

# Teach Yourself...Stuff OR How I Learned to Stop Worrying and Love GeekSpeak

Take charge! As Dan Quayle said, “what a waste it is to lose one’s mind.” Don’t let your job let your brain turn to mush. Grab the reins of your own continuing education, cowboy up and take your technical skills to the next level. Y’know, follow Nike’s advice.

## Ten Ways to Set the Stage

### 1 *Make Your Life Behave*

Get your life on an even keel – you can’t expect to internalize new skills when your job security is in question, or when your childcare situation is falling apart, or while you’re trying to figure out What to Do About Dad. Daily, you juggle your different roles – worker, spouse, parent, child, athlete... “student” puts another ball in the air.

### 2 *Pay Yourself First*

Financial advisors always say, “Pay yourself first.” The debt will absorb everything available – your to-do list will do the same. The tasks you have to do will expand to fill all of your available time. Set yourself a schedule and put learning first – then look at your inbox. The new skills you’re learning make you more efficient, make your output more professional, and give you confidence and energy to tackle the tedious stuff with redoubled vigor.

### 3 *Have a Playground*

When everything has to be approved by corporate, we don’t take chances – why put yourself through the headaches if you’re not sure if you’re in over your head? So have a place to try out new ideas. Some will work, some won’t. It’s not expensive. For an annual domain name registration fee (I paid \$15), and then a monthly hosting fee (I pay \$5/month), you can have complete creative control. Plus, having your own domain name marks you as serious about this stuff.

### 4 *Seek Out Like-Minded Individuals*

Talk to people whose opinion you respect. Get their feedback, trade solutions. If you’re a team of one, find an online forum. Lurk until there’s a question you’re confident about answering, then jump in and be their hero.

### 5 *Breakfast Is the Most Important Meal of the Day*

Start off your day focused on learning. When it’s out of the way, you’ve accomplished something with your day. Let the rest of the day head south, hang onto that morning’s accomplishment to keep your head above water.

### 6 *Identify Time-Suckers or Turn Nothing into Something*

If I ask you, “What are you doing?” And you say, “Nothing.” That’s a time-sucker. They’re everywhere – online games, TV shows, Chatty McTalksalot two cubicles down. Minimize their impact on your day. Pick one online game and give yourself a time limit. Follow one TV show. Have an exit strategy when you go to chat with a real talker. And then fill that time with “something” – read another chapter, add a new feature, try a new design idea.

### 7 *Mimic Ship in Front*

Find sites you like, look at their source code, and see if you can duplicate a similar look and functionality on your site. Do you understand what they did in their code? What’s missing that you have to guess at? Did they code it in the most efficient way?

### 8 *Consistent, Persistent*

Make it a part of your day every day. When my preschooler started piano lessons, his teacher told him to practice 15 minutes a day. It doesn’t seem like much time to ask for, but making it a part of the daily routine leads to real progress.

### 9 *Share and Enjoy*

Base your site’s content on something you enjoy, and it becomes a new aspect of that hobby. You’ll find yourself grinning and saying, “I have to try it out - to support my web site.” Don’t worry that your interest is too obscure a niche. Count on it, there’s a magazine out there targeted at people more crazy into it than you are.

### 10 *Practice, Practice, Practice*

If you haven’t spoken Spanish since high school, you’re going to have trouble getting past <<¿Habla inglés?>> Once you have a new skill under your belt, you need to use it. Wait on using the shortcuts (like templates, auto-typed text, code modules) until you’re really fluent, and you’ve internalized how to do it from scratch.

