Course: _____ Name: _____ Date: _____

Part I: Intro (4)

0. When adding JavaScript to a web page, where can you put your JavaScript code?

[ ] In a separate file that ends in `.js`

[ ] Inside `<link>` tags at the `<head>`

[ ] In `<script>` just before the closing `</body>`

[ ] Both (0) and (2)

1. Why should you put the HTML `<script>` before the closing `</body>` tag?

[ ] It's the only way to add Javascript to HTML

[ ] You shouldn't place your `<script>` tag before the closing `</body>`

[ ] It's recommended because that way the html from your page will be loaded before the Javascript code is read

[ ] Otherwise you will not be able to debug your code

2. What will `console.log("Westernization complete");` do?

[ ] Add a paragraph to the page with the text "Westernization complete"

[ ] Print the text "Westernization complete" to the javascript console

[ ] Create a variable with the string "Westernization complete"

[ ] Return the string "Westernization complete" from the function

3. Which of the following JavaScript commands tells a browser to open a dialog box with a message?

[ ] document.write()

[ ] window.alert()

[ ] console.log()

[ ] window.prompt()

Part II: Variable (5)

0. Which of the following is **NOT a valid variable name in JavaScript??**

[ ] stepsToFinsh

[ ] steps_to_finish

[ ] steps-to-finish

[ ] stepstofinish1

1. To **declare a variable with the name `score` in Javascript we can write:**

[ ] $score

[ ] variable score

[ ] def score

[ ] var score

2. What is the difference between `let` and `const`

[ ] `let` only saves a variable for 2 minutes, `const` saves the variable until the program stops running

[ ] `let` allows us to re-assign a variable, `const` will throw an error if we try to reassign the variable

[ ] There is no such keyword `let` in javascript, `const` will throw an error if we try to reassign the variable

[ ] `let` works with any type of variable, `const` only works with functions

3. Which of the following declares a variable named `count` and stores the number `10` in it?

[ ] let count = 10

[ ] count = 10

[ ] const count = 10

[ ] Both (0) and (2)

4. To **assign a variable in Javascript, we use:**

[ ] /

[ ] :

[ ] =

[ ] Both (0) and (2)

Part III: String (5)

0. Which of the following is a valid string:

[ ] 'You've got to see this'

[ ] "You've got to see this"

[ ] "You've got to see this'

[ ] None of the above

1. To escape a character in JS we would write:

[ ] `It${'}s such a boring day`

[ ] 'It\'s such a boring day'

[ ] 'It's such a boring day'

[ ] 'It"'"s such a boring day'

2. To put two strings together in JS we can use the following operator between them:

[ ] 'Hello' ++ ' World'

[ ] 'Hello' . ' World'

[ ] 'Hello' += ' World'

[ ] 'Hello' + ' World'

3. String **interpolation is another way to put two strings together in ES6. How can we print "Hello World!", assuming that we have declared a variable `let who = "World"`?**

[ ] `Hello who`

[ ] "Hello 'who'"

[ ] `Hello ${who}`

[ ] "Hello ${who}"

4. Assuming that we have declared a variable `let message = "What a GREAT day!"`, which statement will return `'what a great day!'`?

[ ] lowerCase(message)

[ ] message.toLowerCase()

[ ] toLowerCase(message)

[ ] message.lowerCase()

Part IV: Number (6)

0. Which of the following adds 15 to `sum` and reassigns the result back to `sum`?

[ ] sum += 15

[ ] sum + 15

[ ] sum = sum + 15

[ ] Both (0) and (2)

1. `%` is a mathematical operator in JS

[ ] true [ ] false

2. To multiply and divide numbers in Javascript we use:

[ ] `X` for multiplication, `/` for division

[ ] `/` for multiplication, `*` for division

[ ] `*` for multiplication, `/` for division

[ ] `*` for multiplication, `%` for division

3. Which of the following JavaScript methods takes a string and tries to convert it to a float?

[ ] parseInt()

