Introduction

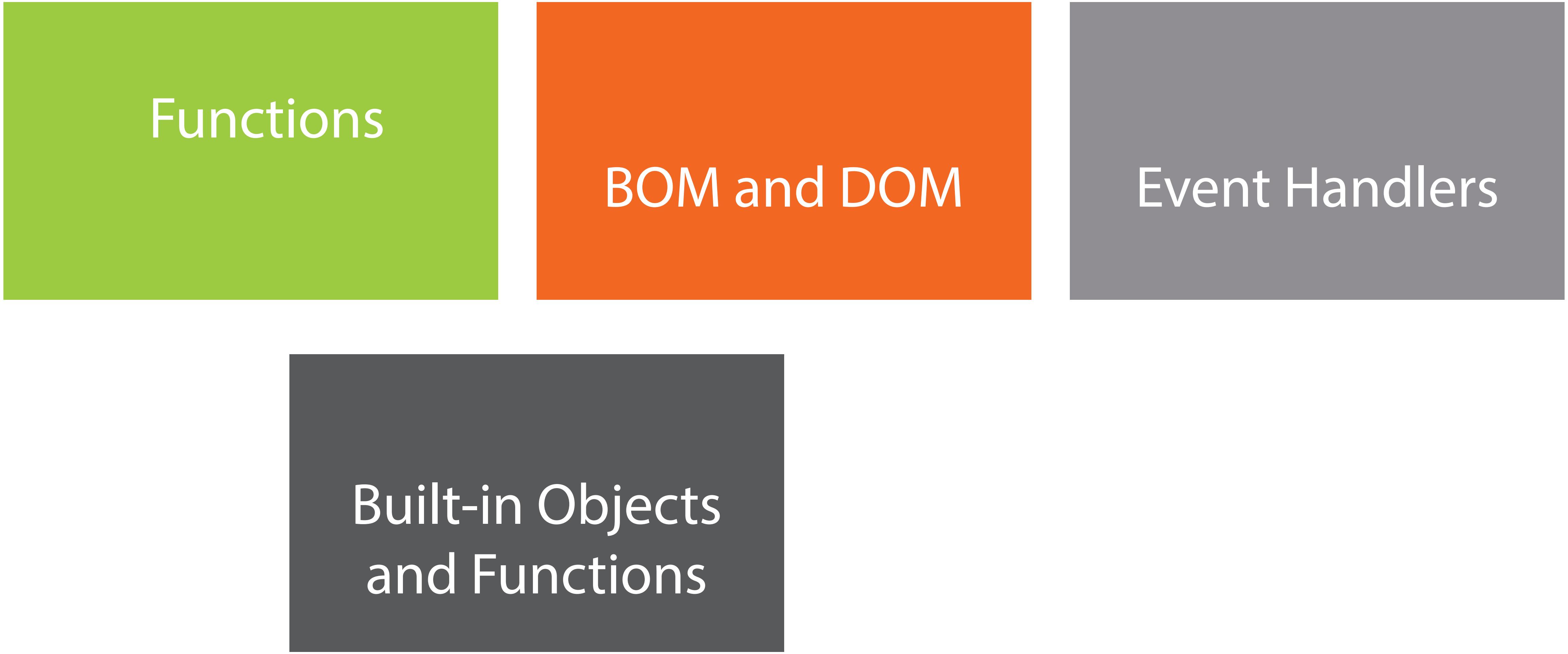
JavaScript Basics

Variables, Types and Scope

Operators

Arrays and Reference Types

Objects, JSON and Prototypes



Functions

BOM and DOM

Event Handlers

Built-in Objects and Functions

Module Overview

<script>

Variables and Types

Functions

Conditionals

Loops

JavaScript in Web Pages

<script>

src attribute

defer attribute

<noscript>

```
<!DOCTYPE html>
<html>
<head>
    <script>
        console.log('Hello!');
    </script>
</head>
<body>
</body>
</html>
```

Question

What shows in the console?

Answer

```
Hello!
```

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
    <h1>Our Page</h1>
    <script>
        console.log('In Body');
    </script>
</body>
</html>
```

Question

What shows in the console?

Answer

`In Body`

File: js/app.js

```
console.log('in app.js');
```

File: index.html

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
    <h1>Our Page</h1>
    <script src="js/app.js"></script>
</body>
</html>
```

Question

What shows in the console?

Answer

```
in app.js
```

File: js/app.js

```
console.log('in app.js');
```

File: index.html

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
    <h1>Our Page</h1>
    <script type="text/javascript" src="js/app.js">
        console.log('Hello!');
    </script>
</body>
</html>
```

Question

What shows in the console?

Answer

```
in app.js
```

File: js/app.js

```
console.log('in app.js');
```

File: index.html

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
    <h1>Our Page</h1>
    <script src="js/app.js" />
</body>
</html>
```

Question

What shows in the console?

Answer

`Unexpected end of file`

File: js/app.js

```
console.log('in app.js');
```

File: js/salesOrder.js

```
console.log('in salesOrder.js');
```

File: index.html

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
    <h1>Our Page</h1>
    <script src="js/app.js" ></script>
    <script src="js/salesOrder.js" ></script>
</body>
</html>
```

Question

What shows in the console?

Answer

```
in app.js
in salesOrder.js
```

File: js/app.js

```
console.log('in app.js');
```

File: js/salesOrder.js

```
console.log('in salesOrder.js');
```

File: index.html

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
    <h1>Our Page</h1>
    <script defer src="js/app.js" ></script>
    <script src="js/salesOrder.js" ></script>
</body>
</html>
```

Question

What shows in the console?

Answer

```
in salesOrder.js
in app.js
```

```html
<!DOCTYPE html>
<html>
<head>
</head>
<body>
    <noscript>
        <h1>This application requires
JavaScript.</h1>
    </noscript>
</body>
</html>
```

Question

What shows in the web page when JavaScript is turned off or isn’t available?

Answer

```
This application
requires JavaScript.
```

Variables

Fundamentals

undefined

Primitive Types

Object and Function Types

```
var orderId = 9001;
console.log(orderId);
```

Question

What shows in the console?

Answer

9001

```
var orderId = "ORD-9001";
console.log(orderId);
```

Question

What shows in the console?

Answer

```
ORD-9001
```

```
var orderId = "ORD-9001";
orderId = 1201;
console.log(orderId);
```

Question

What shows in the console?

Answer

```
1201
```

```
var orderId;
console.log(orderId);
```

Question

What shows in the console?

Answer

`undefined`

```
orderId = 9001;
console.log(orderId);
```

Question

What shows in the console?

Answer

```
9001
```

```
‘use strict’;
orderId = 9001;
console.log(orderId);
```

Question

What shows in the console?

Answer

```
Runtime Error: variable
undefined in strict
mode
```

```
var orderId = 9001;
console.log(typeof orderId);
```

Question

What shows in the console?

Answer

`number`

```
var orderId = 9001.01;
console.log(typeof orderId);
```

Question

What shows in the console?

Answer

`number`

```
var orderId = “ORD-9001”;
console.log(typeof orderId);
```

Question

What shows in the console?

Answer

```
string
```

```
var isActive = true;
console.log(typeof isActive);
```

Question

What shows in the console?

Answer

`boolean`

```
var isActive;
console.log(typeof isActive);
```

Question

What shows in the console?

Answer

```
undefined
```

```
var order = {
    orderId: 9001,
    isActive: true
};
console.log(typeof order);
```

Question

What shows in the console?

Answer

`object`

```
var cancelledOrders = [9001, 9002,
9003];
console.log(typeof cancelledOrders);
```

Question

What shows in the console?

Answer

`object`

```
var order = null;
console.log(typeof order);
```

Question

What shows in the console?

Answer

`object`

```
function cancelOrder(orderId) {
};

console.log(typeof cancelOrder);
```

Question

What shows in the console?

Answer

`function`

Function Fundamentals

Declaring Functions

Arguments

Returning Values

Function Expressions

```
function printOrder() {
    console.log(‘Printing order.’);
};

printOrder();
```

What shows in the console?

```
Printing order.
```

```
function printOrder(orderId) {
    console.log(‘Printing order: ’ + orderId);
};

printOrder(‘9002’);
```

Question

What shows in the console?

Answer

```
Printing order: 9002
```

```
printOrder(‘9002’);

function printOrder(orderId) {
    console.log(‘Printing order: ’ + orderId);
};
```

Question

What shows in the console?

Answer

```
Printing order: 9002
```

```
function calcTotalPrice(quantity, price) {
    console.log('Total price: ' + (quantity * price));
};

calcTotalPrice(2, 4.00);
```

Question

What shows in the console?

Answer

```
Total price: 8
```

```
function calcTotalPrice(quantity, price, currency)
{
    console.log(currency);
};

calcTotalPrice(2, 4.00);
```

Question

What shows in the console?

Answer

`undefined`

```
function calcTotalPrice(quantity, price)
{
    return quantity * price;
};

var totalPrice = calcTotalPrice(2, 4.00);
console.log(totalPrice);
```

Question

What shows in the console?

Answer

8

```
function calcTotalPrice(quantity, price) {
    return quantity * price;
};

var totalPrice = calcTotalPrice(2, 4.00);
console.log(totalPrice);

totalPrice = calcTotalPrice(3, 3.00);
console.log(totalPrice);
```

Question

What shows in the console?

Answer

```
8
9
```

```
function getOrder() {
    // nothing returned
};

var order = getOrder();
console.log(order);
```

Question

What shows in the console?

Answer

`undefined`

```
var activateOrder = function() {
    console.log(‘Order activated.’);
};

console.log(typeof activateOrder);
```

Question

What shows in the console?

Answer

`function`

```
var activateOrder = function() {
    console.log(‘Order activated.’);
};

activateOrder();
```

Question

What shows in the console?

Answer

```
Order activated.
```

```
var activateOrder = function() {
    console.log(‘Order activated.’);
};

activateOrder;
```

Question

What shows in the console?

Answer

`(nothing shows)`

If and Switch Statements

if

if ... else

switch ... case

Advanced cases

```
var total = 99.99;
var freeShipping = false;

if (total >= 50.00)
    freeShipping = true;

console.log(freeShipping);
```

Question

What shows in the console?

Answer

`true`

```
var total = 99.99;
var freeShipping = false;

if (total >= 100.00)
    freeShipping = true;

console.log(freeShipping);
```

Question

What shows in the console?

Answer

`false`

```
var total = 99.99;
var freeShipping;

if (total >= 100.00)
    freeShipping = true;
else
    freeShipping = false;

console.log(freeShipping);
```

Question

What shows in the console?

Answer

`false`

```
var total = 99.99;
var freeShipping;
var savings;

if (total >= 100.00) {
    freeShipping = true;
    savings = 29.99;
}
else {
    freeShipping = false;
    savings = 0;
}
console.log(savings);
```

Question

What shows in the console?

Answer

0

```
var total = 99.99;
var freeShipping;
var savings;

if (total >= 100.00) {
    freeShipping = true;
    savings = 29.99;
};
else {
    freeShipping = false;
    savings = 0;
}
console.log(savings);
```

Question

What shows in the console?

Answer

syntax error

```
var orderType = 'business';
var shipMethod;

if (orderType == 'business')
    shipMethod = 'FedEx';
else if (orderType == 'personal')
    shipMethod = 'UPS Ground';
else
    shipMethod = 'USPS';

console.log(shipMethod);
```

Question

What shows in the console?

Answer

FedEx

```
var orderType = 'business';
var shipMethod;

switch (orderType) {
    case ‘business’:
        shipMethod = 'FedEx';
        break;
    case 'personal‘:
        shipMethod = 'UPS Ground';
        break;
    default:
        shipMethod = 'USPS';
}
console.log(shipMethod);
```

Question

What shows in the console?

Answer

FedEx

```
var orderType = 'business';
var shipMethod;

switch (orderType) {
    case 'business':
        shipMethod = 'FedEx';
        // break;
    case 'personal':
        shipMethod = 'UPS Ground';
        break;
    default:
        shipMethod = 'USPS';
}
console.log(shipMethod);
```

Question

What shows in the console?

Answer

UPS Ground

```
var orderType = 'unknown';
var shipMethod;

switch (orderType) {
    case 'business':
    case 'personal':
        shipMethod = 'UPS Ground';
        break;
    default:
        shipMethod = 'USPS';
}
console.log(shipMethod);
```

Question

What shows in the console?

Answer

USPS

```
var orderType = 'unknown';
var shipMethod;

switch (orderType) {
    case 'business':
    case 'personal':
        shipMethod = 'UPS Ground';
        break;
}
console.log(shipMethod);
```

Question

What shows in the console?

Answer

`undefined`

```
var orderTotal = 99.99;
var discount;

switch (true) {
    case orderTotal >= 50 && orderTotal < 75:
        discount = 10;
        break;
    case orderTotal >= 75 && orderTotal < 100:
        discount = 20;
        break;
    case orderTotal >= 100:
        discount = 30;
        break;
    default:
        discount = 0;
}
console.log(discount);
```

Question

What shows in the console?

Answer

20

while and do...while Statements

while

do...while

```javascript
var lineItemCount = 3;
var currentItem = 0;
while (currentItem < lineItemCount) {
    console.log("item: " + currentItem);
    currentItem++;
}
```

Question

What shows in the console?

Answer

```
item: 0
item: 1
item: 2
```

```
var lineItemCount = 3;
var currentItem = 0;
while (currentItem < lineItemCount)
    console.log("item: " + currentItem++);
```

Question

What shows in the console?

