JavaScript Progressive Enhancement

Back to JS page

Table of Contents

- JavaScript Progressive Enhancement
  - Start With HTML
    - Example 1
  - Add CSS for Better Design
    - Example 2
  - Add JavaScript as an Enhancement
    - Example 3
  - Graceful Degradation vs Progressive Enhancement
  - Modern HTML Helps
  - Document
  - Reference

Start With HTML

Progressive enhancement starts with a solid HTML foundation. Your web page should work and be accessible with HTML alone, before adding any CSS or JavaScript.

This ensures that:

- Content is accessible to all users and devices
- Search engines can index your content
- The page works in older browsers
- Screen readers can navigate the content

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Progressive Enhancement</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Start With HTML</h4>
<p>This page works with HTML only.</p>

<!-- HTML-only navigation -->
<nav>
  <a href="#">Home</a> |
  <a href="#">Products</a> |
  <a href="#">About</a> |
  <a href="#">Contact</a>
</nav>

<!-- HTML-only form -->
<form action="/action_page.php">
  <label for="email">Subscribe to newsletter:</label>
  <input type="email" id="email" name="email" required>
  <input type="submit" value="Subscribe">
</form>

</body>
</html>
```

Example 1

Result View Example

Add CSS for Better Design

After the HTML foundation is solid, add CSS to improve the visual design and layout of the page. CSS enhances the appearance without affecting functionality:

```
<!DOCTYPE html>
<html>
<head>
<style>
.card {
  background-color: #f1f1f1;
  border-radius: 8px;
  padding: 20px;
  margin: 10px 0;
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);
}
.btn {
  background-color: #04AA6D;
  color: white;
  padding: 10px 20px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}
</style>
</head>
<body>

<h2>JavaScript Progressive Enhancement</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Add CSS for Better Design</h4>

<div class="card">
  <h5>Card Title</h5>
  <p>This card is styled with CSS, but the HTML content is still accessible without it.</p>
  <button class="btn" onclick="alert('Button clicked!')">Click me</button>
</div>

</body>
</html>
```

Example 2

Result View Example

Add JavaScript as an Enhancement

After HTML and CSS are in place, add JavaScript to enhance the user experience. JavaScript should add functionality on top of what already works:

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Progressive Enhancement</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Add JavaScript as an Enhancement</h4>

<!-- HTML foundation: basic form validation -->
<form id="myForm" action="/action_page.php">
  <label for="fname">Name:</label>
  <input type="text" id="fname" name="fname" required>
  <br><br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required>
  <br><br>
  <input type="submit" value="Submit">
</form>

<p id="demo"></p>

<script>
// JavaScript enhancement: add live validation feedback
let form = document.getElementById("myForm");
let fname = document.getElementById("fname");
let email = document.getElementById("email");

fname.addEventListener("input", function() {
  if (fname.value.length > 0) {
    fname.style.borderColor = "green";
  } else {
    fname.style.borderColor = "red";
  }
});

email.addEventListener("input", function() {
  if (email.value.includes("@")) {
    email.style.borderColor = "green";
    document.getElementById("demo").innerHTML = "Email looks valid!";
  } else if (email.value.length > 0) {
    email.style.borderColor = "orange";
    document.getElementById("demo").innerHTML = "Email needs @ symbol";
  } else {
    email.style.borderColor = "";
    document.getElementById("demo").innerHTML = "";
  }
});
</script>
```

```
</body>
</html>
```

Example 3

Result View Example

Graceful Degradation vs Progressive Enhancement

Approach	Description
Progressive Enhancement	Start with a basic HTML version that works everywhere. Add CSS and JavaScript to enhance the experience for capable browsers. Content is always accessible.
Graceful Degradation	Build the full-featured version first (with JavaScript). Then try to make it work in older browsers. The user may get a broken experience if JavaScript fails.

Progressive enhancement is generally preferred because it ensures your content is always accessible, regardless of the user's browser or device.

Modern HTML Helps

Modern HTML5 provides many built-in features that reduce the need for JavaScript:

- `<input type="email">` - Built-in email validation
- `<input type="date">` - Native date picker
- `<input type="range">` - Native slider
- `<details>` / `<summary>` - Native expandable sections
- `<dialog>` - Native modal dialogs
- `<progress>` - Native progress bars

Using these HTML features means your page works without JavaScript, following the progressive enhancement approach.

Document

Document in project

You can Download PDF file.

Reference

- W3Schools JavaScript Progressive Enhancement