



Sponsored By

John Delavera

TurboZine: no.1 eZine For Internet Marketers

[Click Here](#)

Copyright © 2005 By John Delavera & The Internet Company LLC
ALL RIGHTS RESERVED.

DISCLAIMER AND LEGAL NOTICES: This ebook is supplied for information purposes only and the material herein does not constitute professional business advice. This publication is designed to provide accurate and authoritative information with regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional advice. The reader is advised to consult with an appropriately qualified business professional before making any business decision. John Delavera & The Internet Company LLC do not accept any responsibility for any liabilities resulting from the business decisions made by purchasers of this book.

Master Resale and Redistribution Rights To This Ebook!

Congratulations!

You now own the master resale and redistribution rights to this ebook **My Early Days On The Net!**

By owning the master resale rights you can reprint, resell or redistribute this ebook for any price you'd like (the suggested price is \$17.00) and keep 100% of the profits! **Be sure to read the license file included to your purchase too.**

[**Click here**](#) to download the website you need to resell this ebook and also learn how to make this ebook your 24/7 "digital salesperson!"

Good Luck!

About the author

John Delavera is known for his Internet Marketing inventions and innovations. His analytical mind and viral marketing strategies have helped many exceed their Internet Marketing goals and expectations, and been applied to the creation of popular marketing products including:

- [JVManager](#): a Universal Payment Solution for accepting orders through multiple third-party payment processors with a built-in Affiliate Manager.
- [TurboReferer](#): an all-in-one solution for running and maintaining an affiliate and referral reward system.
- [Dynamic Pricing Generator](#): a unique script that creates dynamic prices for Paypal® orders.
- [TurboZiner's Anti-Filter tool](#): Net's first anti-filter tool created for allowing ezine owners to reach their audiences by coping with Spam Assassin's filter.
- [Special Offer Manager](#): a one-of-a-kind script for running special offers.
- [Buy Me A Drink](#): A cool way to allow your readers, subscribers and customers to show their appreciation to you.
- [Buy Me A Product](#): The easiest way to create an online Paypal® shop.
- [Forum Survivors](#): A viral twist on the concept of online forums that can make your site addictive to your visitors.
- [OTO: One-Time-Offer](#): A unique solution for offering true one time special deals to your readers, subscribers and customers.

He is also the editor of the following compilations:

- [Best Free Ebook](#): the best and the biggest collection of free ebooks anyone can brand for free.

- [Best Seller Ebook](#): the ultimate collection of ebooks, software, and packages offered online with Master Resale Rights.
- [The Best Of David Vallieres Collection](#): the ultimate collection of David Vallieres' ebooks, audio files, and video files.
- [TurboCD](#): the ultimate turbo-Collection of Internet Marketing Video and Audio files along with MANY Turbo-Bonuses!
- [Together Project](#): Real Stories by John's subscribers sharing 1 thing in common: Passion to Success!

John also conceived and created the Biggest Joint Venture On Internet Marketing: [Santa's Deal Time](#) through which 80+ marketers offered their products and services of more than \$11,000.00 real value to 1 bargain price.

Not only does John excel at Internet Marketing, he is also an excellent consultant, writer and teacher – as described in the dozens of testimonials presented on his site and in his winning [TurboZine](#) Newsletter.

He is also the writer of the [Viral Marketing Tutorial](#) and the editor of the [PillSeries](#) free brandible ebooks.

You can also find John's Blog at [Blog-Times.com](#) and touch base with him through [www.TurboHelpDesk.com](#)

And for now, sit back, buckle up and learn how to speed up your entry and expansion into the Internet marketing arena...!

Table Of Contents

| | |
|---|----|
| Credits | 6 |
| Part 1: My early days on the Net | 7 |
| Part 2: Frank Sinatra, Internet Marketing and me... .. | 14 |
| Part 3: META TAGS, Spiders, and Robots Explained in plain English | 26 |
| Part 4: Write, write, write HTML and learn... and progress..... | 36 |
| Part 5: SSI Magic!..... | 43 |
| Part 6: Clients & Servers, Domain Names, Hosting, Daemons, and FTP... .. | 54 |
| Part 7: When John met Amazon, Allen, Corey, and Ken..... | 74 |
| Part 8: When John met Amazon, Allen, Corey and Ken - <i>continued</i> | 80 |
| Epilogue | 85 |
| \$\$\$ Make Money \$\$\$ | 86 |

Credits

A special “THANK YOU” goes to:

- [TurboZine](#)’s subscribers: Your support gives me the strength and the inspiration needed to continue writing.
- **Joe Chengery III** for his continuous friendship and help to almost any “written” work I present online.
- **Paul Kleinmeulman** for his friendship, Australian humor☺, critical eye and tongue of truth.
- **Viola Tan** for her willingness to always stand by me.
- **Steve Boutelle** for his continuous friendship and “idea brain storming.”
- All others whose names are not included here who trust and partner with me to my online endeavors.

And of course I’d like to gratefully thank:

- **God**, for allowing me to live in this Wonderful Life.
- **My wife** for always being next to me.
- **My parents** for “being there.”
- **YOU**, my dearest reader, for trusting me and also for being an active member of this cyber society of online friends.

Respectfully Yours,

John Delavera

Part 1: My early days on the Net

I am going to tell you how I started using the Net and "computers."

I remember when I connected to the Net for the first time (back in 1994) using a browser called "Mosaic"... For the young surfers: employees of Mosaic later created "Netscape," the first easy-to-use Internet browser. It was at the time Bill Gates had declared, *"Our main software production policy does not include the Internet, so an Internet browser is not in our plans."* Quite ironic, eh? Some years later, Bill included Internet Explorer as the default in Windows™, leading Netscape and other companies to sue him for unfair antagonism.

So when I first connected to the Net, I asked myself,

"And now what?"

That was how my adventure began...

I remember surfing from site A in the US to site B in Japan, back to A, and then to site C in Africa, and so on...

After some months of surfing I became "information-overloaded..."

I had to find out a way to be more productive.

The secret was to TRACK anything you do.

Keeping a file like a *diary*...

I started TRACKING what I did, SAVING everything in **specific** folders: all the website templates I liked, all new resources, answers to common subjects/questions, etc.

In general, I made my online activity cleaner than my desk:) I used free software to organize the information stored in my PC: **TreePad** (you can find it at <http://www.treepad.com/treepadbiz>)

...also check <http://www.freebyte.com>, which offers some excellent free software.) I find the storage of information in tree-like directories very useful. The files this program creates are small in bytes. Use it and you'll remember me.

Your lesson:

Organize your online activity and keep a SCHEDULE.

Don't let yourself surf and surf without having a **PURPOSE** to what you're doing.

If you want to just play and surf, then put it on the schedule too.
Time is money in our days; always remember that.

Back to my past.

I then learned that I could create web pages.
I could create a webpage and ALL people in the world would read that webpage. That was AMAZING for me and that's how I started learning HTML.

Now, allow me to go back to 1984: That was the time I bought Allan Sugar's Amstrad 6128 (the first computer WITH a screen - green - AND a mouse AND a... hard drive!) Until then, I had to save data on tapes (cassettes). Trying to figure out where each game started on the cassette was indeed horrible... So, with a hard drive in my computer, I could SAVE data on floppy disks. And then, I could re-use data and update data and ... even create MY "programs"... That's how I started learning "Basic," the first programming

language for computers. Note that PCs' "IBM compatibles" had not yet been invented.

Do you know WHY I wanted to learn the "Basic" language? Not only because I wanted to create my "programs" (silly lines across the screen and pixels shooting and shooting to nowhere...), but MAINLY because ***I wanted to BE ABLE to KNOW HOW programs (software) work(s) the way they (it) seem(s) to work.***

You see, in those days, you had to "type" all the source code of the game in a file (the source code was some thousands of lines with strange symbols), save it into your computer, and then type "run" and the name of the program to play it! Primitive ages, yes, indeed. So, I did not want to just type the code... I wanted to KNOW what I was typing...

That's what happened when I came upon HTML too.

I learned to use the HTML language because not only did I want to learn how to write my web pages, but also because **I wanted to KNOW what was BEHIND the objects on my web pages.** I thought: "This way, I can MANIPULATE my web pages better." And that was correct. HTML is like a bicycle. Once you learn it, you never forget it.

Your lesson: **If you want to succeed in what you do, you MUST first LEARN HOW IT WORKS.**

If you want to become a racecar driver, you must know how the car engine works.

If you want to become a writer, you must read all the "classics" first and then find your own style.

Knowing your subject is **not** the END of your journey; it's where you begin to create YOUR work.

Speaking about Internet Marketing, you must know about the products you suggest to others. You must know how that software works before sending your affiliate link to your list. You must know the difference between ebook A and ebook A (that's A, yes – you see there are 1000s of ebooks but most of them are talking about THE SAME THING!) and be able to say why these two ebooks are different even if they have the same name: it happens all the time on the web; one person adds some bonuses and tutorials to an ebook, while another person adds different bonuses and tutorials to the same ebook, resulting in different products.

So...

I was trying to learn the HTML language (the **code**) and wanted to be able to write code quickly and easily.

But then I thought that someone EVENTUALLY would invent software that would allow you to write your web pages like the way you write your documents in Word™. So, even if I was becoming an HTML expert, I would lose VALUABLE time because, ONE DAY, I could use a software that would allow me to do the same things quicker, easier, and better.

I was right.

Netscape first included an HTML Editor in their browser. Then Microsoft® Frontpage™ Express came, followed by Frontpage™ being included in the Microsoft® Office® programs, and you know the rest of the story.

Your next lesson:

Try to GUESS the evolution – what it will come next. Try to guess what the future will bring. **Learn and use things that will STAND alone & LAST in the future.** And always use up-to-date tools to deal with them.

For example, when I started using a text processor, I chose "Ami-Pro," but after a few years, I realized that all those hundreds of files I had written could not be used by the famous (and most usable by then) Microsoft® Word™. I would have saved valuable time if I had used Microsoft® Word™ from the beginning...

Another example: I was thinking of creating a membership site with hundreds of software programs and ebooks. But then, while I was planning the site and trying to upload all those GigaBytes, I ...stopped. Do you know why? Because I thought MY idea was NOT unique. I should have thought of this first: "why don't you first check what exists on the Net." That's how I came [Bryan Winters' Push Button Publishing](#). With all of these existing sites, it'd be a complete waste of time and a DEAD END for me if I tried to create a similar collection. Bryan Winters would always be MILES ahead of me...

Note added after 2 years: Eventually I created that collection: the [Best Seller Ebook](#). Why? Because I noticed that while many membership sites or plain collections existed that time, NONE of them offered a SYSTEM to their members for RE-SELLING that collection... That was a **UNIQUE** idea and I embarked upon it... That's how the BSE had been created allowing people to resell the access to the collection and also resell the resale rights, the master resale rights, etc. This was a NEW MARKETING TWIST (Bryan Winters admitted it too!!) that worked for ALL others that created a similar SYSTEM: a HUGE collection offered WITH resale rights...

