**HTML (Hypertext Markup Language) is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user. Each individual markup code is referred to as an element (but many people also refer to it as a tag). Some elements come in pairs that indicate when some display effect is to begin and when it is to end.**

**XHTM**

**XHTML stands for EXtensible HyperText Markup Language and is the next step in the evolution of the Internet. The XHTML1.0 is the first document type in the XHTML family.**

XHTML was developed by the W3C to help web developers make the transition from HTML to XML. By migrating to XHTML today, web developers can enter the XML world with all of its attendant benefits, while still remaining confident in their content's backward and future compatibility.

Developers who migrate their content to XHTML1.0 will realize the following benefits:

* XHTML documents are XML conforming. As such, they are readily viewed, edited, and validated with standard XML tools.
* XHTML documents can be written to operate better than they did before in existing browsers as well as in new browsers.

XHTML documents can utilize applications like scripts and applets that rely upon either the HTML Document Object Model or the XML Document Object Model.

**DHTML**

**DHTML stands for 'Dynamic HTML'. Many people are familiar with Dynamic HTML, but many don't know what it really is. Before we get into what DHTML is, we should first point out what DHTML is not.**

**XML**

**XML (Extensible Markup Language) is a flexible way to create common information formats and share both the format and the data on the World Wide Web, intranets, and elsewhere. For example, computer makers might agree on a standard or common way to describe the information about a computer product (processor speed, memory size, and so forth) and then describe the product information format with XML. Such a standard way of describing data would enable a user to send an intelligent agent (a program) to each computer maker's Web site, gather data, and then make a valid comparison. XML can be used by any individual or group of individuals or companies that wants to share information in a consistent way.**

**HTML5**

HTML5 is the newest hyper text markup language for websites from the World Wide Web Consortium (W3C). The first draft was made public in2008, but not much happened until2011. In2011, HTML5 was released and people started writing about it and using it, but the support in different browsers was still poor. Today all major browsers (Chrome, Safari, Firefox, Opera, IE) offer HTML5 support, therefore the newest HTML technology can be used at its best today.

**CSS**

CSS was first developed in1997, as a way for Web developers to define the look and feel of their Web pages. It was intended to allow developers to separate content from design so that HTML could perform more of the function that it was originally based on - the markup of content, without worry about the design and layout.

CSS didn't gain in popularity until around2000, when Web browsers began using more than the basic font and color aspects of CSS. And now, all modern browsers support all of CSS Level1, most of CSS Level2, and some aspects of CSS Level3.

Web Designers that don't use CSS for their design and development of Web sites are rapidly becoming a thing of the past. And it is arguably as important to understand CSS as it is to know HTML - and some would say it was more important to know CSS.

# Difference between HTML, XHTML, DHTML and XML

##### **Key difference: HTML stands for HyperText Markup Language. It is a well known mark up language used to develop web pages. It has been around for a long time and is commonly used in webpage design. XHTML stands for Extensible HyperText Markup Language. It is a markup language written in XML. Essentially, it is a hybrid between HTML and XML specifically designed for Net device displays. It is HTML defined as an XML application. DHTML is essentially Dynamic HTML. It is a new way of looking at and controlling the standard HTML codes and commands. DHTML is a collection of technologies that are used to create interactive and animated web sites. XML stands for Extensible Markup Language. It is a specification developed by the W3C. It is a markup language designed especially for Web documents. It defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.**

HTML stands for HyperText Markup Language. It is a well known mark up language used to develop web pages. It has been around for a long time and is commonly used in webpage design. XML or Extensible Markup Language defines a set of rules for encoding documents in a format that can be read by both, human and computer.

HTML is written using HTML elements, which consist of tags, primarily and opening tag and a closing tag. The data between these tags is usually the content. The main objective of HTML is to allow web browsers to interpret and display the content written between the tags. The tags are designed to describe the page content. HTML comes with predefined tags. They allow one to insert images, text, videos, forms and other pieces of content together into a cohesive webpage.

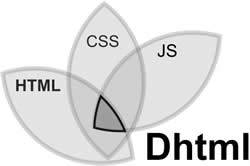
Elements of HTML are the basic building blocks of all websites. HTML allows images and objects to be embedded in the webpage. It can also be used to create interactive forms. HTML also provides the means to create structured documents. It does this by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. However these days, web pages are rarely designed using only HTML. HTML allows for the programmer to embed scripts written in languages such as JavaScript, which many often do. This changes the look and behavior of the HTML web pages.

