1

एच.टी.एम.एल. का परिचय
Introduction to HTML

1.1 परिचय (Introduction) :

अधिकांशतः व्यक्ति और संगठन एच.टी.एम.एल. (HTML) में वेब साईट बनाते हैं। यहां पर वेबसाइट बनाने के बहुत तरीके हैं, लेकिन वेब साईट बनाने के लिये एच.टी.एम.एल. (HTML) बहुत ज्यादा उपयोग में आता है। हम विश्वास करते हैं कि एच.टी.एम.एल. (HTML) वेब पेज बनाने का सबसे अच्छा तरीका है, जब आप विभिन्न कम्प्यूटर जो सारे विश्व में फैले हैं के द्वारा एक बड़े टेक्स्ट को आपस में जोड़ते हैं तब आप मकड़ी के जाले जैसा लिंक प्राप्त करते हैं, जिसे वर्ड वाइड वेब (World Wide Web) कहते हैं, इसमें इन्टरनेट से जुड़े विभिन्न कम्प्यूटर आपस में जुड़े रहते हैं। ये सभी वेब पेज एक विशेष फॉरमेट में लिखे जाते हैं। यह फॉरमेट या भाषा एच. टी.एम.एल. (H.T.M.L. – Hyper Text Markup Language) कहलाती है। दस्तावेज (Document) के तार्किक संगठन की व्याख्या करने के लिए HTML को विशेष हाईपर टेक्स्ट विस्तार के साथ डिजाइन किया जाता है। यह वर्ड प्रोसेसर (Word Processor) जैसी भाषा नहीं बनी है, जैसे वर्ड या वर्ड परफेक्ट। सभी हम एच.टी.एम.एल. (HTML) का उपयोग ज्यादा करते हैं, क्योंकि विभिन्न क्षमता के विभिन्न ब्राउजर्स (Browsers) में समान एच.टी.एम.एल. (HTML) दस्तावेज को देखा जा सकता है। उदाहरण के लिए एच.टी.एम.एल. (HTML) आपको टेक्स्ट जैसे टाईटल और अनुच्छेद को अलग से चिन्हित करता है और तब यह चिन्हित तथ्यों को ब्राउजर के द्वारा व्याख्या करने के लिए छोड़ देता है। उदाहरण के लिए एक ब्राउजर नए परिच्छेद को बनाता है जबकि दूसरा केवल एक खाली लाइन छोड़ता है।

HTML निर्देश दस्तावेज के टेक्स्ट को खण्डों में विभक्त करता है इन्हें तत्व कहते हैं। यह बड़े रूप से दो भागों में बांटे जा सकते हैं, जो यह बताते हैं कि ब्राउजर के द्वारा प्रधान अंश (बॉडी) के दस्तावेजों को किस तरह दिखाया जा सकता है। जैसे टाईटल या अन्य दस्तावेजों में संबंध। अगले अनुच्छेदों में HTML दस्तावेजों का सम्पूर्ण विवरण दिया जायेगा।

HTML (टेग/तत्व के नाम, उनका कैसे उपयोग होगा) दूसरी भाषा स्टैण्डर्ड जनरलाइज्ड मार्कअप भाषा (SGML-Standard Generelized Markup Language) का उपयोग करते हुए

बनाये गए हैं। SGML बहुत कठिन है तथा यह बहुत ज्यादा दस्तावेजों को एकत्रित करने के लिए नेम अक्षरों की श्रेणी है जिसको बिन्दु के द्वारा पृथक किया जाता है। उदाहरण के लिए : www.yahoo.com" और "www.railways.org.gov" IP address नम्बर की जगह इन नामों का उपयोग करना सरल, व्यवस्थित तथा अनुकूल है।

1.2 यूनिवर्सल रिसोर्स आइडेन्टीफायर (Universal Resources Identifiers) का परिचय :

1.2.1 वर्ल्ड वाइड वेब (World Wide Web - WWW) :

वर्ल्ड वाइड वेब सूचनाओं के स्त्रोत का नेटवर्क है। वेब तीन तरीके से संभावित विस्तृत उपयोगकर्ता के लिए स्त्रोत उपलब्ध कराते हैं।

