

JS Briefing

Total points 45/50 ?

part- 1

Email *

ashwiniraghuveer41@gmail.com



✓ If **isHungry** equals **true**, which of the following expressions evaluates to **true**? *5/5

- ☐ !isHungry === true
- ☐ !isHungry
- ☐ isHungry === false
- ☒ isHungry !== false



Feedback



Correct!



✓ What will the following code print to the console? *

5/5

```
let num = 10;  
num *= 3;  
console.log(num);
```

- ☐ 'num'
- ☒ 30
- ☐ 3
- ☐ 10



Feedback



*Correct! *= will multiply the num by 3 and then reassign the value of num to that result.*



✓ What is the correct way to call the **random** method on the **Math** global object? *5/5

- ☐ Math(random)
- ☒ Math.random()
- ☐ random.Math()
- ☐ math.random()



Feedback



Nice work! This is the correct syntax.



✓ What is the outcome of this statement? *

```
console.log('hi!'.length);
```

5/5

- ☒ 3 is printed to the console.
- ☐ 'hi!'.length will be printed to the console.
- ☐ 1 is printed to the console.
- ☐ hi! is printed to the console.



Feedback



Nice work! .length will access the length property of hi! which is 3 characters long.



✓ What is string interpolation? *

5/5

- ☐ Changing the value of a variable.
- ☒ Using template literals to embed variables into strings.
- ☐ Joining multiple strings together using operators like +
- ☐ Printing a string to the console.



Feedback



Correct! String interpolation is when we insert, or interpolate, variables into strings using template literals.



✓ What will the code block log to the console? *

5/5

```
let groceryItem = "apple";

switch (groceryItem) {
  case "tomato":
    console.log("Tomatoes are $0.49");
    break;
  case "lime":
    console.log("Limes are $1.49");
    break;
  case "papaya":
    console.log("Papayas are $1.29");
    break;
  default:
    console.log("Invalid item");
    break;
}
```

- ☐ Tomatoes are \$0.49
- ☐ Papayas are \$1.29
- ☒ Invalid item
- ☐ Limes are \$1.49



Feedback





Correct! Since `groceryItem = "apple"`, it does not match any of the cases, so the default block will run.

✓ How would you properly refactor this code block using the ternary operator? *5/5

```
if (walkSignal === 'Walk') {  
  console.log('You may walk!');  
} else {  
  console.log('Do not walk!');  
}
```

- ☐ `walkSignal ? console.log('You may walk!') : console.log('Do not walk!');`
- ☐ `walkSignal === 'Walk' ? ('You may walk!') : ('Do not walk!');`
- ☒ `walkSignal === 'Walk' ? console.log('You may walk!') : console.log('Do not walk!');` ✓
- ☐ `walkSignal === 'Walk' : console.log('You may walk!') : console.log('Do not walk!');`

Feedback



Correct!



✓ What is the correct way to call a string's built-in method? *

5/5

- ☐ toUpperCase('codecademy');
- ☐ 'codecademy'.toUpperCase;
- ☒ 'codecademy'.toUpperCase();
- ☐ toUpperCase('codecademy');



Feedback



Nice work! .toUpperCase() is appended to the string to call it.



✗ What will the following code log to the console? *

.../5

```
let needTacos = true;
```

```
if (needTacos) {  
  console.log("Finding tacos");  
} else {  
  console.log("Keep on keeping on!");  
}
```

☐ Keep on keeping on!

☒ Finding tacos

✗

No correct answers



✓ What will the code block log to the console? *

5/5

```
let runTime = 35;  
let runDistance = 3.5;  
  
if (runTime <= 30 && runDistance > 3.5) {  
  console.log("You're super fast!");  
} else if (runTime >= 30 && runDistance <= 3) {  
  console.log("You're not making your pace!");  
} else if (runTime > 30 || runDistance > 3) {  
  console.log("Nice workout!");  
} else {  
  console.log("Keep on running!");  
}
```

- ☒ Nice workout! ✓
- ☐ You're not making your pace!
- ☐ You're super fast!
- ☐ isHungry !== false

Feedback



Correct!



This form was created inside of Mr. & Mrs. Cloud.

Google Forms