Answer

```
item: 0
item: 1
item: 2
```

```
var lineItemCount = 3;
var currentItem = 0;
while (currentItem < lineItemCount)
    console.log("item: " + currentItem);
```

Question

What shows in the console?

Answer

```
item: 0
item: 0
item: 0
(endlessly)
```

```
var lineItemCount = 3;
var currentItem = 0;
do {
    console.log("item: " + currentItem);
    currentItem++;
} while (currentItem < lineItemCount);
```

Question

What shows in the console?

Answer

```
item: 0
item: 1
item: 2
```

```
var lineItemCount = 3;
var currentItem = undefined;
do {
    console.log("item: " + currentItem);
    currentItem++;
} while (currentItem < lineItemCount);
```

Question

What shows in the console?

Answer

```
item: undefined
```

for and for...in

for

break

continue

for...in

```javascript
var lineItemCount = 3;

for (var i = 0; i < lineItemCount; i++)
    console.log(i);
```

Question

What shows in the console?

Answer

```
0
1
2
```

```
var lineItemCount = 3;
var i = 0;

for (; i < lineItemCount; i++)
    console.log(i);
```

Question

What shows in the console?

Answer

```
0
1
2
```

```
var lineItemCount = 3;
var i = 0;

for (; i < lineItemCount;)
    console.log(i++);
```

Question

What shows in the console?

Answer

```
0
1
2
```

```
var lineItemCount = 5;

for (var i = 0; i < lineItemCount; i++) {
    console.log(i);
    if (i == 1)
        break;
}
```

Question

What shows in the console?

Answer

```
0
1
```

```
var lineItemCount = 5;

for (var i = 0; i < lineItemCount; i++) {
    if (i == 1)
        continue;
    console.log(i);
}
```

Question

What shows in the console?

Answer

```
0
2
3
4
```

```
var lineItemCount = 5;

outerLoop:
for (var i = 0; i < lineItemCount; i++) {
    for (var j = 0; j < 3; j++) {
        if (j == 1)
            continue outerLoop;
    }
    console.log(i);
}
```

Question

What shows in the console?

Answer

(nothing)

```
var lineItem = {
    product: 'Widget 1',
    quantity: 4,
    price: 9.50
};

for (var field in lineItem)
    console.log(field);
```

Question

What shows in the console?

Answer

```
product
quantity
price
```

```
var lineItem = {
    product: "Widget 1",
    quantity: 4,
    price: 9.50
};

for (var field in lineItem)
    console.log(field + " : " + lineItem[field]);
```

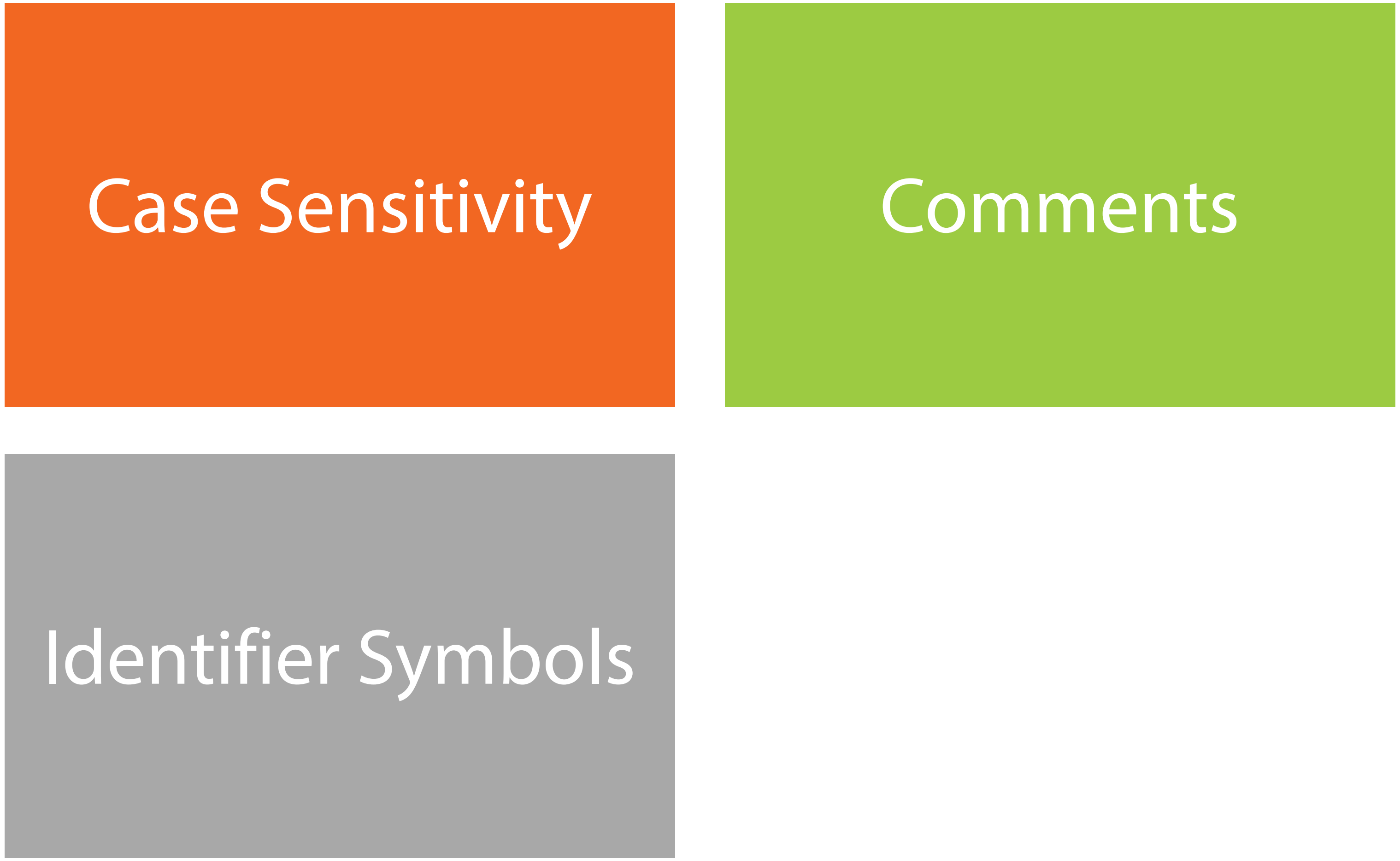
Question

What shows in the console?

Answer

```
product : Widget 1
quantity : 4
price : 9.50
```

JavaScript Usage Features



Case Sensitivity

Comments

Identifier Symbols

```
var orderId = 5005;
console.log(OrderId);
```

Question

What shows in the console?

Answer

```
ReferenceError: OrderId
is not defined
```

```
function submitOrder() {
    console.log('Submitting Order...');
}

submitorder();
```

Question

What shows in the console?

Answer

```
ReferenceError:
submitorder is not
defined
```

```
/*
console.log(1);
*/
console.log(2);
// console.log(3);
```

Question

What shows in the console?

Answer

2

```
/*
console.log( /*1*/);
*/
console.log(2);
// console.log(3);
```

Question

What shows in the console?

Answer

```
SyntaxError: Unexpected token )
```

```
var @product = 'PRD-2000';
console.log(@product);
```

Question

What shows in the console?

Answer

`SyntaxError`

```
var 2product = 'PRD-2000';
console.log(2product);
```

Question

What shows in the console?

Answer

`SyntaxError`

```
var $product = 'PRD-4000'
$product += 'X2'
console.log($product)
```

Question

What shows in the console?

Answer

```
PRD-4000X2
```

Summary

- <script>
- Variables and Types
- Functions
- Conditional Statements
- Loops
- JavaScript Usage

Module Overview

Hoisting

Primitive Types

undefined and null

Global Scope

Function Scope

Block Scope

Hoisting

Two Pass Compilation in JavaScript

Pass 1

All declarations are located and identifiers are known by the compiler.

Pass 2

Execution occurs in Pass 2. Because all variables are known from Pass 1, they may be accessed before they're declared.

Hoisting

The word "hoisting" is used to describe this behavior because it's as if the declarations are lifted (hoisted) to the top of the function before execution.

Hoisting

Hoisted variables are initialized to undefined.

```
console.log(productId);
```

Question

What shows in the console?

Answer

```
ReferenceError:
productId is not
defined
```

```
console.log(productId);

var productId = '9000';
```

Question

What shows in the console?

Answer

`undefined`

```
var total = price * quantity;
var price = 3.00;
var quantity = 5;

console.log(total);
```

Question

What shows in the console?

Answer

NaN

```
showProduct();

function showProduct() {
    console.log('Showing a Product');
}
```

Question

What shows in the console?

Answer

```
Showing a Product
```

```
showProduct();

var showProduct = function () {
    console.log('Showing a Product');
}
```

Question

What shows in the console?

Answer

```
TypeError: showProduct
is not a function
```

```
var showProduct = function () {
    console.log('Showing a Product');
}

showProduct();
```

Question

What shows in the console?

Answer

```
Showing a Product
```

```
showProduct();

function showProduct() {
    console.log('Showing a Product');
}
```

Question

What shows in the console?

Answer

```
Showing a Product
```

```
var quantity = 2;
var price = 0.99;
console.log(quantity * price);
```

Question

What shows in the console?

Answer

```
1.98
```

```
var price = "0.99";
console.log(typeof price);
```

Question

What shows in the console?

Answer

`string`

```
var oct = 070;
console.log(oct);
```

Question

What shows in the console?

Answer

56

```
var oct = 088;
console.log(oct);
```

Question

What shows in the console?

Answer

88

```
'use strict';
var oct = 070;
console.log(oct);
```

Question

What shows in the console?

Answer

`SyntaxError`

```
var hex = 0x10;
console.log(hex);
```

Question

What shows in the console?

Answer

16

```
var value = 3.2e4;
console.log(value);
```

Question

What shows in the console?

Answer

```
32000
```

```
var value = 3.2e-4;
console.log(value);
```

Question

What shows in the console?

Answer

```
0.00032
```

```
console.log(Number.MAX_VALUE);
```

Question

What shows in the console?

Answer

```
1.7976931348623157e+308
```

```
console.log(Number.MIN_VALUE);
```

Question

What shows in the console?

Answer

```
5e-324
```

```
var n = Number.MAX_VALUE * 2;
console.log(n);
```

Question

What shows in the console?

Answer

`Infinity`

```
var n = Infinity;
console.log(typeof n);
```

Question

What shows in the console?

Answer

`number`

```
var n = -Infinity;
console.log(n);
```

Question

What shows in the console?

Answer

```
-Infinity
```

```
console.log(Number.POSITIVE_INFINITY + ' ' +
            Number.NEGATIVE_INFINITY);
```

Question

What shows in the console?

Answer

```
Infinity -Infinity
```

```
var count;
var price = 5.00;

console.log(price / count);
```

Question

What shows in the console?

Answer

NaN

```
console.log(typeof NaN);
```

Question

What shows in the console?

Answer

```
number
```

```
var n = NaN;
if (isNaN(n))
    console.log('NaN!');
```

Question

What shows in the console?

Answer

NaN!

```
console.log("Hello" + ' world!');
```

Question

What shows in the console?

Answer

```
Hello world!
```

```
console.log("The order is \"In Progress\"");
```

Question

What shows in the console?

Answer

```
The order is "In
Progress"
```

```javascript
console.log('Nike\'s Order');
```

Question

What shows in the console?

Answer

```
Nike's Order
```

```
var productType = 'Hardware';
console.log(productType.length);
```

Question

What shows in the console?

Answer

8

```
var productType = '  Hardware   ';
console.log('[' + productType.trim() + ']');
```

Question

What shows in the console?

Answer

```
[Hardware]
```

```
var productType = '  Hardware   ';
productType.trim();
console.log('[' + productType + ']');
```

Question

What shows in the console?

Answer

```
[  Hardware  ]
```

```
var productType = 'Hardware';
console.log(productType.toLowerCase());
```

Question

What shows in the console?

Answer

```
hardware
```

```
var productType = 'Hardware';
console.log(productType.toUpperCase());
```

Question

What shows in the console?

Answer

HARDWARE

```
var isSpecial = true;
if (isSpecial)
    console.log("It's true!");
```

Question

What shows in the console?

Answer

```
It’s true!
```

```
var isSpecial = True;
if (isSpecial)
    console.log("It's true!");
```

Question

What shows in the console?

Answer

```
ReferenceError: True is
not defined
```

```
console.log( Boolean("Hello") );
```

Question

What shows in the console?

Answer

```
true
```

```
console.log( Boolean("") );
```

Question

What shows in the console?

Answer

```
false
```

```
console.log( Boolean(" ") );
```

Question

What shows in the console?

Answer

```
true
```

```
console.log( Boolean(6) );
```

Question

What shows in the console?

Answer

```
true
```

```
console.log( Boolean(0) );
```

Question

What shows in the console?

Answer

```
false
```

```
console.log( Boolean(undefined) );
```

Question

What shows in the console?

Answer

```
false
```

```
console.log( Boolean(null) );
```

Question

What shows in the console?

Answer

```
false
```

```
var value = 99.99;
console.log(typeof !!value );
```

Question

What shows in the console?

Answer

`boolean`

```
var value = "no value";
console.log( !!value );
```

Question

What shows in the console?

Answer

```
true
```

undefined and null

```
var productId;

console.log(typeof productId);
```

Question

What shows in the console?

Answer

`undefined`

```
//var productId;

console.log(typeof productId);
```

Question

What shows in the console?

Answer

undefined

```
var productId = null;

console.log(typeof productId);
```

Question

What shows in the console?

