**Fun With JavaScript** **☺** (by- Tejas Shailesh Sabunkar)  
  
===========PROBLEM-1======================

var foo = function foo() {

console.log(foo === foo);

};

foo();

Ans -

True

False

TypeError

===========PROBLEM-2======================

Number("1") - 1 == 0;

Ans-

function

number

object

undefined  
TypeError

===========PROBLEM-3======================

function aaa() {

return

{

test: 1

};

}

alert(typeof aaa());

Ans-

true

false  
TypeError

===========PROBLEM-4======================

(true + false) > 2 + true;

Ans-

true

false

TypeError

NaN

===========PROBLEM-5======================

function bar() {

return foo;

foo = 10;

function foo() {}

var foo = '11';

}

alert(typeof bar());

Ans-

number

function

undefined

string

TypeError

===========PROBLEM-6======================

"1" - - "1";

Ans-

0

2

11

"11"  
TypeError

===========PROBLEM-7======================

var x = 3;

var foo = {

x: 2,

baz: {

x: 1,

bar: function() {

return this.x;

}

}

}

var go = foo.baz.bar;

clg(go());

clg(foo.baz.bar());

Ans-

What is the output ?

>

===========PROBLEM-8======================

new String("This is a string") instanceof String;

Ans-

true

false

TypeError

===========PROBLEM-9======================

new String("") === typeof(string)

Ans-

true

false

TypeError

===========PROBLEM-10======================

[] + [] + 'foo'.split('');

Ans-

"f, o, o"

TypeError

["f", "o", "o"]

[][]["f", "o", "o"]

===========PROBLEM-11======================

new Array(5).toString();

Ans-

",,,"

[]

"[]"

TypeError

===========PROBLEM-12======================

var myArr = ['foo', 'bar', 'baz'];

myArr.length = 0;

myArr.push('bin');

console.log(myArr);

Ans-

What is the output ?

>

===========PROBLEM-13======================

String('Hello') === 'Hello';

Ans-

true

false

TypeError

===========PROBLEM-14======================

var x = 0;

function foo() {

x++;

this.x = x;

return foo;

}

var bar = new new foo;

console.log(bar.x);

Ans-

Referrence Error

TypeError

undefined

0

1

===========PROBLEM-15======================

"This is a string" instanceof String;

Ans-

True

False

TypeError

===========PROBLEM-16======================

var bar = 1,

foo = {};

foo: {

bar: 2;

baz: ++bar;

};

foo.baz + foo.bar + bar;

Ans-

TypeError

ReferrenceError

Undefined

NaN

null

4

5

===========PROBLEM-17======================

var myArr = ['foo', 'bar', 'baz'];

myArr[2];

console.log('2' in myArr);

Ans-

What is the output?

>

===========PROBLEM-18======================

var arr = [];

arr[0] = 'a';

arr[1] = 'b';

arr.foo = 'c';

alert(arr.length);

Ans-

1

2

3

undefined

===========PROBLEM-19======================

10 > 9 > 8 === true

Ans-

What is the output?

>

===========PROBLEM-20======================

function foo(a, b) {

arguments[1] = 2;

alert(b);

}

foo(1);

Ans-

What is the output of alert?

>

===========PROBLEM-21======================

NaN === NaN;

Ans-

Ans-

What is the output?

>

===========PROBLEM-22======================

a = 10;

function foo(){

a = 30;

return {

a

}

}

console.log(a);

foo()

console.log(foo.a)

console.log(foo().a)

Ans-

What is the output ?

>

===========PROBLEM-23======================

console.log('op1',a)

foo = () =>{

a = 30;

}

console.log('op2',a)

a = 50;

console.log('op3',a)

function foo(){

a = 10;

}

console.log('op4',a)

Ans-

Write the output in sequence ?

>

===========PROBLEM-24======================

function Cat(cname, ccolor) {

this.catName = cname;

this.catColor = ccolor;

catEyes = 'blue';

}

Cat.prototype.age = 12;

var catC = new Cat("Fluffy", "White");

console.log('op1',Cat.catName);

console.log('op2',catC.catName);

console.log('op3',Cat.prototype.age);

console.log('op4',catC.prototype.age);

console.log('op5',Cat.\_\_proto\_\_.age);

console.log('op6',catC.\_\_proto\_\_.age);

Ans-

Write the output in sequence ?

>

===========PROBLEM-25======================

var bottle = function bottle() {­

bottleName = 'Kinely';

this.bottleId = 12;

}

console.log('op1', bottle().bottleId)

console.log('op2', bottle().bottleName)