




Explore

EN






JavaScript Essentials



JavaScript is one of the most popular and powerful language among developers. This course covers all the JavaScript essentials needed to create an interactive web ap...More

Start

Learning Progress

Status: Completed

 **Completion Certificate**



Overview

Contents

Discussions

What you will learn

Create interactive HTML web pages using JavaScript. Create web applications with modular code using Functions and Modules. Develop a web app including Object-Oriented Programming concepts such as Classes in JavaScript. Implement web app with validations in HTML document and handle the events. Implement web app with Asynchronous Programming using JavaScript Async and Await. Implement a secure web app by incorporating mitigations for common security vulnerabilities in JavaScript. Integrate JavaScript with HTML Using Web APIs like - XHR (for AJAX - Asynchronous

Authors/Creators

AS


Angad Singh Bindra


R


Raghavendra02


Show Curators/Contacts


At a glance


 Course


 19h 42m


 Beginner Level

 Free

 Infosys Wingspan

 EN

 JavaScript, JS, javascript, callbacks, Promises, DOM, Notification, Web Storage, node, scripting language, ES6, ecmaascript, WebAPI, AJAX, OOP, object oriented programming, Jasmine, Testing, Karma, async & await, JavaScript security



Thank you. Your test submitted.

You have cleared this assessment.

Obtained Percentage


Obtained Marks

68 %

17 / 25

Best Attempt Score:68 % on 15-02-2025

Review Your Attempt

 Sections	<div>Result: CLEARED</div> <div>Overall Percentage: 68% (17/25)</div>
<div>> JavaScript Essentials: Assessment</div>	<div>JavaScript Essentials: Assessment</div> <div>Section Score: 17/25</div> <div><div><div></div> Correct</div><div><div></div> Incorrect</div><div><div></div> Unattempted</div></div>

Q1 of 25

What will be the output of code snippet given below?

```
var lis = [  
  { alterEgo: 'Bruce Wayne', city: 'Gotham City' },  
  { alterEgo: 'Oliver Queen', city: 'Star City' },  
  { alterEgo: 'Barry Allen', city: 'Central City' }  
];  
var [, , darkKnight] = lis;  
function cityHunter({ alterEgo: x, heroName: y, powers: k, city: z }) {  
  document.write(x + '-' + y + '-' + k + '-' + z);  
}  
cityHunter(darkKnight);
```

- ☒ Barry Allen-undefined-undefined-Central City
- ☐ Bruce Wayne-null-null-Gotham City
- ☐ Error in Line 7 : Property mismatch
- ☐ Barry Allen-Central City

Q2 of 25

What will be printed in console on successful execution of the below code snippet?

```
function test() {  
  console.log(a);  
  console.log(foo());  
  var a = 1;  
}  
function foo() {  
  return 2;  
}  
test();
```

- ☐ This code will throw an error
- ☐ 2
- ☐ 1 2
- ☒ undefined 2

Q3 of 25

What will be the output in the console for the following code?

```
function man() {  
    power = 10;  
    console.log("Man's power is:",power);  
  
}  
function bat(spider) {  
    var power = 30;  
    try {  
        spider();  
    } catch(e) {  
        power = 20;  
    }  
    console.log("Bat's power is:",power);  
}  
function spider() {  
    bat(man);  
}  
spider();
```

- ☒ Man's power is: 10 Bat's power is: 30
- ☐ Man's power is: 10 Bat's power is: 10
- ☐ Man's power is: 30 Bat's power is: 10
- ☐ Man's power is: 30 Bat's power is: 30

Q4 of 25

What will be the output in the console for the following code?

```
var i=666;  
function myFunc() {  
  for(var i=1; i<10; i++) {  
    } console.log(i);  
  }  
  myFunc();
```

- ☐ Error
- ☒ 10
- ☐ 11
- ☐ 666

Q5 of 25

What will be the output on console when below code gets executed?

```
const myFunction = () => console.log("Hello ") || "World!";  
let returnValue = myFunction();  
console.log(returnValue);
```

- ☐ Compilation Error
- ☐ Hello
- ☐ World!
- ☒ Hello World!

