

# IT510 \* Module 3 Reading

Please read through this PDF, as it will introduce you to some of the requirements for the Module and help guide you in completing the Assessment. The last few pages should especially be helpful if you are a career-changer.

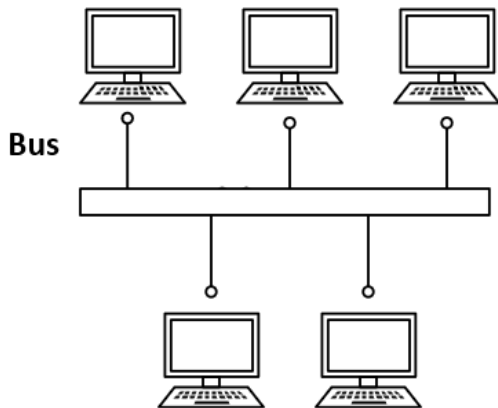
## Part 1: Networking

### Basic Topologies

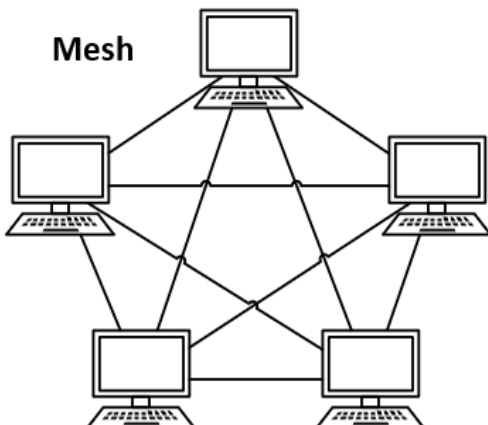
The ebook reading explains networking basics, including topologies and the hardware needed in addition to computers. Please read the additional information provided here as well.

It will be important to remember that a network might very well be a combination of topologies. Perhaps one department will be developed using Mesh, and another could be a simpler Star; the purpose of a department or a set of employees can help you make a decision about what may be best. The physical layout, cabling, overall costs, and other facts must be considered.

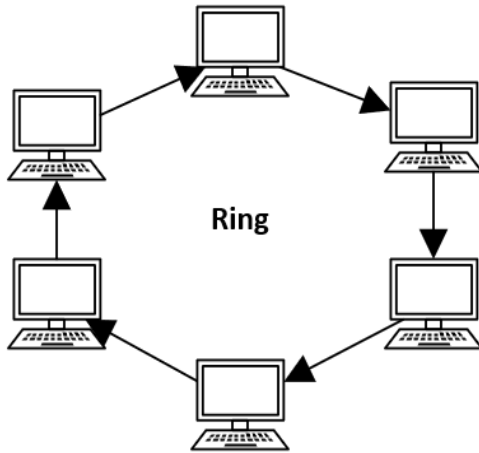
#### LAN (Local Area Network) Topologies



**Bus:** All devices in the bus topology are connected to a linear cable ("trunk," "segment," or "backbone"). This works very well for small networks, with a limited amount of cabling and ease of expansion. A limitation is that it tends to be slow.

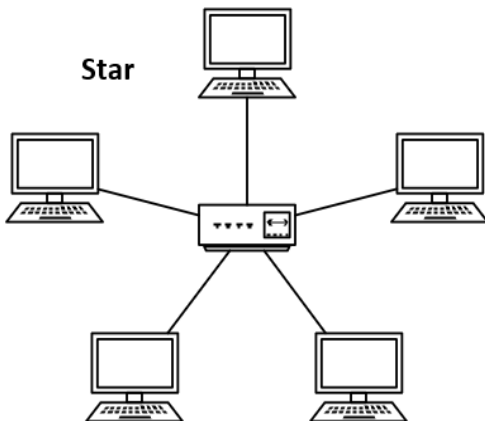


**Mesh:** This is a "point-to-point" method in which every device is connected to every other device. Data flows either through routing or flooding. It is easy to troubleshoot but difficult to install, and much more cable is necessary than in most other methods.