XHTML, on the other hand, stands for Extensible HyperText Markup Language. It is a markup language written in XML. It is a collection of XML markup languages that mirror or extend versions of HTML. Essentially, it is a hybrid between HTML and XML specifically designed for Net device displays. It is HTML defined as an XML application.

XHTML uses three XML namespaces that correspond to three HTML 4.0 DTDs: Strict, Transitional, and Frameset. These namespaces are used to qualify element and attributes names by associating them with namespaces identified by URI references. The namespaces also prevent identically custom-named tags, which may be used in different XML documents from being read the same way. Hence, each custom tag is read differently.

XHTML markup must conform to the markup standards defined in a HTML DTD. In fact, XHTML is almost identical to HTML 4.01. However, it is a stricter and cleaner version of HTML 4.01.

As XHTML is in part HTML, it is supported by almost all major browsers. However, in order to be compatible with the said Net devices, XHTML must go through a modularization process. In this, the device designer will specify which elements are supported by using standard building blocks. The content creators can then target these building blocks or modules. These modules conform to certain standards. Hence, XHTML extensibility makes sure that the layout and presentation stays more-or-less the same over various platforms.

Whereas, DHTML is essentially Dynamic HTML. It is a new way of looking at and controlling the standard HTML codes and commands. DHTML is a collection of technologies that are used to create interactive and animated web sites. DHTML gives more control over the HTML elements. It allows one to incorporate a client-side scripting language, such as JavaScript, a presentation definition language, such as CSS, and the Document Object Model in HTML web pages.

DHTML also allows the pages to change at any time, without returning to the Web server first. It allows scripting languages to change a web page's look and function after the page has been fully loaded and during the viewing process. It also allows the user to add effects to their pages that are otherwise difficult to achieve.

Wikipedia list additional DHTML features, such as DHTML allows the developers to:

* Animate text and images in their document, independently moving each element from any starting point to any ending point, following a predetermined path or one chosen by the user.
* Embed a ticker that automatically refreshes its content with the latest news, stock quotes, or other data.
* Use a form to capture user input, and then process, verify and respond to that data without having to send data back to the server.
* Include rollover buttons or drop-down menus.

Moreover, XML stands for Extensible Markup Language. It is a specification developed by the W3C. It is a markup language designed especially for Web documents. It defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It allows designers to create their own customized tags. It also enables the definition, transmission, validation, and interpretation of data between applications and organizations.

XML is a text-based data format with strong support via Unicode for languages. It emphasizes simplicity, generality, and usability over the Internet. It is also widely used for the representation of arbitrary data structures, especially in web services. Programmers often use APIs while processing XML data and schema systems to aid in the definition of XML-based languages.

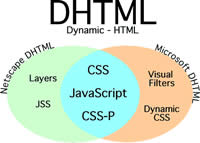
The XML syntax has formed the basis for many document formats, such as RSS, Atom, SOAP, and XHTML. In fact, XML-based formats have become the default for many office-productivity tools, including Microsoft Office, OpenOffice.org and LibreOffice, and Apple's iWork.

##### **Key Difference: DHTML (Dynamic HTML) is not a coding language but actually a collection of technologies that are used with HTML in order to make HTML more interactive. XHTML (Extensible Hypertext Markup Language) is coding language that is identical to HTML, however with a stricter set of syntactic rules that are based on the XML standard.**

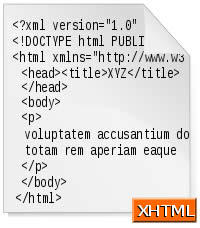
A lot goes into creating a simple web page, such as coding, placing, dynamic links, etc. During the start up stages of web browsers and web pages, HTML coding language would do the trick as the pages were simple and it did not require much interaction. However, as the pages became more complex and interactive, HTML could not fulfill all the requirements and additional technologies were added in order to make the pages more interactive, such as DHTML. DHTML and HXTML are related in the world of computing, but refer to two different things.

HTML (Hypertext Markup Language) is the original programming language that was used in order to develop web sites. It provided a simple tag-based syntax in order to create forms, em

bed pictures, links and other attributes. HTML is written in the form of tags, enclosed in angle brackets <>, which describe the tag and add content to it. HTML tags such as <h1>; <p1> lets the browser known what is enclosed in the tag, such as heading or paragraph. The tag is ended using the </h1> or </p1>. The first type of tag is known as start tag or opening tag, while the second tag is known as closing tag or end tag. The simplicity of HTML tags is what made it weaker as the lenient tags would put more work on the browser to decipher the page. As different browsers decipher pages differently, a coder designing a web page for one browser could not make the web page work properly in another browser.