Q6 of 25

What will be the output in the console for the following code?

```
function bat(man, maxPower) {  
  power = maxPower;  
  try {  
    man();  
  } finally {  
    let power = 400;  
  }  
  console.log(power);  
}  
function man() {  
  try {  
    throw new Error();  
    power = 100;  
  } catch(e) {  
    power = 200;  
  } finally {  
    power = 300;  
  }  
}  
bat(man, 100);
```

- ☐ 300
- ☐ 200
- ☐ 100
- ☒ 400

Q7 of 25

What will be the console output for the code given below?

```
function spider(man) {  
    console.log("Spider");  
    return man;  
}  
function man() {  
    console.log("Man");  
}  
function bat(man) {  
    console.log("Bat");  
    spider(man)();  
}  
bat(man);
```

- ☐ Bat Spider
- ☒ Bat Spider Man
- ☐ Bat Man Spider
- ☐ Bat Man

Q8 of 25

Can you predict the output of below-mentioned code?

```
const result = [];  
function check(par1, ...par2) {  
    return result.push(par1, par2);  
}  
console.log(check("Hello", "World"));
```

- ☐ undefined
- ☒ 2
- ☐ HelloWorld
- ☐ [Hello,World]

Q9 of 25

What will be the output in the console for the following code?

```
let power = 666;
function man() {
  power = 10;
  console.log("Man's power is:",power);
}
function bat(spider) {
  let power = 30;
  try {
    spider();
  } catch(e) {
    power = 20;
  }
  console.log("Bat's power is:",power);
}
function spider() {
  bat(man);
}
spider();
```

- ☐ Man's power is: 10 Bat's power is: 10
- ☐ Man's power is: 30 Bat's power is: 10
- ☒ Man's power is: 10 Bat's power is: 30
- ☐ Man's power is: 30 Bat's power is: 30

What will be the output in the console for the following code?

```
function bat(man, maxPower) {  
    power = maxPower;  
    try {  
        man();  
    } finally {  
        power = 400;  
    }  
    console.log(power);  
}  
function man() {  
    try {  
        throw new Error();  
        power = 100;  
    } catch(e) {  
        power = 200;  
    } finally {  
        power = 300;  
    }  
}  
bat(man, 100);
```

- ☐ 100
- ☐ 200
- ☐ 300
- ☒ 400

Q11 of 25

What will be the output of the code snippet given below?

```
var myList=["Edison","Tesla","Einstein"];  
var [,a]=myList;  
document.write(a);
```

- ☒ Error in Line 2
- ☐ Tesla
- ☐ Edison
- ☐ Einstein

Q12 of 25

What will be the output on console when below code gets executed?

```
var myNumber = 111;  
function updateMyNumber() {  
    var myNumber = 222;  
    myNumber = myNumber + 111;  
}  
updateMyNumber();  
console.log(myNumber);
```

- ☒ 222
- ☐ 111
- ☐ 333
- ☐ Compilation Error

Q13 of 25

Ravi has developed the below code to find any string in an array which contains "Script" in it. What will be the outcome of below code snippet?

```
let output = []
var component = (...par1) => {
  output = [...par1];
  return output.filter((a) => a.includes('Script'));
}
console.log(component('ES6', 'JavaScript', 'CoffeeScript', 'React'));
```

- ☐ Compilation Error
- ☐ undefined
- ☒ ['JavaScript', 'CoffeeScript']
- ☐ JavaScriptCoffeeScript

Q14 of 25

Observe the below code snippet and choose the appropriate option.

```
class Person {  
    constructor(friends) { this.friends = friends; }  
}  
class Employee extends Person {  
    constructor(friends) { super(friends) }  
    displayFriends() {  
        for (let friend of friends) {  
            console.log(friend)  
        }  
    }  
}  
let friendList = ["Alex", "John", "Chris", "Annie"];  
let x = new Employee(friendList);  
x.displayFriends();
```

- ☐ Error as displayFriends() method cannot use friends as it is defined in Base class - Person
- ☐ Error as friends is not defined in child class Employee
- ☒ Error as friends is not referred using this keyword in displayFriends() method.
- ☐ The code snippet will give the output as ["Alex", "John", "Chris", "Annie"]

Q15 of 25

Consider the below module definition in ES6:

```
// mymodule.js
export function a() {
  // some logic here
}
export function b() {
  // some logic here
}
```

- ☐ import * from "./mymodule"
- ☐ import * from ",/mymodule.js"
- ☐ import as "./mymodule"
- ☒ import * as mymodule from "./mymodule"

Q16 of 25

What will be the output of below code?

```
counts = [1, 2, 3, 4];  
counts.map(_ => Math.pow(_, 2));  
console.log(counts);
```

- ☐ [1, 4, 9, 16]
- ☐ []
- ☐ [1, 2, 3, 4]
- ☒ Compilation Error

Q17 of 25

Which of the followings is/are **TRUE** about ES6 Arrow function syntax?