Your lesson: **Be Unique.**

Now, do not misunderstand this lesson.

By saying “Be Unique” I am not saying that you *must* INVENT something. Not at all. There is no need to invent the wheel – any wheel - again.

You can get the wheel add 3 more wheels and make a car☺

You can add some wings, a powerful engine, a new shape, a special fuel and ...create a plane. Creating a successful product by just by “copying” a successful idea, making it a step better, adding a NEW MARKETING TWIST always works - believe me.

Summarizing:

1. **Organize your life and your online activities.** By the way, one of the best ebooks I have read on this subject is an ebook written by Jim Edwards and his father Dallas. It's called "[The Lazy Man's Guide to Online Business](#)" - That's not the best title they could have used, but the ebook guides you on how to be productive when dealing with online activities.
2. Always use a **schedule** to spare/save your time online.
3. **Know your subject** and how things work **FROM INSIDE**.
4. Use the most **current tools** for working towards your end.
5. **Don't waste your time trying to re-invent the wheel** (use the wheel instead – make it better, more powerful!)

6. Try to **predict the future** and work for a unique product that will stand and make a difference. That product will **sell**.

[top](#)

Part 2: Frank Sinatra, Internet Marketing and me...

Cute title, eh?

I love Sinatra's songs.

In most cases, I don't pay attention to what the song says, but this doesn't happen with Frank's songs.

So what's the relationship between Sinatra's song, "New York" and "Internet Marketing" --- *and also to ANY difficult target in your life?*

Franks says in "New York":

"If I can make it there, I'm gonna make it anywhere."

Now, apply it to Internet Marketing:

***"If I can master Internet Marketing,
I'll be able to master ANY niche."***

That's 100% true, **because if you master Internet Marketing, and in particular "Viral Marketing"** (some people say that's my expertise), then **you can prevail in ANY other niche.**

That is because Internet Marketing is one of the **TOUGHEST** markets. With all these BIG names around, how could a little person make the difference??? (Okay, I did it, but it wasn't that easy...)

Back to 1995.

I realized that knowing how to write HTML language is must-have knowledge.

Yes, it was a must, but I was not a programmer. I thought that it'd be stupid to consume my time to become an expert in a subject in which OTHERS have a University Degree. So, I decided to learn HTML but **not to become an expert in it**; I'd be able to read the code of any page I want to create and nothing else.

Your lesson:

**Do NOT try to compete PROFESSIONALS.
Be a Professional To What You're Doing Instead...**

I did it. I downloaded ALL HTML tutorials that existed at that time, printed them, and read them all. HTML was a new language. I was learning it while all others were learning it too...

I was learning Version 1.0 and some of the "tags" were so new that the manual was saying, "More information soon"☺

But I did it.

Today I understand how HTML works.

Do you want me to make everything clear for you? I'll try it and I'm sure you'll know how to write HTML by the end of this article! That's a promise I am going to keep.

Remember these lines:

- ANY HTML document is a combination of **TWIN TAGS**.
- ANY tag starts/opens with <taghere> and ends/closes with its twin </taghere>. The difference is the slash / that denotes the end of the tag, the end of WHAT THIS TAG does.

- ALL tags **DO** something to the text included between them. What <taghere> does to your text finishes with </taghere>... And then the next tag comes...the next TWIN tags in particular.

I repeat the word TWIN TAGS because I want you to understand how tags work:

If a tag appears and its TWIN tag doesn't, then we have a problem in code.

Imagine each tag as a "Romeo and Juliet" case.

Romeo is the 1st tag.

Juliet is the Romeo tag with the slash /.

Or imagine each tag as an ENTRANCE/EXIT combination. ENTRANCE starts the function of the tag while EXIT finishes what the tag does.

Did you get it?

Good.

All tags **DO** something. It's like writing in Microsoft® Word™. You select the text you want and click **B**. The text becomes **BOLD**. You select the text you want and click **I**. The text becomes *ITALIC*; and so on.

So, if you learn what each tag does, then all you have to do is

- open a tag
- type your text
- close the tag

If you want to apply more than one *style* to your text, then you must open and close the TWIN tags in a row. For example, open 1st tag, open 2nd tag, type your text, close 2nd tag, close 1st tag.

Example: You want to type in **BOLD** and **RED** letters the phrase:

"I want to succeed."

Here is the "code:"

```
<open BOLD tag>  
<open RED LETTERS tag>  
I want to succeed.  
</close RED LETTERS tag>  
</close BOLD tag>
```

If you'd like the same phrase in ITALICS too, then here it is:

```
<open BOLD tag>  
<open RED LETTERS tag>  
<open ITALICS tag>  
I want to succeed.  
</close ITALICS tag>  
</close RED LETTERS tag>  
</close BOLD tag>
```

All you have to do is learn (memorize) the HTML (TWIN) tags and you have your code ready.

For our example:

```
<b></b> stand for BOLD  
<font color=red></font> stand for RED LETTERS  
<i></i> stand for italics
```

Here is the HTML code for our example:

```
<b>
<font color=red>
<i>
I want to succeed.
</i>
</font>
</b>
```

And this is what it'll be displayed on the screen:

I want to succeed.

Isn't HTML easy??? ☺

Here are some quick tips in order to de-mystify HTML:

- ALL HTML documents start with <html> tag; thus they end with its twin </html>.
- There are some tags that constitute a **family** and follow a **hierarchy**. For example, like in Life:
 - o **GrandFather** comes first, then
 - o Your **Father** and then
 - o **You**.

The same applies to these **FAMILY TAGS**. The **TABLE** is a "FAMILY TAG" and a good example.

<table></table> is the GrandFather.

<tr></tr> is the Father.

<td></td> is You. :-)

<table> tag opens the family,

<tr> prepares the ground for you to come, and everything between <td> tags is visible on the web.

Here is an example of a basic table in your web page:

```
<table>
<tr>
<td>
Your Text here...
</td>
</tr>
</table>
```

While **TABLE** "FAMILY TAG" is a "GrandFather-Father-Son" type of tag, **UL** tag that stands for a "bulleted list" and **OL** tag that stands for a "numbered list" are a "**Father-Son**" type of tag – no Grandfather exists. UL comes, and then LI appears. All LI end, then UL disappears too.

Here is a bulleted list for your webpage:

```
<ul>
<li>bullet one</li>
<li>bullet two</li>
<li>bullet three</li>
</ul>
```

And here is a numbered list:

```
<ol>
<li>bullet one</li>
<li>bullet two</li>
<li>bullet three</li>
</ol>
```

Easy?

And, of course, even the **BREAK** or the **PARAGRAPH** or the **ALIGNMENT** in your HTML document needs a tag too.

`
` stands for the **BREAK**.

`<p>` stands for the **PARAGRAPH** tag. Actually 1 `<p>=2
`.

Thus you can also have this:

`

` instead of `<p>`.

The **ALIGNMENT** in your document can be defined inside the `<p>` tag. Normally, you don't need to close the `<p>` tag with a **TWIN** `</p>`; the only exception is when you want to control the alignment of your text. Thus:

`<p align=center></p>`

`<p align=left></p>`

`<p align=right></p>`

As it also happens with Microsoft® Word™, you can define your own **TYPE** of **FONT**. I advise you **NOT** to use any font other than "**Times New Roman**" because Search Engines **LOVE** this font.

Why?

Because Times New Roman is the **DEFAULT** font for all web pages. Since Search Engines do **NOT** read **ALL** the web page, if you will **NOT** include special tags for formatting your text in a font other than the default one, then Search Engines **WILL READ MORE** of your web page and this is how your chances for **MORE TEXT** to be indexed are maximized.

If you decide to use another font, consider **Verdana** or **Arial** or **Tahoma**. These fonts exist on **ANY** computer so your page will be displayed the same way on all machines.

Here is the **FONT TWIN** tag:

Your text here.

Examples:

Your Times New Roman text here.

Verdana text here.

Your Arial text here.

The default font is **Times New Roman**, so even if you do not define a font style, the "Times New Roman" style will be used.

As you saw above, you can also include the **COLOR** of the font inside that tag:

Your red text here.

Your green text here.

The default color is **BLACK**, so even if you don't define a Color, the black color will be used.

You can also define the **SIZE** of the font inside that tag:

Your smaller text here.

Your normal text here.

Your larger text here.

The default size is 2, so even if you don't define a font size, the number 2 size will be used.

Now... There is a CERTAIN template for ANY HTML document and this template includes ALL the basic TWIN tags the HTML documents need in order to be VISIBLE on the web. Here is the philosophy first:

- Open an HTML document.
- Open the HEADER section. Everything in that section (except the title tag) will not be displayed on the web, and for that reason, ALL the tags in that section - except for the title tag - should include the word META.
 - o For the HTML document, META means, *"Do not display this data on the web, but instead, use this tag for sending information to the Search Engine Robots about the identity of this specific web page."*
- Close the HEADER section.
- Open the BODY section. Everything in this section will be visible on the web.
- Close the BODY section.
- Close the HTML document.

That is the structure of ANY web page on the Net. And here is the code:

```
<html>
<header>
<title>My title here</title>
<meta .....tag continues here>
```

<metaanother meta tag continues here - META tags have NO "Juliet"! They open but do not close. Consider all META tags as bachelors:-))>

<metaanother META tag continues here>

</header>

<body>

Everything you want here

</body>

</html>

That's your HTML document.

If you study and understand all the above, then the only thing to do is to learn the HTML tags for creating links, inserting images, etc... and you can write your first HTML document!

I am sure you have heard MANY things about these "**Bachelor**" tags, *the "Romeo with No Juliet" tags*... The **META** tags.

Nothing mysterious about these tags exists. You only need to remember just 2 META tags:

KEYWORDS

DESCRIPTION

Each META tag has a 'name' and a 'content' - like human beings:-)

So, here they are:

<meta name="KEYWORDS" content="you place your keywords here, separated by commas">

<meta name="DESCRIPTION" content="you place your description here">

Believe it or not, these 2 META tags and your TITLE TWIN tag (<title></title>) are responsible for the placement of your web page in the Search Engines.

However, DO NOT believe that **ONLY** these tags will rank your page in the Top 10 results. This is a **BIG LIE**. There is a certain way to WRITE your web page for making it a "Top 10 result" in a Search Engine. META tags and the TITLE tag help, but there must be a **SYNERGY** of ALL TAGS in your HTML document. This is a big subject and cannot be analyzed further here, but remember this: Search Engines are stupid machines. Humans program them. And programming has its rules. So, one of the basic rules that has not changed and will NOT change in the future is this:

You can get HIGH scores in search engines if you choose 1 keyword for 1 web page you want to promote and then "tell" your tags to create a **CONSPIRACY** for promoting this keyword. What I mean by CONSPIRACY is this: **ALL tags in the HTML code MUST promote this keyword.**

In case you're wondering if *I* can teach YOU Search Engine Marketing (too), then please believe me: I can!

