1. What is the starting point of a basic `for` loop in JavaScript?

   - 
     **A)** `for (let i = 1; i < 10; i++)`

   - 
     B) `for (let i = 0; i < 10; i++)`

   - 
     C) `for (let i = 0; i <= 10; i++)`

   - 
     D) `for (let i = 1; i <= 10; i++)`

2. What part of the `for` loop controls how many times the loop executes?

   - 
     **A)** Initialization

   - 
     B) Condition

   - 
     C) Increment/Decrement

   - 
     D) All of the above

3. What will be the output of the loop?

   ```
   for (let i = 0; i < 5; i++) {
     console.log(i);
   }
   ```

   - 
     **A)** 0, 1, 2, 3, 4

   - 
     B) 1, 2, 3, 4, 5

   - 
     C) 1, 2, 3, 4

   - 
     D) 0, 1, 2, 3, 4, 5

4. Can a `for` loop be used without a condition?

   - 
     **A)** Yes

   -

B) No

5. What is the significance of the increment part of the `for` loop?

    –

        **A)** It checks the condition.

    –

        B) It initializes the loop.

    –

        C) It changes the loop variable each iteration.

    –

        D) It ends the loop.

6. What will be the output of the following code?

```
for (let i = 5; i > 0; i--) {
  console.log(i);
}
```

    –

        **A)** 5, 4, 3, 2, 1

    –

        B) 1, 2, 3, 4, 5

    –

        C) 4, 3, 2, 1

    –

        D) 5, 4, 3, 2, 1, 0

7. Can the loop variable be modified inside the `for` loop body?

    –

        **A)** Yes

    –

        B) No

8. What happens if the condition in the `for` loop is always `true`?

    –

        **A)** Loop runs indefinitely.

    –

        B) Loop will never run.

    –

        C) Loop runs once.

D) Loop exits immediately.

9. What does the loop do?

```js
for (let i = 3; i < 10; i += 2) {
  console.log(i);
}
```

**A)** 3, 5, 7, 9

B) 4, 6, 8, 10

C) 5, 7, 9

D) 3, 4, 5, 6

10. What will happen in this code?

```js
for (let i = 0; i < 3; i++) {
  let i = 5;
  console.log(i);
}
```

**A)** 0, 1, 2

B) 5, 5, 5

C) Error

D) Infinite loop

**Intermediate Level:**

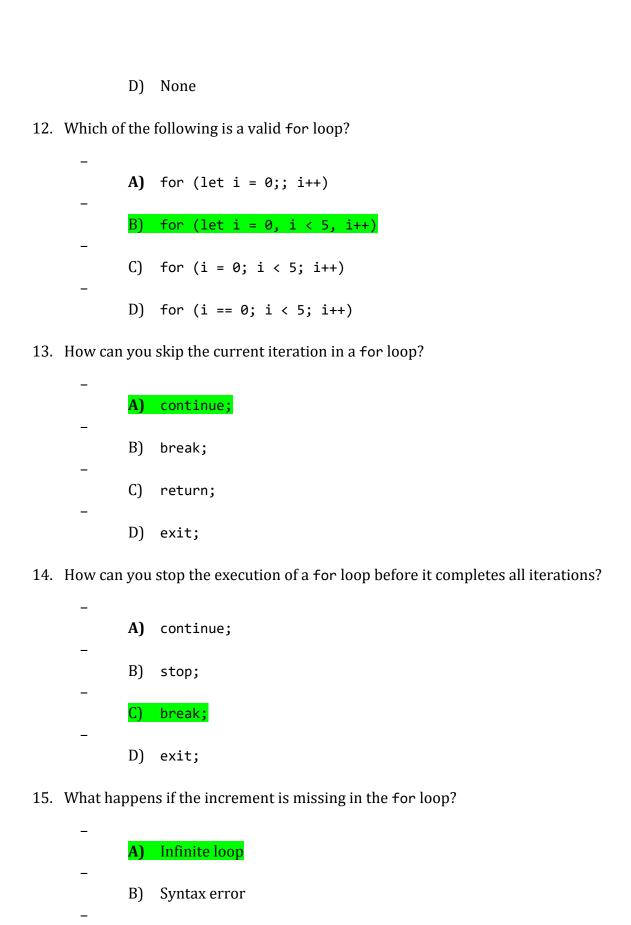11. What is the scope of the variable i in a for loop?