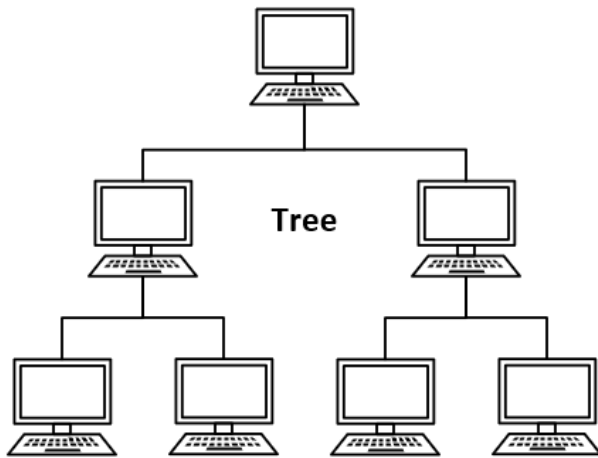


**Ring:** In this topology, a cable loop connects all devices, and data is sent via tokens. This means each computer is directly connected to exactly two other computers, with data flowing from one to another. If one device goes down, the data flow is stopped. "Repeaters" help deliver larger amounts of data, as it might have to go through the ring more than once.

Although this unidirectional (data flows in one direction), it can be reworked with two cables between devices for the ability to flow in *both* directions, thereby creating what is called a "dual ring." This method is considered inexpensive and easy to expand, but troubleshooting problems can still be difficult, and adding a device will still temporarily disturb the network.



**Star:** A hub or switch distributes data to all devices connected to the network in this commonly used topology. Data flow is fast, and it is easy to upgrade and troubleshoot. If a node fails, the rest of the network is still functional, but the hub must be reliable.



**Tree:** A hierarchy is developed, with each node (computer) connected to two others, which are then each connected to two more, and so on. This looks like a family tree in "descendancy" format, showing an ancestor and then descendants and their children, etc. This works well for computers used in groups or separate departments. Expansion and troubleshooting are easy, but is considered expensive due to the amount of cabling and maintenance issues. Importantly, if the original node fails, the entire network goes down.

Most networks are a combination of these topologies. Consider connecting all workstations with the same purpose to each other and remember that the idea is to control access.

## Other Network Components

Computers are not the only devices needed for a network, of course. First, the novice networker or systems analyst needs to know the difference between hubs, switches, and routers. While they serve some of the same purpose of providing connections to multiple devices, there are differences. In order from simple to complex, these are:

- **Hub:** Connects multiple computers using ethernet ports but has no security measures or limitation on data flow. This can be inefficient and slow.
- **Switch:** Connects multiple computers using ethernet ports and provides some security through data packet filtering and the ability to identify individual devices in the connection. This is faster than using a hub, and bandwidth can also be controlled.
- **Router:** Connects networks and controls data packets like a switch, but also uses DHCP (Dynamic Host Configuration Protocol) and NAT (Network Address Translation). This is considered an "intelligent" device, as it learns from input and output. A router controls access via firewall and may also log network activity.

Other items need to be included in network diagrams.