I'll give you just 1 proof – there are many but this is not the purpose of this ebook:

Go to ANY search engine and search for "free autoresponder," or "free autoresponders" - with or without the quotes. My site, autoresponders.org, comes in the first Top 10-20 results in ALL Search Engines. Plus: that site has been listed in the Top 10 results since 2001. As you can imagine, that did not happen overnight and it's NOT because I am lucky. I do not believe in luck.

Actually, that's why I learned HTML and that's why you should learn - at least how to read - HTML code. Because you must

CONTROL your web page and its standings in the Search Engines. Search Engines constitute one of the **BIGGEST** Traffic Magnets for your web page - you know it & you need them.

And here is a **secret**:

You can create your web pages with a **WYSIWYG** (What You See Is What You Get) HTML editor and then "clean" the code (by editing it in the NotePad) and make it search-engine friendly by creating the SYNERGY I am talking about...

One could wonder... Who is John Delavera, any way... He is telling us that he knows HTML, he knows about website promotion, he knows about viral marketing, about creating ebooks, configuring autoresponders and all other scripts, promoting a website, and now mastering the Search Engines --- what next?! Come on... Who is he anyway...?

As I wrote in the beginning of this document, I am a lover of Sinatra's songs and of Internet Marketing. *"If you can master Internet Marketing, then you'll be able to master ALL aspects of the Internet."*

I just can master Internet Marketing...

[top](#)

Part 3: META TAGS, Spiders, and Robots Explained in plain English

When HTML language first appeared on the web, META TAGS did not exist.

Search Engines created the need for META TAGS creation.

Search engines use META TAGS in order to "stamp" an html document in a quick and easy way and "file" it into their records.

I'll give you an example (don't you love examples?:-))

Every Saturday and Sunday, I buy 3-5 newspapers, basically because there are some "gifts," like comics, books, and special publications with the weekend issues. However, on Saturday, I write this Newsletter; thus, I don't have much time to do anything else. On Sundays, I try to relax and take a walk, go out for lunch, etc.... However, even if I do not read the newspapers I buy, I don't throw them away; I keep them.

As you can imagine, I have a pile of newspapers in my house waiting for me to read them. When I have some time, I get as many of them as I can and start "scanning" the titles with my eyes. When an article is about the Internet, I keep it.

Again, as you can imagine, I have some 1000s of newspaper articles in my house waiting for me to categorize them. So when I have some time, I get as many of them as I can and start.

Mmm... *E-commerce*, pile on the left.
Internet Marketing... pile in the middle.
Cyber Security... pile on the right.

How am I "categorizing" the articles?

By reading the headlines.

Now, let's apply this example to our case.

There are MILLIONS of web pages online, like the 1000s of articles in my house. Search Engines need some headlines to be able to categorize these web pages, like what I do when I categorize the articles by their headlines.

META TAGS is the creation that allows Search Engines to categorize web pages.

META TAGS categorize an HTML document and help Search Engines to file it into their online libraries.

As I told, META TAGS are "ORPHAN" or ...*Bachelor* tags. They don't need any TWIN tag to exist (no Juliet☺) since they work alone.

A human being has a NAME and a CONTENT.

I am *John* (NAME) and I am a *GOOD PERSON* (CONTENT)

Each META TAG consists of a NAME and CONTENT too. The NAME of the META TAG tells the Search Engine what's included in the CONTENT.

I bet you're now wondering...

"Hey John, HOW come META TAGS can SPEAK TO Search Engines?"

Good question; time to introduce you to the **SPIDERS** and the **ROBOTS**.

You probably know that WWW stands for the World Wide Web. The word "Web" has been used because ALL computers that are connected TO the Net are somehow connected to each other.

But where a "web" exists, "spiders" also exist.

There are many types of "spiders" on the Net. Spiders are computer programs that "crawl" the web and do what their programmer told them to do.

For example, there is a "spider" for collecting e-mail addresses from the web pages. Another "spider" checks the links of a web site.

Search Engines use "spiders" to collect information about the web pages.

Thus, at specific periods of time, Search Engines send their "spiders" out and the spiders start sending information back to the Search Engines.

Now, **ROBOTS** are *the servants of SPIDERS*.

Spiders are the creatures that BUILD the directory of the Search Engines, while Robots maintain this directory and basically report to the Search Engines whether what the Spiders found and reported to them before still applies and exists as it did before or not.

Imagine...

- Spider as a *Spacecraft*,
- Search Engines as *NASA*, and

- Robots as *Satellites*.

NASA sends the Spacecraft "Spider" to Mars.

Spider lands and transmits information, collects samples of land, takes photos, etc., and then returns to the base and feeds NASA with its findings.

NASA creates a record for Mars and puts all the information there.

After some time, NASA commands Satellite "Robot" to pass by Mars and see if the specific findings "Spider" reported are still the same.

Satellite "Robot" goes to Mars and checks the information NASA has. *"Nothing changed..."* "Robot" reports. Or... *"I see John Delavera fishing..."*

Then NASA adds in Mars's record: *"July 5, 2003. 'Robot' reported that John Delavera appeared on Mars fishing. Since 'Spider' reported that no sea exists on Mars, schedule 'Spider' to visit Mars again during the next space journey. End of the report."*

Now, if "Robot" reports, *"Nothing changed,"* then NASA will **not** send "Spider" to Mars again until there is a special reason; it's a matter of cost too, as you can understand.

That's how Search Engines, Spiders, Robots and Meta Tags work. Of course, Search Engines do not send any Spacecrafts or any other creatures to your site! :-)) Spiders are the robotic visitors to your site and do no harm.

However, SOME of the Search Engines have instructed their Spiders to NOT ONLY examine the META TAGS, but also SOME other elements of the web pages.

For example, Google® says to her Spider:

“Go and see what the META TAGS say, but do not leave the site until you check to see if all the information given by the META TAGS agrees with the information described INSIDE the content of the web page.”

Yes, Google has some good Spiders.

Moreover, Google says to her Spider:

*“If the META TAGS contain information that is irrelevant to what the web page is dealing with, mark that web page with a **Red X**.”*

Ooops!

Bad news for your site... No listing at Google.com

Now, you're getting closer to my points:

META TAGS are used to help search engines categorize the web pages. **Try to TRICK the search engines by inserting irrelevant information into your META TAGS, and you'll never see your web page listed in the SEs.**

Also Google says,

“Compare the information you find inside the META TAGS to what is described in the TITLE, and also check if this information is repeated by the first paragraph of the text inside the <body></body> tags, but IF the same information is repeated TOO many times, then mark that web page with a Red X. ”

Ouch!

Another command of Google:

“Get the information you find inside the META TAGS and check to see if the content of the 'keywords' and the 'description' tags is repeated throughout the page, how many times this happens and also check this and that...”

Thus, please keep the following in mind:

- Fake META TAGS can ban your page and you will never see your website listed in the SEs.
- Good META TAGS that reflect exactly what's described in your web page can make the SEs rank your web page in their Top Listings.

With all the above said, here are the META TAGS you can use:

NAME="Reply-to"

Example:

```
<META NAME="Reply-to"  
CONTENT="delavera@turboreferer.com">
```

Some search engines will spider your site and if they find a "Reply-To" in the meta-tags, the engine will automatically email you letting you know that your site was spidered. Personally, I have never used this tag and have never heard of anyone receiving such an email. Plus, giving your email inside your html code will enable other email-spiders to capture your email address and increase the spam you receive.

My suggestion: **DO NOT use this tag.**

NAME="Author"

Example:

```
<META NAME="Author" CONTENT="John Delavera">
```

This field is what tells the engines who wrote/designed the website or what your name is. This will help build your reputation and brand your business name.

My suggestion: **Use this tag.**

NAME="Subject"

Example:

```
<META NAME="Subject" CONTENT="Internet Marketing">
```

This is the subject of your website. It's recommended that you choose one of your sites' main keywords and place it in this field. This will help the engines to place your listing in the right category.

My suggestion: **Use this tag.**

NAME="Description"

Example:

```
<META NAME="Description" CONTENT="Description goes here">
```

This is a simple description of your website and what it offers. It's recommended that you enter no more than 25 words (or 255 characters) in your site description. This is also what is shown to people in the search engines under or next to your site link.

My suggestion: **ALWAYS use this tag.**

NAME="Keywords"

Example:

```
<META NAME="Keywords" CONTENT="Keywords">
```

In this field you enter your site's keywords. DO NOT use the same keywords more than once and always try to make sure that each keyword appears inside your webpage at least twice. You can also use keyword phrases as well (examples: internet marketing, website promoting.)

My suggestion: **ALWAYS use this tag.**

NAME="Language"

Example:

```
<META NAME="Language" CONTENT="English">
```

In this field, you can enter what language your site is in. Some engines give visitors a language choice and this will help the engines know how to index your site based on its language.

My suggestion: **If you use a language other than English, use this tag.**

NAME="Revisit-after"

Example:

```
<META NAME="Revisit-after" CONTENT="15 days">
```

This tag tells the engines WHEN to send their ROBOTS to check if your content has been changed or not.

My suggestion: **ALWAYS use this tag.**

NAME="Rating"

Example:

```
<META NAME="Rating" CONTENT="GENERAL">
```

This tag tells the engines whether your site is family-friendly or adult-based. If your site is family-friendly, enter = “GENERAL”. If your site is adult-based, enter = “ADULT”.

My suggestion: **ALWAYS** use this tag.

NAME="Robots"

Example:

```
<META NAME="Robots" CONTENT="All">
```

Here are the Robots again! :-)

This tag tells the engines if the robots should index your page or not. You can insert "Index" to have your page listed. You can insert "No Index" to have your page NOT listed. You can insert "Follow" to also have all links inside your webpage listed. If you choose "No Follow," then the robots will not follow the links inside your web page, and thus those links will not be listed. And you can choose "All" to have both your page and the links inside your page listed too.

My suggestion: **ALWAYS** use this tag.

NAME="Classification"

Example:

```
<META NAME="Classification" CONTENT="Internet  
Services">
```

This field tells the engines what type of website you have. This is important because some engines will index your site this way.

My suggestion: **Use this tag.**

I remind you that ALL META TAGS are inserted between the <head></head> TWIN TAGS, which come after the <html> tag, and before the <body> tag.

And here are the META TAGS discussed in this "lesson."

```
<META NAME="Reply-to"
CONTENT="delavera@turboreferer.com">
<META NAME="Author" CONTENT="John Delavera">
<META NAME="Subject" CONTENT="Internet Marketing">
<META NAME="Description" CONTENT="Description - less
than 255 characters">
<META NAME="Keywords" CONTENT="Keywords - never
repeat the same keyword more than 2 times">
<META NAME="Language" CONTENT="English">
<META NAME="Revisit-after" CONTENT="15 days">
<META NAME="Robots" CONTENT="All">
<META NAME="Rating" CONTENT="GENERAL">
<META NAME="Classification" CONTENT="Internet
Services">
```

[top](#)

Part 4: Write, write, write HTML and learn... and progress

Back in 1995...

