

WEBデザイナーという肩書きを使う場合にあったらいいかもなスキルとWEBで見れる無料教材とリソース集。

[プログラマが1ヶ月でWEBデザイナーに転身する方法](#) というエントリを見て、プログラマの方々のWEBデザインに関する関心の高さをみて書いてみます。

WEBデザイナーといっても色々な知識が必要で、かつ、最近はフリーのテンプレートも豊富にある中、どれだけ出来ればWEBデザイナーとして成り立つのだろうか？という疑問がありましたので考えてみました。プログラマの方で、WEBデザインをやりたい。また、全くの素人だけどWEBデザイナーを仕事にされたいという方の参考になれば幸いです。

最低限の知識編

1. HTMLとレイアウト

まず、知っている人には物凄く当たり前の事ですが、HTMLとCSSによってどういうレイアウトが出来るのか？という点について知っておくことが基本ですね。

驚くべきことに、PhotoshopだけをやってWEBデザイナーというのがあるようですが、そういう方は今の会社を出た際に大変だと思うので、知っておいたほうがよいでしょう。

それには、HTMLとCSSについて基本を書籍などで学んだ上で、以下のレイアウトサイト集なんかを見て、どんなことが出来るのか知りつつ、ダウンロードしていじって遊びつつ可能なことを知っておくといかも。

CSSレイアウトのサンプル集



42個のフリーCSSレイアウト-サンプル集「Free Faux Column CSS Layouts」

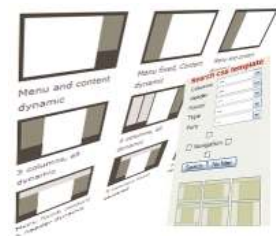
Faux Column CSS Layouts



フリーのWEBデザインテンプレート集



フリーのCSSテンプレート&テンプレート配布サイトまとめ



基本を知っている方は、これらをカスタマイズして遊ぶ過程でレイアウトを覚えていくのが一番楽しくて、リファレンスを読むより速く身につくのかもしれません。

最近ではWYSIWYGにてレイアウトを作るツールなんかもあるので、それらで生成されたコードを読んだりしても、レイアウトについて色々学べるでしょう。

WYSIWYG頼みでは対応しきれないことがあるので、HTMLだけでかけるようになっておいた方がよいのは言うまでもなくて、会社によっては、そのソフトを使ってはいけいないなんてこともあるのでテキストエディタだけでもレイアウトでき、付加的にそういうツールで効率化できるようになっていると良さそうですね。

2. Photoshop等グラフィックツールによるデザイン力

Photoshopによるデザインを学ぶ教材というのは探せばいくらでもあって、本が要らないぐらいネットにはあふれています。
これらをこなすだけで、それなりにレイアウトのやり方を学ぶことが出来るはずです。

そもそものサンプルをちょこっと変えてみるだけでもいいかもしれません。数をこなせば、それなりにデザインを作れるような気がします。
元々のセンスは必要ですが、WEBデザインは情報をうまく見せられればいいので、芸術的な側面はそれ程必要ないですね。
情報のうまい見せ方という点におけるデザイン力があれば、ちょっとカッコ悪くてもよいのかもしれない。

まず、Photoshopの使い方が分からない、という方は入門書から始めるのがよいでしょう。
サイトだけの情報だと、ちゃんと基本を覚えるのは難しいと思います。

Photoshopはそこそこ使えるという方はチュートリアルで実践していきたいですね
以下、Photoshopデザインのチュートリアル集

[プロによる美麗サイトデザインを構築するPhotoshopチュートリアル集](#)



[チュートリアルでここまでやる？というプロレベルのWEBレイアウト作成チュートリアル](#)

Create a Modern Web Design from Scratch in Photoshop
マシンのレイアウト、テキスト、色、フォント、グラフィックをすべて作成する。



内容のほとんどがデザインに集中しています。



[プロレベルのサイト作成時に使える実用Photoshopチュートリアル集](#)