Answer

`object`

```
console.log(typeof null);
```

Question

What shows in the console?

Answer

```
object
```

```
console.log(undefined == null);
```

Question

What shows in the console?

Answer

```
true
```

```
console.log(undefined === null);
```

Question

What shows in the console?

Answer

```
false
```

Global Scope

```
var productId = 'PRD-2000';

console.log(productId);
console.log(window.productId);
```

Question

What shows in the console?

Answer

```
PRD-2000

PRD-2000
```

```
console.log(this === window);
```

Question

What shows in the console?

Answer

```
true
```

```
var description = 'hardware product';

function updateProduct() {
    description = 'updated product';
}

updateProduct();
console.log(description);
```

Question

What shows in the console?

Answer

```
updated product
```

```
//var description = 'hardware product';

function updateProduct() {
    description = 'updated product';
}

updateProduct();
console.log(description);
```

Question

What shows in the console?

Answer

```
updated product
```

```
'use strict';

//var description = 'hardware product';

function updateProduct() {
    description = 'updated product';
}

updateProduct();
console.log(description);
```

Question

What shows in the console?

Answer

```
ReferenceError:
description is not
defined
```

Function Scope

```
var description = 'original product';

function updateProduct() {
    var description = 'updated product';
}

updateProduct();
console.log(description);
```

Question

What shows in the console?

Answer

`original product`

```
function updateProduct() {

    var description = 'updated product';
    var updateProductId = function () {
        console.log(description);
    }

    updateProductId();
}

updateProduct();
```

Question

What shows in the console?

Answer

```
updated product
```

```
var description = 'original product';

function updateProduct() {
    var description = 'updated product';
    var updateProductId = function () {
        console.log(description);
    }

    updateProductId();
}

updateProduct();
```

Question

What shows in the console?

Answer

updated product

```
var description = 'original product';

function updateProduct() {

    var updateProductId = function () {
        console.log(description);
    }

    updateProductId();
}

updateProduct();
```

Question

What shows in the console?

Answer

```
original product
```

```
// var description = 'original product';

function updateProduct() {

    var updateProductId = function () {
        console.log(description);
    }

    updateProductId();
}

updateProduct();
```

Question

What shows in the console?

Answer

```
ReferenceError:
description is not
defined
```

Block Scope

```
'use strict';

try {
    throw 123;
}
catch (e) {
    console.log(e);
}
```

Question

What shows in the console?

Answer

123

```
'use strict';
console.log(e);
try {
    throw 123;
}
catch (e) {
}
```

Question

What shows in the console?

Answer

```
ReferenceError: e is
not defined
```

```
'use strict';

try {
    console.log('in try:' + e);
    throw 123;
}
catch (e) {
    console.log(e);
}
```

Question

What shows in the console?

Answer

```
ReferenceError: e is
not defined
```

```
'use strict';

console.log(insideCatch);
try {
    throw 123;
}
catch (e) {
    var insideCatch = 'insideCatch';
    console.log(e);
}
```

Question

What shows in the console?

Answer

```
undefined

123
```

```
'use strict';
try {
    throw 'myArtificialBlockLevelVar'
}
catch (blockLevelVar) {
    // blockLevlVar is in block scope
    // …
};

console.log(blockLevelVar);
```

Question

What shows in the console?

Answer

```
ReferenceError:
blockLevelVar is not
defined
```

Summary

- Hoisting
- Primitive Types (number, string and boolean)
- undefined and null
- Global Scope
- Function Scope
- Block Scope

Module Overview

+ - * / %

Unary

Bitwise

Boolean

Equality

Relational

```
var total = 5 + 3;
console.log(total);
```

Question

What shows in the console?

Answer

8

```
var total = 5.1 + 3.3;
console.log(total);
```

Question

What shows in the console?

Answer

```
8.39999999999999
```

```
var total = 5.1 + 3.3;
console.log(total.toFixed(2));
```

Question

What shows in the console?

Answer

```
8.40
```

```
var id = "PRD" + "X99";
console.log(id);
```

Question

What shows in the console?

Answer

PRDX99

```
var id = "PRD" + 2000;
console.log(id);
```

Question

What shows in the console?

Answer

```
PRD2000
```

```
var id = 2000 + "PRD";
console.log(id);
```

Question

What shows in the console?

Answer

```
2000PRD
```

```
var id = "2000" + "500";
console.log(id);
```

Question

What shows in the console?

Answer

```
2000500
```

```
var id = 2000 + "500";
console.log(id);
```

Question

What shows in the console?

Answer

```
2000500
```

```
var id = 2000 + undefined;
console.log(id);
```

Question

What shows in the console?

Answer

NaN

```
var id = "PRD" + undefined;
console.log(id);
```

Question

What shows in the console?

Answer

```
PRDundefined
```

```
var id = 2000 + null;
console.log(id);
```

Question

What shows in the console?

Answer

```
2000
```

```
var id = "PRD" + null;
console.log(id);
```

Question

What shows in the console?

Answer

```
PRDnull
```

```
var id = 2000 + NaN;
console.log(id);
```

Question

What shows in the console?

Answer

NaN

```
var id = "PRD" + NaN;
console.log(id);
```

Question

What shows in the console?

Answer

PRDNaN

```
var total = 30 - 20;
console.log(total);
```

Question

What shows in the console?

Answer

```
10
```

```
var total = 3.8 - 2.1;
console.log(total);
```

Question

What shows in the console?

Answer

```
1.6999999999999997
```

```
var total = 3.8 - 2.1;
console.log(total.toFixed(3));
```

Question

What shows in the console?

Answer

```
1.700
```

```
var total = "300" - "200";
console.log(total);
```

Question

What shows in the console?

Answer

```
100
```

```
var total = "PRD300" - "ITEM200";
console.log(total);
```

Question

What shows in the console?

Answer

NaN

```
var total = 300 - undefined;
console.log(total);
```

Question

What shows in the console?

Answer

NaN

```
var total = 300 - null;
console.log(total);
```

Question

What shows in the console?

Answer

```
300
```

```
var obj = {
    valueOf: function () { return 100; }
};
var total = 300 - obj;
console.log(total);
```

Question

What shows in the console?

Answer

200

```
var total = 300 - NaN;
console.log(total);
```

Question

What shows in the console?

Answer

NaN

```
var total = 300 - "";
console.log(total);
```

Question

What shows in the console?

Answer

```
300
```

```
var total = 9 * 3;
console.log(total);
```

Question

What shows in the console?

Answer

27

```
var total = 1.1 * 1.1;
console.log(total);
```

Question

What shows in the console?

Answer

```
1.2100000000000002
```

```
var total = 1.1 * 1.1;
console.log(total);
if (total != 1.21)
    console.log('Big Problem!');
```

Question

What shows in the console?

Answer

```
1.2100000000000002

Big Problem!
```

```
var total = 1.1 * 1.1;
console.log(total);
if (total.toFixed(2) == 1.21)
    console.log('Fixed!');
```

Question

What shows in the console?

Answer

```
1.2100000000000002

Fixed!
```

```
var total = 20 * NaN;
console.log(total);
```

Question

What shows in the console?

Answer

NaN

```
var total = 20 * undefined;
console.log(total);
```

Question

What shows in the console?

Answer

NaN

```
var total = 20 * null;
console.log(total);
```

Question

What shows in the console?

Answer

```
0
```

```
var obj = {
    valueOf: function () { return 100; }
};
var total = 4 * obj;
console.log(total);
```

Question

What shows in the console?

Answer

```
400
```

```
var total = 4 * Infinity;
console.log(total);
```

Question

What shows in the console?

Answer

```
Infinity
```

```
var total = -4 * Infinity;
console.log(total);
```

Question

What shows in the console?

Answer

```
-Infinity
```

```
var total = 4 * "XYZ";
console.log(total);
```

Question

What shows in the console?

Answer

NaN

```
var total = 9 / 0;
console.log(total);
```

Question

What shows in the console?

Answer

```
Infinity
```

```
var total = -9 / 0;
console.log(total);
```

Question

What shows in the console?

Answer

```
-Infinity
```

```
var total = 9 / "3";
console.log(total);
```

Question

What shows in the console?

Answer

3

```
var total = 9 % 4;
console.log(total);
```

Question

What shows in the console?

Answer

1

```
var total = 9 % "   4   ";
console.log(total);
```

Question

What shows in the console?

Answer

1

Unary Operators

++   --   +value   -value

```
var level = 5;
++level;
console.log(level);
```

Question

What shows in the console?

Answer

6

```
var level = 5;
level++;
console.log(level);
```

Question

What shows in the console?

Answer

6

```
var level = 5;
var nextLevel = ++level;
console.log(nextLevel);
```

Question

What shows in the console?

Answer

6

```
var level = 5;
var nextLevel = level++;
console.log(nextLevel);
```

Question

What shows in the console?

Answer

5

```
var level = 5;
var nextLevel = --level;
console.log(nextLevel);
```

Question

What shows in the console?

Answer

4

```
var level = 5;
var base = 10;
var nextLevel = level-- * base;
console.log(nextLevel);
```

Question

What shows in the console?

Answer

```
50
```

```
var level = 5;
var base = 10;
var nextLevel = --level * base;
console.log(nextLevel);
```

Question

What shows in the console?

Answer

```
40
```

```
var level;
console.log(++level);
```

Question

What shows in the console?

Answer

NaN

```
var level = null;
console.log(++level);
```

Question

What shows in the console?

Answer

1

```
var value = 9;
value = +value;
console.log(value);
```

Question

What shows in the console?

Answer

9

```
var value = -9;
value = +value;
console.log(value);
```

Question

What shows in the console?

Answer

```
-9
```

```
var value = 'calc';
value = +value;
console.log(value);
```

Question

What shows in the console?

Answer

NaN

```
var value = 42;
value = -value;
console.log(value);
```

Question

What shows in the console?

Answer

```
-42
```

```
var value = -42;
value = -value;
console.log(value);
```

Question

What shows in the console?

Answer

42

```
var value = 'calc';
value = -value;
console.log(value);
```

Question

What shows in the console?

Answer

NaN

Bitwise Operators

```
var num1 = parseInt('1010', 2);
var num2 = parseInt('0110', 2);
var total = num1 & num2;
console.log(total.toString(2));
```

Question

What shows in the console?

Answer

```
10
```

```
var num1 = parseInt('1010', 2);
var num2 = parseInt('0110', 2);
var total = num1 | num2;
console.log(total.toString(2));
```

Question

What shows in the console?

Answer

```
1110
```

```
var num1 = parseInt('1010', 2);
var num2 = parseInt('0110', 2);
var total = num1 ^ num2;
console.log(total.toString(2));
```

Question

What shows in the console?

Answer

```
1100
```

```
var num1 = parseInt('0001', 2);
num1 = num1 << 2;
console.log(num1.toString(2));
```

Question

What shows in the console?

Answer

```
100
```

```
var num1 = parseInt('1000', 2);
num1 = num1 >> 3;
console.log(num1.toString(2));
```

Question

What shows in the console?

Answer

1

```
var num1 = -parseInt('1000', 2);
num1 = num1 >> 3;
console.log(num1.toString(2));
```

Question

What shows in the console?

Answer

-1

```
var num1 = parseInt('1000', 2);
num1 = num1 >>> 1;
console.log(num1.toString(2));
```

Question

What shows in the console?

Answer

```
100
```

```
var num1 = -parseInt('1000', 2);
num1 = num1 >>> 1;
console.log(num1.toString(2));
```

Question

What shows in the console?

Answer

```
11111111111111111111111
11111100
```

Boolean Operators

```
var value = !true;
console.log(value);
```

Question

What shows in the console?

Answer

```
false
```

```
var value = !false;
console.log(value);
```

Question

What shows in the console?

Answer

```
true
```

```
var value = !99;
console.log(value);
```

Question

What shows in the console?

Answer

`false`

```
var value = !0;
console.log(value);
```

Question

What shows in the console?

Answer

```
true
```

```
var value = !"";
console.log(value);
```

Question

What shows in the console?

Answer

```
true
```

```
var value = !"A";
console.log(value);
```

Question

What shows in the console?

Answer

```
false
```

```
var value = !" ";
console.log(value);
```

Question

What shows in the console?

Answer

`false`

```
var value = !new Object();
console.log(value);
```

Question

What shows in the console?

Answer

```
false
```

```
var value = !null;
console.log(value);
```

Question

What shows in the console?

Answer

```
true
```

```
var value = !undefined;
console.log(value);
```

Question

What shows in the console?

Answer

```
true
```

```
var value = !NaN;
console.log(value);
```

Question

What shows in the console?

Answer

```
true
```

```
var value = !!false;
console.log(value);
```

Question

What shows in the console?

Answer

```
false
```

```
var value = true && true;
console.log(value);
```

Question

What shows in the console?

Answer

```
true
```

```
var value = true && false;
console.log(value);
```

Question

What shows in the console?

Answer

`false`

```
var value = false && true;
console.log(value);
```

Question

What shows in the console?

Answer

```
false
```

```
var value = false && false;
console.log(value);
```

Question

What shows in the console?

Answer

```
false
```

```
var obj = {
    calc: 'Logical AND'
};
var value = obj && 99;
console.log(value);
```

Question

What shows in the console?

Answer

99

```
var obj = {
    calc: 'Logical AND'
};
var value = obj && 0;
console.log(value);
```

Question

What shows in the console?

Answer

```
0
```

```
var obj = {
    calc: 'Logical AND'
};
var value = obj && obj;
console.log(value);
```

Question

What shows in the console?