1. वेब पर स्त्रोत ढूंढ़ने के लिए एक समान योजना (eg. URIs)
2. वेब पर स्त्रोत के नाम द्वारा एक्सेस करने के लिए प्रोटोकॉल (eg. HTTP)
3. स्त्रोतों के बीच में सरल आपसी संबंध के लिए हाईपरटेक्स्ट

ये तीनों तरीके आपस में एक साथ उपयोग में आते हैं।

1.2.2 URIs:

HTML दस्तावेज, इमेज, विडियो क्लिप, प्रोग्राम आदि प्रत्येक स्त्रोत वेब पर उपलब्ध होते हैं, यह युनिवर्सल रिसोर्स आइटेन्टीफायर (Universal Resource Identifier) या **URI** के द्वारा इनके एड्रेस को खोजा जा सकता है। यूनिफॉर्म रिसोर्स लोकेटर (Uniform Resource Locator - URL) URI का एक रूप है।

URI के तीन भाग है :

1. स्त्रोत एक्सेस करने के लिए एक नाम योजना का उपयोग करते हैं।
2. स्त्रोत का स्वयं का नाम, पथ के नाम के साथ।
3. स्त्रोत के मशीन होस्ट का नाम

उदाहरण :

http://www.microsoft.com/download/ieupdate.html

यह URI निम्न प्रकार से पढ़ा जा सकता है :

यहां पर एक दस्तावेज http प्रोटोकॉल के द्वारा www.microsoft.com मशीन पर / download पथ के द्वारा उपलब्ध होता है। HTML दस्तावेजों को लिंक करने के लिए अन्य स्कीम जो है वह "mailto" ईमेल के लिए और FTP (File Transfer Protocol) फाइल ट्रांसफर करने के लिए सम्मिलित कर सकते हैं।

यहां पर URI का अन्य उदाहरण है। यह उपयोगकर्ता मेल–बॉक्स के लिए है।

For all comments, please send email to

<A href = "mailto:anilgupta09@yahoo.co.in> anil gupta </A>

नोट : ज्यादातर पाठक URL टर्म को जानते हैं लेकिन URI टर्म को नहीं। URL रूप सामान्यतया URI नाम योजना का उपसमूह है। URI इन्टरनेट पर सामान्य एक एड्रेस है। जब आप URL को एन्टर करते हैं तब आप ब्राउजर को यह बताते हैं कि जो पेज आप ढूंढ़ना चाहते हैं उसको प्राप्त करने के लिए वास्तव में कहां जाना है। जैसा कि पत्र पर लिखा हुआ पता डाकिये को यह

बताता है कि इस पत्र को कहां जाना चाहिये। अन्य विभिन्नताएं भी संभव हो सकती हैं और एड्रेस कई स्तर नीचे तक जा सकता है, लेकिन इस उदाहरण में URL का मूल स्वरूप दर्शाया गया है।

1.2.3 फ्रेगमेन्ट आईडेन्टीफायर (Fragment Identifier) :

कुछ URI स्त्रोत के अन्दर लोकेशन को इंगित करते हैं। फ्रेगमेनटेड आई डी (ID) URI के सम्पूर्ण ऑब्जेक्ट का अनुसरण करते हैं, जो कि हैश साईन (#) से पृथक होते हैं तथा एक एंकर आईडेन्टीफायर का अनुसरण करते हैं जिसको **फ्रेगमेंट आईडेन्टीफायर** कहते हैं। यदि फ्रेगमेन्ट आईडी शून्य (रिक्त) है तो हैश साईन के बाद लिखे शब्द को छोड़ देते हैं। एक रिक्त फ्रेगमेन्ट आईडी हैश साईन के साथ तथा बिना साथ होने का मतलब है कि URI सम्पूर्ण आब्जेक्ट को इंगित करता है।

वहां पर किसी तरह की दिक्कत नहीं होती है जहां कि फ्रेगमेन्ट ID कुछ भी इंगित करे जो ऑब्जेक्ट में से निकाला गया है। यह इसका अपना अधिकार है। यह हो सकता है उदाहरण के लिए ऑब्जेक्ट के अन्दर अपने आप को किसी भी बिन्दु पर इंगित करे।

उदाहरण : यहां पर URI Section_2 एंकर नाम को इंगित करता है।

http//www.somesite.com/html/top.html#section_2

top.html के अन्दर Section_2 दिखाई देगा जब ब्राउजर के अन्दर उपरोक्त एड्रेस निष्पादित होगा।

