

Question : 1 of 30

If we run the following Node.js program what will be printed on the console?

```
let func = ()=>{
    let pets = ["Dog", "Cat", "Fly", "Bug", "Ox"];
    pets = pets.slice(2, 4);
    return pets[1];
}
console.log(func());
```

1. Dog

2. Cat

3. Fly

4. Bug

5. Ox

If we run the following Node.js program what will be printed on the console?

```
let functionVariable = function(list, check){  
    var matchFound = false;  
    for (let i = 0; i < list.length; i++){  
        if (check === list[i]) {  
            matchFound = true;  
            break;  
        }  
    }  
    return matchFound;  
}  
const cities = ["Cheyenne", "Santa Fe", "Tucson", "Great Falls", "Honolulu"];  
console.log(functionVariable(cities, "Karachi"));
```

1. True

2. False

3. Nothing

4. There will be a runtime error

Question : 3 of 30

If we run the following Node.js program what will be printed on the console?

```
let z = (x, y) => x + y;  
console.log(z(3, 1));
```

1. 1

2. 2

3. 3

4. 4

5. There will be a runtime error

Question : 4 of 30

If we run the following Node.js program what will be printed on the console?

```
console.log(((x, y) => x + y)(3,4));
```

1. 1

2. 3

3. 4

4. 7

5. There will a runtime error

If we run the following Node.js program what will be printed on the console?

```
let z = (x)=>{
    if (x > 2)
    {
        x = x + 1; // Add 1 to x
        if (x < 3)
            x = x + 2; // Add 2 to x
    }
    else
    {
        x = x + 2;
        if (x == 4)
            x = x + 1;
    }
    return x;
}
console.log(z(2));
```

1. 2

2. 3

3. 4

Question : 6 of 30

I want to create a condition for an if statement that will succeed only if variable x is not within the range 10 to 20 (inclusive) and variable y is not within the range 20 to 30 (inclusive). Which of the following is the correct condition to use?

- 1. $((x < 10 \&\& x > 20) \parallel (y < 20 \&\& y > 30))$
- 2. $(x \leq 10 \parallel x \geq 20 \&\& y \leq 20 \parallel y \geq 30)$
- 3. $(x < 10 \parallel x > 20 \&\& y < 20 \parallel y > 30)$
- 4. $((x \leq 10 \parallel x \geq 20) \&\& (y \leq 20 \parallel y \geq 30))$
- 5. $((x < 10 \parallel x > 20) \&\& (y < 20 \parallel y > 30))$

NEXT



```
if (x < 5)
    a = 6;
else
    a = 7;
```

Which of the following code segments has exactly the same effect as that extract?

1. if (x >= 5)
 a = 6;
else
 a = 7;

2. if (x <= 5)
 a = 7;
else
 a = 6;

3. if (x > 5)
 a = 6;
else
 a = 7;

4. if (x >= 5)
 a = 7;
else
 a = 6;

5. if (x > 5)
 a = 7;
else
 a = 6;

In this multiple if statement, one of the statements can never be executed. Which statement is it?

```
if (m > 2 && n < 10) {  
    if (m > 3 || n > 1)  
        console.log("Statement A");  
    else  
        console.log("Statement B");  
}  
else {  
    if (m > 3 && n < 9)  
        console.log("Statement C");  
    else  
        console.log("Statement D");  
    console.log("Statement E");  
}
```

1. Statement A

2. Statement B

3. Statement C

4. Statement D

If we run the following Node.js program which consists of two files what will be printed on the console?

```
// File addTwo.mjs
export default function addTwo(num) {
    return num + 2;
}

// File app.mjs
import addTwo from './addTwo.mjs';

console.log(addTwo(4));
```



- 1. There will be a runtime Syntax Error
- 2. 6
- 3. 2
- 4. 4

Question : 10 of 30

If we run the following Node.js program what will be printed on the console?

```
let z = (x, y) => x > y ? 3 : 4;  
console.log(z(5,6));
```

1. 4

2. 3

3. 5

4. 6



```
    .then(function(data) {
      console.log('Success:', data.message);
    })
    .catch(function(data) {
      console.log('Error:', data.message);
    });
  