[ ] parseFloat()

[ ] Number.toFixed()

[ ] Math.round()

4. Math.ceil(5.2) will return:

[ ] 0.2

[ ] 6

[ ] 5

[ ] NaN

5. parseInt() and parseFloat() do the same thing

[ ] true [ ] false

Part V: Boolean (10)

By each expression below, mark if it's `true` or `false`

	true	false
`5 > 2`	[ ]	[ ]
`'a' === 'A'`	[ ]	[ ]
`'hello' < 'world'`	[ ]	[ ]
`1 === 1 && 2 !== 2`	[ ]	[ ]
`1 === 1 || 2 !== 2`	[ ]	[ ]
`!false`	[ ]	[ ]
`undefined`	[ ]	[ ]
`3 % 3 === 0`	[ ]	[ ]
`score && !score`	[ ]	[ ]
`4 <= [1, 2, 3, 4].length`	[ ]	[ ]

Part VI: Conditionals and Loops (Program Flow) (8)

0. You must add an `else` clause to a conditional statement.

[ ] true [ ] false

When a browser runs the pieces of code listed below, what will the user see?

Question 1

```
let lives = 0;

if (lives === 0 ) {
  alert('The game is OVER!');
} else {
  alert('I guess you\'re still alive');
}
```

[ ] An alert dialog that says 'I guess you're still alive'

[ ] This is not a valid conditional statement

[ ] An alert dialog that says 'The game is OVER!'

[ ] Nothing, this code will not run since `alert()` is not a function

Question 2

```
let lives = 0;
let score = 10;

if ( score < 5 ) {
  alert('Not quite, better luck next time.');
} else {
  alert('You passed.');
} else if ( score  > 7) {
  alert('Good job!');
}
```

[ ] An alert dialog that says 'You Passed.'

[ ] This is not a valid conditional statement

[ ] An alert dialog that says 'Good Job!'

[ ] An alert dialog that says 'Not quite, better luck next time.'

3. When should you use loops?

[ ] When you want to repeat code over and over a certain number of times.

[ ] When you want to store multiple items in an single variable.

[ ] When you want to run certain code only when a particular condition is true.

[ ] When you want to run many lines of code by simply executing a single statement.

Example A:

```
let num = 0;

while (num > 20 ) {
  alert(num);
  num += 1;
}
```

4. Why will this loop never run:

[ ] Because there's a syntax error in this code.

[ ] The counter variable `num` is never increased

[ ] The condition asks if that variable is greater than 20 and the variable is 0

[ ] Because the code doesn't use a variable named `counter`

Example B:

```
let counter = 0;

while (counter < 5) {
  console.log('The counter is now: ' + counter);
  counter += 1;
}
```

5. Which of the code blocks below does the same thing as this while statement:

[ ]
```
for (let i=1; i < 5; i += 1) {
  console.log('The counter is now: ' + i);
}
```

[ ]
```
for (let counter=0; counter < 10; counter += 1) {
  console.log('The counter is now: ' + counter);
}
```

[ ]
```
for (let i=0; i < 5; i += 2) {
  console.log('The counter is now: ' + i);
}
```

[ ]
```
for (let i=0; i < 5; i += 1) {
  console.log('The counter is now: ' + i);
}
```

6. Which of the following statements will make the JavaScript interpreter exit a loop even when the loop condition is still true?

[ ] stop;

[ ] return;

[ ] break;

[ ] continue;

7. Is it possible for a loop to run forever?

[ ] yes [ ] no

Part VII: Function (8)

0. How would you "call" a function named sayHello?

[ ] run sayHello();

[ ] sayHello;

[ ] sayHello();

[ ] function(sayHello);

1. Which of the following code snippets correctly shows how to create a function named sayHello which opens an alert dialog with the string "Hello" in it?