1.2.4 संबंधित URIs (Relative URIs) :

संबंधित URI कोई भी नाम योजना की सूचना नहीं रखता है, इसका पथ सामान्यतया उसी मशीन पर वर्तमान दस्तावेज के स्त्रोतों को इंगित करता है। संबंधित URI सम्बंधित पथ को रखता है। उदाहरण : ".." का मतलब पथ में लिखे हुए हायआरची में एक स्तर ऊपर जो कि पथ द्वारा परिभाषित रहता है।

संबंधित URI मूल URI का उपयोग करते हुए सम्पूर्ण URI बनते हैं। संबंधित URI का पृथक्करण करने के लिए उदाहरण है, आप मानते हैं मूल URI "http://www.methi.com/support/intro.html है निम्न संबंधित URI हाइपरलिंक के लिये मार्कअप है।

<A href = "suppliers.html">suppliers</A>

इसका विस्तृत URI http://www.methi.com/support/suppliers.html होगा। जब कि इमेज के लिए मार्कअप रिलेटिव URI निम्न है।

<IMG src "../icon/logo.gif"alt="logo>

इसका विस्तृत URI http://www.methi.com/icons/logo.gip होगा।

HTML में URI निम्न उपयोग आते हैं।

- स्त्रोतों और दस्तावेजों को अन्य से जोड़ना।
- स्क्रिप्ट तथा बाह्य स्टाईल शीट को जोडना।
- इमेज चित्र बनाना।
- फॉर्म को सबमिट (Submit) करना।

1.3 HTML का इतिहास (History of HTML) :

अधिकांशतः लोग इस बात से सहमत थे कि HTML दस्तावेजों को विभिन्न ब्राउजर्स और प्लेटफॉर्म के लिए कम लागत पर अच्छा कार्य करना चाहिये। HTML के प्रत्येक वर्जन के

लिए उद्यम खिलाड़ियों के मध्य बहुत सहमति हुआ करती थी ताकि विषय सामग्री बनाने के लिए लागत बर्बाद नहीं हो तथा उनके दस्तावेज कम समय में अपठनीय नहीं हो जायें।

HTML इस विचार के साथ विकसित हो चुका था कि वेब पर सभी प्रकार की डिवाईसेज सूचना को उपयोग करने में सक्षम है। PC के साथ कई प्रकार का ग्राफिक्स रिजोल्यूशन तथा रंग की गहराई, सेल्यूलर फोन, हाथ में पकड़ने वाले डिवाईस, आवाज के लिए, इनपुट और आउटपुट के लिए यंत्र, कम और ज्यादा बेन्ड विड्थ (Bandwidth) और इसी प्रकार के कई यंत्र।

1969 में चारले गोल्डफर्ब IBM रिसर्च प्रोजेक्ट इन्टीग्रेटेड लॉ ऑफिस इनफोरमेशन सिस्टम (Integrated law office information system) पर कार्य कर रहे थे और इनके साथ एडवर्ड मोसर एण्ड रेमण्ड लारी थे जिन्होंने जनरलाइज्ड मार्कअप लैंग्वेज (Generalized Markup Language -GML) की खोज की थी। GML टेक्स्ट एडिटिंग, फोरमेटिंग तथा सूचना प्राप्त करने के लिए सबसिस्टम के दस्तावेजों को बाँटता है। स्टैण्डर्ड का पहला वर्किंग ड्राफ्ट (working draft) 1980 में प्रकाशित हुआ था।

यह सत्य है कि 1990 की शुरूआत में वेब की खोज हुई थी। उन दिनों के दौरान कम्युनिकेशन तकनीक में विकास हो रहा था। इसका मतलब यह था कि जल्दी और देर से वेब पर कुछ घटित होना था। शुरूआत में हाईपरटेक्स्ट प्रचार में आया था और कम्प्युटर पर उपयोग में आने लगा था। इन्टरनेट पर भी उपयोगकर्ता की संख्या में तेजी से विस्तृत हो रहा था। डिस्ट्रीब्युटेड सूचना के लिए भी उपयोगकर्ता बढ़ रहे थे। इन्टरनेट पर मशीन का एड्रेस डोमेन नाम स्कीम के द्वारा बहुत ही आसानी से दिया जा सकता है।