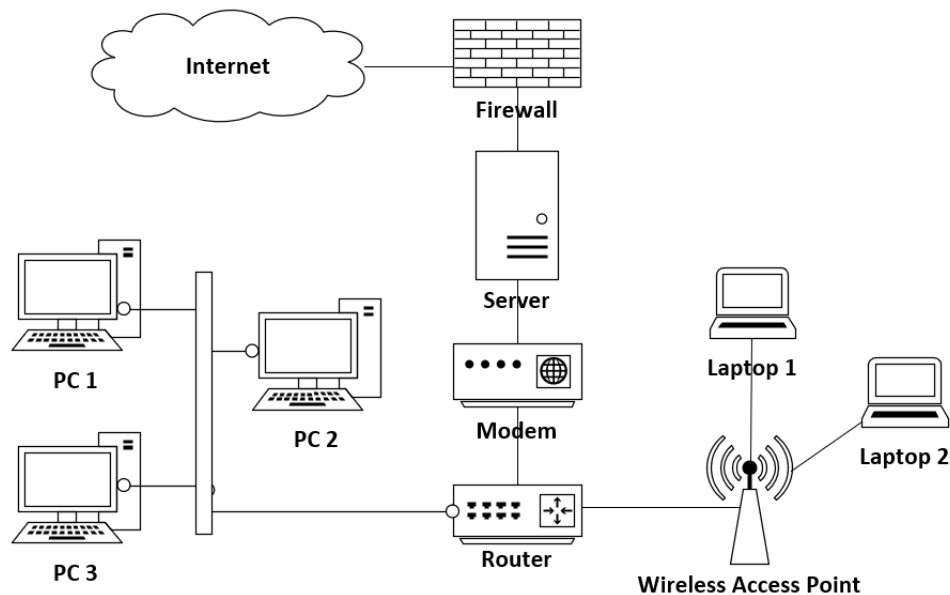
- **The Internet:** Typically diagrammed as a cloud shape in a network diagram, this is necessary for a LAN to be able to work with other networks (WAN).
- **Server:** A location for storage and retrieval of data and files. This is essential for LAN. Multiple servers are typically set on "racks."
- **Firewall(s):** Hardware/software that monitors data input and output. A set of given rules will block messages that do not meet the required criteria. A network without a firewall will be subject to a wide variety of attacks; leaving this important feature out of a network is unacceptable.

- **Modem:** Allows data to be transmitted over cables or telephone lines; this connects the Internet (or servers) to the switch or router. The term "modem" is short for modulator-demodulator.
- **Wireless Access Point:** A rather simple device that sends wireless signals. It acts as a kind of hub, without limiting data packets to anyone who has a password (assuming it is password-protected). A router is often used in conjunction with a WAP to provide better control.

You may also have to consider **printers, projectors, bridges**, and other devices. If you are not sure of the purpose of any of these, a quick online search should teach you what you need to know and can net some good example diagrams to review.

Lastly, consider which items may be used with wireless connections. Laptops, tablets, and phones are the most commonly used.

Your diagram will be more detailed than the example below, but this is provided so you understand the basic visual representation that is required. Again, do not copy this; you need your own structure, but reading diagrams helps with understanding the concepts.



Make sure you follow all instructions carefully as usual, and properly embed your network diagram in your Word document.

## Part 2: The Graphical User Interface

### Analyzing Websites

The web has a plethora of opportunities to evaluate the graphical user interface (GUI); the principles for design and usability are the same as for software applications. Part of this module's Assessment will focus on a bookseller's website for the reasons explained in the Assessment instructions, and the information below will help you understand what you need to know.

#### 1. Coding Error Detection

The World Wide Web Consortium (W3C) is an organization headed by Sir Tim Berners-Lee, the inventor of the web. The W3C provides recommendations for web coding; concerns include

ensuring that sites are accessible to people with disabilities and that coding is up to date. While some errors will not affect what a page looks like today, old code may not be readable on all modern browsers in the future, so maintaining current standards is important.

HTML and CSS are the primary languages used, as explained above. Web developers use *validators* to check code for errors. To find the HTML validator, look in the Reading page for this module. You have some options:

- Insert the URL of the page you wish to check. This will not search the entire site — just the page that resides at that particular address. (This is sufficient for the Assessment's home page error check.)
- Upload a file for review. This is only an option for the developer, as you will not have possession of the actual files.
- Insert source code itself for review. This is good for the developer, but you can also retrieve code yourself. Visit a page and use the keyboard shortcut Control+U (Command+U for Macs). The code will appear, typically in a new tab, and you can then copy/paste into the validator.