DHTML (Dynamic HTML) is not a coding language but actually a collection of technologies that are used with HTML in order to make HTML more interactive. DHTML is an umbrella term for a collection of technologies such as HTML, XHTML, JavaScript, CSS and Document Object Model. A combination of these objects are used together in order to create more interactive and animated websites. One major difference between standard HTML and DHTML is once a page is loaded in standard HTML, it will not change until it receives another request from the server, however in DHTML, the elements are allowed to change at any time, without having to return to the server. Four primary features of DHTML include changing the tags and properties, real-time positioning, dynamic fonts specifically for Netscape Communicator and data binding for Internet Explorer. During the late1990s and early 2000s, DHTML was popularly used in order to create browser- based action games, however due to problems in loading for different servers, it proved difficult.



XHTML (Extensible Hypertext Markup Language) is coding language that is identical to HTML, however with a stricter set of syntactic rules that are based on the XML standard. It operates similarly to HTML, but it follows a strict set of rules, which when not followed causes the browser to generate an error code. For example, in HTML some tags can be left open and the browser will generate the data to its best ability, but in XHMTL all tags must have a proper opening and closing tag. XHTML is a combination of HTML and XML, a strict subset of SMGL. The proper tags allows easier for web pages to be rendered in almost any browser. It is a much harder language to learn language for beginners, but is still considered widely beneficial. It is considered better to code dynamic web pages using XHTML as it provides a better scalability and less cross-browser problems.

HTML is a Hyper Text Markup Language which is used to display the internet website contents on the internet browsers. This is the common language to develop the web site pages.

DHTML is Dynamic HTML(DHTML) which will be used to display the dynamic web site pages. It is not a standard defined by the World Wide Web Consortium (W3C), it is a "marketing term" that was used by Netscape and Microsoft  
to describe the new technologies the 4.x generation browsers would support. Dynamic HTML is a combination of technologies to make Web pages dynamic. To most people, Dynamic HTML means a combination of HTML 4.0, Style Sheets and JavaScript.

XHTML is similar to the HTML but follows the rules of XML. XHTML is used to be compatible with XML programming. Following the rules now would make it possible to include XML programming in the future. It is not difficult to change HTML pages to XHTML, but it can be time-consuming. Finding all line breaks and images to include closing tags, converting any uppercase to lowercase and any other incompatibility can be a nuisance. Using a find and replace program can allow you to edit your code faster, but you still have to reupload all those changes. It is recommended that programmers try to remember these rules to comply with W3C recommendations, so the web pages appear correctly in most browsers. The main differences between HTML and XHTML is the case-sensitivity, the need to use closing tags for all tags, the need to use quotes around all attribute values and that all attributes must be in lowercase as XML requires.

Keyfeature

## New Doctype

Still using that pesky, impossible-to-memorize XHTML doctype?

|  |  |
| --- | --- |
| 1  2 | <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"      "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>"> |

If so, why? Switch to the new HTML5 doctype. You'll live longer -- as Douglas Quaid might say.

|  |  |
| --- | --- |
| 1 | <!DOCTYPE html> |

In fact, did you know that it truthfully isn't even really necessary for HTML5? However, it's used for current, and older browsers that require a specified doctype. Browsers that do not understand this doctype will simply render the contained markup in standards mode. So, without worry, feel free to throw caution to the wind, and embrace the new HTML5 doctype.

## 2. The Figure Element

Consider the following mark-up for an image:

|  |  |
| --- | --- |
| 1  2 | <img src="path/to/image" alt="About image" />  <p>Image of Mars. </p> |

There unfortunately isn't any easy or semantic way to associate the caption, wrapped in a paragraph tag, with the image element itself. HTML5 rectifies this, with the introduction of the <figure> element. When combined with the <figcaption> element, we can now semantically associate captions with their image counterparts.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <figure>      <img src="path/to/image" alt="About image" />      <figcaption>          <p>This is an image of something interesting. </p>      </figcaption>  </figure> |

## 3. <small> Redefined

Not long ago, I utilized the <small> element to create subheadings that are closely related to the logo. It's a useful presentational element; however, now, that would be an incorrect usage. The small element has been redefined, more appropriately, to refer to small print. Imagine a copyright statement in the footer of your site; according to the new HTML5 definition of this element; the <small> would be the correct wrapper for this information.