Having basic knowledge about HTML, I started writing my first web pages. I still had many things to learn... I had to memorize all HTML tags and what each tag does.

I thought that if I used an HTML editor, then I could speed up my work. I could then select a block of text, hit B, and the TWIN TAGS `text here` would automatically be inserted by the HTML editor program (for making the text **BOLD**).

What a discovery!

That's how I started using HotDog HTML editor. That was the most popular HTML editor at that time and future versions also included a browser so I could see how the code would be displayed online. You can still find this editor in any Tucows collection (<http://www.tucows.com>.)

HotDog was a good solution, but I felt that what I needed was a WYSIWYG (What You See Is What You Get) editor. Even if you know ALL html tags and you can write the code easily, it's by far much easier to create your web pages using a WYSIWYG editor. It's like writing a document in Word™.

The first browser I used was **Mosaic**.

A few people that worked on the development of Mosaic left that company and created their own browser: the revolutionary "**Netscape®**." Netscape was a great browser and its future versions offered what I had been waiting so long for: the 1st WYSIWYG

HTML editor built-in Netscape browser. I started creating my web pages like crazy. With Netscape's WYSIWYG editor, html-writing was **so easy**! But there have been times when I needed to edit the code manually.

There have been times when I needed to edit the code simultaneously for some dozens of HTML documents. For example: I created the website A. Website A finally consisted of 12 web pages. When I thought I finished the site, I found out that I needed to change the text in one paragraph. Thus, I had to open all 12 pages and manually find where the mistake (typo, error) was and then edit it in all 12 files. So, what I needed was a program that would allow me to OPEN ALL 12 files at once and CHANGE the content in ALL 12 files by using just 1 command (the 'Replace' command.)

That's how I found the ARACHNOPHILIA software. I still use this program. I need this program and you need it too. You can find it at <http://www.arachnoid.com/arachnophilia/index.html>. There are many other similar programs like TextPad, NotePad Pro, etc., but I found Arachnophilia to be the friendliest program of all. Plus it's 100% free - no restrictions at all.

For example, you typed 'hlp' instead of 'help' in all of your web pages, or you just want to exclude a paragraph from ALL of your pages. Just open all pages with Arachnophilia and edit them simultaneously. It's so easy!

So what I did was this: I was creating my web pages with Netscape WYSIWYG editor and then editing the code or changing the code in ALL my web pages with Arachnophilia. That's what I am doing today too!

Future versions of Netscape did not include the editor.
Do you know why?

Because the competition had just started.

In those early years of the WWW, Bill Gates had declared: *"Internet is not a target market for our products,"* but after a while, Microsoft® introduced Internet Explorer. IE *was* a bad browser. Nobody could ever imagine at that time that IE would become the standard and the most popular browser and that Netscape would lose its glory. But it happened. Bill had been releasing his new versions of Windows™ one after another: 3.1, 3.11, Windows 95, and then "Plug'n'Play" Windows 98! Having collected million of users all over the world, it was easy for Bill to create a product that could kill Netscape by including IE in his Windows. That's what he did. Netscape was finished as the #1 browser when Bill Gates decided to include IE in Windows. That's how FrontPage Express™ had been created, too: the best and the easiest WYSIWYG editor I had ever seen...

FrontPage Express™ made so much sense that Microsoft® decided to create a better version of it and included it as FrontPage™ in the Microsoft Office™ Professional Version edition. Today, I use FrontPage™ and Arachnophilia for creating my web sites. I create all pages with FrontPage™ and then edit the code with Arachnophilia. I also use Arachnophilia for massive changes to .html files and .txt files. I know there are MANY other great HTML editors like Adobe®'s products, NetObjects Fusion™, Macromedia's products, etc., but I find Frontpage to be the easiest one.

After creating some 1000s of web pages, I got tired of... making changes! Just imagine it: You create a web site with 5 main pages, so you include a navigation menu on all pages with links to these 5 main pages plus a link back to your homepage.

By the way: What is a "homepage"?

A homepage is the page that opens automatically when the user types the URL of your domain name. For example, if your domain is `www.mydomain.com`, then when a user types `http://www.mydomain.com`, a web page is displayed. THAT'S your homepage, and in most cases, it's called `index.html` or `index.htm`.

So suppose, you created a web site with these pages:

- Your homepage (`index.html`)
- A page with information about yourself (`about.html`)
- A page with your products (`products.html`)
- A page with a FAQ - Frequently Asked Questions (`faq.html`) and
- A page where the visitors order your product (`order.html`)

I am going to tell you how to apply a **STYLE** to your website in another lesson (it's a big chapter indeed), but in all cases, you must include a navigation menu on ALL pages. Thus, all 5 pages (`index.html`, `about.html`, `products.html`, `faq.html` and `order.html`) must have a NAVIGATION MENU (both at the top and at the bottom of each page) with LINKS to ALL pages. This menu:

```
<a href="http://www.mydomain.com/index.html">Homepage</a>
|<a href="http://www.mydomain.com/about.html"></a> |
<a href="http://www.mydomain.com/products.html"></a> |
<a href="http://www.mydomain.com/faq.html"></a> |
<a href="http://www.mydomain.com/order.html"></a>
```

Now, what if after a few days, you want to include a NEW page, say a page for your NEW products and name it `news.html`. You must manually edit all 5 pages and include this new page in your navigation menu. That was a problem... That's how I got tired... I had to find a solution, and I found it. This is YOUR solution too.

SSI tags.

SSI stands for "Server Side Includes" tags.

I learned about SSI tags in 1997 and I have been creating ALL my web pages using these tags since then.

I am going to teach you how to use SSI tags (exactly: not more than 5 -that's FIVE only- SSI tags) and you'll thank me for the rest of your life. Yes, SSI tags are THAT powerful.

With SSI tags, you create a page with your navigation menu (say menu.html) and include a link to that page on all other... 129 web pages on your web site.

Whenever you want to add a new page, you change just 1 page (menu.html) and the navigation menu is updated on ALL 129 pages of your website! Powerful tags!

You can do the same with your header/footer sections of your web pages. Usually, these sections include the same code on all pages. For example, suppose your page starts with:

“Welcome to my page... ”

and ends with

“Come back soon.”

Between these two phrases, you can have anything you want, tables, images, audio files ... anything. What you can do by using SSI is create a header file (header.html) and include the HTML code from the first HTML TAG <html> up to the word "Welcome..." and a footer file (footer.html) that includes all the

HTML code after the word "...soon" up to the last HTML tag
</html> - *Read again above how HTML TAGS work.*

Can you imagine what you can do with SSI tags?

If you create your site using a header and a footer, then you can
CHANGE the LAYOUT of ALL of your web pages by just
changing these 2 files!

Here is a basic HTML structure for a plain web page, your
homepage: **index.html**

```
<html>
<header>
<title>Your title here</title>
<meta tag here - read my article on Meta Tags in Issue 43>
</header>
<body>
All content here...
</body>
</html>
```

Here is your header file: **header.html**

```
<html>
<header>
<title>Your title here</title>
<meta tag here - read my article on Meta Tags in Issue 43>
</header>
<body>
```

And here is your footer file: **footer.html**

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

And here is the code for your new index.html file:

```
<SSI tag here that calls header.html>  
All content here...  
<SSI tag here that calls footer.html>
```

For your new index.html to work you must **RENAME** it from index.html to **index.shtml**.

Did you notice the “s” in “.shtml” extension? Yep! That means that web page (index.shtml) uses SSI tags. Isn't SSI powerful?

However, in order to use SSI tags, you must be able to understand where your files reside on your server and understand the structure of your server.

[top](#)

Part 5: SSI Magic!

More about SSI tags...

Again: SSI stands for "Server Side Include" type of commands.

If you host your site on a server that allows SSI tags to be executed, then YOU are very lucky indeed. Most of the (good) hosts allow SSI tags, but here is a test so you can see if you're lucky or not:)

Open your notepad and copy and paste this phrase:

SSI works!

Save this file as **ssi.txt**

Upload ssi.txt to your server. Upload it to your root folder, so if your domain is <http://mydomain.com>, you can access this file here: <http://mydomain.com/ssi.txt>.

Again, open your notepad and copy & paste this code:

```
<html>
<body>

<!--#include virtual="/ssi.txt"-->

</body>
</html>
```

Now, save this file as ssi.shtml.

The extension **.shtml** tells the server that the file ssi.shtml *includes* an SSI tag. In most cases, when you use SSI tags, you'd name the pages that include these tags as something that ends in .shtml.

Upload ssi.shtml to your server. Again, upload it to your root folder where you uploaded ssi.txt, so if your domain is <http://mydomain.com>, you can access this file here:
<http://mydomain.com/ssi.shtml>.

Now, open your browser and type the URL address for the ssi.shtml file. Again, if your domain name is <http://mydomain.com>, type into your browser <http://mydomain.com/ssi.shtml> - just remember to change mydomain.com to your domain name:)

Okay?

What do you see?

If you see

SSI works!

...then you now know how to use SSI tags!

If you see nothing or an error message, double-check that you followed my instructions and try again. If nothing works, then you should ask your host to allow you to run SSI tags in your server.

Just copy and paste this email message and send it to your host:

[---MESSAGE STARTS HERE---]

Hello!

I'd like to use SSI tags for my account, YOUR DOMAIN NAME.

Could you please tell me if this is possible, and if yes, please configure my server so I can run files ending in .shtml.

Thanks

YOUR NAME

YOUR DOMAIN NAME

[---MESSAGE ENDS HERE---]

Now, suppose everything is ready and you can use SSI tags.

Pay attention to the following:

Your server is like your computer (the 'server' is a computer too, right?)

(Actually YOUR computer is a LOCAL server.

It even has an IP (Internet Protocol) address, this one:

127.0.0.1)

I am sure that you're familiar with the tree structure your computer uses. There are...

folders on your computer,

.....**files** inside those folders,

.....**subfolders** inside the folders,

.....**files** inside the subfolders, and so on.

You can create directories and subdirectories on your server the same way you create folders and subfolders on your computer.

Every file on your computer has a **LOCAL ADDRESS** in the same way every file on your server has an **INTERNET ADDRESS**, which is called the **URL** (Uniform Resource Locator.)

The 'root' address of your computer is this one: C:\

C: stands for the name of your hard drive. Even if you have many drives on your computer, there will always be C: as the name of your hard drive.

For your information, **D:** is your 'root' address for your CD ROM drive, **E:** for your DVD drive, etc... You can see the names of your drives when you double-click the 'My Computer' icon on your Desktop.

The equivalent 'root' address for your server is YOUR DOMAIN NAME. For example: <http://mydomain.com>

Thus, if you save a file named 'test.txt' on your computer and save it in **C:**, then this file will have this **LOCAL** address: **C:\test.txt**.

If you create a folder named 'My Folder' on your computer and save it in C:\, then this folder will have this LOCAL address: C:\My Folder.