```

1. Success, Success, Error, Error caught, Success: test

2. Success, Defeat, Error, Success: test

3. Success, Success, Error caught, Success: test

4. Success, Error, Error, Error caught, Success: test

5. Error, Error, Error caught, Error: test

6. Success, Defeat, Error, Error caught, Success: test

```
let promise = job(true);

promise.then(function(data) {
    console.log(data);
    return job(true);
})

.then(function(data) {
    if (data !== 'victory') {
        throw 'Defeat';
    }
    return job(true);
})

.then(function(data) {
    console.log(data);
})

.catch(function(error) {
    console.log(error);
    return job(false);
})

.then(function(data) {
    console.log(data);
    return job(true);
})

.catch(function(error) {
    console.log(error);
    return 'Error caught';
})

.then(function(data) {
    console.log(data);
    return new Error('test');
})

.then(function(data) {
```

Question : 11 of 30

If we run the following Node.js program what will be printed on the console?

```
function job(state) {  
    return new Promise(function(resolve, reject) {  
        if (state) {  
            resolve('Success');  
        } else {  
            reject('Error');  
        }  
    });  
  
    let promise = job(true);  
  
    promise.then(function(data) {
```

Question : 12 of 30

Consider the following line which has been extracted from a program:

```
for (var count = 0; count < 31; count+=2)
```

How many times does this loop execute?

1. 15

2. 16

3. We cannot tell just from the line given above.

4. 30

5. 31

```
a=0;
function func1(callback) {
    setTimeout(function() {
        a+=5;
        callback();
        a+=9;
    }, 200);
}
a += 2;
func1(function() {
    console.log(a);
});
a += 3;
```

1. 19

2. 5

3. Nothing

4. 2

5. 3

If we run the following Node.js program what will be printed on the console?

```
let a = 0;  
function func1(callback) {  
    setTimeout(function() {  
        a+=5;  
        callback();  
        a+=9;  
    }, 200);  
}  
a += 2;  
func1(function() {  
    console.log(a);  
});  
a += 3;
```

1. 19

2. 5

3. Nothing

4. 2

If we run the following Node.js program what will be printed on the console?

```
const promise = new Promise((resolve, reject) => {
    resolve(1);
})
.then(value => {
    console.log(value);
    return 2;
})
.then(value => {
    console.log(value);
    return 3;
})
.then(value => {
    console.log(value);
    return 4;
})
.then(value => {
    console.log(value);
    return 5;
})
.then(value => {
    console.log(value);
})
.catch(value => {
    console.log(value);
})
```

```
.then(value => {
    console.log(value);
})
.catch(value => {
    console.log(value);
})
```

Reload Current Page



Do you really want to

OK

1. 5, 4, 3, 2, 1

2. 2, 3, 4, 5

3. 1, 2, 3, 4, 5

4. 1, 2, 3, 4

Question : 15 of 30

Given the following code:

```
var a = 5;  
var c = function(b) {  
    a += 3;  
    return b + a;  
}(2);  
console.log(c);
```

What will be printed on the console?

1. undefined

2. 10

3. 8

4. 7

K2

Quiz : Fundamentals of JavaScript and Node.js

Question : 16 of 30

Can you use the following statement to define a variable?

let 5variable = 5;



1. No



2. Yes

If we run the following Node.js program what will be printed on the screen?

```
function job(state) {  
    return new Promise(function(resolve, reject) {  
        if (state) {  
            resolve('Success');  
        } else {  
            reject('Error');  
        }  
    });  
}  
  
let promise = job(true);  
  
promise.then(function(data) {  
    console.log(data);  
    return job(false);  
})  
.catch(function(error) {  
    console.log(error);  
    return 'Error caught';  
})  
.then(function(data) {  
    console.log(data);  
    return job(true);  
})  
.catch(function(error) {  
    console.log(error);  
})
```

```
})
.then(function(data) {
  console.log(data);
  return job(true);
})
.catch(function(error) {
  console.log(error);
});
```