Answer

```
Object {calc: "Logical
AND"}
```

```
var obj = {
    calc: 'Logical AND'
};
var value = true && obj;
console.log(value);
```

Question

What shows in the console?

Answer

```
Object {calc: "Logical
AND"}
```

```
var value = null && 99;
console.log(value);
```

Question

What shows in the console?

Answer

```
null
```

```
var value = "Z" && null;
console.log(value);
```

Question

What shows in the console?

Answer

```
null
```

```
var value = "Z" && undefined;
console.log(value);
```

Question

What shows in the console?

Answer

undefined

```
var value = "Z" && NaN;
console.log(value);
```

Question

What shows in the console?

Answer

NaN

```
var value = false && productId;
console.log(value);
```

Question

What shows in the console?

Answer

```
false
```

```
var productId = 5;
var value = false && ++productId;
console.log(productId);
```

Question

What shows in the console?

Answer

5

```
var value = true || true;
console.log(value);
```

Question

What shows in the console?

Answer

`true`

```
var value = true || false;
console.log(value);
```

Question

What shows in the console?

Answer

`true`

```
var value = false || true;
console.log(value);
```

Question

What shows in the console?

Answer

```
true
```

```
var value = false || false;
console.log(value);
```

Question

What shows in the console?

Answer

```
false
```

```
var obj = {
    calc: 'Logical OR'
};
var value = obj || 99;
console.log(value);
```

Question

What shows in the console?

Answer

```
Object {calc: "Logical
OR"}
```

```
var obj = {
    calc: 'Logical OR'
};
var value = false || 99;
console.log(value);
```

Question

What shows in the console?

Answer

99

```
var defaultSettings = {
    who: 'default'
};
var userSettings = {
    who: 'user'
};
var value = userSettings || defaultSettings;
console.log(value);
```

Question

What shows in the console?

Answer

```
Object {who: "user"}
```

```
var defaultSettings = {
    who: 'default'
};
var userSettings;
var value = userSettings || defaultSettings;
console.log(value);
```

Question

What shows in the console?

Answer

```
Object {who: "default"}
```

Equality Operators

== !== === !===

```
if (true == 1)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
true
```

```
if (true == 2)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
false
```

```
if (false == 0)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
true
```

```
if (42 == '42')
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
true
```

```
var obj = {
    valueOf: function () { return 42; }
}
if (42 == obj)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
true
```

```
var obj = {
    toString: function () { return 42; }
}
if (42 == obj)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
true
```

```
var obj = {
    name: 'Trigger'
};
var obj2 = {
    name: 'Trigger'
};
if (obj == obj2)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
false
```

```
var obj = {
    name: 'Trigger'
};
if (obj == obj)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
true
```

```
var obj = {
    name: 'Trigger'
};
var obj2 = obj;
if (obj == obj2)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
true
```

```
if (null == undefined)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
true
```

```
if (undefined == 0)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
false
```

```
if (null == 0)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
false
```

```
if (NaN == NaN)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
false
```

```
if (42 != 'forty-two')
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
true
```

```
if (55 === '55')
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

false

```
if (55 === 55)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
true
```

```
if (false === 0)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
false
```

```
if (null === undefined)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
false
```

```
if (55 !== 55)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
false
```

Relational Operators

< <= > >=

```
if (55 < 66)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

`true`

```
if (55 <= 55)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
true
```

```
if (55 > 55)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
false
```

```
if (55 >= 55)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
true
```

```
if ("alpha" < "beta")
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
true
```

```
if ("alpha" < "alph")
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
false
```

```
if ("beta" < "Alpha")
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
false
```

```
if ("Beta" < "alpha")
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
true
```

```
if ("Beta".toLowerCase() < "alpha".toLowerCase())
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
false
```

```
if ('42' < 55)
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
true
```

```
if ('42' < '142')
    console.log('true');
else
    console.log('false');
```

Question

What shows in the console?

Answer

```
false
```

```
var total = NaN;
if (total < 5)
    console.log('less than 5');
if (total >= 5)
    console.log('greater than or equal to 5');
```

Question

What shows in the console?

Answer

(nothing)

```
var total = 6;
total += 4;
console.log(total);
```

Question

What shows in the console?

Answer

```
10
```

```
var total = 6;
total -= 4;
console.log(total);
```

Question

What shows in the console?

Answer

2

```
var total = 6;
total *= 4;
console.log(total);
```

Question

What shows in the console?

Answer

24

```
var total = 6;
total /= 3;
console.log(total);
```

Question

What shows in the console?

Answer

2

```
var total = 6;
total %= 4;
console.log(total);
```

Question

What shows in the console?

Answer

2

```
var total = 1;
total <<= 2;
console.log(total);
```

Question

What shows in the console?

Answer

4

```
var total = 4;
total >>= 1;
console.log(total);
```

Question

What shows in the console?

Answer

2

```
var total = 6;
total *= 4 + 1;
console.log(total);
```

Question

What shows in the console?

Answer

```
30
```

```
var total = 99, tax = 9;
console.log(total);
console.log(tax);
```

Question

What shows in the console?

Answer

```
99

9
```

```
var total = (99, 88, 44);
console.log(total);
```

Question

What shows in the console?

Answer

44

Summary

- Basic Arithmetic Operators: + - * / %
- Unary
- Bitwise
- Boolean
- Equality
- Relational
- Miscellaneous

Overview

Reference Types Explained

Reference Type Examples

Arrays

Dates

Regular Expressions

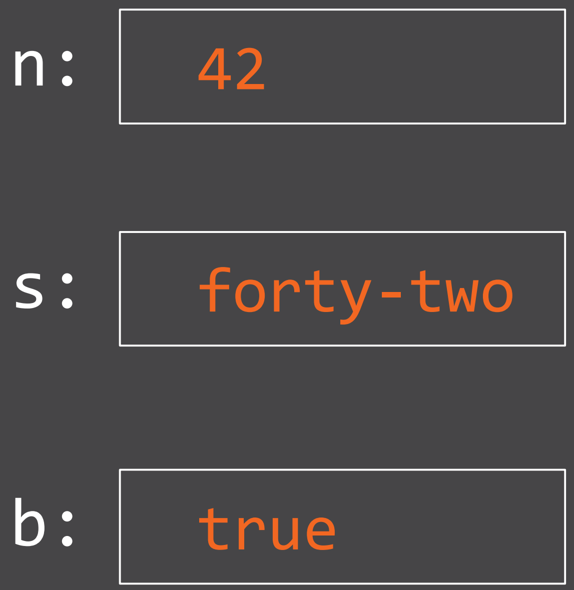
Understanding Reference Types

Primitive Type Storage

```
var n = 42;

var s = "forty-two";

var b = true;
```



n: 42

s: forty-two

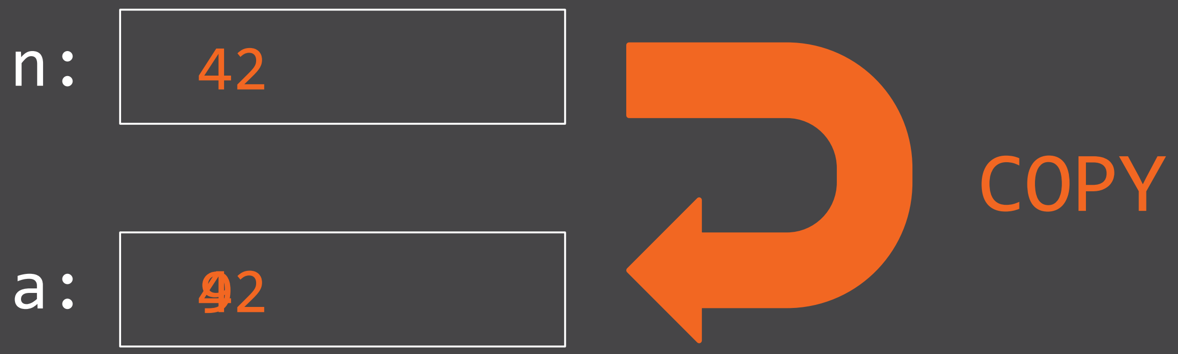
b: true

Primitive Type Storage

```
var n = 42;

var a = n;

a = 9;
```

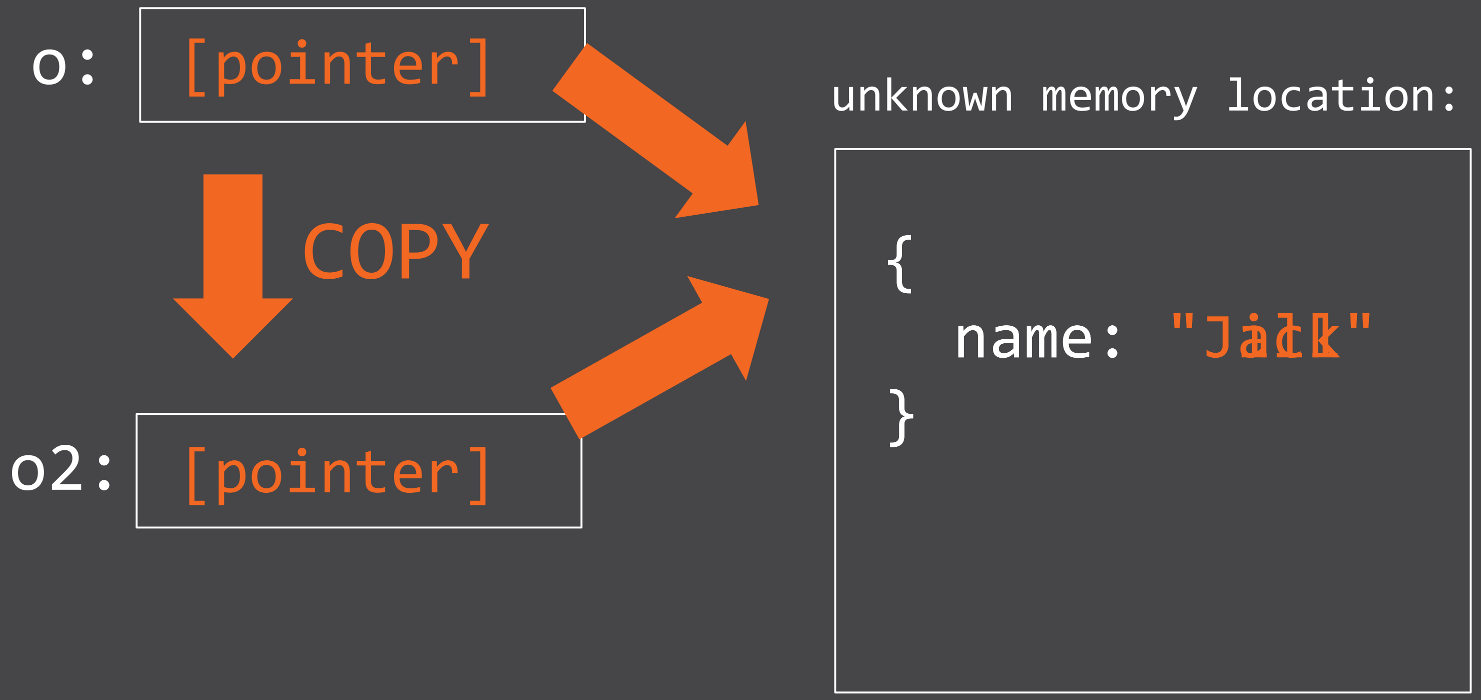


Reference Type Storage

```
var o = {
    name: "Jill"
};


var o2 = o;


o2.name = "Jack";
```



o: [pointer]

COPY

o2: [pointer]

unknown memory location:

```
{
   name: "Jack"
}
```

Common Reference Types

- Objects
- Arrays
- Dates
- RegExp's
- Functions
- Primitives (via Wrappers)

Reference Type Examples

```
var blog = {
    name: 'Ski Utah'
};
var updatedBlog = blog;
console.log(updatedBlog.name);
```

Question

What shows in the console?

Answer

`Ski Utah`

```
var blog = {
    name: 'Ski Utah'
};
var updatedBlog = blog;
blog.name = 'Rocky Mountain Skiing';
console.log(updatedBlog.name);
```

Question

What shows in the console?

Answer

```
Rocky Mountain Skiing
```

```
var blog = {
    name: 'Ski Utah'
};
var updatedBlog = blog;
updatedBlog.name = 'Rocky Mountain Skiing';
console.log(blog.name);
```

Question

What shows in the console?

Answer

```
Rocky Mountain Skiing
```

```
var blog = {
    name: 'Ski Utah'
};
function changeBlogName(localBlog) {
    localBlog.name = 'no name';
}
changeBlogName(blog);
console.log(blog.name);
```

Question

What shows in the console?

Answer

`no name`

```
var a1 = [1, 2, 3];
var a2 = a1;
a1[0] = 99;
console.log(a2[0]);
```

Question

What shows in the console?

Answer

99

Array Fundamentals

```
var entries = new Array('Trains', 'Plains',
    'Automobiles');
console.log(entries instanceof Array);
```

Question

What shows in the console?

Answer

```
true
```

```
var entries = new Array('Trains', 42, true);
console.log(entries instanceof Array);
```

Question

What shows in the console?

Answer

```
true
```

```
var entries = Array('Trains', 'Plains',
    'Automobiles');
console.log(entries instanceof Array);
```

Question

What shows in the console?

Answer

```
true
```

```
var entries = ['Trains', 'Plains',
    'Automobiles'];
console.log(entries instanceof Array);
```

Question

What shows in the console?

Answer

```
true
```

```
var entries = ['Trains', 'Plains',
    'Automobiles'];
console.log(entries[0]);
```

Question

What shows in the console?