**A)** Global

B) Local to the block

C) Local to the function

D) None

12. Which of the following is a valid `for` loop?

**A)** `for (let i = 0;; i++)`

B) `for (let i = 0, i < 5, i++)`

C) `for (i = 0; i < 5; i++)`

D) `for (i == 0; i < 5; i++)`

13. How can you skip the current iteration in a `for` loop?

**A)** `continue;`

B) `break;`

C) `return;`

D) `exit;`

14. How can you stop the execution of a `for` loop before it completes all iterations?

**A)** `continue;`

B) `stop;`

C) `break;`

D) `exit;`

15. What happens if the increment is missing in the `for` loop?

**A)** Infinite loop

B) Syntax error

C) Loop runs once

–

D) No iteration occurs

16. What will be the output?

```
for (let i = 0; i < 5; i += 2) {
  console.log(i);
}
```

–

**A)** 0, 1, 2, 3, 4

–

B) 0, 2, 4

–

C) 0, 2

–

D) 2, 4

17. What will happen?

```
for (let i = 0; i < 3; i++)
  for (let j = 0; j < 2; j++)
    console.log(i + j);
```

–

**A)** 0, 1, 1, 2, 2, 3

–

B) 0, 1, 1, 2, 2, 3, 3

–

C) 0, 1, 1, 2, 2, 3, 3, 4

–

D) 0, 1, 2, 3, 4

18. What will the following loop output?

```
for (let i = 0; i <= 10; i++) {
  if (i === 5) break;
  console.log(i);
}
```

–

**A)** 0, 1, 2, 3, 4

–

B) 0, 1, 2, 3, 4, 5

–

C) 0, 1, 2, 3, 4, 5, 6

D) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

19. What will the following code do?

```
for (let i = 1; i <= 10; i *= 2) {
  console.log(i);
}
```

**A)** 1, 2, 4, 8

B) 1, 3, 5, 7, 9

C) 2, 4, 8

D) 1, 2, 3, 4

20. What will this for loop do?

```
for (let i = 10; i >= 0; i -= 3) {
  console.log(i);
}
```

**A)** 10, 7, 4, 1

B) 10, 8, 6, 4

C) 10, 9, 8, 7

D) Error

**Advanced Level:**
21. How many times will the following loop run?

```
for (let i = 0; i !== 10; i += 3) {
  console.log(i);
}
```

**A)** 4

B) 5

C) 6

D) Infinite

22. What will this loop print?

```javascript
for (let i = 0; i < 3; i++) {
  if (i % 2 === 0) continue;
  console.log(i);
}
```

**A)** 1, 2

B) 0, 2

C) 1

D) 0, 1, 2

23. Can the `for` loop header be empty?

```javascript
for (;;) {
  // some code
}
```

**A)** Yes

B) No

24. How can the following code be corrected?

```javascript
for (let i = 0; i < 5 i++) {
  console.log(i);
}
```

**A)** Add a semicolon after `i++`.

B) Remove `let`.

C) Add a comma between statements.

D) Add parentheses around i++.

25. What is the output?

```
for (let i = 0; i <= 3; i++, console.log(i));
```

- 
    **A)** 1, 2, 3, 4
- 
    B) 0, 1, 2, 3, 4
- 
    C) 0, 1, 2, 3
- 
    D) 1, 2

26. What will be the final value of i in this code?

```
let i;
for (i = 0; i < 5; i++) {}
console.log(i);
```

- 
    **A)** 4
- 
    B) 5
- 
    C) 6
- 
    D) Undefined

27. What will happen in this loop?

```
for (let i = 0; i < 10; i = i + 3) {
  console.log(i);
}
```

- 
    **A)** 0, 3, 6, 9
- 
    B) 1, 4, 7, 10
- 
    C) Infinite loop

D) Syntax error

28. How many times will this loop run?

```
for (let i = 5; i <= 10; i += 2) {}
```

A) 3

B) 4

C) 5

D) Infinite

29. What will the following code print?

```
for (let i = 0; i < 10; i++) {
  if (i % 3 === 0) {
    console.log(i);
  }
}
```

A) 1, 2, 3, 6, 9

B) 3, 6, 9

C) 0, 3, 6, 9

D) 0, 2, 4, 6

30. What happens when the loop has no initialization?

```
for (; i < 5; i++) {
  console.log(i);
}
```

A) Error

B) Infinite loop

C)  0, 1, 2, 3, 4

D)  Depends on the scope of `i`

31.  What will the following code do?

```javascript
for (let i = 1; i < 20; i *= 3) {
  console.log(i);
}
```

**A)**  1, 3, 9

B)  1, 2, 4, 8

C)  1, 3, 6, 9

D)  Infinite loop

32.  How does the `for` loop differ from the `while` loop in terms of syntax?

**A)**  Initialization happens outside the loop in `while`.

B)  `while` loop is more optimized.

C)  `for` loop doesn't require a condition.

D)  `while` loop can't be nested.

33.  What will happen if there is no update statement in the `for` loop?

**A)**  Loop will run once.

B)  Infinite loop.

C)  Loop won't run at all.

D)  Syntax error.