SGML के उपयोग :

SGML की उपयोगिता छोटे संगठन या छोटे संघ के उपयोगकर्ता के लिए बहुत तेजी से विकसित हुई। शुरूआती दो एप्लीकेशन, बहुत बड़ी हिस्सेदारी के साथ विकसित हुई थी वह है एसोसिएशन ऑफ अमेरिका पब्लिशर (AAP) के लिए इलेक्ट्रोनिक्स मेन्यूस्क्रिप्ट तथा डॉकूमेन्टेशन कम्पोज्ड करने के लिए यू.एस. डिपार्टमेन्ट ऑफ डिफेन्स के द्वारा शुरू कम्प्युटर एडेड एग्वीजिशन एण्ड लॉजिस्टिक सपोर्ट (CALS)।

HTML जिसको टिम ने विकसित की थी वह मजबूती से SGML पर आधारित थी। वे विश्व स्तर पर इस बात पर सहमत हुये थे कि टेक्स्ट को एक संरचना इकाई बनाया जाये। जैसे – अनुच्छेद, शीर्षक, सूची अवयव और कई इकाईयां। HTML किसी भी मशीन पर इम्प्लीमेंट कर सकते हैं। यह विचार था कि यह भाषा फोरमेटर्स (ब्राउजर या अन्य कोई देखने वाला सॉफ्टवेयर) से मुक्त हों जो कि वास्तव में टेक्स्ट स्क्रीन पर दिखता है। इसमें टेग के युग्म का उपयोग करते हैं। जैसे <TITLE> और </TITLE> यह सीधे ही SGML से लिया गया है जो कि बिल्कुल समान है। SGML के तत्व जो टिम की HTML में उपयोग होते हैं वह हैं <P> (Paragraph) <H1> से <H6> (शीर्ष स्तर 1 से शीर्ष स्तर 6 तक) <OL> (ऑरडर्ड लिस्ट) <UL> (अनऑडरर्ड लिस्ट) <LI> (लिस्ट अवयव) और अन्य बहुत।

HTML 1.0 और 2.0 :

1992 में बरनर्स ली और सेर्न दल ने HTML 1.0 को रिलिज किया जिसको 1993 में अंतिम रूप दिया गया। इसका विशेष विवरण बहुत सरल था तथा यह कागज के पन्ने पर एक तरफ प्रिन्ट हो सकता लेकिन यह मूल विचार रखता था तथा HTML की तुरन्त खोज में सहायक

था। जो कि तार्किक संरचनाओं तथा प्रदर्शित तत्वों के मध्य पृथक्कीकरण था। यह एक मुख्य अकेला विचार था जो HTML को सीखने में सहायक था।

HTML 2.0 (नवम्बर 1995) 1994 के अन्त में सामान्य अभ्यास के लिए इन्टरनेट इंजीनियरिंग टास्क फोर्स (IETF) की सहायता से विकसित हुआ था।

ब्राउजर और HTML :

नेटस्केप एक उपलब्ध ब्राउजरों में से था। मोसेइक (MOSAIC) NCSA द्वारा प्रस्वावित था। LYNIX (लाइनेक्स) यूनिक्स मशीन पर उपलब्ध था और कुछ अन्य कम्पनियां ब्राउजर को बना रही थीं। उनमें से एक स्पाइग्लास (SPYGLASS) था जिसको माइक्रोसॉफ्ट ने खरीद लिया था और यह इन्टरनेट एक्स्प्लोरर के लिए आधार बना। प्रत्येक ब्राउजर में रेन्डरिंग इंजन (rendering engine) होता है जो कि कोड होता है जो आपके HTML को बताता है कि उसको कैसे लेना है और इसको बदलकर आपको स्क्रीन पर दिखाता है।

HTML 3.0 और HTML 3.2 :