1. Success, Error, Success, Error

Reload Current Page

2. Error, Error caught, Success



Do you rea

3. Error, Success, Error caught

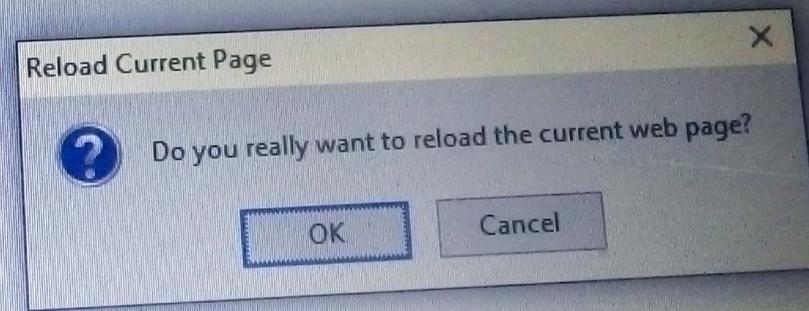
4. Success, Success, Success

5. Success, Success

6. Success, Error, Error caught

If we run the following Node.js program what will be printed on the console?

```
function saySomething(x) {  
    return new Promise(resolve => {  
        setTimeout(() => {  
            resolve("something" + x);  
        }, 2000);  
    });  
}  
function talk(x) {  
    const words = await saySomething(x);  
    console.log(words);  
}  
talk(2);
```



1. It will print a runtime SyntaxError

2. 2000

3. 2002

Quiz : Fundamentals of JavaScript and Node.js

Question : 19 of 30

Which of the following are valid Node Package Manager commands to create package.json and Node.js projects?

(You can select more than one option)

1. npm create

2. npm init -y

 3. npm init

4. node init

Reload Current Page X



Do you really want to reload the current web page?

OK

Cancel

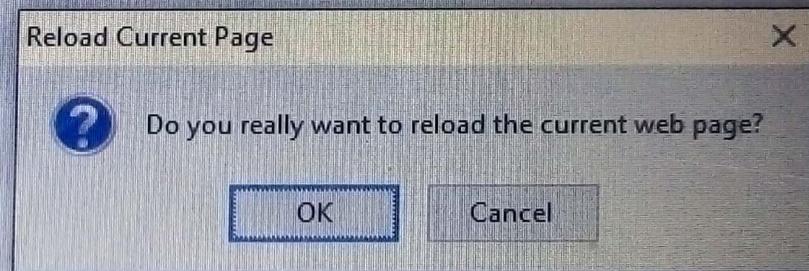
Question : 20 of 30

If we run the following Node.js program what will be printed on the console?

```
function job() {
    return new Promise(function(resolve, reject) {
        reject();
    });
}

let promise = job();

promise.then(function() {
    console.log('Success 1');
})
.then(function() {
    console.log('Success 2');
})
.then(function() {
    console.log('Success 3');
})
.catch(function() {
    console.log('Error 1');
})
.then(function() {
    console.log('Success 4');
});
```



```
    success4() {
      console.log('Success 4');
    }
  }
}
```

- 1. Error 1
- 2. Success 1, Success 2, Success 3, Success 4
- 3. Error 1 Success 4
- 4. Error 1, Success 1
- 5. Error 1, Success 1, Success 2, Success 3, Success 4
- 6. Success 1, Success 2, Success 3, Error 1, Success 4

When you run the following Node Package Manager command:

`npm install dependency`

The command does the following?
(You can select more than one option)

- 1. Downloads the dependency package in the `node_modules` folder.
- 2. Adds reference to the dependency module in the `package.json` file.
- 3. Finds the dependency module from `npmjs.org` and installs it in the project.
- 4. Downloads the sub dependency packages in the `node_modules` folder.

