

# What is the difference between progressive enhancement and graceful degradation?

Asked 14 years, 8 months ago   Modified 3 years, 1 month ago

Viewed 23k times



66

I'm confused on what the difference is between **progressive enhancement** and **graceful degradation**. To me they seem like the same thing.



Can you please explain to me the differences between the two and in which situation I would use one over the other?



javascript

progressive-enhancement

graceful-degradation

Share

Improve this question

Follow

edited Oct 27, 2021 at 0:05



Peter Mortensen

31.6k ● 22 ● 109 ● 133

asked Mar 31, 2010 at 4:05



zeckdude

16.2k ● 44 ● 146 ● 194

This is my understanding as well. – SeanJA Mar 31, 2010 at 4:07

## 7 Answers

Sorted by:

Highest score (default)



They *are* almost exactly the same thing, but they differ in context.

89



There is a class of browsers called "A Grade Browsers".

These are your typical audience members that (probably) make up the majority of your visitors. You'll start with a baseline of these users. Call this **best modern practices**.



If you want to **enhance** the experience for anyone happening to use Firefox 3.6 or Safari 4 or some other whizbang developer nightly WebKit what have you, you'll want to do awesome things like

- rounded corners via CSS
- shadowed text (but please god, not *too* much)
- drop shadows (see above parenthetical)

These make your site kick-ass looking, but won't break it. This is **progressive enhancement**. Embracing the future from the point of **best practices**.

On the other hand, your niche Nintendo site attracts a fair number of [Internet Explorer 5](#) users. Poor you, but you also want to make sure they keep coming back. You might provide an alternative to your Ajax behavior by including the Ajax script in an external file and if their JavaScript isn't turned on, maybe your links update the

whole page. Etc. From the point of view of **best modern practices**, you're making sure that certain historical markets are being catered by **some semblance of a functional site**. This is **graceful degradation**.

They are mostly identical, but they differ in terms of priority for many development teams: progressive enhancement is quite nice if you have the time, but graceful degradation is often **necessary**.

Share Improve this answer

edited Nov 5, 2021 at 20:50

Follow

answered Mar 31, 2010 at 4:17



Alex Mcp

19.3k ● 12 ● 61 ● 94



37



If your site looks equally good on all browsers, but certain browsers get, say, dancing ponies because they support dancing ponies, then that's progressive enhancement. It works in all browsers, but certain browsers get something extra. Usually that term is applied to certain JavaScript features that may enhance usability beyond "raw HTML".



If your site only looks the way you intend to on browsers that fully support, say, CSS 3, and Internet Explorer 8- will display the same page without, say, rounded corners, then that's graceful degradation. The site is really meant for state-of-the-art browsers, but it's still usable in older browsers, just not as fancy.

In the end, they're really the same thing, looked at from two different perspectives.

Share Improve this answer

Follow

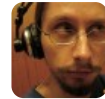
edited Nov 5, 2021 at 20:14



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Mar 31, 2010 at 4:14



deceze ♦

521k ● 88 ● 793 ● 936



The direction from a chosen **baseline** for each concept is different.

25



Graceful degradation starts at an ideal user experience level and decreases depending on user agent capabilities down to a minimum level, catering for agents that don't support certain features used by the baseline.



Progressive enhancement starts at a broad minimum user experience and increases depending on user agent capabilities up to a more capable level, catering for agents that support more advanced features than the baseline.

I think that one could employ both concepts if time/budget permit. If not then graceful degradation would be preferred.

Share Improve this answer

Follow

edited Nov 5, 2021 at 20:28



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Mar 31, 2010 at 9:56



Dean Burge

3,461 ● 28 ● 22



11



Although I agree with both Alex Mcp and deceze in a way, the terms "graceful degradation" and "progressive enhancement" have slightly different meanings from where I stand.

**Graceful degradation**, a lot of the time (in my opinion), seems to be more of a stick to beat an application into submission after it's been built badly in the first place in my experience. Like someone building out some vast JavaScript object that provides the user with something really cool to play with, until a manager comes along, tests the thing and everyone runs screaming throwing their arms around when it comes to their attention that their application doesn't work in 35% of browsers.