*The small element now refers to "small print."*

## 4. No More Types for Scripts and Links

You possibly still add the type attribute to your link and script tags.

|  |  |
| --- | --- |
| 1  2 | <link rel="stylesheet" href="path/to/stylesheet.css" type="text/css" />  <script type="text/javascript" src="path/to/script.js"></script> |

This is no longer necessary. It's implied that both of these tags refer to stylesheets and scripts, respectively. As such, we can remove the type attribute all together.

|  |  |
| --- | --- |
| 1  2 | <link rel="stylesheet" href="path/to/stylesheet.css" />  <script src="path/to/script.js"></script> |

## 5. To Quote or Not to Quote.

...That is the question. Remember, HTML5 is not XHTML. You don't have to wrap your attributes in quotation marks if you don't want to you. You don't have to close your elements. With that said, there's nothing wrong with doing so, if it makes you feel more comfortable. I find that this is true for myself.

|  |  |
| --- | --- |
| 1 | <p class=myClass id=someId> Start the reactor. |

Make up your own mind on this one. If you prefer a more structured document, by all means, stick with the quotes.

## 6. Make your Content Editable

The new browsers have a nifty new attribute that can be applied to elements, called contenteditable. As the name implies, this allows the user to edit any of the text contained within the element, including its children. There are a variety of uses for something like this, including an app as simple as a to-do list, which also takes advantage of local storage.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16 | <!DOCTYPE html>    <html lang="en">  <head>      <meta charset="utf-8">      <title>untitled</title>  </head>  <body>      <h2> To-Do List </h2>       <ul contenteditable="true">          <li> Break mechanical cab driver. </li>          <li> Drive to abandoned factory          <li> Watch video of self </li>       </ul>  </body>  </html> |

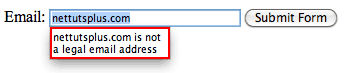
Or, as we learned in the previous tip, we could write it as:

|  |  |
| --- | --- |
| 1 | <ul contenteditable=true> |

## 7. Email Inputs

If we apply a type of "email" to form inputs, we can instruct the browser to only allow strings that conform to a valid email address structure. That's right; built-in form validation will soon be here! We can't 100% rely on this just yet, for obvious reasons. In older browsers that do not understand this "email" type, they'll simply fall back to a regular textbox.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16 | <!DOCTYPE html>    <html lang="en">  <head>      <meta charset="utf-8">      <title>untitled</title>  </head>  <body>      <form action="" method="get">          <label for="email">Email:</label>          <input id="email" name="email" type="email" />            <button type="submit"> Submit Form </button>      </form>  </body>  </html> |



*At this time, we cannot depend on browser validation. A server/client side solution must still be implemented.*

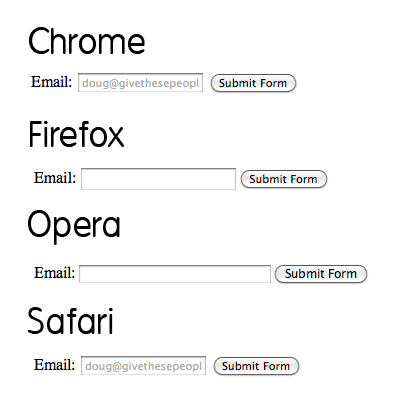
It should also be noted that all the current browsers are a bit wonky when it comes to what elements and attributes they do and don't support. For example, Opera seems to support email validation, just as long as the name attribute is specified. However, it does not support the placeholder attribute, which we'll learn about in the next tip. Bottom line, don't depend on this form of validation just yet...but you can still use it!

## 8. Placeholders

Before, we had to utilize a bit of JavaScript to create placeholders for textboxes. Sure, you can initially set the value attribute how you see fit, but, as soon as the user deletes that text and clicks away, the input will be left blank again. The placeholder attribute remedies this.

|  |  |
| --- | --- |
| 1 | <input name="email" type="email" placeholder="doug@givethesepeopleair.com" /> |

Again, support is shady at best across browsers, however, this will continue to improve with every new release. Besides, if the browser, like Firefox and Opera, don't currently support the placeholder attribute, no harm done.

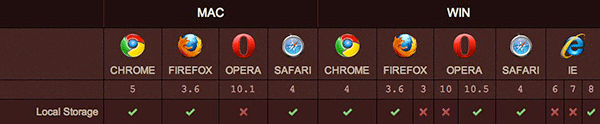


## 9. Local Storage

Thanks to local storage (not officially HTML5, but grouped in for convenience's sake), we can make advanced browsers "remember" what we type, even after the browser is closed or is refreshed.

