

Function Hoisting in JavaScript Quiz

```
hello();

var hello = function() {
  console.log("hello!");
};
```

What type of error will be thrown when the above code snippet is run?

☐ ReferenceError: Cannot access 'hello' before initialization

☐ No error will be thrown from the above code snippet.

☒ TypeError: hello is not a function

EXPLANATION

In the above code snippet the named var declared variable is hoisted to the top of the scope with the value of `undefined`. The first line of the code snippet above will then attempt to invoke `undefined` resulting in a `TypeError` because the value of `hello` is not a function and therefore cannot be invoked.

```
let hello = "hello";

function hello(num) {
  console.log("hello!");
}

console.log(hello);
```

What will happen when the above code snippet is run?

☐ TypeError: hello is not a function

☐ "hello" will be printed to the console

☒ `ReferenceError: Identifier 'hello' has already been declared`

EXPLANATION

Attempting to define a `let` declared variable and a function declaration with the same name in the same scope will throw a `ReferenceError` because the name cannot be declared twice.

```
console.log(goodNight());
```

```
var goodNight = function() {  
  return "Good Night!";  
};
```

What will happen when the above code snippet is run?

☐ "Good Night" will be printed to the console

☐ `ReferenceError: Cannot access 'goodNight' before initialization`

☒ `TypeError: goodNight is not a function`

EXPLANATION

The `goodNight` function is a function expression defined using `var`. A `var` declared variable will have its name hoisted to the top of its scope and its value set to `undefined`. So in the above code snippet we are attempting to invoke `undefined` so we receive a `TypeError`.

```
console.log(goodNight());
```

```
let goodNight = function goodNight() {
```

```
    return "Good Night!";  
};
```

What will happen when the above code snippet is run?

- ☐ TypeError: goodNight is not a function
- ☒ ReferenceError: Cannot access 'goodNight' before initialization
- ☐ "Good Night" will be printed to the console

EXPLANATION

The `goodNight` function is a function expression defined using `let`. Since any `let` variable declared variable won't be accessible until the value of the function is assigned we receive a `ReferenceError`.

```
console.log(shoutWord("apple"));
```

```
function shoutWord(word) {  
    return word.toUpperCase();  
}
```

What will happen when the above code snippet is run?

- ☐ ReferenceError: Cannot access 'shoutWord' before initialization
- ☒ "APPLE" will be printed to the console
- ☐ TypeError: shoutWord is not a function

EXPLANATION

The `shoutWord` function is a named function declaration so it will be hoisted in memory and available in the above scope.

Retake Quiz