For the Assessment, it is not important to know how to fix the coding errors, but simply to see the number of errors present. You might compare to other pages to see how your chosen website fares.

## 2. Design Basics

There are many design parameters that a web developer considers. As a systems analyst, you need to know some of the basics, which are presented below.

### *Color and Branding*

- **Light/dark contrast:** Are the colors for text and background sufficiently different so that readability is high? The most readable combination is black text on white background (as you see here in this document). Look online for a "color contrast checker" to learn more.
- **Color dependencies:** Is the screen free from color dependency, where the user must depend on knowing a color in order to understand content or complete an action? (See Accessibility below.)
- **Branding:** Do the colors match a company logo or official image? How is this color combination used?
- **Consistency:** Are the colors used on the screen consistently? (In other words, are all subheadings of the same size presented with the same coloring? etc.)

### *Text and Readability*

- **Font family:** What are the main fonts used? There are several main font *families*:

Family	Description	Example
Serif	Letters have "feet" and the lines have variable width	Times New Roman
Sans-serif	Simple lines of consistent width	Arial
Monospace	All letters have the same width; used typically to show coding	Courier New
Script	Looks somewhat like handwriting	Comic Sans MS
Fancy	Does not fit any other category; wide variety possible. Sometimes called "Fantasy" fonts	<i>Blackadder JTC</i>

Sans-serif fonts provide better readability on a screen, especially if the font is rather small. Serif fonts are acceptable if large enough that the serifs (feet) do not become too pixelated. Monospace fonts are mostly used for displaying code. Script and fancy fonts should be used sparingly if at all, as readability can be poor.

- **Consistency:** Are font choices used consistently?
- **Size:** Is the size of all font presented on the screen big enough to be easily read?
- **Line length:** Is the line length of paragraphs short enough for quick readability?
- **Alignment:** Are paragraphs aligned left? Centered paragraphs are hard to read, but centering headings is okay.

### 3. Accessibility

Not all people have the same sensory capacity (especially hearing and sight); the responsibility of the interface designer is to make sure that content is accessible for people who must use a screen reader or other technologies. Accessibility also helps those who are unable to hear a video because they work in a loud environment, to view images even if their browser fails to render it, etc.

The W3C provides recommendations for accessibility through its Web Content Accessibility Guidelines which are linked at this module's main Reading page. Here are a few major accessibility issues to contemplate:

- Videos should have closed captioning or subtitles. If these are not available, a full transcript may suffice.
- Audio files should have written transcripts available.
- Images need "alt" (alternative) text when coded in a webpage. You can check this by reviewing the code as explained above. You might also be able to check this by right-clicking and "inspecting" or otherwise checking the image.
- Font must be large and simple enough for easy readability.
- Color contrast (as mentioned above) is also important for readability, and color dependency must be avoided.
- Navigation should rely on word labels, not just images.

You will discover more as you review accessibility ideas for the Assessment.

### 4. Jakob Nielsen's Usability Heuristics

Usability expert Jakob Nielsen developed "**10 Heuristics for User Interface Design**" in the early 1990s and updated his list in late 2020. The primary source for this is linked in the reading list on this module's main Reading page.

## XML

While HTML identifies the purpose of content on a webpage, and CSS provides style, XML is a kind of coding that is meant to simply "transport" data. XML is used for search engine optimization, a technique that helps a website rise towards the top of search results, and for other purposes.

Unlike with HTML and CSS, the person coding XML gets to make up tags! They must be *descriptive* of the content placed between beginning and end tags. The descriptive wording is placed in angle brackets.

The top line of an XML document contains the **prolog** (or prologue), which identifies the code that follows as XML. Note the location of all punctuation, spaces, and capitalizations:

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Tag Rules and Case

Tags *must* be presented in pairs; there are no single tags in XML. Note the placement of the forward slash for the end-tag in this pair:

```
<show>Game of Thrones</show>
```

The descriptive tags you write yourself cannot begin with "xml" or a number and cannot contain spaces.