Photoshop Pages Tutorial from Scratch



[How to Create a Grunge Web Design](#)



[Create an Awesome Band Website Template](#)



センスがないからダメだ、という方がいますが、その前に、上記教材なんかをみて、手を動かしてみましよう。案外出来たりするものです。

3. Photoshop→HTMLに落とし込む技術

Photoshopのスキルがすごくても、実際にHTMLに落とし込むことが出来なければWEBデザイナーとはいえないのかも。
ここら辺は、実際に手を動かしてPhotoshop→HTMLを実践していくしかないように思います。

[Photoshop→HTMLにする有料サービス](#)なんかもあったりして、ここはなかなか技術を求められるところですが、数をこなすしかないのかもしれない。
まあ、そもそも、CSS+HTMLについて詳しい方はここは苦にならないかもしれませんね。というわけで、(1)の部分をみっちりやっておくことは重要なんだろう。

生成されたHTMLが以下のタグチェッカーなんかでよい点を取れるようになるのもよさそう。

[Another HTML-lint gateway](#)

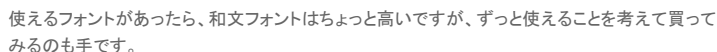


psd2css なんていうオンラインツールも存在するみたいですが、生成コードは機械的だし、やっぱり自分で出来るようにはなっていたいですね。



色々なフォントを知っておくことも重要ですが、上場している企業なんかのフォントをよく見てみると、ゴシック体のようなすっきりとしたフォントがよく使われているようで、与える印象もいいものが多いです。

ikesai.com なんかで色々なサイトのフォントについて研究してみたりもいいかもしれません



フリーでも過去に色々と紹介しました。

萌え系フォントのまとめ

[美しいフリーフォント45セット](#)

[フリーフォント大量配布/紹介サイトいろいろ](#)

ウェブ、DTPに使えるクールなフォント集

手書き風フォント2「アイコ」

フリーフォントを入手するためのトップ5サイト

最近では、Windowsユーザの方だと、標準のメイリオなんかも手軽に使えてよさそう。

5. 素材(写真・アイコン)の確保

素材を手に入れるのは意外と簡単で、[素材時点フォトバイブル20000](#)なんかを手に入れるのが手っ取り早く高品質な素材を大量に得る方法ですね。

あとは、フリーのものを使うという手もありますね。そういう場合に色々とリソースサイトとして知っておいたほうがよいのは以下。

商用利用、加工を含めてフリーで使える「ゆんフリー写真素材集」



[フリーフォト、無料写真素材サイト「足成」](#)



[5000枚以上のパブリックドメイン写真素材集「Free Stock Photos」](#)



[ロイヤリティフリーで使える素材写真「NYYH Stock Image Library」\(現在見れないかも\)](#)



素材があれば、Photoshopでフィルタをかけてそれっぽく見せるのに時間はほとんどかかりませんように思います。というわけで、素材は重要です。

アイコンリソースについても覚えておくと非常に活用できます。

[iconPot - クオリティの高いアイコンまとめサイト](#)



[アイコンを探すのに便利なサイト色々](#)



その他

[ウェブサイトに使えるアイコンセットのまとめサイト](#)

[フリーで超高品質なアイコンやボタン配布サイトまとめ](#)

[アイコンを探すのに便利なサイト色々](#)

[12500以上のハイクオリティなアイコンをまとめたサイト「IconArchive」](#)

6. SEO絡みのHTMLの知識

HTMLに落とし込む際に、SEOの知識を持ち合わせていれば顧客から喜ばれることは間違いなさそうです。

デザインの美麗さよりも、SEOによってアクセスを集められる方が、利益を上げられるわけで、WEBデザイナーにとって、ビジネス的に売りになる、コアな知識のように思います。