Answer

Trains

```
var entries = ['Trains', 'Plains',
    'Automobiles'];
console.log(entries[2]);
```

Question

What shows in the console?

Answer

```
Automobiles
```

```
var entries = ['Trains', 'Plains',
    'Automobiles'];
console.log(entries[3]);
```

Question

What shows in the console?

Answer

`undefined`

```
var entries = new Array(5);
console.log(entries.length);
```

Question

What shows in the console?

Answer

5

```
var entries = new Array("5");
console.log(entries.length);
```

Question

What shows in the console?

Answer

1

```
var entries = [];
console.log(entries.length);
```

Question

What shows in the console?

Answer

```
0
```

```
var entries = [,,,];
console.log(entries.length);
```

Question

What shows in the console?

Answer

`3 (or 4 in IE 8 and lower)`

```
var entries = ['Planes', 'Trains'];
entries.length = 10;
console.log(entries.length);
```

Question

What shows in the console?

Answer

```
10
```

```
var entries = ['Planes', 'Trains'];
entries.length = 1;
console.log(entries[1]);
```

Question

What shows in the console?

Answer

`undefined`

```
var entries = ['Planes', 'Trains',
    'Automobiles'];
entries[2] = 'Cars';
console.log(entries[2]);
```

Question

What shows in the console?

Answer

```
Cars
```

```
var entries = ['Planes', 'Trains',
    'Automobiles'];
entries[42] = 'Cars';
console.log(entries[42]);
```

Question

What shows in the console?

Answer

```
Cars
```

```javascript
var entries = ['Planes', 'Trains',
    'Automobiles'];
entries[42] = 'Cars';
console.log(entries[42]);
console.log(entries.length);
```

Question

What shows in the console?

Answer

```
Cars

43
```

```
var entries = ['Planes', 'Trains',
    'Automobiles'];
entries[entries.length] = 'Bicycles';
console.log(entries.length);
```

Question

What shows in the console?

Answer

4

```
var entries = ['Planes', 'Trains',
    'Automobiles'];
console.log(entries.toString());
```

Question

What shows in the console?

Answer

```
Planes,Trains,Automobiles
```

```
var ratings = [5, 2, 4];
console.log(ratings.valueOf());
```

Question

What shows in the console?

Answer

```
[5, 2, 4]
```

```
var ratings = [5, 2, 4];
console.log(ratings.join('|'));
```

Question

What shows in the console?

Answer

5|2|4

Array Features

```
var ratings = [];
ratings.push(5);
ratings.push(2, 4);
console.log(ratings.length);
```

Question

What shows in the console?

Answer

3

```
var ratings = [];
ratings.push(5);
ratings.push(2, 4);
console.log(ratings.pop());
```

Question

What shows in the console?

Answer

4

```
var ratings = [];
ratings.push(5);
ratings.push(2, 4);
ratings.pop();
console.log(ratings.length);
```

Question

What shows in the console?

Answer

2

```
var ratings = [];
ratings.push(5);
ratings.push(2, 4);
var firstRating = ratings.shift();
console.log(firstRating);
```

Question

What shows in the console?

Answer

5

```
var ratings = [];
ratings.push(5);
ratings.push(2, 4);
var firstRating = ratings.shift();
console.log(ratings.length);
```

Question

What shows in the console?

Answer

2

```
var ratings = [];
ratings.push(5);
ratings.push(2, 4);
var newCount = ratings.unshift(3);
console.log(ratings[0]);
console.log(newCount);
```

Question

What shows in the console?

Answer

3

4

```
var ratings = [1, 2, 3];
var newRatings = ratings.concat([4, 5, 6]);
console.log(newRatings.length);
```

Question

What shows in the console?

Answer

6

```
var ratings = [1, 2, 3];
var newRatings = ratings.concat([4, 5, 6], 7, 8);
console.log(newRatings.length);
```

Question

What shows in the console?

Answer

8

```
var ratings = [1, 2, 3];
var newRatings = ratings.concat([4, 5, 6], 7, 8);
console.log(ratings.length);
```

Question

What shows in the console?

Answer

3

```
var ratings = [1, 2, 3, 4];
var newRatings = ratings.slice(2);
console.log(newRatings.toString());
```

Question

What shows in the console?

Answer

```
3,4
```

```
var ratings = [1, 2, 3, 4];
var newRatings = ratings.slice(1,3);
console.log(newRatings.toString());
```

Question

What shows in the console?

Answer

```
2,3
```

```
var ratings = [1, 2, 3, 4];
var newRatings = ratings.slice(1,3);
console.log(ratings.toString());
```

Question

What shows in the console?

Answer

```
1,2,3,4
```

```
var ratings = [1, 2, 3, 4];
var newRatings = ratings.slice(-2);
console.log(newRatings.toString());
```

Question

What shows in the console?

Answer

```
3,4
```

```
var ratings = [1, 2, 3, 4];
ratings.splice(1, 2);
console.log(ratings.toString());
```

Question

What shows in the console?

Answer

```
1,4
```

```
var ratings = [1, 2, 3, 4];
var removedRating = ratings.splice(0, 1);
console.log(removedRating.toString());
```

Question

What shows in the console?

Answer

1

```
var ratings = [1, 2, 3, 4];
ratings.splice(2, 0, 99);
console.log(ratings.toString());
```

Question

What shows in the console?

Answer

```
1,2,99,3,4
```

```
var ratings = [1, 2, 3, 4];
ratings.splice(2, 0, 99, 100, 101);
console.log(ratings.toString());
```

Question

What shows in the console?

Answer

```
1,2,99,100,101,3,4
```

```
var ratings = [1, 2, 3, 4];
ratings.splice(2, 1, 99);
console.log(ratings.toString());
```

Question

What shows in the console?

Answer

```
1,2,99,4
```

```
var ratings = [1, 2, 3, 4];
ratings.splice(2, 1, 99, 100, 101);
console.log(ratings.toString());
```

Question

What shows in the console?

Answer

```
1,2,99,100,101,4
```

```
var ratings = [1, 2, 3, 4];
ratings.reverse();
console.log(ratings.toString());
```

Question

What shows in the console?

Answer

```
4,3,2,1
```

```
var ratings = [4, 1, 3, 2];
ratings.sort();
console.log(ratings.toString());
```

Question

What shows in the console?

Answer

```
1,2,3,4
```

```
var ratings = [4, 1, 3, 2, 10];
ratings.sort();
console.log(ratings.toString());
```

Question

What shows in the console?

Answer

```
1,10,2,3,4
```

```
var ratings = [4, 1, 3, 2, 10];
ratings.sort(function (value1, value2) {
    return value1 - value2;
});
console.log(ratings.toString());
```

Question

What shows in the console?

Answer

```
1,2,3,4,10
```

```
var ratings = [1, 2, 3, 4, 5, 4, 3, 2, 1];
console.log(ratings.indexOf(3));
```

Question

What shows in the console?

Answer

2

```
var ratings = [1, 2, 3, 4, 5, 4, 3, 2, 1];
console.log(ratings.indexOf('3'));
```

Question

What shows in the console?

Answer

```
-1
```

```
var ratings = [1, 2, 3, 4, 5, 4, 3, 2, 1];
console.log(ratings.indexOf(66));
```

Question

What shows in the console?

Answer

```
-1
```

```
var ratings = [1, 2, 3, 4, 5, 4, 3, 2, 1];
console.log(ratings.lastIndexOf(1));
```

Question

What shows in the console?

Answer

8

Date Fundamentals

```
var dt = new Date();
console.log(dt);
```

Question

What shows in the console?

Answer

```
Fri Sep 25 2015 11:20:11 GMT-0700 (Pacific Daylight Time)
```

```
var dt = new Date(0);
console.log(dt);
```

Question

What shows in the console?

Answer

```
Wed Dec 31 1969 16:00:0
0 GMT-
0800 (Pacific Standard
Time)
```

```
var dt = new Date('2/13/2015');
console.log(dt);
```

Question

What shows in the console?

Answer

```
Fri Feb 13 2015 00:00:0
0 GMT-
0800 (Pacific Standard
Time)
```

```
var dt = new Date('March 1, 2015');
console.log(dt);
```

Question

What shows in the console?

Answer

```
Sun Mar 01 2015 00:00:0
0 GMT-
0800 (Pacific Standard
Time)
```

```
var dt = new Date('Hi Mom!');
console.log(dt);
```

Question

What shows in the console?

Answer

```
Invalid Date
```

```
var dt = Date.parse('Hi Mom!');
console.log(dt);
```

Question

What shows in the console?

Answer

NaN

```
var dt = Date.UTC(2000, 0);
console.log(dt);
```

Question

What shows in the console?

Answer

```
946684800000
```

```
var dt = new Date(Date.UTC(2000, 0));
console.log(dt);
```

Question

What shows in the console?

Answer

```
Fri Dec 31 1999 16:00:0
0 GMT-
0800 (Pacific Standard
Time)
```

```
var dt = Date.now();
console.log(dt);
```

Question

What shows in the console?

Answer

```
1443206120689
```

```
var dt = new Date();
console.log(dt.toDateString());
```

Question

What shows in the console?

Answer

```
Fri Sep 25 2015
```

```
var dt = new Date();
console.log(dt.toTimeString());
```

Question

What shows in the console?

Answer

```
11:38:13 GMT-0700 (Pacific Daylight Time)
```

```
var dt = new Date();
console.log(dt.getFullYear());
```

Question

What shows in the console?

Answer

2015

```
var dt = new Date();
dt.setFullYear(2025);
console.log(dt.getFullYear());
```

Question

What shows in the console?

Answer

```
2025
```

```
add website page for date/time fns
```

Question

Answer

Regular Expressions

```
var blogText = "Sam I Am";
var pattern = new RegExp('am', 'g');
console.log(pattern instanceof RegExp);
```

Question

What shows in the console?

Answer

```
true
```

```
var blogText = "Sam I Am";
var pattern = /am/g;
console.log(pattern instanceof RegExp);
```

Question

What shows in the console?

Answer

```
true
```

```
var blogText = "Sam I Am";
var pattern = /am/g;
console.log(pattern.test(blogText));
```

Question

What shows in the console?

Answer

```
true
```

```
var blogText = "Sam I Am";
var pattern = /bam/g;
console.log(pattern.test(blogText));
```

Question

What shows in the console?

Answer

```
false
```

```
var blogText = "Sam I Am";
var pattern = /m/g;
console.log(pattern.test(blogText));
console.log(pattern.test(blogText));
console.log(pattern.test(blogText));
```

Question

What shows in the console?

Answer

```
true
true
false
```

```
var blogText = "Sam I Am";
var pattern = /m/g;
var result = pattern.exec(blogText);
console.log(result);
```

Question

What shows in the console?

Answer

```
["m", index: 2, input:
"Sam I Am"]
```

```
var blogText = "Sam I Am";
var pattern = /m/g;
var result = pattern.exec(blogText);
console.log(result[0]);
```

Question

What shows in the console?

Answer

m

```
var blogText = "Sam I Am";
var pattern = /m/g;
var result = pattern.exec(blogText);
console.log(result.index);
```

Question

What shows in the console?

Answer

2

```
var blogText = "Sam I Am";
var pattern = /m/g;
var result = pattern.exec(blogText);
console.log(result.input);
```

Question

What shows in the console?

Answer

```
Sam I Am
```

```
var blogText = "Sam I Am";
var pattern = /m/g;
var result = pattern.exec(blogText);
console.log(result.index);
result = pattern.exec(blogText);
console.log(result.index);
```

Question

What shows in the console?

Answer

2

7

```
var blogText = "Sam I Am";
var pattern = /m/g;
var result = pattern.exec(blogText);
while (result) {
    console.log(result.index);
    result = pattern.exec(blogText);
}
```

Question

What shows in the console?

Answer

2

7

```
var blogText = "Sam I Am";
var pattern = /m/g;
var result = blogText.match(pattern);
console.log(result);
```

Question

What shows in the console?

Answer

```
["m", "m"]
```

```
var blogText = "Sam I Am";
var pattern = /am/g;
var result = blogText.match(pattern);
console.log(result);
```

Question

What shows in the console?

Answer

```
["am"]
```

```
var blogText = "Sam I Am";
var pattern = /am/gi;
var result = blogText.match(pattern);
console.log(result);
```

Question

What shows in the console?

Answer

```
["am", "Am"]
```

```
var blogText = "Sam I Am";
var pattern = /[aA]m/g;
var result = blogText.match(pattern);
console.log(result);
```

Question

What shows in the console?

Answer

```
["am", "Am"]
```

```
var blogText = "Sam I Am";
var pattern = /.m/g;
var result = blogText.match(pattern);
console.log(result);
```

Question

What shows in the console?

Answer

```
["am", "Am"]
```

Summary

- Reference Types Explained
- Reference Type Examples
- Arrays
- Dates
- Regular Expressions

Overview

Simple Objects and JSON

Understanding Prototypes

Working With Prototypes

Object.create()

Object.defineProperty()

Simple Objects and JSON

```
var project = new Object();
project.name = 'Project Phoenix';
console.log(project.name);
```

Question

What shows in the console?

Answer

```
Project Phoenix
```

```
var project = new Object();
project.name = 'Project Phoenix';
project.securityLevel = 15;
project.updateDueDate = function () {
    return true;
};
console.log(project.updateDueDate());
```

Question

What shows in the console?

Answer

```
true
```

```
var project = new Object();
project.name = 'Project Phoenix';
project.securityLevel = 15;
project.updateDueDate = function () {
    return true;
};
console.log(project['securityLevel']);
```

Question

What shows in the console?