There is one very special set of tags called the **root tag pair**. The beginning root tag is placed directly under the prolog. The end tag is placed at the very end of the XML.

Tags are case-sensitive. This means the beginning and end tag capitalizations *must* match. You can choose any of these case styles mentioned previously; consistency is crucial:

- All small letters      authorlastname
- Pascal                  AuthorLastName
- Camel                  authorLastName
- Underscore            author\_last\_name

Do not write all-caps or use periods, colons, or hyphens

Example of non-matching case — this XML will *not* work:

```
<brandName>Magnavox</brandname>
```

## Nesting

All tags must be **nested** properly. If you remember small children's nesting cups or Russian nesting dolls, this term will make sense: one pair must fit within another.

The following example shows the prolog properly followed by a root tag pair (*favorites*), and nested tags:

```
<?xml version="1.0" encoding="UTF-8"?>
<favorites>
  <tvshow>Game of Thrones</tvshow>
  <musicgroup>Beatles</musicgroup>
  <food>Pizza</food>
  <kidsbook>Charlotte's Web</kidsbook>
</favorites>
```

Further nesting can also be done: for example:

```
<musicgroup>
  <name>Beatles</name>
  <album>Sgt. Pepper</album>
</musicgroup>
```

## More Rules

Notice that while content inserted into tag pairs such as *Game of Thrones* is written in normal human language with spaces between words, XML tag names cannot have spaces in them.

You can, however, insert **attributes** into a tag. Attributes have values presented in double quotes. Note that the tag name itself does not have a space in it, but there is a space before the attribute *type*. Double quotes must be used for the value:

```
<food type="dinner">Pizza</food>
```

## Validation

XML is considered "well formed" when every rule is followed. This is crucial, as even one error will mean that the XML code will not execute; these rules are absolute. The good news is that there are several easily found online validators to help you check coding.

## Examples

In this first example, *favorites* is the root tag pair. All small letter case was used and is consistent:

```
<?xml version="1.0" encoding="UTF-8"?>
<favorites>
  <tvshow>Game of Thrones</tvshow>
  <musicgroup>
    <name>Beatles</name>
    <album>Sgt. Pepper</album>
  </musicgroup>
  <food>Pizza</food>
  <kidsbook>Charlotte's Web</kidsbook>
</favorites>
```

In this example, *Devices* is the root tag pair and Pascal case was chosen. You can see that tags can be reused (except for root tags) so that long lists can be created:

```
<?xml version="1.0" encoding="UTF-8"?>
<Devices>
  <CellPhone>
    <Brand>LG</Brand>
    <Model>MagnaPhone 5</Model>
    <ScreenSize>6.0 inches</ScreenSize>
    <Network>5g</Network>
    <Memory>6 GB RAM</Memory>
    <TalkTime>20 hours</TalkTime>
    <Camera>front facing</Camera>
    <OperatingSystem>Snapdragon</OperatingSystem>
  </CellPhone>
  <CellPhone>
    <Brand>Samsung</Brand>
    <Model>Perfecto 9.2</Model>
    <ScreenSize>6.5 inches</ScreenSize>
    <Network>5g</Network>
    <Memory>10 GB RAM</Memory>
    <TalkTime>17.5 hours</TalkTime>
```



```
<Camera>front facing</Camera>
<OperatingSystem>Nougat</OperatingSystem>
</CellPhone>
</Devices>
```

## Part 3: Additional Information

### Recording Your Voice

One part of this module's Assessment takes the form of an audio file. To do this, you have several **recording options**, including recording in PowerPoint and then saving as an mp3 or M4A file, going to an online recording site, creating a file on your phone and then transferring it to your computer, or using any number of free software options. Not that your final file must be attached to the initial post; do not attempt to link to a file in an online repository. If you need assistance with recording options, reach out to your professor for guidance.