If we run the following Node.js program what will be printed on the console?

```
let f = (nr) => {
    console.log(nr);
    if (nr > 0) {
        f(-nr);
    }
}
f(3);
```

1. 4 3 2 1

2. 3 2 1 0

3. 1 2 3 4

4. 0 1 2 3

Q. If we run the following Node.js program which consists of two files what will be printed on the console?

```
// File addTwo.mjs
function addTwo(num) {
    return num + 2;
}
export { addTwo };

// File app.mjs
import addTwo from './addTwo.mjs';

console.log(addTwo(4));
```

1. There will be a runtime Syntax Error

2. 4

3. 6

4. 2

If we run the following Node.js program what will happen:

```
let myPromise = new Promise(function(myResolve, myReject) {  
    setTimeout(function() { myResolve("I love Pakistan"); }, 3000);  
});  
  
myPromise.then(function(value) {  
    console.log(value);  
});
```

- 1. Will print "3000" immediately
- 2. Will print "I Promise that I love Pakistan" immediately
- 3. Will print "I love Pakistan" immediately
- 4. Will print "I love Pakistan" after some time

Given the following code:

```
let a = "55" + 5 - 5;
```

What value will be in the variable a?

1. undefined

2. null

3. 555

4. 550

5. 55

6. NaN

Given the following code:

```
let a = "We are the best developers in the World!";
let b = "the World";
let c = a.indexOf(b);
if (c !== -1) {
    a = a.slice(0, c) + "Asia" + a.slice(c + b.length);
}
```

What string value will be in the variable a?

- 1. We are the best developers
- 2. We are the best developers in the Asia!
- 3. We are the best developers in Asia!
- 4. We are the best developers in the World!
- 5. We are the best developers in the!

Question : 27 of 30

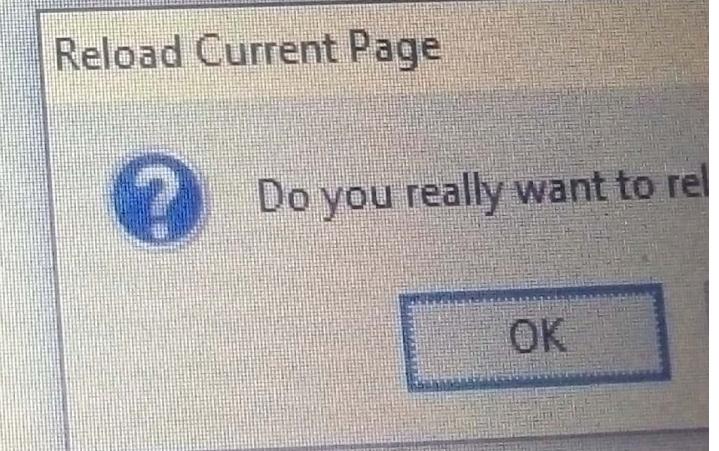
How many times is the string Hello! written on the screen?

```
for (let x = 1; x < 10; x++)  
    for (let y = 1; y < 5; y++)  
        console.log("Hello!");
```

- 1. The program does not give us enough information to work it out.
- 2. 45
- 3. 36
- 4. 50
- 5. 40

If we run the following Node.js program what will be printed on the console?

```
let a = 0;  
function func1(callback) {  
    setTimeout(function() {  
        a+=5;  
        callback();  
        a+=9;  
    }, 200);  
}  
a += 2;  
(function(a) {  
    func1(function() {  
        console.log(a);  
    })  
})(a);  
a += 3;
```



1. 19

2. 0

3. 3

4. 10

5. Nothing

6. 2

7. 5

Given the following code:

```
let dayOfWk = "Sat";
switch(dayOfWk) {
    case "Sat" :
        console.log("Whoopee");
    case "Sun" :
        console.log("Aha");
    case "Fri" :
        console.log("TGIF!");
    default :
        console.log("Shoot me now!");
}
```

What will be printed on the console?

(You can select more than one option)

1. Aha

2. Whoopee

3. TGIF!

4. Shoot me now!

Given the following code:

```
function printArray(a) {  
    let len = a.length, i = 0;  
    if (len == 0)  
        console.log("Empty Array");  
    else {  
        do {  
            console.log(a[i]);  
        } while (++i < len);  
    }  
}
```

What does the above code result in?

- 1. Prints the numbers in the array in order
- 2. Prints 0 to the length of the array
- 3. Prints the numbers in the array in the reverse order
- 4. Prints "Empty Array"