- ☐ Curly Brackets { } are optional for multiple lines of code.
- ☒ Curly Brackets { } are mandatory for multiple lines of code.
- ☒ Parentheses () are optional for single parameter.
- ☐ Parentheses () are mandatory for single parameter.

Observe the code below:

```
class Person {
    constructor(name, age) { this.pname = name; this.page = age; }
    displayMasterDetails() {
        console.log("name :", this.pname);
        console.log("age :", this.page);
    }
}

class Employee extends Person {
    constructor(name, age, empid) {
        //.....(1).....
        this.empID = empid;
    }
    displayAllDetails() {
        //.....(2).....
        console.log("Employee Id : " + this.empID)
    }
}
```

What should be filled in blank (1) to make constructor call to base class- Person?

What should be filled in blank (2) to make displayMasterDetails() function call of base class Person in class Employee?

- ☒ blank 1 - super(name,age) , blank 2 - super.displayMasterDetails();
- ☐ blank 1 - super(name,age) , blank 2 - this.displayMasterDetails();
- ☐ blank 1 - super(name,age) , blank 2 - displayMasterDetails();
- ☐ blank 1 - super(name,age) , blank 2 - Person.displayMasterDetails();

Q19 of 25

What will be the output of the following snippet?

```
function emailFormat(email){
  try{
    if(email.match(/^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}$/)){
      console.log("Matching the email format!!")
      return true;
    }
    else{
      throw new Error("Email invalid!!");
    }
  }catch(error){
    console.log(error.message);
  }finally{
    console.log("Into the finally block!!");
  }
}
console.log(emailFormat("hello.123@example.com"));
```

- ☐ Matching the email format!! Into the finally block!!
- ☐ Email invalid!! Into the finally block!!
- ☒ Matching the email format!! Into the finally block!! true
- ☐ true Matching the email format!! Into the finally block!!

Q20 of 25

Which of the followings RegEx will properly validate mobile number? It should be of 10 digits and it should start with 6/7/8/9. (Select any two options)

- ☐ /^[6789]\d{9}\$/
- ☐ /^\d{10}\$/
- ☐ /^[^6-9]\d{9}\$/
- ☒ /^[6-9][0-9]{9}\$/

Q21 of 25

What will be the output in the console for the following code?

```
let i=666;  
function myFunc() {  
  for(let i=1; i<10; i++) {  
    } console.log(i);  
  }  
myFunc();
```

- ☒ Error
- ☐ 666
- ☐ 10
- ☐ 11

Q22 of 25

Which of the following regular expressions will be valid for the following list of words?

bunny, benny, bunt, bonny, binny, banny, benty

- ☒ /b[a-z](nn | nt)y/
- ☐ /b[a-z]+(nn | nt)^y/
- ☐ /b[a-z]?[a-z]*(nn | nt)\$y/
- ☐ /\$b[a-z]{1}(nt | nn)^y/

Q23 of 25

What will be the output in the console for the following code?

```
let power = 666;
function man() {
  power = 10;
  console.log("Man's power is:",power);
}
function bat(spider) {
  power = 30;
  try {
    spider();
  } catch(e) {
    power = 20;
  }
  console.log("Bat's power is:",power);
}
function spider() {
  bat(man);
}
spider();
```

- ☐ Man's power is: 10 Bat's power is: 10
- ☐ Man's power is: 20 Bat's power is: 10
- ☒ Man's power is: 10 Bat's power is: 20
- ☐ Man's power is: 30 Bat's power is: 10

Q24 of 25

Which of the following regular expressions will correctly match the string ABCDE1234F? (Select any two options)

- ☐ /^[A-Z]{5}[\d]{4}[A-Z]\$/
- ☐ /^[A-Z](5)[\d](4)[A-Z]?\$/
- ☐ /^[w]{5}[\d]{4}[A-Z]{1}\$/
- ☒ /^(A-Z){5}(0-9){4}(A-Z){1}\$/

Q25 of 25

What will be the output of the following snippet?

```
var str = "The best things3 are free in life";  
var pattern = new RegExp("[a-z]+[0-9]");  
var result = pattern.exec(str);  
console.log(result);
```

- ☐ true
- ☐ s3
- ☐ The best things3 are free in life
- ☒ things3