Your first step should be to write a **transcript**. This helps avoid "ums" and long pauses; your presentation of ideas will run more smoothly. Writing a transcript also helps you confirm that your content is complete. A third reason for writing a transcript is that not everyone is able to hear. Some clients may be deaf or have some hearing loss. Other listeners may be trying to access your work in a loud environment or in a quiet place where listening to audio would not be appropriate. A transcript allows everyone access to the same information.

Find a **quiet environment** to record your audio file. Turn off the television, tell your family you need some quiet for a few minutes, choose a different part of the house to avoid your dogs' barking, or take whatever steps you need in order to avoid background noises.

**Practice** with the microphone. If it is too close to your mouth, plosives such as "p" may "pop" and be distracting. If the microphone is too far away, the listener might have to turn up the volume or the sound might be "fuzzy."

Practice with your **vocal technique** as well:

- Maintain consistent vocal volume.
- Avoid monotone (same-pitch) delivery, instead choosing important words and syllables for slight emphasis.
- Practice pacing so you do not talk too quickly or too slowly.
- Enunciate (speak clearly).
- Check to make sure any unusual words are pronounced correctly.
- Sit (or stand) with good posture and use a relaxed breath flow to help avoid vocal strain.

Do not be afraid to rerecord if you do not like your first attempt.

### Using PowerPoint

Another part of the Assessment requires PowerPoint. While the instructions tell you specific requirements, this short list should help as well, especially if you have not used PowerPoint recently:

- Use a design theme: look for options by visiting the Design ribbon.
- Keep it simple, without transitions or animation, and follow all instructions. The focus here will be content, so adding images other than what is included in the chosen design theme is not allowed.

- Follow the rules of good design with readable text and reasonable color choices.

Remember you can add slides by clicking in the left-side pane and pressing Control+M (or Command+M for Macs). You can also right-click to choose "New Slide." Follow instructions carefully as usual, and if you need help with the software, contact the Tech Tutors.

## Quizzes

You are encouraged to take the non-graded quizzes. In this module, the first one will solidify your understanding of networking basics, and the second one covers XML coding. The questions are all multiple choice (with just one potential answer) and true-false.

## Getting Help

To find the Academic Success Center, look for My Studies > Academic Success Center from your home page (where your courses are listed, not inside this classroom). There you will find a plethora of information for writing, math, science, business, and technology. You can also connect with tutors. This is a free service for Purdue Global students; if you have not investigated it prior to this term, it is a good idea to check it out and see what great help is available.

\* \* \* \* \*

## If You Are New to IT

This section is presented for those who are new to the field or just wish to solidify understanding of computer concepts relevant to the module or the course. It is a good idea for seasoned professionals to scan this information, too, in case there is something new to learn.

## Supplemental Reading: Networks

The ebooks listed below are in the course's Library list as "optional." Click on More Tools > Library to look for these items.

Meyers, M., Jernigan, A., & Lachance, D. (2019). *CompTIA IT fundamentals+ all-in-one exam guide (exam fc0-u61)* (2nd ed.). McGraw Hill.  
<https://libauth.purdueglobal.edu/sso/skillport?context=144895>

- *Chapter 10: "Configuring Network and Internet Connectivity"*

Meyers, M. (2019). *CompTIA A+ certification all-in-one exam guide (exams 220-1101 & 220-1102)* (10th ed.). McGraw Hill. <https://libauth.purdueglobal.edu/sso/skillport?context=144455>

- *Chapter 18: "Essentials of Networking"*
- *Chapter 19: "Local Area Networking"*
- *Chapter 20: "Wireless Networking"*

## Area Networks

In addition to the topology (the physical setup), there are many *kinds* of computer networks, each of which serves a purpose and may be restricted in scope. These include:

- **PAN:** Personal Area Network
  - A small network where devices are within 100 meters of each other. Bluetooth and other technologies can be used.
- **LAN:** Local Area Network
  - A network set up in a small geographic location such as a company building (or two). These are controlled by individual businesses, schools, etc.
- **WLAN:** Wireless Local Area Network
  - As you can expect, a LAN that works principally on wireless connections for the customers/users.
- **MAN:** Metropolitan Area Network
  - The connection of several LANs into one larger network which is accessible to an entire campus or a city. Routers are needed to connect the LANs together.
- **WAN:** Wide Area Network
  - A network that covers an even larger geographic region (even potentially an entire country) and is made up of many LANs. The Internet itself could be identified as a WAN.

