JS Institute

JSE-40-01

JSE – Certified Entry-Level JavaScript Programmer

QUESTION & ANSWERS

QUESTION: 1

Which of the following loop instructions is intended only to loop

through all elements of the indicated array?

Option A :

do ... while

Option B :

for ... in

Option C :

for ... of

Option D :

for

Correct Answer: C

Explanation/Reference:

Topic: for of over array

Try it yourself:

```
let people = ["Peter", "Paul", "Mary"];

for (let p of people) {

console.log(p);

}

// Peter

// Paul
```

// Mary

Explanation:

The for of statements iterates over the values of any iterable.

The code block inside the loop is executed once for each value.

https://www.w3schools.com/jsref/jsref_forof.asp

QUESTION: 2

Review the following code (note the variable name)let height = 170;

height = height + 10;

console.log(Height);

As a result of its execution, the following should appear in the console:

Option A :

"Height"

Option B :

180

Option C :

"Uncaught ReferenceError: Height is not defined"

Option D :

170

Correct Answer: C

Explanation/Reference:

Topic: ReferenceError

Try it yourself:

let height = 170;

height = height + 10;

console.log(Height);

// Uncaught ReferenceError: Height is not defined

Explanation:

JavaScript is case sensitive

and therefore height is not the same as Height with a capital "H".

The ReferenceError object represents an error when a variable that

doesn't exist (or hasn't yet been initialized) in the current scope is referenced.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/ReferenceError

QUESTION: 3

Analyze the following code:

```
let point = {x: 10, y: 20};
for (let f in point) {
    console.log(point[f]);
}
```

What will appear on the console as a result of its execution?

Option A :

x

y

Option B :

10

20

Option C :

point

Option D :

10

Correct Answer: B

Explanation/Reference:

Topic: for in over objects

Try it yourself:

```
let point = {x: 10, y: 20};

for (let f in point) {
    console.log(point[f]);
    // 10
    // 20
}
```

Explanation:

The for in statements combo iterates (loops) over the properties of an object.

The code block inside the loop is executed once for each property.

https://www.w3schools.com/jsref/jsref_forin.asp

QUESTION: 4

If a variable stores the value false then the variable:

Option A :

will no longer be used in the program.

Option B :

is of the Math type.

Option C :

is of the Logical type.

Option D :

is of the Boolean type.

Correct Answer: D

Explanation/Reference:

Topic: Boolean

Try it yourself:

```
let variable = false;

console.log(typeof variable);  // boolean
```

Explanation:

A Boolean is a logical data type that can have only the values true or false

https://developer.mozilla.org/en-US/docs/Glossary/Boolean

QUESTION: 5

Which of the following is not a loop instruction in JavaScript?

Option A :

for ... of

Option B :

for ... in

Option C :

do ... while

Option D :

if ... else

Correct Answer: D

Explanation/Reference:

Topic: if else

Try it yourself:

```
let num;

num = Math.random();

console.log(num) // e.g. 0.5882571338770821

if (num < 0.5) console.log("Low");

else console.log("High");
```

Explanation:

if else only happens once and therefore it is not a loop.

The if else statement executes a block of code if a specified condition is true

If the condition is false another block of code can be executed.

https://www.w3schools.com/jsref/jsref_if.asp

The for in statements combo iterates over the properties of an object.

The code block inside the loop is executed once for each property.

https://www.w3schools.com/jsref/jsref_forin.asp

The for of statements combo iterates over the values of any iterable.

The code block inside the loop is executed once for each value.

https://www.w3schools.com/jsref/jsref_forof.asp

The do while statements combo defines a code block to be executed once,

and repeated as long as a condition is true.

The do while is used when you want to run a code block at least one time.

https://www.w3schools.com/jsref/jsref_dowhile.asp

QUESTION: 6

Analyze the following code:

let x = 10;

ocnsole.log(x);

What exception will be thrown as a result of its execution attempt?

Option A :

RangeError

Option B :

SyntaxError

Option C :

TypeError

Option D :

ReferenceError

Correct Answer: D

Explanation/Reference:

Topic: ReferenceError

Try it yourself:

let x = 10;

ocnsole.log(x);

// Uncaught ReferenceError: ocnsole is not defined

Explanation:

There is a spelling error in console

The ReferenceError object represents an error when a variable that

doesn't exist (or hasn't yet been initialized) in the current scope is referenced.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/ReferenceError

QUESTION: 7

Analyze the following code:

let msg "Hello World";

console.log(msg);

What exception will be thrown as a result of its execution attempt?