If you save the file 'test.txt' on your computer and save it in C:\My Folder, then this file will have this LOCAL address: C:\My Folder\test.txt.

And also, if you create a subfolder named 'pages' inside the 'My Folder' subfolder and save the file 'test.txt' in that new sub-sub-folder, then that file will have this LOCAL address: C:\My Folder\pages\test.txt.

Again, EVERYTHING inside your computer has an ADDRESS in the same way EVERYTHING on the Net has an INTERNET ADDRESS.

So, if your domain name is <http://mydomain.com> and you upload the test.txt file into your 'root', then the address of this file on the Net will be <http://mydomain.com/test.txt>.

Did you notice the difference in slashes?

The LOCAL addresses on your computer use this slash \ for separating the files in the address, while the Internet addresses (URLs) use this slash /.

Accordingly, if you create a sub-directory on your server named 'My Folder,' the Internet address of this folder would be: http://mydomain.com/My_Folder

And if you upload the test.txt file to it, then it'll get this URL: http://mydomain.com/My_Folder/test.txt

If you remember what you read above, your homepage is usually called index.html or index.htm or even index.shtml (if it includes SSI tags), and so, by saying "homepage," we mean the file that opens automatically when you type your URL, for example: <http://mydomain.com>

So when I type <http://mydomain.com>, the file "index.shtml" opens automatically. This is how the web works.

BUT if I miss putting an index.html or an index.htm or an index.shtml file inside a folder, then that folder will REVEAL ALL its content.

...I have a reason for explaining all this... just reach the end and you'll understand the whole picture...

For example, if I upload the files test1.txt, test2.txt, test3.txt to my 'root' server (<http://mydomain.com>) and then type the address <http://mydomain.com> into my browser, I will get a blank screen with the contents of <http://mydomain.com>, like this:

...

test1.txt

test2.txt

test3.txt

BUT if I also upload a BLANK file named index.html, or even better, an index.html file that contains only the word "Hello!", then when I type <http://mydomain.com> into my browser, I won't see the above listing of files test1.txt, test2.txt, and test3.txt. I'll see a white page with only the word, "Hello!" That's because, when I type <http://mydomain.com>, my browser will automatically search for an index.* file and open it (index.* is an abbreviation to stand for any of the three forms: index.html, index.htm, or index.shtml). Only if the browser can't find a file named index.*, will it then list the contents of the "folder."

I hope you understood all of the above.

That's why you ALWAYS need to upload an index.* file (even if it's a blank file) inside all of the directories on your server **so that the browser does not *list* the contents of that folder.**

With all the above explained, you can now understand how SSI will work. The syntax of an SSI command is this:

```
<!--#include virtual="/ssi.txt"-->
```

What this command says to your browser:

The browser reads

<!--# and understands:

"This is the start of an SSI command."

The browser reads

"include virtual =" and understands:

"I must open the file, which resides on the server, of the address that follows the = symbol, get its content, and place it here."

The browser reads

"/ssi.txt" and understands:

"Okay, the file is named ssi.txt and resides inside the 'root' folder." Now, I will open the ssi.txt file and put everything I find inside the file here."

The browser reads "-->" and understands:

"Everything is fine. The SSI command is well typed and now I finished what I had been instructed to do. Over!"

If I want to include the content of the file test.txt (that has the address 'http://mydomain.com/pages/test.txt') inside the index.html file (that has the address 'http://mydomain.com/index.html'), then can you imagine how the SSI tag would be?

Here it is:

```
<!--#include virtual="/pages/test.txt"-->
```

What would the browser understand by reading the above command?

"Open the file with the name 'test.txt' that is inside the folder 'pages' and place everything that is inside that file here."

That's how you can include the contents of ANY file in your page. Just give the CORRECT 'path' of the file and the browser will understand it.

There is also another useful (and easier) SSI tag. This one:

```
<!--#include file="ssi.txt"-->
```

See the difference? Instead of the word 'virtual', here we have the word 'file'.

You can use this command when the file you want opened by the browser resides in the SAME folder where your .shtml page is.

For example, if your ssi.shtml is at

<http://mydomain.com/pages/ssi.shtml>

then by using the command <!--#include file="ssi.txt"-->, the browser will take as given that the ssi.txt file is located inside the directory 'pages' - if not, you'll get an error message.

So, if you want to use a file that is located in another folder, use the 'virtual' SSI tag. If you want to use a file that is inside the same folder where your .shtml page is, then use the 'file' command.

There are many other SSI commands you can use to automate many tasks on your web page.

Feel free to copy and paste these commands inside a file that ends in .shtml

Current Date:

```
<!--#config timefmt="%A %B %d %Y" -->  
<!--#echo var="DATE_LOCAL" -->
```

This will display the date, like this:

Current Date and Time:

Saturday November 1 2004

Here are some variations of this powerful SSI command:

If you want your date to look like this: Sat 19 Jul 03

```
<!--#config timefmt="%a %d %b %y" -->  
<!--#echo var="DATE_LOCAL" -->
```

If you want your date to look like this: 07/19/03

```
<!--#config timefmt="%m/%d/%y" -->  
<!--#echo var="DATE_LOCAL" -->
```

IP Address:

```
<!--#echo var="REMOTE_ADDR" -->
```

This will display the IP address of your visitor, like this:

Your IP Address:

213.16.196.104

Browser Type and OS:

```
<!--#echo var="HTTP_USER_AGENT" -->
```

This will display the kind of browser your visitor uses, like this:

Browser Type and OS:

Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)

Web Page File Name:

```
<!--#echo var="DOCUMENT_NAME" -->
```

This will display the name of the file opened by the browser, like this:

Web Page File Name:
ssi.shtml

Web Page File Size:

```
<!--#fsize file="ssi.shtml" -->
```

This will display the size of the file opened by the browser, like this:

Web Page File Size:
16KB

This Web Page's Last Modified Date:

```
<!--#flastmod file="ssi.shtml" -->
```

This will display the last date the ssi.shtml file has been modified, like this:

This Web Page's Last Modified Date:
Thursday June 19 2003

As you can understand, SSI commands and tags are powerful. You can create your website by using only 1 file for your header.txt and another file for your footer.txt and call these 2 files from any page

you want. Then you'll be able to change the layout of ALL of your web pages by just editing these 2 files! ☺

[top](#)

Part 6: Clients & Servers, Domain Names, Hosting, Daemons, and FTP

You have read how HTML, Meta-Tags and SSI work.

If you study all the above, you'll be able to write your html pages and understand what these html tags are doing. You'll also be able to create search-engine-friendly web pages; and by exploiting the SSI commands, you'll be able to create a web site you can update easily (by changing 1-2 files instead of 100s of webpages.)

Now, I'm going to tell you how to use FTP in order to upload your pages onto a server.

Let's go again some years back in the past.

Tim Berners Lee invented the World Wide Web in 1989, an internet-based hypermedia initiative for global information sharing while working at CERN, the European Particle Physics Laboratory. He wrote the first web client (browser-editor) and server in 1990. You can read more about Tim at

<http://www.w3.org/People/Berners-Lee/>

Tim created the HTTP, which stands for the Hypertext Transfer Protocol. He drafted the rules under which the web can work; however, his basic idea was a) the creation of a language (HTML) that would allow anybody to write a 'page' for the web, and b) the creation of a 'browser' that would allow anybody to 'read' this page. The revolutionary discovery Tim offered was the creation of (Hyper)Links. This would be the way of "jumping" from one page to another no matter where these pages resided.

The full description and analysis of all the above constitutes a '**protocol**.' Thus, a protocol is a series of terms and rules under which a certain mechanism works.

Tim's mechanism was the web and his protocol was the Hypertext Transfer Protocol, or HTTP.

FTP stands for the "**File Transfer Protocol**."

In Tim's HTTP, browsers "read" the web pages, while in FTP, special software, the FTP-clients, allow the use of this Internet Service. Keep in mind that it would be easier for me to create a multimedia presentation than to show you how FTP works, but I'll try to describe this service in plain English and in text. You'll be able to start FTPing. That's a promise!

So, FTP is a very common method of moving files between two computers. FTP was invented and in wide use long before the advent of the World Wide Web and was originally used via a text-only interface.

In order to understand how FTP works, you must be able to understand the difference between "**servers**" and "**clients**."

Here is the difference:

- They both are computers.
- They both are connected to the Internet.
- Servers are connected on a 24/7/365 basis, that is 24 hours per day, 7 days per week, 365 days per year.
- Clients are connected through dial-up.

Imagine a "Server" as a **HOTEL** and a "Client" as a **customer/client** of that Hotel. The customer comes to the Hotel, pays, sleeps, and goes.

Your PC is a "**client**" and the Internet is the "Hotel."

Actually, the Internet, as the "Hotel" for our example, consists of MANY computers connected to each other. Imagine Hotels A, B, and C. All of them are connected through halls. You enter Hotel A and then walk into a hall and enter Hotel B, then Hotel C.

That's the Internet:)

Your **Internet Service Provider** (ISP) is one of the "Hotels." You dial up your ISP, enter your username and password, and connect to THIS SPECIFIC COMPUTER - "Hotel" for our example.

I repeat it: when you dial up your ISP, you are connected **NOT** to the Internet, but actually to your ISP's computer, which in turn is connected to the Internet. For the sake of our example, if you went to your "Hotel" and found locked doors and you didn't have the right key, then you could not enter into your room, right? Your "keys" for the 'Hotel' are your username and password. So, when you use your username and password, you enter into your ISP's computer.

The Internet is all of these 'Hotels' connected to each other; all these **SERVERS** connected to each other with wires (in most cases, these wires are located under the sea.)

BECAUSE your ISP's computer is a **SERVER** - that means it's **ALWAYS connected to the Internet** - you can connect to it whenever you want and access the rest of the ("Hotels"-) Servers on the Internet too. If your ISP's computer breaks down, you'll not be able to connect to the rest of the Servers online because you'll not be able to reach your "Hotel."

I hope you understood the example above.

So, your computer is a **CLIENT** on the Net.

Using your connection, you first visit your ISP's computer (Server), and then **THROUGH IT**, you visit other computers too.

Now pay attention:

YOUR use of the other servers depends on the **PROTOCOL** you use. Each protocol needs a special software so you can access the services it provides.

You can use **HTTP** by using the HTTP clients (browsers.)

You can use **FTP** by using the FTP clients (more below.)

You can use **EMAIL** by using the EMAIL clients (Outlook, Eudora, etc.)

The term '**client**' again denotes that *"you use it while you're online, do your job, and leave it."*

Actually, the relationship between 'server and client' governs **ALL** aspects of the Net. **Almost EVERYTHING online works on that basis.** YOU are a client for the Net. Your computer is a client too. Your Internet Explorer, your Outlook or Eudora, are clients too.

No matter what we're talking about - **HUMANS** or **COMPUTERS** - this specific distinction **ALWAYS** exists and it always means this very simply:

CLIENT comes and leaves.