## The Cloud

The cloud is a global collection of servers networked so that a wide variety of services can be offered. This can include the following as well as other services that are all accessible thanks to the Internet. To learn more about these services, do an online search or check the features offered by these providers:

Acronym	Meaning	Example(s)
IaaS	Infrastructure as a Service	Amazon Web Services (AWS)
PaaS	Platform as a Service	Windows Azure, Salesforce Lightning Apps
SaaS	Software as a Service	Office 365, Adobe's Creative Cloud software
DBaaS	Database as a Service	Microsoft Azure SQL Database
BaaS	Backup as a Service	Acronis, Carbonite
SECaaS	Security as a Service	Oracle Cloud Access Security Broker (CASB)
STaaS	Storage as a Service	Dropbox, Google Drive
DRaaS	Disaster Recovery as a Service	Carbonite Cloud Backup

In addition, there are a few different kinds of implementation:

- **Public:** A provider offers services to anyone who wishes to purchase them.
- **Private:** Cloud services are owned by a company or small amalgamation of companies who wish to share the costs.
- **Hybrid:** As you imagine, this is a combination, where organizations can more flexibly use public and private cloud implementations.

## More Hardware

- **Gateway:** As the name suggests: it is the entry point of a network. It serves as a bridge between networks and can be either software or hardware.
- **Repeater:** Takes a weak signal and repeats it until it is strong enough for data transmission.
- **Bridge:** Connects LAN segments; it works somewhat like a router.

## Supplemental Reading: Web Development

Wood, K. (2018). *Confident web design: Master the fundamentals of website creation and supercharge your career*. Kogan Page. <https://libauth.purdueglobal.edu/sso/skillport?context=134874>

- Chapter 2: "HTML Part 1"
- Chapter 4: "CSS Part 1"

## HTML: HyperText Markup Language

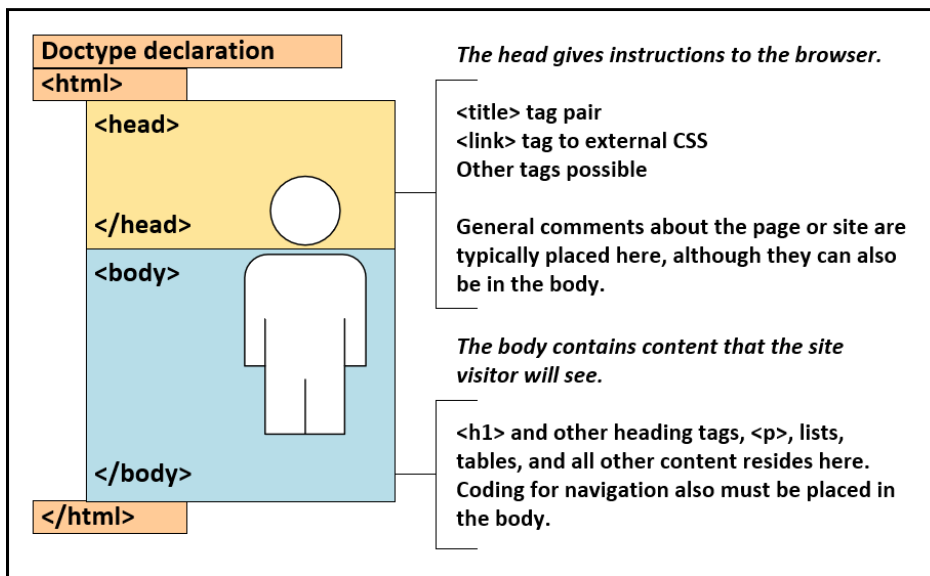
This is the language used for displaying content on webpages; it is also used in other applications for the same purpose. The word "markup" indicates that it is not really doing anything other than identifying the content.