Option A :

TypeError

Option B :

SyntaxError

Option C :

ReferenceError

Option D :

RangeError

Correct Answer: B

Explanation/Reference:

Topics: SyntaxError assignment operator

Try it yourself:

// let msg "Hello World";

// Uncaught SyntaxError: unexpected token: string literal

console.log(msg);

Explanation:

The assignment operator is missing.

The SyntaxError object represents an error

when trying to interpret syntactically invalid code.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/SyntaxError

QUESTION: 8

The msg variable contains a String type value.

Information about the number of characters of this string can be obtained using:

Option A :

msg.length

Option B :

msg.length()

Option C :

msg.charsAt()

Option D :

msg.chars

Correct Answer: A

Explanation/Reference:

Topic: string.length

Try it yourself:

```
let msg = "Hello";

console.log(msg.length);  // 5
```

Explanation:

The string.length property returns the length of a string.

https://www.w3schools.com/jsref/jsref_length_string.asp

QUESTION: 9

Most errors in JavaScript code that the interpreter encounters

while the program is running cause exceptions to be thrown.

What do unhandled exception do?

Option A :

The interpreter ignores unhandled exceptions,

the program continues to execute,

and no information is received.

Option B :

An error message appears in the console

and the execution of the program is aborted.

Option C :

An error message appears in the console

and program execution continues.

Option D :

The interpreter removes the erroneous piece of code,

replaces it with corrected one,

and continues the program execution.

Correct Answer: B

Explanation/Reference:

Topic: unhandled exception

Try it yourself:

console.log(hello);

// Uncaught ReferenceError: hello is not defined

console.log("I am to late!");

Explanation:

If an unhandled exception is raised n error message appears in the console

and the execution of the program is aborted.

Uncaught exception

https://en.wikipedia.org/wiki/Exception_handling#Uncaught_exceptions

QUESTION: 10

Analyze the following code:

```
function fn(a) {
    console.log('test');
    a--;
    if(a>2) fn(a);
}
fn(6);
```

How many times will test appear on the screen?

Option A :

One hundred twenty

Option B :

One

Option C :

Five

Option D :

Four

Correct Answer: D

Explanation/Reference:

Topic: recursive function postfix decrement operator

if greater than operator

Try it yourself:

```
function fn(a) {
    console.log('test: ' + a);
    a--;
    if(a>2) fn(a);
}
fn(6);
// test: 6
// test: 5
// test: 4
// test: 3
```

Explanation:

The fn() function decrements the parameter

and if the parameter is still greater than 2

it will call itself again.

A function that calls itself is called a recursive function

https://www.programiz.com/javascript/recursion

The decrement operator decrements its operand and returns a value.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Decrement

The if statement executes a block of code if a specified condition is true

https://www.w3schools.com/jsref/jsref_if.asp

The greater than operator returns true if the left operand is

greater than the right operand, and false otherwise.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Greater_than

QUESTION: 11

The result of the operation false || "false" will be:

Option A :

false

Option B :

true

Option C :

false

Option D :

0

Correct Answer: A

Explanation/Reference:

Topic: short-circuit evaluation with OR

Try it yourself:

```
x = false || "false";

console.log(x);         // false

console.log(typeof x);  // string
```

Explanation:

JavaScript has a short circuit evaluation.

For the logical OR operator that means,

if the first operand is falsy the second operand will be returned.

In this case the string "false"

The logical OR operator returns the value of the first truthy operand

encountered when evaluating from left to right,

or the value of the last operand if they are all falsy.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Logical_OR#short-circuit_evaluation

QUESTION: 12

Analyze the following code:

```
let x = 10;

function test() {

   let x = 20;
```

}

test();

console.log(x);

What will be displayed in the console as a result of its execution?

Option A :

10

Option B :

undefined

Option C :

Nothing will show up.

Option D :

"x"

Correct Answer: A

Explanation/Reference:

Topics: global scope function scope shadowing

Try it yourself:

let x = 10;

function test() {

  let x = 20;

}

test();

console.log(x); // 10

Explanation:

The function test() could access the global variable x

but if there is a new variable x inside of the function

it will be a different entity that is only valid inside of the function

and does not effect the global variable of the same name.

A variable declared outside a function and a { } block,

becomes a global variable with global Scope.