Answer

15

```javascript
var project = new Object();
project.name = 'Project Phoenix';
project.securityLevel = 15;
project.updateDueDate = function () {
    return true;
};
var field = 'securityLevel';
console.log(project[field]);
```

Question

What shows in the console?

Answer

15

```
var project = {};
project.name = 'Project Phoenix';
console.log(project.name);
```

Question

What shows in the console?

Answer

```
Project Phoenix
```

```
var project = {
    name: 'Project Phoenix',
    securityLevel: 15,
    updateDueDate: function () {
        return true;
    }
};
console.log(project.name);
```

Question

What shows in the console?

Answer

```
Project Phoenix
```

```
var project = {
    name: 'Project Phoenix',
    securityLevel: 15,
    updateDueDate: function () {
        return true;
    },
    team: ['iguanaboy', 'cactusgirl',
            'Buffy']
};
console.log(project.team[2]);
```

Question

What shows in the console?

Answer

Buffy

```
var project = {
    tasks: [
        {
            taskName: 'first'
        },
        {
            taskName: 'second'
        }
    ]
};
console.log(project.tasks[1].taskName);
```

Question

What shows in the console?

Answer

second

Understanding Prototypes

Every JavaScript Object has a prototype property.

However, we don't always have access to it.

The prototype property is simply an object.

```
var project = anyObject;
project.someFunction();
```

How does JavaScript locate / resolve someFunction?

- `project.someFunction()`
- `project.prototype.someFunction()`
- `project.prototype.prototype.someFunction()`
- `project.prototype.prototype.prototype.someFunction()`
- `etc.`

In most browsers, we can access the prototype property by using the symbol

__proto__ .

(not safe!)

The prototype property is most easily accessed on constructor functions.

(Covered in the functions module of this course)

Working with Prototypes

```
var project = {
    name: 'Project Phoenix'
};
console.log(project.foo());
```

Question

What shows in the console?

Answer

```
TypeError: project.foo
is not a function
```

```
var project = {
    name: 'Project Phoenix'
};
console.log(project.toString());
```

Question

What shows in the console?

Answer

```
[object Object]
```

```
console.log(typeof Object.prototype);
```

Question

What shows in the console?

Answer

```
object
```

```
console.log(typeof Object.prototype.toString);
```

Question

What shows in the console?

Answer

```
function
```

```
var project = {
    name: 'Project Phoenix'
};
console.log(project.prototype);
```

Question

What shows in the console?

Answer

undefined

```
var project = {
    name: 'Project Phoenix'
};
console.log(typeof project.__proto__);
```

Question

What shows in the console?

Answer

object

```
var project = {
    name: 'Project Phoenix'
};
console.log(project.__proto__ === Object);
```

Question

What shows in the console?

Answer

```
false
```

```
var project = {
    name: 'Project Phoenix'
};
console.log(project.__proto__ === Object.prototype);
```

Question

What shows in the console?

Answer

```
true
```

Object.create() and Prototypes

```
var project = {
    securityLevel: 2
};
var secretProject = Object.create(project);
console.log(secretProject.securityLevel);
```

Question

What shows in the console?

Answer

2

```
var project = {
    securityLevel: 2
};
var secretProject = Object.create(project);
console.log(typeof secretProject.toString);
```

Question

What shows in the console?

Answer

`function`

```
var project = {
    securityLevel: 2
};
var secretProject = Object.create(project);
console.log(secretProject.__proto__ === Object.prototype);
```

What shows in the console?

```
false
```

```
var project = {
    securityLevel: 2
};
var secretProject = Object.create(project);
console.log(secretProject.__proto__.__proto__ === Object.prototype);
```

Q What shows in the console?

A `true`

Object.defineProperty()

```
var task = { };
Object.defineProperty(task, 'text', {
    value: 'Get this job done!'
});
console.log(task.text);
```

Question

What shows in the console?

Answer

```
Get this job done!
```

```
'use strict';
var task = { };
Object.defineProperty(task, 'text', {
    value: 'Get this job done!',
    writable: true
});
task.text = task.text + ' ... NOW!';
console.log(task.text);
```

Question

What shows in the console?

Answer

```
Get this job done! ...
NOW!
```

```
'use strict';
var task = { };
Object.defineProperty(task, 'text', {
    value: 'Get this job done!'
    // writable: true
});
task.text = task.text + ' ... NOW!';
console.log(task.text);
```

Question

What shows in the console?

Answer

```
TypeError: Cannot
assign to read only
property...
```

```
'use strict';
var task = { };
Object.defineProperty(task, 'text', {
    value: 'Get this job done!',
    enumerable: true
});
for (var f in task)
    console.log(f);
```

Question

What shows in the console?

Answer

```
text
```

```
'use strict';
var task = { };
Object.defineProperty(task, 'text', {
    value: 'Get this job done!'
    //enumerable: true
});
for (var f in task)
    console.log(f);
```

Question

What shows in the console?

Answer

```
(nothing shows)
```

```
'use strict';
var task = { };
Object.defineProperty(task, 'text', {
    value: 'Get this job done!',
    configurable: false
});
Object.defineProperty(task, 'text', {
    value: 'Done!'
});
console.log(task.text);
```

Question

What shows in the console?

Answer

```
TypeError: Cannot
redefine property: text
```

```
'use strict';
var task = { };
Object.defineProperty(task, 'text', {
    value: 'Get this job done!',
    configurable: true
});
Object.defineProperty(task, 'text', {
    value: 'Done!'
});
console.log(task.text);
```

Question

What shows in the console?

Answer

Done!

```
'use strict';
var task = {
    _dueDate: '1/15/16'
};
Object.defineProperty(task, 'dueDate', {
    get: function () {
        return this._dueDate;
    }
});
console.log(task.dueDate);
```

Question

What shows in the console?

Answer

1/15/16

```
'use strict';
var task = {
    _dueDate: '1/15/16'
};
Object.defineProperty(task, 'dueDate', {
    get: function () {
        return this._dueDate;
    },
    set: function (newValue) {
        this._dueDate = newValue;
    }
});
task.dueDate = '2/2/16';
console.log(task.dueDate);
```

Question

What shows in the console?

Answer

2/2/16

```
'use strict';
var task = { };
Object.defineProperties(task, {
    'text': {
        value: 'New Task'
    },
    'dueDate': {
        value: '1/15/16'
    }
});
console.log(task.text + ' Due: ' +
   task.dueDate);
```

Question

What shows in the console?

Answer

```
New Task Due: 1/15/16
```

```
'use strict';
var task = { };
Object.defineProperties(task, {
    'text': {
        value: 'New Task'
    }
});
var descriptor = Object.getOwnPropertyDescriptor(task, 'text');
console.log(descriptor);
```

Q

What shows in the console?

A

```
Object {value: "New Task", writa
ble: false, enumerable: false, c
onfigurable: false}
```

Miscellaneous Object Functions

```
'use strict';
var project = {
    name: 'Top Secret Project',
    dueDate: '1/1/2016'
};
console.log(project.hasOwnProperty('name'));
```

Question

What shows in the console?

Answer

```
true
```

```
'use strict';
var project = {
    name: 'Top Secret Project',
    dueDate: '1/1/2016'
};
console.log(project.hasOwnProperty('toString'));
```

Question

What shows in the console?

Answer

```
false
```

```
'use strict';
var project = {
    name: 'Top Secret Project',
    dueDate: '1/1/2016'
};
console.log(project.__proto__.hasOwnProperty('toString'));
```

Q What shows in the console?

A `true`

```
'use strict';
var project = {
    name: 'Top Secret Project',
    dueDate: '1/1/2016'
};
console.log(Object.prototype.isPrototypeOf(project));
```

Q What shows in the console?

A `true`

```
var project = {
    securityLevel: 2
};
var secretProject = Object.create(project);
console.log(project.isPrototypeOf(secretProject));
```

What shows in the console?

```
true
```

```
var project = {
    securityLevel: 2
};
var secretProject = Object.create(project);
console.log(Object.prototype.isPrototypeOf(secretProject));
```

Q What shows in the console?

A `true`

```
var project = {
    securityLevel: 2
};
var secretProject = Object.create(project);
console.log(project.__proto__.isPrototypeOf(secretProject));
```

Q What shows in the console?

A `true`

```
var project = {
    securityLevel: 2
};
var secretProject = Object.create(project);
console.log("securityLevel" in secretProject);
```

Q What shows in the console?

A `true`

```
var project = {
    securityLevel: 2
};
var secretProject = Object.create(project);
console.log("toString" in secretProject);
```

Q What shows in the console?

A `true`

```
var project = {
    securityLevel: 2
};
var secretProject = Object.create(project);
console.log("securityLevel" in Object.prototype);
```

Q What shows in the console?

A `false`

Summary

- Simple Objects and JSON
- Understanding Prototypes
- Working With Prototypes
- Object.create()
- Object.defineProperty()
- Miscellaneous Object Functions

Overview

Function Expressions

Constructor Functions

The "this" Keyword

Calling Functions (call and apply)

Closures

IIFE's

Naming Function Expressions

```
var hireEmployee = function (name) {
};
console.log(typeof hireEmployee);
```

Question

What shows in the console?

Answer

```
function
```

```
var hireEmployee = function (name) {
    throw ('Error');
};
hireEmployee('JJ');
```

Question

What shows in the console?

Answer

```
Uncaught Error -
 hireEmployee
```

```
var hireEmployee = function (name) {
    throw ('Error');
};
var action = hireEmployee;
action('JJ');
```

Question

What shows in the console?

Answer

```
Uncaught Error -
 hireEmployee
```

```
var hireEmployee = function myHireEmployeeFn(name) {
    throw ('Error');
};
var action = hireEmployee;
action('JJ');
```

Q What shows in the console?

A Uncaught Error - myHireEmployeeFn

Constructor Functions

```
console.log(typeof Object);
```

Question

What shows in the console?

Answer

```
function
```

```
var Employee = function (name) {
    this.name = name;
};
var newEmployee = new Employee('JJ');
console.log(typeof newEmployee);
```

Question

What shows in the console?

Answer

```
object
```

```
var Employee = function (name) {
    this.name = name;
};
var newEmployee = new Employee('JJ');
console.log(newEmployee.name);
```

Question

What shows in the console?

Answer

`JJ`

```
var Employee = function (name, boss) {
    this.name = name;
    this.boss = boss;
};
var newEmployee = new Employee('JJ', 'JD Hogg');
console.log(newEmployee.boss);
```

Question

What shows in the console?

Answer

```
JD Hogg
```

```
var Employee = function (name) {
    this.name = name;
};
var e1 = new Employee('JJ');
var e2 = new Employee('JV');
console.log(e1 === e2);
```

Question

What shows in the console?

Answer

`false`

```
var Employee = function (name) {
    this.name = name;
};
var e1 = new Employee('JJ');
var e2 = new Employee('JV');
console.log(e1.__proto__ === e2.__proto__);
```

Question

What shows in the console?

Answer

```
true
```

```
var Employee = function (name) {
    this.name = name;
    this.giveRaise = function () {
    };
};
var e1 = new Employee('JJ');
var e2 = new Employee('JV');
console.log(e1.giveRaise === e2.giveRaise);
```

Question

What shows in the console?

Answer

```
false
```

```
var Employee = function (name) {
    this.name = name;
    this.giveRaise = function () {
    };
};
console.log(typeof Employee.prototype);
```

Question

What shows in the console?

Answer

`object`

```
var Employee = function (name) {
    this.name = name;
    this.giveRaise = function () {
    };
};
console.log(Employee.prototype === Object.prototype);
```

What shows in the console?

```
false
```

```
var Employee = function (name) {
    this.name = name;
};
Employee.prototype.giveRaise = function () {
};
var e1 = new Employee('JJ');
var e2 = new Employee('JV');
console.log(e1.giveRaise === e2.giveRaise);
```

Q What shows in the console?

A `true`

```
var Employee = function (name) {
    this.name = name;
    this.salary = 50000;
};
Employee.prototype.giveRaise = function (raise) {
    this.salary += raise;
};
var e1 = new Employee('JJ');
var e2 = new Employee('JV');
e1.giveRaise(100000);
console.log(e1.salary);
console.log(e2.salary);
```

Q What shows in the console?

A

```
150000
50000
```

The "this" Keyword

```
console.log(typeof this);
```

Question

What shows in the console?

Answer

```
object
```

```
console.log(this === window);
```

Question

What shows in the console?

Answer

```
true
```

```
var name = 'Jeff';
console.log(this.name);
```

Question

What shows in the console?

Answer

```
Jeff
```

```
var updateSalary = function () {
    console.log(this === window);
};
updateSalary();
```

Question

What shows in the console?

Answer

`true`

```
var employee = {
    name: 'Jeff',
    updateSalary: function () {
        console.log(this);
    }
};
employee.updateSalary();
```

Question

What shows in the console?

Answer

```
Object {name: "Jeff"}
```

```
var employee = {
    name: 'Jeff',
    updateSalary: function () {
        var fn = function () {
            console.log(this);
        };
        fn();
    }
};
employee.updateSalary();
```

Question

What shows in the console?

Answer

```
Window {...}
```

```
var employee = {
    name: 'Jeff',
    updateSalary: function () {
        var fn = function () {
            console.log(this === window);
        };
        fn();
    }
};
employee.updateSalary();
```

Question

What shows in the console?

Answer

```
true
```

```
var Address = function (line1) {
    this.line1 = line1;
    console.log(this);
};
var addr = new Address('123 State St.');
```

Question

What shows in the console?