SERVER is always there.

Now, let's go further...

Imagine a 3D **BOOK**.

You open the book and try to read it.

You read NOTHING because your eyes cannot see the 3D models inside the book. What do you do?

You get some specific GLASSES.
You wear the glasses and open the book.
Voila!

A new 3D world comes in front of your eyes!
Magic? No.
Just technology.

Now, imagine a file named **something.doc**

You open your Windows Explorer, double click the "something.doc" file, and your computer automatically opens Microsoft Word™. That's how you read what's inside the file.

Try to open an *image* with Microsoft Word™.
Try to open any file that ends in *.gif* or *.jpg*
You'll see 100s of strange symbols...
Nothing readable...

Use Adobe PhotoShop™ and open a file named *delavera.jpg*

Voila!
Here I am, my .jpg picture!

Try to open a file that ends in *.pdf* with Microsoft Word™.

Again, you'll see nothing readable.
Use Adobe Acrobat™ and you'll be able to read what's inside the *whatever.pdf* file.

Conclusion:

Every file on your computer needs a specific **READER.**

The same happens on the Internet.
Every service provides you with files - data-bytes.
You need specific readers to open these files.

The same happens to Internet Services.
In order to USE a service, you need THIS SERVICE'S READER.
In Internet terms, you need THIS SERVICE'S CLIENT!

Internet Explorer™ is an HTTP client.
That's how you can view what's on the web.

Outlook™ is an EMAIL client.
That's how you can read your email.

If you've understood all the above, I am sure you can now understand how FTP works.

As I already told you, a SERVER is a computer connected to the Net on a permanent basis.

What do you have inside your computer?

Documents, Outlook, an image editor, an MP3-reader, a video-reader, etc.

Servers also carry MANY software packages inside them.
Servers serve you. That's why they are called "servers."
Servers serve you with their software packages and their hardware that allows the software to run.

I do not know WHY, but the basic software packages a server uses to serve you are called ... DAEMONS. Strange eh?

Here is how Webopedia.com defines a 'daemon':

(<http://webopedia.internet.com/TERM/d/daemon.html>)

daemon

Pronounced DEE-mun or DAY-mun. A process that runs in the background and performs a specified operation at predefined times or in response to certain events. The term daemon is a UNIX term, though many other operating systems provide support for daemons, though they're sometimes called other names. Windows, for example, refers to daemons as System Agents and services. Typical daemon processes include print spoolers, e-mail handlers, and other programs that perform administrative tasks for the operating system. The term comes from Greek mythology, where daemons were guardian spirits.

To tell you the truth, I'd prefer them to be called "Gods" or "Spirits" or even "Spells"... Anyway...

Here is my definition: **A DAEMON is a software package that can serve MANY people at the same time through a server.**

I told you that YOUR computer is a CLIENT. It's a computer served by another computer: the server. But, did you know that your computer could become a server too?

Suppose that you connect to the Net and leave that connection open, or even better, have DSL and have a permanent online connection. You can transform your computer from a CLIENT into a SERVER.

Do you know how?

By installing *DAEMONS* inside your computer. That is, by installing software packages that will allow you to *SERVE* people who visit your computer.

What is Windows 95, Windows 2000, Windows XP?

Do you remember DOS?

A computer needs an Operating System (OS) to work. An OS is the platform on which you can build your other applications. Each system - each OS - offers its own applications. For example, you cannot run programs designed for the OS of Windows XP on a computer where an Apple OS has been installed. Also, you cannot run perl and cgi scripts on your computer because they need a UNIX OS.

The most recognizable OS for home computers is Windows (95, 98, 2000, XP.)

Servers have their own *OS*.

Microsoft has created Windows NT. This is the OS you need if you want to make your computer a *SERVER*. That is: if you install Windows NT on your computer, then you can install other applications (daemons), and whenever you are connected to the Net, your computer can act like a server.

NOTE: Personally, I do not prefer Windows Servers. I love UNIX, which is a completely different OS. It's a big chapter to analyze here; just remember that UNIX (or Linux) is also an *OS* that will allow you to transform your computer from a home computer (Client) into a *SERVER*.

There is a specific daemon (software) for each service you want to offer through a server.

If you install an email daemon on your server, it will allow you to offer email services. Thus, people can get an email address of a domain you offer, and create and access their email through your email server/daemon.

If you install an **IRC** (Internet Relay Chat) daemon on your Server, it will allow you to offer chat services, so people will come to your system and chat.

It's not that easy to run a server and I'd say **FORGET** it, but at least have this knowledge in order to understand how all these Internet services work.

Thus, if you install an **FTP** daemon, you can allow people to connect to your computer and download files from a folder you have defined as a "shared" folder.

Your house has an address. The same is with your computer - any computer.

Each computer connected to the Net has an address.
Suppose your address is "2012 Big Road Street."

On the Internet, the 'address' is called an **IP**: Internet Protocol. **BOTH** servers **AND** clients have an Internet Protocol address (IP address).

IP is a number that consists of 4 numbers separated by a dot (.)

ALL computers that are connected to the Net correspond to an IP address.

Since a **SERVER** is connected to the Net on a 24/7/365 basis, all servers have a standard IP address - each server has its own unique

IP. YOUR computer also has an IP address for the time you are online. Each time you connect to the Net, you get a different IP address. Your IP address is a number that identifies you on the web while you surf. Here is a site you can use if you want to know your IP address each time you're connected to the Net:

http://www.urgentclick.com/whats_my_ip_address.php

Each time you connect to the Net, you get a different IP - a set of 4 numbers separated by dots.

I repeat it: **ALL computers that are connected to the Net have an IP address.** IP addresses are like *fingertips*: all are different from each other. So, your IP address is your digital fingertip, thus your **identity** online.

Now you can also understand how **HOSTING** works.

You can register a domain name, say `www.mydomain.com`, but if you don't HOST your domain name on a Server - somewhere on the Net - you can't do anything with it. Since the Internet is millions of computers connected to each other and each one of them has an IP address, you can easily imagine that each IP corresponds to a DOMAIN NAME.

Suppose there is a company named 'My Hosting Web.'

This company has a Server.

UNIX OS has been installed on this server.

There is also a FTP daemon installed.

Suppose that the domain name this company uses is 'My Hosting Web' at `http://my-hosting-web.com` (this is just an example) and that this domain corresponds to an IP address, say `198.22.34.09...`

Using specific daemons, 'My Hosting Web' company can SELL hosting to you. For the reason that 'My Hosting Web' company sells space for hosting, YOU will use 'My Hosting Web's' IP address in order to DECLARE YOUR website's address on the Net too - your PERMANENT website's address.

A HOSTING SERVER's IP is named "**Name Server.**"

The distinction between a simple "IP address" and a "Name Server" lies in the fact that a **COMPUTER-server which has a NAME SERVER can ASSIGN IP addresses to OTHER domain names too.** That's how HOSTING works.

So, ALL domain names that are HOSTED by a server obtain a specific IP address too. Actually, "domain names" were invented by some brilliant minds for commercial reasons because a "name" is something a Client (you) can remember. You cannot remember numbers like 198.88.92.0, but you can certainly remember www.mydomain.com, right?

Again, if you don't HOST a domain name on a (hosting) server, the domain name will have NO IP address and you WON'T be able to do ANYTHING with it.

Back to our example with 'My Hosting Web' company. Say that the "Name Servers" this company uses are 198.22.34.09 and 198.22.34.10.

Name Servers are like domain names too. They can be assigned to specific combinations of words. Thus the above IP addresses can be replaced by NS1.MY-HOSTING-WEB and NS2.MY-HOSTING-WEB accordingly.

Now, suppose you want to register a domain name, say www.mybigwebsite.com.

You go at <http://www.software4profit.com/domains> (a GoDaddy.com affiliate site), search for that domain name, and find out that mybigwebsite.com is available. During the registration process, you'll be asked what you want to do with that domain name.

Do you want to PARK it for free, meaning that you will NOT use it at all?

Or do you want to HOST that domain name? If you choose to host this domain name, you'll then be asked to fill in the NAME SERVERS of the hosting company you'll use.

I'll give you an example with the Name Servers of the hosting company I use for the majority of my sites. I use Host4Profit.com

What I did with the domain name software4profit.com:

I went to Godaddy.com, registered that domain, and when I was asked for the Name Servers, I gave these: NS.MY-EHOST.COM and NS2.MY-EHOST.COM. These are the actual Name Servers of the Host4Profit.com hosting company.

Then I went at Host4Profit.com and ordered a hosting package. I told them that I had already registered my domain name and also had declared their Name Servers. What that company did was ASSIGN an IP address to the domain name 'software4profit.com'

The activation of my domain name and my hosting package meant that:

- a) When someone would type the IP address or software4profit.com into their browser, my site would be displayed; and

- b) That I could use the IP address or the domain name software4profit.com as an address and use my username and password in order to CONNECT to HOST4PROFIT.COM's SERVER, and furthermore, to MY AREA on THAT Server.

Do you remember the ***PATHS*** of the computer I explained to you already?

I told you that **C:** is the ROOT path of your computer, so if you created a file, test.txt, and saved it to the ROOT, then its address would be C:\test.txt. If you created this file, test.txt, and saved it to the folder My Computer, then the PATH to this file would be C:\My_Computer\test.txt.

I also told you that the same thing happens when you store your documents onto the server. If your domain name is www.mybigwebsite.com, and you save a file test.html to the ROOT section, then its address will be <http://www.mybigwebsite.com/test.html>.

If you create a folder, 'demo', on the server that hosts your pages and save the file, test.html, in the 'demo' folder, then you can access that file by typing <http://www.mybigwebsite.com/demo/test.html>.

In order to BE CONNECTED to the hosting company's server, you need an ***FTP client***, a software you install on your computer. I use the famous free FTP client **WS FTP LE** (LE means 'Light Edition'). You can download it from here: <http://www.ipswitch.com>

Once you install it, open it: double click the icon of the program through your Windows Explorer or find it through Start/Programs.

When you open it, you'll see that it asks you to type the identity of the computer you want to be connected to. Actually, it asks for:

Profile Name:

Host name/Address:

Host type: Automatic (default - never change it)

User ID:

Password:

Account: - never use it, and

Comments

If you have read carefully what I have described so far, you can now understand that this program asks for your IP address (Host name/Address:) - this is the IP address of your domain name; and also asks for your username and password (User ID: and Password:).

Your hosting company provides you with all this information. So type them in, write anything you want in 'Profile Name:' (just to name this connection and distinguish it from any others you'll create in the future), and check the option 'Save Pwd' so you'll not have to type your password again. Then click 'OK.'

WS FTP LE will start working and you can actually see the procedure of connecting your computer to your hosting company's server, asking for the login name (username) and the password, and thus, allowing your computer to 'enter' the server. Then you'll see on the right side of the program the contents of the server: this is your online area.