All { } blocks and functions can access it.

https://www.w3schools.com/js/js_scope.asp

JavaScript has function scope

Each function creates a new scope. Variables defined inside a function

are not accessible (visible) from outside the function.

https://www.w3schools.com/js/js_scope.asp

Shadowing

When a variable is declared in a certain scope

having the same name defined on its outer scope

and when we call the variable from the inner scope,

the value assigned to the variable in the inner scope

is the value that will be stored in the variable in the memory space.

https://www.geeksforgeeks.org/variable-shadowing-in-javascript

QUESTION: 13

We want to declare a protocol constant and initialize it with the value "http"

What should such a declaration look like?

Option A :

let const protocol = "http";

Option B :

let protocol; const protocol = "http";

Option C :

const protocol; protocol = "http";

Option D :

const protocol = "http";

Correct Answer: D

Explanation/Reference:

Topic: constant

Try it yourself:

const protocol1 = "http";

// let const protocol2 = "http";

// Uncaught SyntaxError: unexpected token: keyword 'const'

// const protocol3; protocol3 = "http";

// Uncaught SyntaxError: missing = in const declaration

// let protocol4; const protocol4 = "http";

// Uncaught SyntaxError: redeclaration of let protocol4

Explanation:

The value of a constant can't be changed through reassignment

and it can't be redeclared.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/const

QUESTION: 14

Analyze the following code:

let x 10;

console.log(x);

What exception will be thrown as a result of its execution attempt?

Option A :

ReferenceError

Option B :

RangeError

Option C :

TypeError

Option D :

SyntaxError

Correct Answer: D

Explanation/Reference:

Topic: SyntaxError

Try it yourself:

// let x 10;

// Uncaught SyntaxError: unexpected token: numeric literal

console.log(x);

Explanation:

The assignment operator is missing.

The SyntaxError object represents an error

when trying to interpret syntactically invalid code.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/SyntaxError

QUESTION: 15

Analyze the code below:

```
let a = 10;
do {
   console.log(--a);
} while (a > 3);
```

Which statement can replace the do ... while from the example above?

Option A :

```
while (a >= 3)
   console.log(a--);
```

Option B :

```
while (a > 2)
   console.log(--a);
```

Option C :

```
while (a > 3)
    console.log(--a);
```

Option D :

```
while (a > 3)
   console.log(a--);
```

Correct Answer: C

Explanation/Reference:

Topics: while do while

Try it yourself:

```
let a = 10;
do {
   console.log(--a);  // 9 8 7 6 5 4 3
} while (a > 3);
console.log("a1:");
let a1 = 10;
while (a1 > 3)
  console.log(--a1);  // 9 8 7 6 5 4 3
console.log("a2:");
let a2 = 10;
```

```
while (a2 > 3)
  console.log(a2--);  // 10 9 8 7 6 5 4
console.log("a3:");
let a3 = 10;
while (a3 > 2)
  console.log(--a3);  // 9 8 7 6 5 4 3 2
console.log("a4:");
let a4 = 10;
while (a4 >= 3)
  console.log(a4--);  // 10 9 8 7 6 5 4 3
```

Explanation:

It is the while loop with the same condition

and the same content.

The while statement creates a loop (around a code block)

that is executed while a condition is true

https://www.w3schools.com/jsref/jsref_while.asp

The do while statements combo defines a code block to be executed once,

and repeated as long as a condition is true

The do while is used when you want to run a code block at least one time.

https://www.w3schools.com/jsref/jsref_dowhile.asp

QUESTION: 16

A recursive function is a function that:

Option A :

does not use global variables.

Option B :

does not return any value.

Option C :

uses loop instructions.

Option D :

calls itself again while running.

Correct Answer: D

Explanation/Reference:

Topic: recursive function

Try it yourself:

```
function factorial(n) {
    if (n == 1) return 1;
    return n * factorial(n -1);
}
console.log(factorial(5));  // 120
```

Explanation:

A function that calls itself is called a recursive function.

https://www.programiz.com/javascript/recursion

QUESTION: 17

We have initialized the name variable with the value "Alice"

then we try to write the number 100 into it.

let name = "Alice";

name = 100;

Option A :

the variable will retain the value "Alice"

(we cannot modify the value that the variable has been initialized with).

Option B :

the variable will contain the string "100"

Option C :

the program will be aborted due to an error

(we are trying to insert a value of a different type into the variable

than the value with which it was initialized).

