.

```
var library = [  
  {  
    author: 'Bill Gates',  
    title: 'The Road Ahead',  
    readingStatus: true  
  },  
  {  
    author: 'Steve Jobs',  
    title: 'Walter Isaacson',  
    readingStatus: true  
  },  
  {  
    author: 'Suzanne Collins',  
    title: 'Mockingjay: The Final Book of The Hunger Games',  
    readingStatus: false  
  }  
];  
  
ans  
  
for (var i = 0; i < library.length; i++)  
  {
```

```

    var book = "" + library[i].title + "" + ' by ' + library[i].author + ".";

    if (library[i].readingStatus) {

    console.log("Already read " + book);

    } else

    {

    console.log("You still need to read " + book);

    }

}

```

Q3. What will be the output of the following code?

```

var Employee =

{

company: 'Rohit'

}

var Emp1 = Object.create(employee);

delete Emp1.company Console.log(emp1.company);

ans output will be error as javascript is case sensitive

```

Q4. Consider the two functions below. Will they both return the same thing? Why or why not?

```

function foo1()

{

return {

bar: "hello"

};

```

```

}

function foo2()

{

return

{

bar: "hello"

};

}

ans

```

they both will not return the same thing because javascript is a single threaded language that is it will try to execute the given piece of code before going to next line hence in 2nd case it will try to execute the return value before moving to next line whereas in 1st case it will look for the closing end of }

```

5.var arr = [2, 56, 78, 34, 65];

var new_arr = arr.map(function(num) {

return num / 2;

});

print(new_arr);

```

What will be the output for the above code?

```

// [object Array] (5)

[1,28,39,17,32.5]

```

```

6.const set = new Set(['Beethoven', 'Mozart', 'Chopin', 'Chopin'])

```

How to delete 'Beethoven' from set

```

ans-set.delete('Beethoven')

```

Q7. Create a nested array object called operatingSystem and add below key and values

Name - Ubuntu , version - 18.4 , license - open source

```
ans: const operatingSystem = {  
    name: 'Ubuntu',  
    version: 18.04,  
    license: 'Open Source'  
};
```

```
const output= Object.entries(operatingSystem);
```

```
console.log(output);
```

```
=>
```

```
[  
    ["name", "Ubuntu"]  
    ["version", 18.04]  
    ["license", "Open Source"]  
]
```

```
8.const name = { firstName: 'Philip',
```

```
lastName: 'Fry' };
```

```
const details = {
```

```
job: 'Delivery Boy',
```

```
employer: 'Planet Express'
```

```
};
```

How to copy values from one object to another in above mentioned example , merge and

print them?

ans :

```
const name = { firstName: 'Philip',  
lastName: 'Fry' };  
  
const details = {  
job: 'Delivery Boy',  
employer: 'Planet Express'  
};  
  
const output=Object.assign(name, details);  
  
console.log(output);
```

```
9.const gimli =  
{ name: "Gimli",  
race: "dwarf",  
weapon: "axe",  
greet: function() { return `Hi, my name is ${this.name}!`; }, },  
gimli.greet()
```

10.Consider below as an example of constructor function where we want name and level to be

referred to a function itself, is below example correct or not? Please explain?

```
function Hero(name, level)  
{
```

```
name = name;
```

```
level = level;
```

```
}
```

ans: it is not correct, as we want name and level to be referred to a function itself, correct way is

```
function Hero(name, level)
```

```
{
```

```
  this.name = name;
```

```
  this.level = level;
```

```
}
```

here this.property is used for refering to the function itself.