Answer

```
Address {line1: "123 St
ate St."}
```

```
var Address = function (line1) {
    this.line1 = line1;
};
Address.prototype.updateZipCode = function () {
    console.log(this);
};

var addr = new Address('123 State St.');
addr.updateZipCode();
```

Q What shows in the console?

A

```
Address {line1: "123 State
 St."}
```

Calling Functions

Using call() and apply()

```
var updateZipCode = function () {
    console.log(this);
};
updateZipCode();
```

Question

What shows in the console?

Answer

```
Window { ... }
```

```
var updateZipCode = function () {
    console.log(this);
};
updateZipCode.call({});
```

Question

What shows in the console?

Answer

```
Object {}
```

```
var updateZipCode = function () {
    console.log(this);
};
updateZipCode.call({ zip: '11787'});
```

Question

What shows in the console?

Answer

```
Object {zip: "11787"}
```

```
var updateZipCode = function () {
    console.log(this);
};
var zipCode = {
    zip: '11787'
};
updateZipCode.call(zipCode);
```

Question

What shows in the console?

Answer

```
Object {zip: "11787"}
```

```
var updateZipCode = function (newZip, country) {
    console.log(newZip + ' ' + country);
};
var zipCode = {
    zip: '11787'
};
updateZipCode.call(zipCode, '11888', 'us');
```

Q What shows in the console?

A `11888 us`

```
var updateZipCode = function (newZip, country) {
    console.log(newZip + ' ' + country);
};
var zipCode = {
    zip: '11787'
};
updateZipCode.apply(zipCode, ['11888', 'us']);
```

Q What shows in the console?

A `11888 us`

```
var updateZipCode = function (newZip, country) {
    console.log(newZip + ' ' + country);
};
var zipCode = {
    zip: '11787'
};
updateZipCode.apply(zipCode, '11888', 'us');
```

Q What shows in the console?

A

```
Uncaught TypeError: Function.prototype.apply: Arguments list has wrong type
```

Closures

```
var salaryUpdater = function (salary) {
    var currentSalary = salary;
    var generator = function () {
        currentSalary *= 2;
        return currentSalary;
    };
    return generator;
};

var updateFn = salaryUpdater(50000);

console.log(updateFn());
```

Q What shows in the console?

A `100000`

```
var salaryUpdater = function (salary) {
    var currentSalary = salary;
    var generator = function () {
        currentSalary += 10000;
        return currentSalary;
    };
    return generator;
};

var updateFn = salaryUpdater(50000);
updateFn();
console.log(updateFn());
```

Q What shows in the console?

A

```
70000
```

IIFEs

Immediately Invoked Function Expressions

```
(function () {
    console.log('Executed!');
})();
```

Question

What shows in the console?

Answer

```
Executed!
```

```
(function () {
    console.log('Executed!');
}());
```

Question

What shows in the console?

Answer

```
Executed!
```

```
(function () {
    var employeeName = 'Jill';
})();

console.log(employeeName);
```

Question

What shows in the console?

Answer

```
Uncaught ReferenceError
: employeeName is not
defined
```

```
var app = {};
(function (ns) {
    ns.name = 'None';
})(app);

console.log(app.name);
```

Question

What shows in the console?

Answer

None

```
var app = {};
var jQuery = {};

(function (ns, $) {
    ns.name = 'None';
    console.log($ === jQuery);
})(app, jQuery);
```

Question

What shows in the console?

Answer

```
true
```

```
var app = {};
var jQuery = {};

(function (ns, $, undefined) {
    ns.name = 'None';
    console.log(undefined);
})(app, jQuery);
```

Question

What shows in the console?

Answer

undefined

```
+function () {
    var employeeName = 'Jill';
    console.log('Executed!');
}();

console.log(employeeName);
```

Question

What shows in the console?

Answer

```
Executed!

Uncaught ReferenceError
: employeeName is not
defined
```

Recursion

```
var orgChart = {
    name: 'Michael', subordinates: [
        {
            name: 'Andy', subordinates: [
                {
                    name: 'Dwight', subordinates: []
                },
                {
                    name: 'Kevin', subordinates: []
                }
            ]
        }
    ]
};

var fn = function (topEmployee) {
    console.log(topEmployee.name);
    for (var i = 0; i < topEmployee.subordinates.lengt
h; i++)
        fn(topEmployee.subordinates[i]);
};

fn(orgChart);
```

Question

What shows in the console?

Answer

```
Michael
Andy
Dwight
Kevin
```

```
var orgChart = {
    name: 'Michael', subordinates: [
        {
            name: 'Andy', subordinates: [
                {
                    name: 'Dwight', subordinates: []
                },
                {
                    name: 'Kevin', subordinates: []
                }
            ]
        }]
};
var fn = function (topEmployee) {
    console.log(topEmployee.name);
    for (var i = 0; i < topEmployee.subordinates.lengt
h; i++)
        fn(topEmployee.subordinates[i]);
};
var fn2 = fn;
fn = null;
fn2(orgChart);
```

Question

What shows in the console?

Answer

```
Michael

Uncaught TypeError: fn
is not a function
```

```
var orgChart = {
    name: 'Michael', subordinates: [
        {
            name: 'Andy', subordinates: [
                {
                    name: 'Dwight', subordinates: []
                },
                {
                    name: 'Kevin', subordinates: []
                }
            ]
        }]
};
var fn = function showAllEmployees (topEmployee) {
    console.log(topEmployee.name);
    for (var i = 0; i < topEmployee.subordinates.lengt
h; i++)
        showAllEmployees(topEmployee.subordinates[i]);
};
var fn2 = fn;
fn = null;
fn2(orgChart);
```

Question

What shows in the console?

Answer

```
Michael
Andy
Dwight
Kevin
```

Summary

- Function Expressions
- The "this" Keyword
- Calling Functions Using Call and Apply
- Closures
- IIFE's
- Recursion

Overview

The window Object

System Dialogs

The location Object

Document Basics

Query Selectors

The window Object and Timers

```
console.log(window.screenLeft + ', ' + window.screenTop);
```

What shows in the console?

```
1, 1
```

```
console.log(window.innerWidth + ', ' + window.outerWidth);
```

Q What shows in the console?

A

```
1189, 1201
```

```
window.open('http://www.abcd.com', '_blank');
```

Q Will a window open?

A (popups must be enabled)

```
console.log(new Date().getSeconds());
setTimeout(function () {
    console.log(new Date().getSeconds());
}, 1000);
```

Question

What shows in the console?

Answer

6

7

```
console.log(new Date().getSeconds());
var id = setTimeout(function () {
    console.log(new Date().getSeconds());
}, 1000);
clearTimeout(id);
```

Question

What shows in the console?

Answer

31

```
console.log(new Date().getSeconds());
var id = setInterval(function () {
    var secs = new Date().getSeconds();
    console.log(secs);
    if (secs === 10)
        clearInterval(id);
}, 1000);
```

Question

What shows in the console?

Answer

```
7
8
9
10
(stops)
```

System Dialogs

```
alert('Hello World!');
```

Question

What's the result in the browser?

Answer

A system dialog says "Hello World!".

```
if (confirm('Delete EVERYTHING?!')) {
    console.log('You asked for it!')
}
else {
    console.log('Maybe next time...');
}
```

Question

What's the result in the browser and console?

Answer

A system dialog asks "Delete EVERYTHING?!".

```
var result = prompt('Your name?');
console.log(result);
```

Question

What shows in the browser and console?

Answer

null, or whatever was entered in the prompt

The location Object

```
console.log(location.href);
```

Question

What shows in the console?

Answer

```
http://localhost:55138/
M2-html.html
```

```
console.log(location.host);
```

Question

What shows in the console?

Answer

```
localhost:55138
```

```
console.log(location.port);
```

Question

What shows in the console?

Answer

```
55138
```

```
console.log(location.protocol);
```

Question

What shows in the console?

Answer

```
http:
```

```
location.assign('http://www.abcd.com');
```

Question

What occurs in the browser?

Answer

(directs browser to abcd.com)

Document Basics

```
// body of html page
<article id="article1">
    <h3>Title goes here</h3>
    <p>Paragraph 1</p>
    <p>Paragraph 2</p>
    <p>Paragraph 3</p>
</article>

// javascript
var element = document.getElementById('article1');
console.log(element);
```

Q What shows in the console?

A `<article ... </article>`

```
// body of html page
<article id="article1">
    <h3>Title goes here</h3>
    <p>Paragraph 1</p>
    <p>Paragraph 2</p>
    <p>Paragraph 3</p>
</article>

// javascript
var elements = document.getElementsByTagName('p');
console.log(elements);
```

Q What shows in the console?

A `[p, p, p]`

```
// body of html page
<article id="article1">
    <h3>Title goes here</h3>
    <p>Paragraph 1</p>
    <p class="special">Paragraph 2</p>
    <p class="special">Paragraph 3</p>
</article>

// javascript
var elements = document.getElementsByClassName('special');
console.log(elements);
```

Q What shows in the console?

A

```
[p.special, p.special]
```

```
// body of html page
<article id="article1" class="my-custom-class">
    <h3>Title goes here</h3>
    <p>Paragraph 1</p>
    <p class="special">Paragraph 2</p>
    <p class="special">Paragraph 3</p>
</article>

// javascript
var a = document.getElementById('article1');
console.log(a.getAttribute('class'));
```

Q What shows in the console?

A my-custom-class

```
// body of html page
<article id="article1" class="my-custom-class">
    <h3>Title goes here</h3>
    <p>Paragraph 1</p>
    <p class="special">Paragraph 2</p>
    <p class="special">Paragraph 3</p>
</article>

// javascript
var a = document.getElementById('article1');
a.setAttribute('class', 'dynamic-style');
console.log(a.getAttribute('class'));
```

Q What shows in the console?

A

```
dynamic-style
```

```
// body of html page
<article id="article1" class="my-custom-class">
    <h4>Hello!</h4>
    <p>Paragraph 1</p>
</article>

// javascript
var a = document.getElementById('article1');
console.log(a.childNodes);
```

Q What shows in the console?

A `[text, h4, text, p, text]`

Query Selectors

```
// body of html page
<article id="article1" class="my-custom-class">
    <h3>Title goes here</h3>
    <p>Paragraph 1</p>
    <p class="special">Paragraph 2</p>
    <p class="special">Paragraph 3</p>
</article>

// javascript
var element = document.querySelector('article');
console.log(element);
```

Q What shows in the console?

A `<article... </article>`

```
// body of html page
<article id="article1" class="my-custom-class">
    <h3>Title goes here</h3>
    <p>Paragraph 1</p>
    <p class="special">Paragraph 2</p>
    <p class="special">Paragraph 3</p>
</article>

// javascript
var element = document.querySelector('#article1');
console.log(element);
```

Q What shows in the console?

A `<article... </article>`

```
// body of html page
<article id="article1" class="my-custom-class">
    <h3>Title goes here</h3>
    <p>Paragraph 1</p>
    <p class="special">Paragraph 2</p>
    <p class="special">Paragraph 3</p>
</article>

// javascript
var elements = document.querySelector('.special');
console.log(elements);
```

Q What shows in the console?

A

```
<p class="special">Paragraph 2</p>
```

```
// body of html page
<article id="article1" class="my-custom-class">
    <h3>Title goes here</h3>
    <p>Paragraph 1</p>
    <p class="special">Paragraph 2</p>
    <p class="special">Paragraph 3</p>
</article>

// javascript
var elements = document.querySelectorAll('.special');
console.log(elements);
```

Q What shows in the console?

A `[p.special, p.special]`

Summary

- The window Object
- System Dialogs
- The Location Object
- Document Basics
- Query Selectors

Overview

- Events
- The Event Object
- Handling Events
- Event Listeners
- Event Bubbling

Events

User Interface Events

load

unload

abort

error

select

resize

scroll

Mouse and Mouse Wheel Events

click

dblclick

mousedown

mouseenter

mouseleave

mousemove

mouseout

mouseover

mouseup

mousewheel

Keyboard and Text Events

keydown

keypress

keyup

textInput

Focus Events

blur

focus

focusin

focusout

HTML5 Events

contextmenu

beforeunload

Touch Events

touchstart

touchmove

touchend

touchcancel

The Event Object

Event Object Properties

bubbles

cancelable

currentTarget

defaultPrevented

detail

eventPhase

preventDefault()

stopImmediatePropagation()

stopPropagation()

target

trusted

type

Handling Events

```
// body of HTML
<input type="button" value="Submit"
          onclick="alert('foo')" />
```

Question

What shows in the browser when the button is pressed?

Answer

```
System Dialog says:
"foo"
```

```
// body of HTML
<input id="submit1" type="button"
          onclick ="submitForm()" />


// JavaScript file
function submitForm() {
    console.log('in submitForm');
}
```

Question

What shows in the console when the button is pressed?

Answer

```
in submitForm
```

```
// body of HTML
<input id="submit1" type="button"
             onclick="submitForm(this)" />


// JavaScript file
function submitForm(element) {
    console.log(element.id);
}
```

Question

What shows in the console when the button is pressed?

Answer

```
submit1
```

```
// body of HTML
<input id="submit1" type="button"
   onclick="submitForm(this, event)" />


// JavaScript file
function submitForm(element, event) {
    console.log(event.type);
}
```

Question

What shows in the console when the button is pressed?

Answer

```
click
```

```
// body of HTML
<input id="submit1" type="button" />


// JavaScript file
var button =
    document.getElementById('submit1');

button.onclick = function () {
    console.log('Button Clicked');
}
```

Question

What shows in the console when the button is pressed?