Option D :

the variable will contain the number 100

Correct Answer: D

Explanation/Reference:

Topic: weak typing

Try it yourself:

```
let name = "Alice";

name = 100;

console.log(name);        // 100

console.log(typeof name);  // number
```

Explanation:

Javascript is weakly typed and therefore you can assign

different data types to the same variable.

https://en.wikipedia.org/wiki/Strong_and_weak_typing

QUESTION: 18

The string "1024" has been written into the str variable (let str = "1024";)

Then the operation str = -str is performed.

As a result, the variable str will contain:

Option A :

-1024

Option B :

NaN

Option C :

"-1024"

Option D :

"1024"

Correct Answer: A

Explanation/Reference:

Topics: unary negation operator

Try it yourself:

let str = "1024";

str = -str

console.log(str); // -1024

console.log(typeof str); // number

Explanation:

The unary negation operator can convert a non-number into a number.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Unary_negation

QUESTION: 19

Analyze the following code:

```
let id = "100";
{
   let id = 200;
   id = id + 1;
   console.log(id)
}
```

What will appear in the console as a result?

Option A :

1001

Option B :

201

Option C :

101

Option D :

200

Correct Answer: B

Explanation/Reference:

Topics: scope shadowing

Try it yourself:

```
let id = "100";
{
   let id = 200;
   id = id + 1;
   console.log(id) // 201
}
```

Explanation:

Shadowing

When a variable is declared in a certain scope

having the same name defined on its outer scope

and when we call the variable from the inner scope,

the value assigned to the variable in the inner scope

is the value that will be stored in the variable in the memory space.

https://www.geeksforgeeks.org/variable-shadowing-in-javascript

Scope

https://www.w3schools.com/js/js_scope.asp

QUESTION: 20

The minimum JavaScript online development environment will consist of:

(Select two correct answers)

Option A :

a forum for developers to exchange information only

Option B :

an online code editor

Option C :

an interactive page with JavaScript documentation

Option D :

the runtime environment

Correct Answer: B,D

Explanation/Reference:

Topics: code editor runtime environment

Explanation:

A source-code editor is a text editor program designed specifically

for editing source code of computer programs.

https://en.wikipedia.org/wiki/Source-code_editor

https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Available_text_editors

A runtime environment, primarily implements portions of an execution model.

https://en.wikipedia.org/wiki/Runtime_system

QUESTION: 21

Analyze the code snippet:

let n = 5 + 2 ** 2 * 3;

The result stored in the variable n is:

Option A :

42

Option B :

117649

Option C :

84

Option D :

17

Correct Answer: D

Explanation/Reference:

Topics: operator precedence exponentiation operator

addition operator multiplication operator

Try it yourself:

```
let n = 5 + 2 ** 2 * 3;

console.log(n);                // 17
```

```
console.log(5 + 2 ** 2 * 3);    // 17

console.log(5 + (2 ** 2) * 3);  // 17

console.log(5 + 4 * 3);         // 17

console.log(5 + (4 * 3));       // 17

console.log(5 + 12);            // 17

console.log(17);                // 17
```

Explanation:

The exponentiation operator has the highest operator precedence

and is executed first.

The multiplication operator has a higher operator precedence

than the addition operator and is executed next.

The addition operator is last.

The exponentiation operator returns the result

of raising the first operand to the power of the second operand.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Exponentiation

The addition operator produces the sum of numeric operands

or string concatenation.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Addition

The multiplication operator produces the product of the operands.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Multiplication

Operator precedence determines how operators

are parsed concerning each other.

Operators with higher precedence become the operands of operators

with lower precedence.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Operator_Precedence#table

QUESTION: 22

Handling an exception thrown by the interpreter with catch causes:

Option A :

the execution of the program to be automatically aborted,

but we can generate our own error message

(not necessarily in the console)

Option B :

further executing of the program to depend on the code

we place in the execution handler.

Option C :

the program to continue executing until the last instruction,

but we can generate our own error message

(not necessarily in the console).

Option D :

nothing to happen, because catch is not used for exception handling.

Correct Answer: B

Explanation/Reference:

Try it yourself:

// console.log(num);

// Uncaught ReferenceError: num is not defined

```
try {
    console.log(num);
} catch (error) {
    let num = 23;
    console.log(num);  // 23
}
```

Explanation:

If we use num without defining it first,

the interpreter would throw a ReferenceError

But if we do so inside of the try block,

the interpreter would catch that exception and execute the catch block.