WS FTP LE uses two windows: The one on the left shows the contents of YOUR computer. The other one on the right displays the contents of your server (your "area" on that server, because many other users (clients) could have obtained a hosting area on the same server too.) I told you that this is how hosting works: The

hosting company divides the space on the server into accounts and then assigns IP addresses (& domain names) to each account.

So, with WS FTP LE, you can send files (upload) from your computer to your online area and you can also send files (download) from your online area to your computer inside the folder you choose in the left window.

You select the file you want from your computer (window on the left) and hit the 'right arrow' --> for UPLOADING the file to your online area.

You select the file you want from your online area (window on the right) and hit the 'left arrow' <-- for DOWNLOADING the file to your computer.

The file you download will be saved inside the chosen folder on your computer in the window on the left, while the file you upload will be saved inside the chosen folder in your online area in the window on the right.

The structure of your online area is similar to the structure of your computer.

In your computer, the root area is C:\ and there are folders and sub-folders.

In your online area and in most cases, the 'root' is the 'www' folder, or the 'htdocs' folder, or the 'public-html' folder, or the 'public-www' folder. Whatever you UPLOAD inside that folder is VISIBLE online.

If you upload the files test.html and test1.html and then type your domain in your browser, you'll see a blank page with these 2 files listed:

test.html
test1.html

However, if you also upload a file index.html (even a blank one) and then type your domain in your browser, you will see what is included inside the index.html file (read my last article to understand the structure of your area on the server.)

As I told you, I use UNIX, and most of the hosting packages that are offered today are based in UNIX or Linux servers.

Always remember these rules when transferring files on a UNIX system:

- Transfer ALL documents that end in *.htm* (asterisk stands for 'anything here') using the **ASCII** mode.
- Transfer ALL images that end in .gif and .jpg, and all files that end in .doc (Microsoft Word files), in .pdf (Adobe Acrobat files), and in .zip (compressed files) using the **BINARY** mode.

In WS FTP LE, you can choose the transfer mode by checking:

☐ ASCII ☐ Binary ☐ Auto

(**NEVER** use the Auto choice because errors always occur.)

You see, the server wants to know what the file contains. If you "say" to the server that the file is "ASCII," then it will understand that text is included and it will NOT compress the file; it will allow the spaces to exist, the rows, etc... If you upload a text file in BINARY mode, then the same file will lose its text format and it will not be recognizable since the server will compress it as it does

with all BINARY files (actually, the majority of the files uploaded in Binary mode are already compressed.)

Apart from your 'root' folder, you'll see that there are some other folders in your server area, like '**email**' or '**cgi-bin**' or '**stats**' - it depends on the hosting company you choose. The existence of the 'cgi-bin' folder means that you can run .cgi or .perl scripts in your server area. Some companies offer a 'cgi-bin' folder, while others do not. Some companies offer you a 'stats' folder, while others do not allow you to read your server's area statistics.

If you become familiar with the STRUCTURE of your computer, practice transferring files within the structure of your server, and use the ASCII and BINARY modes correctly, you do not need anything else in order to start FTPing and to create your website.

Using WS FTP LE, you can also:

Create directories in your online area: Click 'Mkdir', insert the name of the new folder in the pop-up window, and click OK; the folder will be created in a few seconds.

Rename any file: select the file you want to rename, click the 'Rename' option, insert the new name in the pop-up window, and click OK. The file will be renamed and the server will refresh the contents of your area.

Delete any file: select the file you want to delete, click the 'Delete' option, and the server will refresh the contents of your area by deleting the file you chose.

Edit any file without downloading it onto your computer: select the (text or html) file you want to edit, click the 'Edit' option, and Notepad will open with the contents of the file you chose. You will be asked whether to save the changes or not when closing the

Notepad window so BE SURE of what you're doing. Personally, I started using this option after some years of FTPing and when I felt ready to "risk" any change (and damage:-)):) Always download/backup the changed file onto your computer too so that you don't lose the file if the server breaks down!

You can use the same options (Mkdir, Rename, Delete) for your files inside the window on the left, the files on your computer.

You can also use the "**Refresh**" option whenever you want the contents of the specific window you selected to be updated with the changes you have made.

These are the main features of the WS FTP LE FTP client. It's an easy program to use and I am sure you'll be able to upload and download your files in a few seconds after you've installed it on your computer.

If you don't have your own website and want to test anything you read in this article, you can create a free account at <http://www.freesevers.com> and test how to upload and download files. There are many other 'free hosting' companies online, but it happened that I used that one for my first steps and that's why I suggest it to you.

That was a big lesson indeed.

I know that in part, I went from one subject to another and back and forth again and again. This was not an accident.

In order to understand how FTP works, you need to know everything explained here. I chose this way of writing and I am sure that when you reach the end of this article, you will understand everything. Am I correct? ☺

*Some serious notes need to be added about **HOSTING**.

Host4Profit.com has been the first PAID hosting service I used – it really amazed me. However, years passed and more hosting services appeared. Choosing where to host your domain name and actually YOUR BUSINESS is a very tough decision. While Host4Profit.com and all other hosting services recommended in this ebook are some of the best services offered online, here is what a very good friend and wise marketer REVEALED to me: Choose a host that does NOT offer an affiliate program.

Yes, I know this sounds crazy; however it's 100% the truth and what you must do when choosing a host for your business site. The reason for this is because hosting services that do offer affiliate programs can be blacklisted easily. What that means? It means that there can be affiliates or even customers of those services that will spam people just to make a commission. This is NOT what will happen for sure with all the hosting services that offer an affiliate program, but think about it... if this happens in the future, you will have to move your sites, re-configure your databases and scripts and believe me: that's TOUGH too...

So, trust me and use a hosting service that does not offer an affiliate program for hosting your main business. You can use any other hosting service you want for your secondary sites, or even for using them as back up servers or for delivering your files. But do use a “clean” hosting service for your main business.

Think about it: If an affiliate program did not exist, why someone would suggest a service to others?

There are many reliable hosting services that offer pure hosting without commissions. If I were you I'd select a

Reseller Package, that is: a hosting account through which I could resell hosting too. BUT... I SHOULD NOT use my reseller account for reselling hosting... On the contrary: I'd use it for my own websites. You'll see that you will have to register many domain names in the future and host many different sites for those domains. A host that allows you to resell hosting also allows you to assign hosting accounts to your own domain names.

Thus: do select a Reseller Package from a hosting service that does not offer an affiliate program. This way you can host unlimited domain names, park some of them, or forward them to your server, etc. Use that reseller package FOR YOURSELF. Plus, you can benefit from reselling few accounts to selected customers of yours in the future.

I can suggest you the host that I used for Santa's Deal Time. It's the same host I use for all of my domain names and hosting needs right now. That host facilitated my needs for the site to be available to 1000s of people and ready to accept orders no matter the heavy traffic and 1000s of hits. With THAT host I can easily register a domain name and have the hosting active in less than 15 minutes – yes, this is a FAST service. If you're interested to know about that hosting service (that means you really care about your online business) drop me an email to hosting@turbodays.com

[top](#)

Part 7: When John met Amazon, Allen, Corey, and Ken.

In this ebook, I describe my early days on the Net. As I already told you, I started surfing on the Net in 1994 and since then, I have spent more than 15,000 hours online. By the end of this ebook, you'll be able to call yourself a **WEBMASTER**.

The story continues...

So... I was able to read **HTML** code, to create web pages with **META**-tags and content that could rank my pages in the Top listings of search engines, to use **SSI** tags in order to easily update my pages, to **register** a domain name, to **host** my site using free hosting servers, and to use **FTP** for uploading my pages and testing them online.

I was surfing the net gathering information, reading articles, trying to find out how others use the Internet for profit. I did not know anything about **AFFILIATE** programs at THAT time (1996-1997), and actually, there wasn't any affiliate program to promote. If I remember right, it was only when Amazon.com appeared that they offered to all of us a way of earning money by linking to their books. The concept was simple enough: "just link to us and earn money if someone buys." Clever. I believed that if I created a website with many listings from Amazon.com and then promoted this site to search engines, some 1000s of visitors would come and buy through my affiliate links, thus making me rich!

That was a mistake!

Amazon®'s payout would not make me rich, but I did not realize it at the time. Since there was no other affiliate program to compare

it with, I decided to create a small portal and include some 100s of books from Amazon.com.

I did it. I created a catalog of 1200 books and promoted that website to the search engines. The traffic was good - those pages still attract traffic, but the biggest check I received was less than \$100.00 in 4 months (Amazon pays quarterly.) I had to create some 100s of pages and invest some 100s of hours in order to learn this simple lesson: **you cannot make money from an affiliate program that pays you 2.5% or 5% commissions per sale! :-**(However, I am still getting checks from Amazon. The pages I created some years ago still send commissions to me, but in simple words: Amazon cannot make you millionaire by the commissions offered. Or at least – it did not work for me.

By the way, if you're not familiar with affiliate programs, please visit this site: <http://www.associateprograms.com>

Creating a portal with books from Amazon was not an easy task! That was the time when I learned how **CGI** and **perl** scripts work. I spent some hours reading about cgi scripts, installing and testing cgi scripts, and trying to figure out why these *d**n* things didn't work... I was searching for a script that would allow me to create something like an online catalog. What I did was visit other websites that offered online catalogs and seen how they did it. That's how I discovered the best script offered at that time for creating online libraries with links. The name of the script: "Links" by Gossamer-Threads.com, created by Alex Krohn. This script is offered today at <http://www.gossamer-threads.com>. I have installed "Links" about 200-250 times and still admire it. That script was the inspiration for ALL the similar scripts that are offered today. Alex Krohn was a pioneer.

That was also the time I discovered the web pages of Matt Wright. Matt was a young boy (in 1996) with a talent for writing cgi

scripts. You surely know his scripts. His "WWWboard" script became the basis for all the forthcoming message boards and forum scripts. You can visit his site at <http://scriptarchive.com>. He is also the creator of <http://cgi.resourceindex.com>.

This site, <http://cgi.resourceindex.com>, "Hot Scripts," <http://www.hotscripts.com>, and "The Perl Archive," <http://www.perlarchive.com>, are still my starting points when I want to search for a perl script. And I assure you: I have installed and tested the majority of all the free perl scripts offered at those sites:-) "Big Nose Bird," at <http://bignosebird.com>, is also an excellent resource for HTML, free CGI Scripts, graphics, tutorials, and more. I remind you that I prefer UNIX servers. All cgi scripts have been created to work on UNIX systems, while some of them can also work on Windows systems. The majority of the UNIX/Linux servers today offers you a cgi-bin you can use to upload and test your cgi scripts. [Host4Profit](#) is the hosting service I recommend, so if you need a host, search no further: they provide excellent hosting packages and 5-star customer service!

There is another reason why I suggest this hosting company to you. I know who created it. And here is the story...

As I told you, I was searching for ways to make the Net profitable. **FEW**, very FEW people existed at that time that could provide you with stable suggestions on how to make money on the Net.

Allen Says was (and still is) one of the pioneers of Internet Marketing.