Answer

```
Button Clicked
```

```
// body of HTML
<input id="submit1" type="button" />


// JavaScript file
var button = document.getElementById('submit1');
button.onclick = function () {
    console.log('Button Clicked');
}

button.onclick = null;
```

Question

What shows in the console when the button is pressed?

Answer

```
(nothing shows)
```

Event Listeners

```
// body of HTML
<input id="submit1" type ="button" />

// JavaScript file
var button = document.getElementById('submit1');
button.addEventListener('click', function () {
    console.log('Button Clicked');
});
```

Q What shows in the console when the button is pressed?

A Button Clicked

```
// body of HTML
<input id="submit1" type ="button" />

// JavaScript file
var button = document.getElementById('submit1');
button.addEventListener('click', function () {
    console.log('Button Clicked');
});
button.addEventListener('click', function () {
    console.log('Another Listener Function');
});
```

Q

What shows in the console when the button is pressed?

A

```
Button Clicked
Another Listener Function
```

```
// body of HTML
<input id="submit1" type ="button" />

// JavaScript file
var button = document.getElementById('submit1');

var submitHandler = function formSubmitHandler() {
    console.log('in formSubmitHandler()');
};

button.addEventListener('click', submitHandler);
```

Q What shows in the console when the button is pressed?

A `in formSubmitHandler()`

```
// body of HTML
<input id="submit1" type ="button" />

// JavaScript file
var button = document.getElementById('submit1');

var submitHandler = function formSubmitHandler() {
    console.log('in formSubmitHandler()');
};

button.addEventListener('click', submitHandler);
button.removeEventListener('click', submitHandler);
```

Q

What shows in the console when the button is pressed?

A

(nothing)

Event Bubbling

```
// body of HTML
<div id="div1" style="width:200px; height:200px; background-color: blue;">
        <div id="div2" style="width:100px; height:100px;
             background-color: red;"></div>
</div>

// JavaScript file
var div1 = document.getElementById('div1');
var div2 = document.getElementById('div2');
var clickHandler = function divClickHandler(event) {
    console.log(this.id);
};
div1.addEventListener('click', clickHandler, false);
div2.addEventListener('click', clickHandler, false);
```

Q What shows in the console when div2 is pressed?

A

```
div2
div1
```

```
// body of HTML
<div id="div1" style="width:200px; height:200px; background-color: blue;">
        <div id="div2" style="width:100px; height:100px;
             background-color: red;"></div>
</div>

// JavaScript file
var div1 = document.getElementById('div1');
var div2 = document.getElementById('div2');
var clickHandler = function divClickHandler(event) {
    console.log(this.id);
};
div1.addEventListener('click', clickHandler, true);
div2.addEventListener('click', clickHandler, true);
```

Q What shows in the console when div2 is pressed?

A

```
div1
div2
```

```
// body of HTML
<div id="div1" style="width:200px; height:200px; background-color: blue;">
        <div id="div2" style="width:100px; height:100px;
             background-color: red;"></div>
</div>

// JavaScript file
var div1 = document.getElementById('div1');
var div2 = document.getElementById('div2');
var clickHandler = function divClickHandler(event) {
    console.log(this.id);
    event.stopPropagation();
};
div1.addEventListener('click', clickHandler, true);
div2.addEventListener('click', clickHandler, true);
```

Q What shows in the console when div2 is pressed?

A `div1`

```
// body of HTML
<div id="div1" style="width:200px; height:200px; background-color: blue;">
        <div id="div2" style="width:100px; height:100px;
             background-color: red;"></div>
</div>

// JavaScript file
var div1 = document.getElementById('div1');
var div2 = document.getElementById('div2');
var clickHandler = function divClickHandler(event) {
    console.log(this.id);
    event.stopPropagation();
    event.preventDefault();
};
div1.addEventListener('click', clickHandler, true);
div2.addEventListener('click', clickHandler, true);
```

Q

What shows in the console when div2 is pressed?

A

div1

Summary

- Events
- The Event Object
- Handling Events
- Event Listeners
- Event Bubbling

Overview

Global Functions

Math

String

arguments

Global Functions

```
var value = parseInt('1234');
console.log(value);
```

Question

What shows in the console?

Answer

1234

```
var value = parseInt('b1234');
console.log(value);
```

Question

What shows in the console?

Answer

NaN

```
var value = parseInt('12z34');
console.log(value);
```

Question

What shows in the console?

Answer

12

```
var value = parseInt('1234.9');
console.log(value);
```

Question

What shows in the console?

Answer

1234

```
var value = parseInt(' ');
console.log(value);
```

Question

What shows in the console?

Answer

NaN

```
var value = parseInt('C000', 16);
console.log(value);
```

Question

What shows in the console?

Answer

49152

```
var value = parseFloat('123.9');
console.log(value);
```

Question

What shows in the console?

Answer

```
123.9
```

```
var value = parseFloat('z123.9');
console.log(value);
```

Question

What shows in the console?

Answer

NaN

```
var value = parseFloat('1239e-1');
console.log(value);
```

Question

What shows in the console?

Answer

```
123.9
```

```
var value =
     isFinite(Number.POSITIVE_INFINITY);
console.log(value);
```

Question

What shows in the console?

Answer

```
false
```

```
var value = isFinite(42);
console.log(value);
```

Question

What shows in the console?

Answer

```
true
```

```
var value = isNaN(NaN);
console.log(value);
```

Question

What shows in the console?

Answer

```
true
```

```
var value = isNaN(9 / 0);
console.log(value);
```

Question

What shows in the console?

Answer

```
false
```

```
var value = isFinite(9 / 0);
console.log(value);
```

Question

What shows in the console?

Answer

```
false
```

```
var path = "\\start\\";
console.log(encodeURI(path));
```

Question

What shows in the console?

Answer

```
%5Cstart%5C
```

```
var path = "\\start\\+";
console.log(encodeURI(path));
```

Question

What shows in the console?

Answer

```
%5Cstart%5C+
var path = "\\start\\+";
console.log(encodeURIComponent(path));
```

Question

What shows in the console?

Answer

```
%5Cstart%5C%2B
```

```
var path = "%5Cstart%5C+";
console.log(decodeURI(path));
```

Question

What shows in the console?

Answer

```
\start\+
var path = "%5Cstart%5C%2B";
console.log(decodeURIComponent(path));
```

Question

What shows in the console?

Answer

```
\start\+
var globalVar = 'foo';
var code = 'console.log(globalVar);'
eval(code);
```

Question

What shows in the console?

Answer

foo

The Math Object

```
var value = Math.PI;
console.log(value);
```

Question

What shows in the console?

Answer

```
3.141592653589793
```

```
var value = Math.abs(-42);
console.log(value);
```

Question

What shows in the console?

Answer

42

```
var value = Math.ceil(11.1);
console.log(value);
```

Question

What shows in the console?

Answer

12

```
var value = Math.floor(11.9);
console.log(value);
```

Question

What shows in the console?

Answer

11

```
var value = Math.trunc(42.12);
console.log(value);
```

Question

What shows in the console?

Answer

42

```
var value = Math.max(-12, 0, 12, "88");
console.log(value);
```

Question

What shows in the console?

Answer

88

```
var value = Math.min(-12, 0, 12, "88");
console.log(value);
```

Question

What shows in the console?

Answer

```
-12
```

```
var value = Math.pow(2, 3);
console.log(value);
```

Question

What shows in the console?

Answer

8

```
var value = Math.random();
console.log(value);
```

Question

What shows in the console?

Answer

```
0.2882959560956806
```

```
var value = Math.sqrt(81);
console.log(value);
```

Question

What shows in the console?

Answer

9

```
var value = Math.sqrt(-81);
console.log(value);
```

Question

What shows in the console?

Answer

NaN

The String Object

```
var value = 'My String';
console.log(value.charAt(3));
```

Question

What shows in the console?

Answer

S

```
var value = 'My String';
console.log(value.concat(' Lives!'));
```

Question

What shows in the console?

Answer

```
My String Lives!
```

```
var value = 'My String';
console.log(value.includes(' '));
```

Question

What shows in the console?

Answer

```
true
```

```
var value = 'My String';
console.log(value.endsWith('ing'));
```

Question

What shows in the console?

Answer

```
true
```

```
var value = 'My String';
console.log(value.endsWith('ing '));
```

Question

What shows in the console?

Answer

```
false
```

```
var value = 'My String';
console.log(value.indexOf('M'));
```

Question

What shows in the console?

Answer

```
0
```

```
var value = 'My String';
console.log(value.indexOf('Z'));
```

Question

What shows in the console?

Answer

```
-1
```

```
var value = 'Some String';
console.log(value.lastIndexOf('S'));
```

Question

What shows in the console?

Answer

5

```
var value = 'Some String';
console.log(value.slice(5));
```

Question

What shows in the console?

Answer

```
String
```

```
var value = 'Some String';
console.log(value.slice(5, 8));
```

Question

What shows in the console?

Answer

```
Str
```

```
var value = 'Some String';
console.log(value.slice(-3));
```

Question

What shows in the console?

Answer

```
ing
```

```
var value = 'Some String';
console.log(value.split(' ').length);
```

Question

What shows in the console?

Answer

2

```
var value = 'Some String';
console.log(value.substr(0, 4));
```

Question

What shows in the console?

Answer

```
Some
```

```
var value = 'Some String';
console.log(value.substring(5, 6));
```

Question

What shows in the console?

Answer

S

```
var value = 'Some String';
console.log(value.substr(5, 6));
```

Question

What shows in the console?

Answer

```
String
```

arguments

```
var validateValues = function () {
    console.log(arguments.length);
};
validateValues(1, true, 'Settings');
```

Question

What shows in the console?

Answer

3

```
var validateValues = function () {
    console.log(arguments[0]);
};
validateValues(1, true, 'Settings');
```

Question

What shows in the console?

Answer

1

```
var validateValues = function () {
    arguments[0] = 9;
    console.log(arguments[0]);
};
validateValues(1, true, 'Settings');
```

Question

What shows in the console?

Answer

9

```
var validateValues =
  function validateValuesFn () {
    console.log(arguments.callee);
};
validateValues(1, true, 'Settings');
```

Question

What shows in the console?

Answer

```
function validateValues
Fn() {
  console.log(arguments
.callee);
}
```

Summary

- Global Functions
- Math
- String
- arguments

Overview

Error Handling and Promises

Strict Mode

JSLint and JSHint

Modular JavaScript

Error Handling

try, catch, finally

```
try {
    throw new Error('Custom Error');
}
catch (e) {
    console.log(e.name + ' - ' + e.message);
}
```

Question

What shows in the console?

Answer

```
Error - Custom Error
```

```
try {
    throw new Error('Custom Error');
}
catch (e) {
    console.log(e.name + ' - ' + e.message);
}
finally {
    console.log('Finally done.');
}
```

Question

What shows in the console?

Answer

```
Error - Custom Error
Finally done.
```

```
try {
    var i = 42;
}
catch (e) {
    console.log(e.name + ' - ' + e.message);
}
finally {
    console.log('Finally done.');
}
```

Question

What shows in the console?

Answer

```
Finally done.
```

```
try {
    throw new RangeError('Range Error');
}
catch (e) {
    if (e instanceof RangeError)
        console.log(e.name);
}
```

Question

What shows in the console?

Answer

`RangeError`

Promises and Observables

Strict Mode

```
var actNow = function() {
    'use strict';
    newVar = 9;
};
actNow();
```

Question

What shows in the console?

Answer

```
Uncaught ReferenceError
: newVar is not defined
```

```
var actNow = function() {
    newVar = 9;
};
actNow();
```

Question

What shows in the console?

Answer

```
(no error)
```

```
var actNow = function() {
    var newVar = 9;
    delete newVar;
};
actNow();
```

Question

What shows in the console?

Answer

(no error)

```
var actNow = function () {
    'use strict';
    var newVar = 9;
    delete newVar;
};
actNow();
```

Question

What shows in the console?

Answer

```
Uncaught SyntaxError:
Delete of an unqualified
identifier in strict
mode.
```

```
var actNow = function () {
    var fn = function(a, a) {
    };
};
actNow();
```

Question

What shows in the console?

Answer

```
(no error)
```

```
var actNow = function () {
    'use strict';
    var fn = function(a, a) {
    };
};
actNow();
```

Question

What shows in the console?

Answer

```
Uncaught SyntaxError:
Duplicate parameter name
not allowed in this
context
```

```
var obj = {};
with (obj) {

}
```

Question

What shows in the console?

Answer

```
(no error)
```

```
'use strict';
var obj = {};
with (obj) {

}
```

Question

What shows in the console?

Answer

```
Uncaught SyntaxError:
Strict mode code may not
include a with statement
```

JSLint and JSHint

```
var MyClass = function () {
    'use strict';
    this.id = 9;
};
var c = new MyClass();
console.log(c.id);
```

Question

What shows in the console?

Answer

```
Unexpected 'this'.

Undeclared 'console'.
```

```
var MyClass = function () {
    this.id = 9;
};
var c = new MyClass();
console.log(c.id);
```

Question

What shows in the console?

Answer

```
Expected 'use strict'
before 'this'.
```

```
var myClass = function () {
    'use strict';
    id = 9;
};
var c = new myClass();
console.log(c.id);
```

Question

What shows in the console?

Answer

`Undeclared 'id'.`

Modular JavaScript

JavaScript Module Options

AMD

CommonJS

ES 6

SystemJS

Summary

- Error Handling
  - try, catch, finally
  - promises and observables
- Strict Mode
- JSLint and JSHint
- Modular JavaScript