(1) にて、HTMLで出来ることを知ろう、と書きましたが、HTMLタグをどういふ風にかけば、SEO

的に有利なのか？という点を知っておくことはWEBデザイナーにとっては実は最も重要なことかもしれません。

個人的には、色々なサイトや書籍を読んでみて覚えていったという感じですが、こういう情報は書籍なんかできっちり学ぶ方がよさそうです。
最近では良い書籍も出回っているそうですね。知識を得た上で、サイトを作って実践する、というのも非常によいと思います。

7. ユーザビリティの知識

ユーザビリティ本なんかを1冊ぐらいは読んでユーザビリティの勉強はしておくのがよいですね。
この知識は、SEOと同列にあるようなコアの知識かと思います。いくらSEOによってアクセスを集められても、ユーザビリティが無茶苦茶で顧客が逃げてしまうようなものだったら困りますよね。

ユーザビリティの知識に関してもサイトでなく、やはり情報の良い書籍などで情報を仕入れるのがよさそうです。
サイトだと、以下のサイトなんかが参考になります。

[ユーザビリティ実践メモ](#) - ユーザビリティを主に扱うbebit 社によるブログ。セミナーなども開催している模様。



先日書いた以下のエントリもご参考までに。
[サイト公開前に役立つ25のユーザビリティチェックリスト](#)

Accessibility	Rating	Comments
1. Site load time is reasonable	Good	
2. All pages have an internal search function	Good	
3. From index page, it is easy to find	Good	
4. There is a clear and consistent layout	Good	
5. Images have appropriate file size	Good	

最低限の知識のまとめ

最低限の知識をまとめると1?7の、まずHTML/CSSの知識があつて、SEO、ユーザビリティについての知識・実践力があり、その上で、ユーザビリティやデザイン性を持ったPhotoshopのレイアウトを素材を活用してスピーディにうまく作れて、きちんとしたHTMLに落とし込める、というのが出来て、やっとWEBデザイナーと言えるのかなあと個人的には思います。

また、そういうことができて、やっと1人で仕事を請けて食べることもできるのかなと思います。仮に私が仕事を発注するような場合でも安心できるな、と思います。

これだけのことをするので、WEBデザイナーという肩書きが適切なかどうかは分からなくなってきましたね(汗)
WEBをデザインして見れる形にして顧客に納品し、ビジネスに役立てるという意味では、WEBデザイナーなんでしょうけれども、WEBクリエイターというのが適切なのかも。

付加知識編

以下は、付加的に、出来れば市場価値として上がる、ありがたがられる、といったものについて考えてみます。
あんまり色々やっても、何でも屋ということになりかねないので、全部できるようになる必要はないのかなとも思っていますが、出来るに越したことはない、自身の価値を高められる、というものについて。
まあ、色々出来ると1人でなんでも作れて楽しい、ということもいえます。

8. JavaScriptを知る

JavaScriptは、ブラウザ間で動作に差異があつたりするので、昔の名残を受けて、やっぱりめんどくさいと思う人は多いかも。とはいえ、jQueryならそこら辺はクリアできるので、まずはjQueryからはじめてみてもいいかも。
様々な煩雑なことが単純明快にできるため、JavaScriptを1から学ぶより敷居は低いでしょう。もちろん、複雑なことやるにはJavaScriptの知識も必要になってくるでしょうし、あつたほうがいい、というのは言うまでもないですね。

JavaScriptはプログラマに任せたい、という方でも、どういことが大体できるかっていうことは知っておいたほうがよさそう。

HTMLによるidやclassの適切な振り方とかも分かるはずです。

私のサイトである、[JavaScriptist](#) なんかも参考頂けると幸いです



9. Flash/ActionScriptを知る

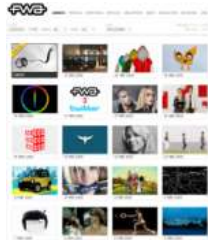
Flashで出来ることを知っておく。JavaScriptでは出来ないことを色々出来るのでそういう場をFlashでカバーできます。

アニメーションなどを入れて、訪問者に印象深いものを与えるようなアニメーションやウィジェット作成にも使えるでしょう。Flashでしか出来ないものすごいサイトも多数存在しています。