[ ]
```
let function = sayHello() {
  alert("Hello");
}
```

[ ]
```
function sayHello() (
  alert("Hello");
)
```

[ ]
```
function sayHello() {
  alert("Hello");
}
```

[ ]
```
function sayHello {
  alert("Hello");
}
```

Example A:

```
function daysOfWeek() {
  return 'Mon', 'Tues', 'Wed', 'Thurs', 'Fri';
}

const days = daysOfWeek();
```

2. A function can return multiple values at once, like in the example above:

[ ] true [ ] false

3. When talking about JavaScript functions, what is a "parameter"?

[ ] A variable in which the function stores information passed to it.

[ ] A value the function returns to the program when the function completes.

[ ] A value that you pass to a function when you call the function.

[ ] The scope in which a function runs.

4. Which of the following shows an example of passing **arguments to a function?**

[ ] `alert("Hello world!")`

[ ]
```
function(order){
  return order + 1
}
```

[ ] `getArea(10, 20, 'sq ft')`

[ ] Both (0) and (2)

5. When you **declare a variable inside a function, that variable is only accessible within that function.**

[ ] true [ ] false

Example B:

```
var message = "Welcome!";

function setMessage() {
  message = "Go away!";
}

setMessage();
alert(message);
```

6. Given the code above, what appears in the alert dialogue when this program runs?

[ ] "Welcome" [ ] "Go Away"

7. What is "scope" in JavaScript?

[ ] Scope is the way you give and get information from functions.

[ ] Scope is the context in which a variable can be accessed, such as within a function, or within the global scope of the entire program.

[ ] Scope is the list of functions in the currently running program.

[ ] Scope is used to determine when a function returns a value.

Part VIII: Data Type (20)

By each expression below, write the letter that represents its data type

b = Boolean, f = Function, n = Number, o = Object, s = String, u = Undefined

[ ] `42`

[ ] `'42'`

[ ] `{ value: 42 }`

[ ] `function(value){ return value; }`

[ ] `true`

[ ] `hello`

[ ] `"hello"`

[ ] `3 + 2 === 5`

[ ] `[1, 2, 3, 4, 5, 6]`

[ ] `3 % 3`

[ ] `` `Hello ${6}` ``

[ ] `3 + "3"`

[ ] `NaN`

[ ] `['good', 'job'].join(' ')`

[ ] `true && false`

[ ] `[1].push(6)`

[ ] `Math`

[ ] `Math.random()`

[ ] `"Hello".split('')`

[ ] `['a', 'b', 'c'].map(letter => letter + '1')`

Part IX: Array Basics (5)

0. Given the following array `const numbers = [ 1, 44, 32, 55, 12, 17];` which snippet will print out the number 17 to the console?

[ ] `console.log(numbers[6])`

[ ] `console.log(numbers)`

[ ] `console.log(numbers[5])`

[ ] `console.log(numbers(5))`

1. Which array method adds an item to the end of an array?

[ ] pop()

[ ] push()

[ ] shift()

[ ] unshift()

2. Which array method returns the first item from an array and removes it from the array?

[ ] pop()

[ ] push()

[ ] shift()

[ ] unshift()

Example A:

```
let temperatures = [ 16, 34, 26, 19, 10];
console.log( temperatures.indexOf( 19 ) );
```

3. When the following code runs what will appear in the JavaScript console?

[ ] -1

[ ] 3

[ ] 0

[ ] 4

4. Which of the following is an example of a two-dimensional array?

[ ]
```
let scores = [
  [ "Andrew", 89, 99, 100 ],
  [ "Lora", 100, 97, 90 ],
  [ "Sam", 75, 80, 82 ],
  [ "June", 77, 81, 80 ]
];
```

[ ]
```
let numbers = [ 1, 2, 3, 4, 5 ];
```

[ ]
```
let monsters = [
  { name: "Godzilla" },
  { name: "Mothra" },
  { name: "Son of Godzilla" }
]
```

[ ]
```
var booleans = [ true, false ];
```

Good Luck!