## Coming Soon to a Theater Near You... Figuring Out What To Learn

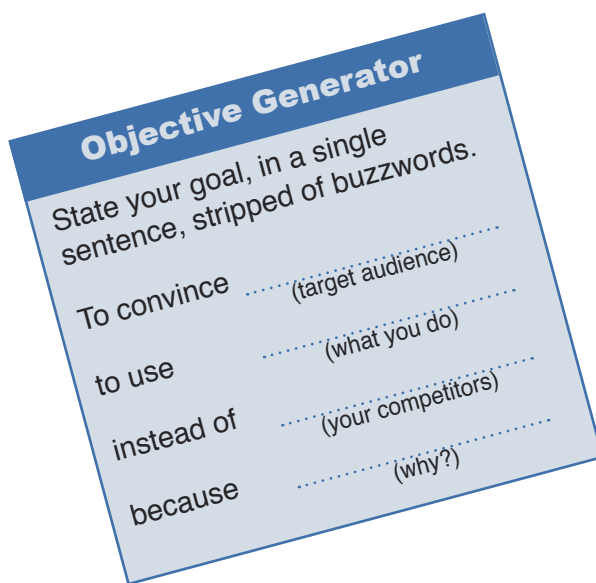
- Look at job descriptions for what you wish you did – what skills are they asking for?  
Keep a list of the skills and technologies that show up consistently.  
How close is your skillset?
- What's the industry standard?  
Talk to vendors, to competitors, to colleagues. What are the trends for your field?  
If you outsource, talk to the vendor who does the work. What software do they use?  
What equipment do they use?  
Is it practical for you to get that on your desk?

## All Your Base Are Belong to Us...Make Your Time

When you've determined what you want to learn, allocate your time. (For more information on "All Your Base", see [Wikipedia article](#).)

- Pick a top priority and dive in to learning one thing.
- Take two or three skills that interest you and divide up your week – try spending Mondays and Wednesdays on CSS and Tuesdays and Thursdays on Flash.
- Set a concrete goal and work towards it: "I want to learn enough JavaScript so that users get an alert before my form submits incomplete data."
- Give yourself a certain amount of time for each thing on your list – make this month "Learn Illustrator Month."

Choose your strategy, take chances, get messy.



## Getting Down to Brass Tacks: Finding Good Resources

- Find a book you like, then read other books by that author. Then read books by the guy who wrote the forward for that guy.
- Find a series of books that works for you.
- Do online tutorials.
- Walk through the tutorials that came bundled with your software. Sometimes the content on these can be really dry (which is the school word for "boring"), but there are people making a living deciding what's essential to know about their software. Give 'em a chance.
- Take a class. What textbook did the instructor choose? Was it good? Are there others in the series, others by the same author?
- What books are your peers reading? What's on your colleague's desk?
- What books/sites come up in the answer threads of web forums?

# How to Get the Respect and Money You Deserve Now That You're a Big Smarty

Ummm... Okay, I got nothin'.  
Somebody else is going to have  
to field this one.

*"Believe me, this would all be  
so much cooler  
if I were getting paid for it."  
-B. Trate*



3

## Tips

### The browser isn't listening to my CSS!

Follow standards – Following the W3C standards really does help your page behave a little more predictably.

Proofread it – HTML calling the CSS and the CSS definitions. Just one little typo can be the culprit.

Validate it – Run your CSS through a CSS validator – maybe you called it “normal” and it’s supposed to be “regular”

Set a border – set a border all the way around the element: it will tell you how big the browser thinks the element is, and if it doesn’t appear, you know that the browser’s not listening to the style definition at all.

### Getting your feet wet with HTML

If you’re new to HTML, then start out with “strict 4.01”. That way you won’t have to unlearn things when “transitional” fades away.

Use lowercase tags. (Current HTML can use uppers or lowers, but going forward, uppers will be deprecated and you’ll have to use lowers, so why not just start out that way?)

Remember every open tag needs a closing tag (okay, almost – a couple tags can stand on their own).

Provide all your images with alt text – a caption that a page reader reads for the vision impaired.

Provide each page with a unique title, that’s what shows up in your history and in your bookmark.

Make sure every page is connected.

Navigation – on every page – should show where you are and how to get back.

Validate your code. When your page doesn’t render the way you want it to, run it through a validator and let it do the work finding where it all falls apart.

As much as makes sense, keep the styling out of the HTML - apply the styling with CSS.

Use indents to make your code easy to read:

```
<table>
  <tr>
    <td>
      Table cell content
    </td>
  </tr>
</table>
```

Be consistent in how you name things (.htm or .html, for example).

### Getting your feet wet with Illustrator

Draw what you see, not what you know is there. Do a google-image-search and find a picture. Take it as a starting point and then modify it to fit your purposes.

Zoom in. Zoom wayyyy in. Now pick a point and use the arrow keys to nudge it to where you want it.

