



[Trent University](#)¹⁰⁰ [For Further Reading](#)¹⁰⁰ [Mozilla Web Documentation](#)¹⁰⁰ [Strategies](#)¹⁰⁰

Progressive Enhancement

With a multitude of browsers comes a multitude of levels of support for the web standards. In fact, no browser has implemented all the standards 100%, and there are always new technologies that are slowly gaining steam. Furthermore, users can set their own browser preferences, so they may have a browser that supports JavaScript but have chosen to turn it off. The point here is that we are faced with a wide range of browser capabilities — from only basic HTML support to all the bells and whistles.

Progressive enhancement is one strategy for dealing with unknown browser capabilities (see **Note**). When designing with progressive enhancement, you start with a baseline experience that makes the content or core functionality available to even the most rudimentary browsers or assistive devices. From there, you layer on more advanced features for the browsers that can handle them. You might finish with some “nice to have” effects, like animation or wrapping text around images in interesting shapes, that enhance the experience for users with the most advanced browsers, but aren't really critical to the brand or message.

Strategies

Progressive enhancement is an approach that informs all aspects of page design and production, including HTML, CSS, and JavaScript:

Authoring strategy

When an HTML document is written in logical order and its elements are marked up in a meaningful way, it will be usable on the widest range of browsing environments, including the oldest browsers, future browsers, and mobile and assistive devices. It may not look exactly the same, but the important thing is that your content is available. It also ensures that [search engines like Google](#)¹⁰⁰ will catalog the content correctly. A clean HTML document with its elements accurately and thoroughly described is the foundation for accessibility.

Styling strategy

You can create layers of experience simply by taking advantage of the way browsers parse style sheet rules. Without going into too much technical detail, you can write a style rule that makes an element background red, but also include a style that gives it a cool gradient (a blend from one color to another) for browsers that know how to render gradients. Or you can use a cutting-edge CSS selector to deliver certain styles only to cutting-edge browsers. The knowledge that browsers simply ignore properties and rules they don't understand gives you license to innovate without bringing older browsers to their knees. You just have to be mindful of styling the baseline experience first, then adding improvements once the minimum requirements are met.

Scripting strategy

As with other web technologies, there are discrepancies in how browsers handle JavaScript (particularly on non-desktop devices), and some users opt to turn it off entirely. The first rule in progressive enhancement is to make sure basic functionality — such as linking from page to page or accomplishing essential tasks like data submission via forms — is intact even when JavaScript is off. In this way, you ensure the baseline experience, and enhance it when JavaScript is available.

Examples

```
:root {  
  --accent-color: #0061d1;  
}  
  
span.term {  
  color: cornflowerblue;  
  color: var(--accent-color);  
}
```

For Further Reading

- There is no better introduction to the progressive enhancement approach than the book [Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement](#) , 2nd Edition, by Aaron Gustafson (New Riders).
- [The Uncertain Web: Web Development in a Changing Landscape](#) by Rob Larson (O'Reilly).
- Once you have more chops, the book [Designing with Progressive Enhancement](#) by Todd Parker, Patty Toland, Scott Jehl, and Maggie Costello Wachs (New Riders) is an excellent deep-dive into techniques and best practices. Read more about it at filamentgroup.com/dwpe/ ¹⁰⁰.

Designing with FlexBox - Neema Kanyi

[Home](#)¹⁰⁰ [Products](#)¹⁰⁰ [Services](#)¹⁰⁰ [Strategies](#)¹⁰⁰