## Truthy and Falsy Values in JavaScript

JavaScript evaluates values in logical contexts (e.g., conditions in `if` statements) as either **truthy** or **falsy**. These concepts help determine how expressions are evaluated in control flow structures.

---

## 1. Falsy Values

A falsy value is one that JavaScript treats as `false` when evaluated in a Boolean context. There are **exactly 7 falsy values** in JavaScript:

1. `false`
    - The Boolean `false` itself.

```
if (false) console.log("Falsy"); // Does not execute
```

2. `0` (Number Zero)
    - Includes both `+0` and `-0`.

```
if (0) console.log("Falsy"); // Does not execute
```

3. `NaN` (Not-a-Number)
    - The result of invalid mathematical operations.

```
if (NaN) console.log("Falsy"); // Does not execute
```

4. `""` (Empty String)
    - Includes both `''` and `""`.

```
if ("") console.log("Falsy"); // Does not execute
```

5. `null`
    - Represents the intentional absence of a value.

```
if (null) console.log("Falsy"); // Does not execute
```

6. `undefined`

- The default value of variables that are declared but not initialized.

```
if (undefined) console.log("Falsy"); // Does not execute
```

7. `document.all`
   - A legacy object for compatibility with older browsers (treated as `undefined` in logical contexts).

```
if (document.all) console.log("Falsy"); // Does not execute
```

---

## 2. Truthy Values

A truthy value is anything that is **not falsy**. In JavaScript, most values are truthy. Here are examples of common truthy values:

### Truthy Examples

1. **Non-zero Numbers**
   - Both positive and negative numbers are truthy.

```
if (42) console.log("Truthy"); // Executes
if (-42) console.log("Truthy"); // Executes
```

2. **Non-empty Strings**
   - Strings with any character, including a single space.

```
if ("hello") console.log("Truthy"); // Executes
if (" ") console.log("Truthy"); // Executes
```

3. **Objects**
   - Any object, including empty ones (`{}` or `[]`).

```
if ({}) console.log("Truthy"); // Executes
if ([]) console.log("Truthy"); // Executes
```

4. `Infinity`
   - Both `Infinity` and `-Infinity` are truthy.

```
if (Infinity) console.log("Truthy"); // Executes
if (-Infinity) console.log("Truthy"); // Executes
```

5. **Functions**
    - Any function is truthy.

```
if (function() {}) console.log("Truthy"); // Executes
```

6. **Other Non-standard Truthy Values**
    - Strings like `"false"` or `"0"` (non-empty strings are always truthy).

```
if ("false") console.log("Truthy"); // Executes
if ("0") console.log("Truthy"); // Executes
```

---

## 3. Examples in Conditional Contexts

### Example 1: Falsy Values

```
let values = [false, 0, NaN, "", null, undefined, document.all];

values.forEach(value => {
    if (!value) {
        console.log(`${value} is Falsy`);
    }
});
```

**Output:**

```
false is Falsy
0 is Falsy
NaN is Falsy
 is Falsy
null is Falsy
undefined is Falsy
[object HTMLAllCollection] is Falsy
```

---

### Example 2: Truthy Values

```
let values = [1, -1, "hello", {}, [], Infinity, " "];

values.forEach(value => {
    if (value) {
        console.log(`${value} is Truthy`);
    }
});
```

**Output:**

```
1 is Truthy
-1 is Truthy
hello is Truthy
[object Object] is Truthy
 is Truthy
Infinity is Truthy
  is Truthy
```

---

## 4. Practical Use of Truthy and Falsy Values

### Default Values with Logical Operators

You can use `||` to assign default values:

```
let name = "";
let displayName = name || "Guest";
console.log(displayName); // "Guest"
```

### Short-circuiting with `&&`

Execute code only if a value is truthy:

```
let isLoggedIn = true;
isLoggedIn && console.log("Welcome back!"); // Executes
```

---

## Summary Table

| Value | Type | Falsy/Truthy |
|---|---|---|
| `false` | Boolean | Falsy |
| `0`, `-0` | Number | Falsy |

| Value | Type | Falsy/Truthy |
|---|---|---|
| `NaN` | Number | Falsy |
| `""`, `''` | String | Falsy |
| `null` | Null | Falsy |
| `undefined` | Undefined | Falsy |
| `document.all` | Special Object | Falsy |
| Non-zero Numbers | Number | Truthy |
| Non-empty Strings | String | Truthy |
| Objects | Object | Truthy |
| Arrays | Object | Truthy |
| `Infinity`, `-Infinity` | Number | Truthy |
| Functions | Object | Truthy |

Understanding these values will help you avoid common pitfalls in logical conditions and write cleaner, more predictable JavaScript code!