***"Someone better provide a fallback for this."***

**Progressive enhancement** though (and it's such a nicer term to say too) would seem to me to be more about building something that just works, on an entry level, everywhere, through the most basic methods available, to provide all the functionality that the user needs. This can then be added to with neat little unobtrusive helpers, styling, etc. that actually enhance the user experience of the application in question, rather than just make it barely usable. ***"That looks cool. Does it work in [Internet Explorer 6](#)? Oh yeah. It does."***

I think maybe giving style as an example of *both* terms in the top two answers here kind of overlooks the real underlying usability issue that **progressive enhancement** often solves by its very nature, where **graceful degradation** ignores until things go wrong.

*Rant over...*

Share Improve this answer

Follow

edited Nov 5, 2021 at 20:36



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered May 9, 2011 at 19:18



BizNuge

938 ● 1 ● 8 ● 20



7



## Graceful degradation

**Graceful degradation** is the ability of a computer, machine, electronic system or network to maintain limited functionality even when a large portion of it has been destroyed or rendered inoperative. The purpose of graceful degradation is to prevent catastrophic failure.

Graceful degradation is one solution. It is the practice of building a web site or application so it provides a good level of user experience in modern browsers. However, it will degrade gracefully for those using older browsers.

The system may not be as pleasant or as pretty, but the basic functionality will work on older systems.

A simple example is the use of 24-bit alpha-transparent PNG images. Those images can be displayed on modern browsers without problems. Internet Explorer 5.5 and Internet Explorer 6 would show the image, but transparency effects would fail (it can be made to work if necessary). Older browsers that do not support PNG would show alt text or an empty space.

Developers adopting graceful degradation often specify their browser support level, e.g. level 1 browsers (best experience) and level 2 browsers (degraded experience).

## Progressive enhancement

**Progressive enhancement** is a strategy for web design that emphasizes accessibility, semantic HTML markup, and external stylesheet and scripting technologies. Progressive enhancement uses web technologies in a layered fashion that allows everyone to access the basic content and functionality of a web page, using any browser or Internet connection, while also providing an enhanced version of the page to those with more advanced browser software or greater bandwidth.

Progressive enhancement is similar concept to graceful degradation but in reverse. The web site or application would establish a base-level of user experience for most browsers. More advanced functionality would then be added when a browser supports it.

Progressive enhancement does not require us to select supported browsers or revert to table-based layouts. We choose a level of technology; i.e. the browser must support HTML 4.01 and standard page request/responses.

Going back to our image example, we might decide that our application should be functional in all graphical browsers. We could use a lower-quality GIF images by default but replace them with 24-bit PNGs when the browser supports them.

---

## Links

Wikipedia: [Progressive enhancement](#) and [Graceful degradation \(fault tolerance\)](#)

Source: [SitePoint Blog](#)

Share Improve this answer

Follow

edited Nov 5, 2021 at 20:42



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Dec 24, 2014 at 11:59





Suresh Karia

18.2k ● 18 ● 68 ● 85



0



I find it tends to be attitudinal - are you saying "okay, my site works with [Lynx](#), users can do everything I want them to be able to do, now let's add some panache", or are you saying "okay, my site works in Firefox, now let's try to fix it for people not willing to use that/who turn off JavaScript/etc."

Share Improve this answer

Follow

edited Nov 5, 2021 at 20:27



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Mar 31, 2010 at 4:20



Damien\_The\_Unbeliever

239k ● 28 ● 355 ● 464



-1



To make it easier, just set your bar right at the top, and then you can ignore progressive enhancement. When a new feature comes out, raise your bar ;)

Or alternatively, set your bar to the lowest level ([Lynx](#) perhaps?) and just use progressive enhancement.

Share Improve this answer

Follow

edited Nov 5, 2021 at 20:38



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Mar 14, 2014 at 2:37



user3206335

35 ● 1 ● 2

---