モバイルやグリッドに出てくるようなゲームもFlashLiteで作ることが出来ますね。

[The FWA: Favourite Website Awards – Web awards at the cutting edge](#)

Flashの物凄いサイトを見たい場合はここが定番



ただ、Flashについては、初期からやっている人も多いと思いますが、個人的に、誤解を恐れずにいうと、WEBデザイナーの方が最低限知っておくべき知識ではないような気がします。もちろん、Flashによる素晴らしいサイトがあるのは事実で、私も大好きなのですが、それ以外に、まず基本となる知識の方をきちんと身につけたほうがいいのではと思っています。

派手なUIやウィジェットよりも、SEOとユーザビリティを考えて売上げに貢献してくれた方が発注者としては有難いはずです。

10. PHPや、Perl、Ruby等フロントエンド言語について知る

SmartyとかPHPのテンプレートぐらいについては知っておくと、PHPベースなどでは、協業の際にありがたがられる＆作業はスピーディーに終わります。

Smartyが出来れば、テンプレート部分の開発をでき、開発者の負担を大きく減らせ、進捗のスピードも上がりそうで有難がられるはずです。

また、PHPでテンプレート開発をすれば、パーツを共有化できて、includeするだけで使いまわしができるので、HTMLを作る際にも応用でき、効率を上げることができます。

このサイトがphpspot開発日誌で私がPHPerということもあって、phpをひいき目に書いてみましたが、PerlのテンプレートやRailsを使っているところも多いと思うので、そこら辺の知識もあったらよいかもしれませんね。

言語はちょっと敷居が高いよ、という方も、Smarty等の各種言語用テンプレートエンジンの記法についてだけでも知っておくのは良さそう。

[Smartyのマニュアル](#)も日本語化されており、調べるのも容易です



私のサイトである、[phpspot](#) でも、[Smartyの使い方](#)についてまとめているので、参考頂ければ

幸いです。



PHPエディタというWindowsネイティブアプリの開発ツールもphpspotにて開発&公開していたりするので、PHPご利用の際はご参考までに。



11. 各種ライブラリ・素材の知識

何をやるにも、1から作るより、ライブラリや素材について知っておけば、1ヶ月かけることが1時間で終わることも多いです。

というわけで、ライブラリや素材を幅広く知っておくことは非常に有用で、ライブラリや素材を色々紹介したような、技術系のブログを日々見ておくってことは重要ですね。

レイアウト集にしてもそうだし、素材集など、情報へのポイントを沢山持っている＝出来ることが多い、という言い方もできるでしょう。

WEB制作系ブログを見て日々アンテナを張っておきましょう。
当ブログでもそうした情報は色々としていきます、と少々宣伝させていただきます。

私自身、情報収集のモチベーションのために、当ブログをやっているようなところもあります。

~~~~~  
こうして考えると、WEBクリエイターというのはなかなか大変です。まあ、大変だからこそ、いろいろ出来るっていう方は市場価値が高く、有難い存在になりそうです。  
考え方を変えてみると色々出来る、というのはなんでも自分で作れてとても楽しく、外注コストも下げられることなので、プログラマの方も是非、足を踏み入れてみるとよいかも。プログラムについては分かっているので敷居はそこまで高くないはずです。

因みに、私自身はWEBデザイン/制作等で食べれるぐらいの収入を得ていたこともあるのですが、今は開発をメインに社会人として働いている身です。  
自分のサイトを作る際にHTMLやデザインを作ったりしますが、それで食べているわけではないです。

そういうわけで、今、WEBデザインを本業に食べている、という人には失礼だったかもしれませんが、1つの考え方ということでご理解下さい。  
また、私自身も全ての事が完璧に出来るわけではないし、過去に作成したサイトのSJISコード&tableレイアウトを、時間の都合上、未だに直せずにいたりする、という現状があるということを、ここで一応お断しておきます。