*"localStorage sets fields on the domain. Even when you close the browser, reopen it, and go back to the site, it remembers all fields in localStorage."  
-*[*QuirksBlog*](http://www.quirksmode.org/blog/archives/2009/06/html5_storage_t.html)

While obviously not supported across all browsers, we can expect this method to work, most notably, in Internet Explorer 8, Safari 4, and Firefox 3.5. Note that, to compensate for older browsers that won't recognize local storage, you should first test to determine whether window.localStorage exists.

via <http://www.findmebyip.com/litmus/>

## 10. The Semantic Header and Footer

Gone are the days of:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <div id="header">      ...  </div>    <div id="footer">      ...  </div> |

Divs, by nature, have no semantic structure -- even after an id is applied. Now, with HTML5, we have access to the <header> and <footer> elements. The mark-up above can now be replaced with:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <header>      ...  </header>    <footer>      ...  </footer> |

*It's fully appropriate to have multiple headers and footers in your projects.*

Try not to confuse these elements with the "header" and "footer" of your website. They simply refer to their container. As such, it makes sense to place, for example, meta information at the bottom of a blog post within the footer element. The same holds true for the header.

**HTML5 Main Features Overview**

HTML5 is in market and almost all the major browser have already started supporting it. Even the Microsoft IE 9 after long period of yes/no has finally agreed to support HTML 5. If you are a web developer it's  right time to start getting grasp over this new sensation in the web world. I am saying this not only because one should be aware of new things in the surrounding, but also because its new features can help you prevent your precious time(which every IT guy is always running short of :)).

Let's start with an overview of new features introduced with HTML5.

**Canvas:**This is the most revolutionary feature HTML5 has come up with, now one can achieve great graphics and visual images on the fly using **<Canvas>** element and JavaScript.

**Video:** Another big shot, Now you no more need third party plug-ins to display your video on web page. This is the one feature that is still evolving and browser community is still working out on standarising the video format. I will discuss more about this tag in detail later in this tutorial.

**Local Storage:**This is very much similar to Cookies, but this is designed to accommodate large amount of information. Also unlike cookies it is not send to server every time the request is made. It remains on the local computer and can be loaded using JavaScript once the page is loaded.

**Web Worker:**This feature provides the browser to run JavaScript in background. This is something similar to threading concept in C#. More on this in the later part of the tutorial

**Offline Mode:**This great feature allow user to create web site that can work even in offline mode.

**Geo Location:**As the name says it all, with this you can figure out the location of the user.More on this in the later part of the article.

**Input Types:**With HTML5 comes more input types, search, email, range, color, telephone(tel),url to name some. These new addition to the input type are going to make the life of developers a lot more simpler.

**PlaceHolderText:**Ever required to create a watermark text for your textbox and ended up writing JavaScript code to handle it. This comes inbuilt with HTML5 placeholderText attribute of input type.

**Form AutoFocus:**New autofocus attribute added to all the controls. This will help add the focus on the control one wants.

**MicroData:** This provides a standardized way to provide additional semantics on the page. More on this later.

Before I go ahead with discussing on each of these topics in details, let me tell how easily we can figure out if our browser support these feature or not. I have created a simple utility that will tell what all new feature of HTML 5 is supported by your browser.

# The Difference Between Dynamic & Static Web Pages

## Static Web Pages

Static Web pages display the exact same information whenever anyone visits it. Static Web pages do not have to be simple plain text. They can feature detailed multimedia design and even videos. However, every visitor to that page will be greeted by the exact same text, multimedia design or video every time he visits the page until you alter that page's source code.

## Dynamic Web Pages

Dynamic Web pages are capable of producing different content for different visitors from the same source code file. The website can display different content based on what operating system or browser the visitor is using, whether she is using a PC or a mobile device, or even the source that referred the visitor. A dynamic Web page is not necessarily better than a static Web page. The two simply serve different purposes.

## Dynamic Page Use

Dynamic pages can serve a variety of purposes. For example, websites run by content management systems allow a single source code file to load the content of many different possible pages. Content creators use a gateway page to submit the material for new pages into the CMS' database. The dynamic page can then load the material for any page in the database, based on parameters in the URL with which a visitor requests the page. Dynamic pages are also what let users log into websites to see personalized content.