### Getting your feet wet with Photoshop

Learn to adjust levels – go to Image-Adjustments-Levels. It brings up a graph (geekspeak: histogram) – your image sorted into colors of pixels stacked up from darkest to lightest. Slide the little black triangle at the bottom of the histogram to the right so it’s underneath some black – that’s telling Photoshop “Everything from here and darker, call that black.” Do the same thing with the little white triangle on the right: “Everything from here and lighter, call that white.”

Keep an eye on Image-Mode. For the Web, you want RGB color, but for print CMYK color.

Use Save Selection - you meticulously selected every pixel, don’t throw that work away.

## Sites to Know

<http://www.alistapart.com>

articles on all aspects of web design

<http://www.webmasterworld.com>

online forums, reference materials

<http://www.w3schools.com/>

online tutorials, reference materials, online certification exams, web standards

<http://www.w3.org/>

world wide web consortium – web standards

<http://validator.w3.org/>

validate your HTML or XHTML markup to make sure it meets web standards

<http://jigsaw.w3.org/css-validator/validator-uri.html>

validate your CSS before you pull your hair out because it isn't rendering properly

[http://webmonkey.wired.com/webmonkey/reference/color\\_codes/](http://webmonkey.wired.com/webmonkey/reference/color_codes/)

<http://www.december.com/html/spec/color.html>

hex color pickers

<http://www.ericmeyeroncss.com/>

companion site to Eric Meyer On CSS, including downloadable files referenced in the book, online extras, hacks and workarounds

<http://inventory.overture.com/d/searchinventory/suggestion/>

yahoo keyword selector tool – helpful with nailing down syntax. You can have the best content in the world, but if everyone else calls it a “gas pedal” and you're calling it a “velocitator” your site traffic is going nowhere fast.

[http://webmonkey.wired.com/webmonkey/reference/special\\_characters/](http://webmonkey.wired.com/webmonkey/reference/special_characters/)

<http://www.tedmontgomery.com/tutorial/HTMLchrc.html>

HTML special characters – all you need to indicate cartoon cussing: @%&◊©™!¢

<http://www.csszengarden.com/>

showcases the capabilities of CSS

<http://www.math.duke.edu/education/ccp/resources/write/design/toc.html>

<http://www.allgraphicdesign.com/whatisgraphicdesign.html>

<http://www.webpagethatsuck.com>

good resources for graphic design, web design



## Recommended Reading

Good book series:

O'Reilly

Visual Quickstart

How to Learn Everything...

Adobe Classroom in a Book

...In a Nutshell

Other Handy Books

*Color Image Scale*, by Shigenobu Kobayashi. ISBN: 477001564X

A great marketing resource concentrating on the messages colors send.

Looking to project an image that's playful? sophisticated? This book can help.

[\(at Amazon\)](#)

*Graphic Design Cookbook: Mix & Match Recipes for Faster, Better Layouts*, by R. Wippo, Leonard Korenthat. ISBN: 0811831809

A book with graphically present page ideas to combat Blank Screen

Syndrome. Jumpstart your design process. [\(at Amazon\)](#)

*Design Essentials for Adobe Photoshop 7 and Illustrator 10 (4th Edition)*, by Luanne Seymour Cohen. ISBN: 0201713632

Need to figure out how to put your favorite politician's head on your Aunt

Tillie's body? Learn how here. This is the newest edition (green cover). [\(at](#)

[Amazon\)](#)

3rd edition: ISBN: 1568304722 [\(at Amazon\)](#)

*Design Basics Index*, by J. Krause. ISBN: 1581805012

Composition, color, design exercises. [\(at Amazon\)](#)

*Quick Solutions to Great Layouts*, by Graham Davis. ISBN: 1581802609

Focused on print layouts. [\(at Amazon\)](#)

*Son of Web Pages That Suck: Learn Good Design by Looking at Bad Design*, by V. Flanders. ISBN: 0782140203

Design pitfalls to avoid. [\(at Amazon\)](#)

*The Non-Designer's Web Book: An Easy Guide to Creating, Designing, and Posting Your Own Web Site*, by R. Williams, J. Tollett. ISBN: 020168859X [\(at](#)

[Amazon\)](#)



Resources



Gretchen H. Grant  
dogwoodtree  
designs

[www.dogwoodtree.net](http://www.dogwoodtree.net)