And further executing of the program would depend on the code

inside of the catch block.

The try catch statements combo handles errors without stopping JavaScript.

https://www.w3schools.com/jsref/jsref_try_catch.asp

QUESTION: 23

What does shadowing mean?

Option A :

Declaring a local variable with the same name

as the previously declared global variable.

Option B :

Declaring a global variable with the same name

as a previously declared global variable.

Option C :

Deleting and rewriting a selected piece of program code.

Option D :

Changing the value of a variable.

Correct Answer: A

Explanation/Reference:

Try it yourself:

```
let x = 23;
{
   let x = 42;
   console.log(x);  // 42
}
console.log(x);  // 23
```

Explanation:

Shadowing

When a variable is declared in a certain scope

having the same name defined on its outer scope

and when we call the variable from the inner scope,

the value assigned to the variable in the inner scope

is the value that will be stored in the variable in the memory space.

https://www.geeksforgeeks.org/variable-shadowing-in-javascript

QUESTION: 24

Which sequence of if ... else statements is incorrect?

Option A :

if ... else if ...

Option B :

if ... else ...

Option C :

if ... else ... else ...

Option D :

if ... else if ... else ...

Correct Answer: C

Explanation/Reference:

Try it yourself:

```
let num;
num = Math.random();
console.log(num)  // e.g. 0.5882571338770821
/*
if (num < 0.34) console.log("Low");
else console.log("Middle");
```

```
else console.log("High");

// Uncaught SyntaxError: expected expression, got keyword 'else'

*/



if (num < 0.5) console.log("Low");

else console.log("High");

if (num < 0.5) console.log("Low");

else if (num >= 0.5) console.log("High");

if (num < 0.34) console.log("Low");

else if (num < 0.66) console.log("Middle");

else if (num >= 0.67) console.log("High");
```

Explanation:

You can have as many else if in the middle as you like,

but only one if at the beginning

and only one else at the ende.

The if else statement executes a block of code if a specified condition is true

If the condition is false another block of code can be executed.

https://www.w3schools.com/jsref/jsref_if.asp

QUESTION: 25

Analyze the following code:

```
const msg = "Hello";

msg += " world";

onsole.log(x);
```

What exception will be thrown as a result of its execution attempt?

Option A :

TypeError

Option B :

SyntaxError

Option C :

TypeError and ReferenceError

Option D :

ReferenceError

Correct Answer: A

Explanation/Reference:

Try it yourself:

const msg = "Hello";

// msg += " world";

// Uncaught TypeError: invalid assignment to const 'msg'

// onsole.log(x);

// Uncaught ReferenceError: onsole is not defined

console.log(x); // Uncaught ReferenceError: x is not defined

Explanation:

You can not reassign a constant

and therefore this script will throw a TypeError

If that wouldn't happen there would be two more ReferenceErrors

There is a typo in console and x was never defined.

But the first error is the TypeError

The TypeError object represents an error when an operation could not

be performed, typically when a value is not of the expected type.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/TypeError

QUESTION: 26

Analyze the following code:

```
let x = false || true;
let y = "true" && "false";
let z = false && true;
console.log(`${x} ${y} ${z}`);
```

What will appear in the console as a result of its execution?

Option A :

false false false

Option B :

true false false

Option C :

false false true

Option D :

false true true

Correct Answer: B

Explanation/Reference:

Topics: short circuit evaluation with AND

logical AND operator logical OR operator

Try it yourself:

```
let x = false || true;      // true

console.log(typeof x)       // boolean

let y = "true" && "false";  // false

console.log(typeof y)       // string

let z = false && true;      // true

console.log(typeof z)       // boolean

console.log(`${x} ${y} ${z}`);  // true  false  false
```

Explanation:

With x and z everything is standard.

But let's have a look at y

JavaScript has a short circuit evaluation.

For the logical AND operator that means,

if the first operand is truthy the second operand will be returned.

In this case the string "false"

The logical AND operator will be true if

and only if all the operands are true

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Logical_AND

The logical OR operator is true if

and only if one or more of its operands is true

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Logical_OR

Short circuit evaluation with AND

The logical AND operator returns the value of the first falsy operand

encountered when evaluating from left to right,

or the value of the last operand if they are all truthy.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Logical_AND#short-circuit_evaluation