## Static vs Dynamic Creation

Developers generally create static pages with HTML, but use languages like PHP, Javascript, or Actionscript to create dynamic pages. They can also use frameworks like Ruby on Rails, Django, or Flex for dynamic pages. Dynamic languages and frameworks also have the technical capacity to create static Web page content. However, doing so creates source code that is unnecessarily complex for its purpose while being more difficult to maintain.

[Ads by Google](https://www.google.com/url?ct=abg&q=https://www.google.com/adsense/support/bin/request.py%3Fcontact%3Dabg_afc%26url%3Dhttp://smallbusiness.chron.com/difference-between-dynamic-static-pages-69951.html%26gl%3DIN%26hl%3Den%26client%3Dca-pub-3235755782694080%26ai0%3DCUFNiBA2GWJzoJ4iboAOX2YeQDZHc8OtFo_v8js0El9r0_QgQASDdj5AYKARg5dLng_wpoAHZj77-A8gBAakCn1yn68Fdgz6oAwGqBLMBT9AhaQFqFbayVhVyf_gWFknGdDO6n0eDem712hfwTZFRQrl6AR4fXqhfikHNb98jSj_SmEJeuGWbpEIkO3D1jtHYuJ71vntrwIGa06ooD7ibs6rFL0-B43CbIp06M0pg9eZF5ixxnHuBsaHNPwZIPWW5a1v2CiZp9w-QDGIx2eby5zQcv6mzr-GwUUb8JL3vnz8J437djBHWSVM7ZwRKqiFCC2Cw6QKiOO3FvC0kkR6TIaOIBgGAB4_wwQGoB6a-G9gHAdIIBQiAYRAB2BMM%26ai1%3DC-u-JBA2GWJzoJ4iboAOX2YeQDeOMnPBH2vzEpuYEwI23ARACIN2PkBgoBGDl0ueD_CnIAQGpAlWcpbR-gE8-qAMBqgSzAU_QMQtSaha2slYVcn_4FhZJxnQzup9Hg3pu9doX8E2RUUK5egEeH16oX4pBzW_fI0o_0phCXrhlm6RCJDtw9Y7R2Lie9b57a8CBmtOqKA-4m7OqxS9PgeNwmyKdOjNKYPXmReYscZx7gbGhzT8GSD1luWtb9gomafcPkAxiMdnm8uc0HL-ps6_hxEZ28CS9758_CeN-3YwR1klTO2cESqohQgtgsOkCojjtxaxgQboekyGjoAZRgAen_vUrqAemvhvYBwHSCAUIgGEQAQ%26ai2%3DCnr4lBA2GWJzoJ4iboAOX2YeQDfnU2o4H-firtbUCwI23ARADIN2PkBgoBGDl0ueD_CmgAef8lMMDyAEBqQJVnKW0foBPPqgDAaoEtAFP0EEZuHAMjYrkpmQb5hobSchjNaCYTJV6M7jRDe1M0RxOu626FSZerRHAS8UA6jrj5IKhXl68L46kHjo3ffWAxt6imf6oezaNioDOq2hCtJnx4cggT4L0Z5Avm3J8TXH29EXtbziBbI69pcdxWE8ocKRhFasbIGvhUYsDYj3d-bT7LR1TK7BmK0l3f06pVko6yf_ifN3IoCDK0ztn8bypocIDnUaqQr2njSMlJ3BHxXShJVSIBgGAB4GD6zyoB6a-G9gHAdIIBQiAYRAB2BMB%26ai3%3DC11HzBA2GWJzoJ4iboAOX2YeQDbutg9cH-8WEmt0CwI23ARAEIN2PkBgoBGDl0ueD_CnIAQGpAvpKgwYSALI-qAMBqgSzAU_QcTwMahC2slYVcn_4FhZJxnQzup9Hg3pu9doX8E2RUUK5egEeH16oX4pBzW_fI0o_0phCXrhlm6RCJDtw9Y7R2Lie9b57a8CBmtOqKA-4m7OqxS9PgeNwmyKdOjNKYPXmReYscZx7gbGhzT8GSD1luWtb9gomafcPkAxiMdnm8uc0HL-ps6_hyB9JsSS9758_CeN-3YwR1klTO2cESqohQgtgsOkCojjtxeh8R9AekyGjgAfD1f1BqAemvhvYBwHSCAUIgGEQAdgTAQ&usg=AFQjCNHArb_6S3rv5xdb0xnGUefMg_WWag)