1993 में HTML+ तथा 1995 में HTML 3.0 बहुत धनी HTML वर्जन प्रस्तावित हुए थे। W3C (World Wide Web Consortium) जो कि HTML के विकास के ऊपर हुई थी उसमें HTML 3.0 पर कुछ मानकों को बनाने का प्रयास हुआ था। लेकिन इसमें क्या सम्मिलित करें इसके लिए बहुत वार्तालाप हुआ था। W3C की कोशिश पर HTML कार्यदल ने 1996 में सामान्य अभ्यास के लिए HTML 3.2 को परिणामस्वरुप बनाया। इसे जनवरी 1997 में अंतिम रूप दिया गया। इसमें मुख्य लक्षणों को जोड़ा गया वह है टेबल और इमेज के चारों तरफ टेक्स्ट का चलना।

HTML 4.0X :

W3C ने 1997 के अन्त में HTML 4.0 की व्याख्या प्रस्वावित की तथा उसके आगे 1999 में HTML 4.01 प्रस्तावित की जिसमें 4.0 वर्जन व्याख्या में हुई कुछ गलतियों को सही किया गया था। इस वर्जन में कुछ विचित्र गलतियों को सही किया गया था। जो 3.2 वर्जन में हो हो गई थी। विशेषकर केसकेडिंग स्टाइल शीट (Cascading Style Sheets) इसमें सम्मिलित की।

XML 1.0 :

यह HTML का अग्रसर है। यहाँ "X" का मतलब एक्सटेनसिबल (eXtensible) है। XML(eXtensible Markup Language) HTML 4.01 का सुधरा हुआ रूप है। यह वेब पेज बनाने के लिए वर्तमान में बहुत मानक है। इसमें कुछ रूचिकर बदलाव कोडिंग मे सम्मिलित किये गए हैं। उदाहरण के लिए : वास्तव मे सभी टेग को अभी तक बन्द करना पड़ता है, जिसमें पैराग्राफ टेग भी शामिल है। अन्य टेग जैसे FONT टेग सभी प्रजेन्टेशन तत्वों को नियंत्रित करने के लिए केसकेडिंग स्टाईल शीट के पक्ष में हटा दिया गया है।

1.4 DHTML :

गतिज शब्द का अर्थ गति या चाल और बदलाव से है। HTML की एक बड़ी कमी उसका स्थिर प्रकृति है। DHTML (Dynamic HTML) तत्वों की सही स्थिति तथा स्तरीकरण

का उपयोग केसकेडिंग (CSS) स्टाईल के साथ नए प्रभाव बनाने के लिए किया जाता है।

1.5 वेब ब्राउजर (Web Browser) :

HTML में बने वेब पेज को दिखने के लिए वेब ब्राउजर की आवश्यकता होती है। बहुत ज़्यादा प्रचलित ब्राउजर इन्टरनेट एक्सपोलर तथा नेटस्केप नेविगेटर है। ब्राउजर्स HTML के सभी टेक्स्ट को एक समान नहीं दिखाता है। जब वेबपेज बनाते हैं तो मुख्य रूप से इसे ध्यान में रखना होता है।

यहाँ वेब वाइड वर्ल्ड पर बहुत सारे वेबपेज उपलब्ध होते है। यह सम्भव होता है कि वेब पर अन्य लोगों के सोर्स कोड देखकर मालूम कर सकते हैं कि उन्होंने एक विशेष लक्षण को अपने वेब पेज पर कैसे लागू किया है। आप स्रोत को निम्न प्रकार देख सकते हैं :

- मैन्यू में से व्यू को चुनो तथा इसे स्रोत पर भेजों।
- वेब पेज पर राइट क्लिक करके व्यूसोर्स को चुनना।

जब उपरोक्त विकल्प का उपयोग HTML पेज के साथ करते है तो नोटपेड या अन्य सामान पैकेज में दिखाता है। आप HTML कोड को आसानी से देख सकते हैं तथा यह प्रकाशनाधिकार (Copyright) का उल्लंघन नहीं है। यह सम्भव है कि आप अन्य पेज बनाने वाले का विचार ले सकते हैं पर उसमें अपना स्वयं का टेक्स्ट होना चाहिए। आपको वेबपेज बनाने वाले को एक ईमेल भेजना चाहिए यदि आप उसका विचार या पिक्चर को उसके वेबपेज से उधार लेते हैं।

(Properties) को तत्व के एट्रिब्युट Attribute नियंत्रित करते हैं। उदाहरण के लिए, पेज के अप्रधान स्थान (Background) रंग के लिए रंग का निर्धारण करना।