Some years ago, I read an [interview](#) Allen gave to Bryan Kumar. Allen created the first "membership site" for "Internet Marketers" and called it: [The Internet Marketing Warriors](#). The "[Warriors](#)" is a SCHOOL. Many BIG names started their businesses by being "[Warriors](#)" first. The funny

thing is, according to what Allen said to Bryan in that interview, he (Allen) had NO idea what he was creating... Can you imagine it? He created a TEAM that counts some 1000s of members today, but that wasn't his initial intention! He just wanted to create something unique and profitable. You can get Allen's interview [here](#).

"[Warriors](#)" was a membership site that offered some 100s of ebooks, reports, and software; most of them had resell rights. The famous "650 reports" package was the first product I sold online:) and I got it from Allen's site. The lifetime subscription to Warriors also included access to the famous Warriors' Forum.

Years passed, [Warriors](#) and Allen himself grew up, Internet Marketing as a subject became bigger, and one day, Allen announced his new service: [Host4Profit](#). This was the first hosting company that offered a very good hosting package and also an affiliate residual program. That means for every client you refer, you're getting paid \$10 per month.

Finally, Allen decided to offer ALL of his products through 1 site: the Warriors' site as it is today. A membership to the [Warriors](#)' site today offers a) access to the [Warriors](#) membership vault full of ebooks, software, reports, and other goodies, b) access to the [Warriors](#)' forum, c) hosting through [Host4Profit](#), etc.

I can assure you that becoming a [Warrior](#) is one of the most important decisions in your marketing life - it's actually a **MUST**.

While I was going from site to site, trying to create some websites with STYLE, I came upon a site titled [marketingtips.com](#). What I liked very much was the first paragraph of the sales copy. It's not the same today, and if I remember it right, it was something like this:

"Today, I am very mad. I am sick of the promises made by the so-called Gurus on how to make money on the Net."

...or something like this.

It was GOOD sales copy...

It gave you the impression that you had discovered the TRUTH about Internet Marketing from a man who had created a profit on the Net!

That was the *longest* sales letter I ever read. The writer offered an expensive Internet Marketing course. By the overall meaning of what he was describing, I decided to get his course. This was how I met [Corey Rudl](#) and how I got his course - [the FIRST Internet Marketing course ever written and still one of the best](#).

Some days after my order, I heard the bell in my office, opened the door, and got a HUGE parcel. I felt so excited that I kept the parcel to remind me of this excitement. I opened it and felt VERY proud of my decision. [Corey's course](#) was my Internet Marketing **BIBLE**. I learned a lot - almost everything an Internet Marketer should learn. It is an "ALL in 1" [course](#) and it describes EVERYTHING you need to know. Corey was right: he knows what he is talking about. Allen Says' membership site ([Warriors](#)) and [Corey's course](#) were my first "marketing-weapons."

After some years, Ken Evoy published his first website with his [MYSS!](#) (Make Your Site Sell!) ebook. [MYSS!](#) was the first commercial **PDF** file on Internet Marketing.

Ken is a good writer indeed and his ebook is cheap. I bought it too. Some people say that you don't have to get Corey's expensive course if you get Ken's ebook since Ken offers almost everything

that is inside Corey's expensive course for a cheaper price in [MYSS!](#) You choose.

They (Ken and Corey) both offer good affiliate programs with high commissions. Ken's 5-Pillar (and now his new 6-Pillar) affiliate program has been described as one of the BEST on the Net. You can find it [here](#). You can also download many of Ken's products for free [here](#).

In summary:

- I learned what an affiliate program was via the hard way: by creating a portal with links to Amazon.com's books. I am still getting commissions from that portal, and I learned how to use cgi and perl scripts.
- By joining the [Warriors'](#) community by Allen Says, I started learning what other marketers knew and got the help I needed as a newbie.
- By reading Corey Rudl's [Internet Marketing Course](#), I learned almost everything about Internet Marketing and got the knowledge I needed on how I could use the web, email, USENET, search engines, forums, etc. for creating and promoting my sites.
- And by reading Ken Evoy's [MYSS!](#), I got additional resources for creating a site that could sell.

All the above resources gave me the knowledge I needed in order to create my sites. Finding out "**what I wanted to do and what I could do**" was **the most difficult task...**

[top](#)

Part 8: When John met Amazon, Allen, Corey and Ken - *continued.*

There I was... able to read HTML code, to create web pages with meta-tags and content that could rank my pages in the top listings of the search engines, to use SSI tags in order to update my pages easily, to register a domain name, to host my site using free hosting servers, and to use FTP for uploading my pages and testing them online. I even bought Allen Says's membership to '[Internet Marketing Warriors](#),' Corey Rudl's [Internet Marketing Course](#), Ken Evoy's '[Make Your Site Sell!](#)' (MYSS!), and understood that there wasn't any 'instant cash' method after I spent some 100s of hours creating a portal with books from Amazon.com just to realize that I could not make a fortune by earning a few pennies through my affiliate commissions.

The story continues...

I must admit that Corey's [Internet Marketing Course](#) was indeed an eye-opener. There were so many tricks inside that course that proved to me how inexperienced I was. You see, there is a way to create success by following 100 steps, but there is also a way to reach the same success level by following just ...3 steps. That's what Corey taught me: to work **smarter** - not necessarily **harder**.

[Ken's ebook](#) is also a GEM because of the resources listed in it. You see, some years later, I understood that what Ken had taught me 5-6 years ago: *A Site Either Sells Or Not.*

REMEMBER IT!

A Site Either Sells Or Not.

There is no middle point on this.

If you have a website, then you are either a freebies provider or you created your site in order to SELL. Quite a simple lesson, but VERY difficult to UNDERSTAND it.

Ken knows his subject and best of all: he is a RESEARCHER - he has a Ph.D. So, after MYSS!, I also ordered [MYKS!](#) That stands for '[Make Your Knowledge Sell!](#)'

In [MYSS!](#), I learned how a good-selling website could be created and how I could make my website stand out from the 1000s of websites; in [MYKS!](#), Ken described how I could create my...

...INFO-PRODUCT!

Now, please remember that Jim Edwards published his ebooks, '[Ebook Secrets Exposed](#)', '[Write And Publish Your Own Ebook In As Little As 7 Days](#)' and his '[Turn Words Into Traffic](#)', some YEARS after Ken published [MYKS!](#)

Ken describes nearly EVERYTHING in [MYKS!](#): where to find your sources and create your ebook, sell it, promote it through other websites, etc. I remember that I was so excited when I read [MYKS!](#) that I wanted to create a button, something like '*This site was created after I read [MYKS!](#)*' for all of the sites I'd create in the future!

Note this: There is information overload online.

You'll find 1000s of sources dealing with the SAME subject.

YOUR role is NOT to get ALL the 1000s of sources - you cannot afford to do this because you'll waste your money and you won't have the time to read everything. So, your role is AND will always be this:

FIND THE FOUNTAIN of the information.

Information is like a river. **You must find where that river starts** and NOT to what sea this river ends. If you do this, you'll save time and money.

Ken's [MYKS!](#) is the fountain of all these ebooks that describe how to create your information product even if you can't write a simple letter. That's true: you can create ANY ebook you want even if you don't know how to write a simple letter. It's the **TECHNIQUE** you'll use to create your ebook and the **STRATEGY** you'll use to promote and sell it.

Again, I do not underestimate any product. I have bought and read almost **every** existing ebook on the Net today that teaches you how to create a product and how to sell it. However, I always recall what Ken describes in [MYKS!](#) - it's the **FOUNTAIN** of this kind of knowledge.

You may think:

'Come on John... Today, you're suggesting [MYKS!](#), in the last issue, you mentioned [MYSS!](#), [Corey's](#) expensive course, and Allen Says' [Warriors](#). Tomorrow you'll tell me about the X ebook, the Y software, and the XYZ package. I believe that I can create anything I want using free resources.'

I agree with you.

That's what I believed in the beginning too.

Actually, you can find ANYTHING you want on the Net today for free. But **if you want to create a successful business based on FREE information, you must have TIME to spend, COURAGE to continue after your failures, and**

PERSISTENCE to eventually learn through this free-information highway what you'd have learned sooner and easier through the resources I am suggesting to you.

You see... **YOU MUST INVEST** in order to get the knowledge you need to go on and succeed. Tom Hua, the creator of the best seller 'Free To Sell' package, also repeats this in one of the interviews he has given:

“Invest if you want to last.”

Now, listen to me.

I do not suggest all these resources here because I want you to buy them. Don't do it. I repeat it: **DO NOT buy anything.**

Here, I am TEACHING you what you need to know in order to survive in this Internet Marketing Era and find your way to riches. Yes, I mean it. You CAN make a fortune on the Net, but you must **invest** TIME to create, test, fail, create again, fail again, and finally find your stable way of making money online.

Did anyone tell you that making money on the Net is easy?

Well, it's not.

It's an **Art**.

It's a **Science**.

It's a matter of **knowledge**.

It's a **“University course.”**

You must **invest** time to pass your lessons, to learn everything, and then to give your business everything you've got. A lawyer needs to spend some years in order to get his degree, then practice law, then find some customers, and then make his own money. The

same happens to mastering the Web. **You must get your lessons, study, practice, and then earn.**

On the other side, there are children and miracle-students that can get a University degree at the age of 13 - I know it, but I am not one of them. **I learned the hard way.**

People like Terry Dean, Frank Garon, and others describe how they quit their jobs in 6 months, 2 years or less after they started on the Internet. I wish **YOU** are one of these 'miracles' - I really wish it, because **I** can learn from **you** and **you** can prosper online easier and sooner than any other 'student.'

But, don't hang your head by admitting you're not a 'miracle.' I admit it – I am not. And for me, this is a way of self-knowledge.

If you have ever been a University student, you know that there are:

- the lessons
- the bibliography
- the assignments

I am giving you the **lessons** and the **bibliography**: all the resources you need to get your '*degree*.' I even suggested some subjects through which you can practice the knowledge you're absorbing.

You need to test your knowledge and prove (*to yourself*) that you know what you're doing. But 'getting your degree,' establishing a professional attitude, and creating your unique position in this market depends completely on **YOU**.

Wishing you ALL the best!

John Delavera

Epilogue

Thus ends **My Early Days on the Net**, but certainly not technological advancements, nor John Delavera's Internet Marketing viral marketing strategies and products.

Keep abreast of the latest Internet marketing trends along with John's own expansion into the Internet marketing arena.

Here are a few ways to do this:

1. Subscribe to his Free, winning [TurboZine](#) Newsletter.
2. Bookmark and return regularly to [John's Blog](#).
3. Contact him through feedback@turbodays.com

Thank you for reading and for your time. Your support and patronage are very much appreciated. If you liked this ebook you can send your testimonial to feedback@turbodays.com

\$\$\$ Make Money \$\$\$

Would you like to know how to make this ebook your 24/7 "digital salesperson" - [click here for more information](#).