Tags typically come in pairs, with a few exceptions. Tags are instructions placed in angle brackets. The end-tag has a forward slash right after the beginning bracket. Here is an example:

```
<h1>Page Heading</h1>
```

This looks a lot like XML, but the tags are pre-determined (the developer cannot compose their own).

Below are the main tags for building an HTML webpage. The Doctype declaration tells the browser that this is to be rendered as an HTML page. Note that there cannot be any coding or content placed between the end-head and beginning-body tags.



## CSS: Cascading Style Sheets

This language is used to make webpages and other screens look nice; color, font style, borders and other effects, and even whole pages are organized using CSS. It is called "cascading" because there are three ways in which you can apply it:

- **External:** separate document loaded to the server
  - A link is typed into the head of an HTML document to connect to the external CSS.
  - This is preferred for multiple-page websites, since one document can control the design of all pages therein.
- **Internal:** inserted in the head of an HTML document
  - This only affects the document into which it is placed.

- **Inline:** inserted into a beginning HTML tag in the body
  - This only affects the content stored within that tag pair.
  - It is not recommended to use much inline CSS, as it is more difficult to locate for design updates.

CSS is constructed differently than HTML. As you learned above, fonts come in families. In any font declaration, you should keep all possibilities in the same family. For example:

```
font-family: Georgia, "Times New Roman", serif;
```

This tells the browser to look for Georgia on the user's computer, and if it is not present, then to find Times New Roman. If the computer has neither font loaded, then the default serif font will show on the page. Multiple-word font names are placed in quotes as you see above.

Colors are even more fun! Hexcode is used, in which numbers 0–9 and letters A–F combine to provide 16 million options. Use six of these characters in CSS color declarations. You can learn more in the reading and by searching on the web for a "color picker" to learn more about RGB colors.

## About Errors

XML is unforgiving; one error will make the entire file not work. In CSS, a missing semicolon, fancy bracket, or extra character can make instructions that follow the error not work, but it would not be a complete failure. HTML is more forgiving, as some errors will not even be noticed by the browser. Since today's ignored HTML errors might produce bad results as browsers evolve, finding mistakes and correcting them is an important task for the web developer. Learning the syntax (rules) for these languages and being careful helps the IT professional pay attention to detail, which is a good skill for any workplace.

The term "**deprecated**" (*deh*-preh-kay-ted) means that a coding practice is considered old and no longer recommended. Examples of deprecated code include typing tags in capital letters or using the <center> tag pair (there are newer and better ways to center text using CSS). Deprecated code make work today, but again, as browsers evolve, might result in problems in the future.

## Programming: Not Really the Same as "Coding"

HTML and CSS are considered *coding* languages, as they do not attempt to provide solutions to problems; that task is for *programming* languages. Various sources will claim that programming is a more general term for all of these; the terminology can be a little confusing.

Some popular programming languages include Python™, Java™, C, C++, C# ("C sharp"), Ruby on Rails, PHP, SQL, MySQL™, and JavaScript. Each has different rules, but there are some basic principles that most of the languages use. Programming is complicated, so if you are a beginner, it is good to start with some definitions:

- **Variables** are values that may change as a program executes. They must be declared and then are assigned to operations in which they are needed.
- **Data types** classify information.
- **Arrays** are like tables of information.
- **Boolean** data produces a binary yes/no response.
- **A string** is a set of characters.

**Common structures:**

- Loop
- While
- If – then
- If – else
- Arrays
- Strings
- Functions

**Common mathematical symbols:**

- Adding +
- Subtracting -
- Multiplying \*
- Dividing /