34.  How many times does this loop run?

```
for (let i = 0; i++ < 5;) {}
```

- **A)** 4

- B) 5

- C) 6

- D) Infinite

35. What is the result of this code?

```
for (let i = 5; i; i--) {
  console.log(i);
}
```

- **A)** 4, 3, 2, 1

- B) 5, 4, 3, 2, 1

- C) 4, 3, 2, 1, 0

- D) Infinite loop

36. What will the following loop print?

```
for (let i = 1; i <= 3; i++) {
  for (let j = 1; j <= 2; j++) {
    console.log(i, j);
  }
}
```

- **A)** 1, 1; 1, 2; 2, 1; 2, 2; 3, 1; 3, 2

- B) 1, 2; 2, 2; 3, 2

- C) 1, 2; 2, 1; 3, 2

- D) Syntax error

37. How many times will the inner loop run in this code?

```
for (let i = 0; i < 4; i++) {
  for (let j = 0; j < 3; j++) {
    console.log(i, j);
  }
}
```

  –

    **A)** 4

  –

    B) 7

  –

    C) 12

  –

    D) 10

38. How many total iterations will both loops run?

```
for (let i = 1; i <= 5; i++) {
  for (let j = 1; j <= i; j++) {
    console.log(i, j);
  }
}
```

  –

    **A)** 15

  –

    B) 10

  –

    C) 12

  –

    D) 20

39. What is printed by the following loop?

```
for (let i = 2; i <= 4; i++) {
  for (let j = 1; j < i; j++) {
    console.log(i, j);
  }
}
```

  –

    **A)** 2, 1; 3, 1; 3, 2; 4, 1; 4, 2; 4, 3

  –

    B) 2, 1; 3, 2; 4, 3

C) 2, 2; 3, 3; 4, 4

D) 1, 2; 2, 3

40. How many iterations will this loop combination execute?

```javascript
for (let i = 0; i < 3; i++) {
  for (let j = i; j < 3; j++) {
    console.log(i, j);
  }
}
```

**A)** 9

B) 6

C) 5

D) 3

41. What is the final output of this nested loop?

```javascript
let result = '';
for (let i = 0; i < 3; i++) {
  for (let j = 0; j < 3; j++) {
    result += i + '' + j + ' ';
  }
}
console.log(result);
```

**A)** "000 011 022"

B) "00 01 02 10 11 12 20 21 22"

C) "012 012 012"

D) Infinite loop

42. What does this code output?

```javascript
for (let i = 1; i <= 2; i++) {
  for (let j = 1; j <= 3; j++) {
```

```
if (i + j === 4) {
  console.log(i, j);
}
}
}
```

- 

    **A)**   1, 3; 2, 2

- 

    B)   1, 1; 2, 3

- 

    C)   2, 1; 1, 2

- 

    D)   1, 2; 2, 1

43. What will happen when the following nested loop is executed?

```
for (let i = 0; i < 3; i++) {
  for (let j = 0; j < 2; j++) {
    if (j === 1) break;
    console.log(i, j);
  }
}
```

- 

    **A)**   0, 0; 1, 0; 2, 0

- 

    B)   0, 1; 1, 1; 2, 1

- 

    C)   0, 1; 1, 0

- 

    D)   Infinite loop

44. What does this loop combination output?

```
for (let i = 1; i <= 3; i++) {
  for (let j = 1; j <= 2; j++) {
    if (i === 2) continue;
    console.log(i, j);
  }
}
```

- 

    **A)**   1, 1; 1, 2; 2, 1; 2, 2; 3, 1; 3, 2

- 

    B)   1, 1; 1, 2; 3, 1; 3, 2

C)   1, 2; 2, 1; 3, 2

D)   Infinite loop

45.  What will the output be for this nested loop setup?

```
let output = '';
for (let i = 1; i <= 3; i++) {
  for (let j = i; j <= 3; j++) {
    output += i + '' + j + ' ';
  }
}
console.log(output);
```

**A)**  "11 12 13 22 23 33"

B)   "12 23 34"

C)   "12 13 23"

D)   Infinite loop

46.  How many times will the inner loop execute in total?

```
for (let i = 0; i < 4; i++) {
  for (let j = i; j >= 0; j--) {
    console.log(i, j);
  }
}
```

**A)**   10

B)   5

C)   4

D)   8

47.  What does this code print?

```
let counter = 0;
for (let i = 0; i < 4; i++) {
```

```
for (let j = 0; j < 2; j++) {
    counter++;
  }
}
console.log(counter);
```

  &mdash;

      **A)** 2

  &mdash;

      B) 4

  &mdash;

      C) 8

  &mdash;

      D) 6

48. What is printed by this code?

```
for (let i = 1; i <= 2; i++) {
  for (let j = 1; j <= 2; j++) {
    if (i === j) break;
    console.log(i, j);
  }
}
```

  &mdash;

      **A)** 1, 2; 2, 1

  &mdash;

      B) 1, 1; 2, 2

  &mdash;

      C) 2, 1

  &mdash;

      D) Infinite loop

49. What happens in the following nested loop?

```
for (let i = 3; i > 0; i--) {
  for (let j = 3; j > i; j--) {
    console.log(i, j);
  }
}
```

  &mdash;

      **A)** 3, 2; 2, 1; 1, 0

  &mdash;

      B) 2, 3; 1, 2; 1, 3

C) 1, 2; 1, 3

D) 3, 2; 3, 1

Ans:-23 13 12

50. How many total times does the loop run in this code?

```javascript
for (let i = 0; i < 3; i++) {
  for (let j = 0; j < 3; j++) {
    if (i === 2) break;
    console.log(i, j);
  }
}
```

**A)** 9

B)  6

C) 4

D) 5