1.6 ब्राउजर तथा खुली विन्डोज के मध्य स्विचिंग (Switching between opened windows and browser) :

HTML कोड लिखने के लिए नोटपेड का उपयोग करते हैं और यह कोड ब्राउजर का उपयोग कर खोला जा सकता है। यहां पर दो तरीकों से खुली विन्डोज तथा ब्राऊजर के मध्य आपस में बदल सकते (Switch) हैं। यह तरीके है :

1. Tab + Alt कुंजी का उपयोग कर विन्डोज को आपस में बदलता हैं।
2. टास्क बार के ऊपर बने विन्डोज के आईकॉन (Icon) के ऊपर माऊस द्वारा क्लिक कर विन्डोज को खोला जा सकता है।

महत्वपूर्ण बिन्दु

1. HTML का पूर्ण रूप हाईपर टेक्स्ट मार्कअप लेंग्वेज (Hyper text Markup Language) होता है।
2. HTML के निर्देश द्वारा टेक्स्ट को खण्ड में विभाजित करते हैं। उन्हें तत्व कहते हैं।

3. यूनिवर्सल रिसोर्स आईडेन्टिफायर (URI) इन्टरनेट पर एड्रेस होता है।
4. URI दो प्रकार के होते हैं – (अ) फ्रेगमेन्टेड URI (ब) रिलेटिव URI
5. HTML CERN (सर्न) में विकसित हुई थी।
6. HTML का प्रथम वर्जन 1992 में रिलिज हुआ था।
7. XML, HTML का अग्रसर है।

अभ्यासार्थ प्रश्न :

बहुचयनात्मक प्रश्न :

1. HTML के प्रोग्राम निष्पादित होते हैं –
   (अ) ब्राउजर मे (ब) DOS प्रोमट पर
   (स) UNIX प्रोमट पर (द) विन्डोज मे
2. इन्टरनेट एड्रेस मिश्रण होता है :
   (अ) URI (ब) HTTP
   (स) HTML पेज (द) उपरोक्त सभी
3. SGML का प्रथम कार्यरूप प्रकाशित हुआ था –
   (अ) 1981 (ब) 1980
   (स) 1979 (द) 1978
4. HTML विकसित हुई थी –
   (अ) ANSI (ब) CERN
   (स) IRS (द) यू.एस. डिपार्टमेन्ट ऑफ डिफेन्स
5. HTML 1.0 किसके द्वारा विकसित हई थी –
   (अ) बरनसली (ब) एन्डर्स
   (स) चार्ली (द) शार्प
6. HTML का पहला वर्जन विकसित हुआ था –
   (अ) 1990 (ब) 1991
   (स) 1992 (द) 1993
7. HTML का क्या मतलब होता है –
   (अ) हाईपर टेक्स्ट मार्क लेंग्वेज (ब) हाईपर टेस्ट मार्क लेंग्वेज
   (स) हाईपर टेक्स्ट मार्कअप लेंग्वेज (द) उपरोक्त में से कोई नहीं

अतिलघुत्तरात्मक प्रश्न :

1. SGML का पूर्ण रूप क्या है ?
2. XML का पूर्ण रूप क्या है ?
3. HTML की कौन सी भाषा अग्रणी है ?

4. रिलेटिव URI क्या है ?

लघुत्तरात्मक प्रश्न :

1. डोमेन नाम क्या है ?
2. WWW क्या है ?
3. फ्रेगमेन्टेड आईडेन्टीफायर क्या है ?
4. HTML कैसे विकसित हुई थी ?
5. SGML के प्रारम्भिक प्रोजेक्ट क्या थे ?
6. DHTML क्या है ?

निबन्धात्मक प्रश्न :

1. URI क्या है ? उदाहरण के साथ समझाइये।
2. HTML के 4.0 तथा 4.01 वर्जन में क्या विशेषताओं को सम्मिलित किया गया है ?

उत्तरमाला

1 (अ) 2 (द) 3 (ब) 4 (ब) 5 (अ) 6 (स) 7 (स)

2
मूल वेबसाइट का बनाना
Designing the Basic Website

2.1 परिचय (Introduction):

इस अध्याय में अनेक प्राथमिक विचारणीय विषय पर पढ़ेंगे जो वेबसाइट बनाने में सहायक होते हैं। इस अध्याय के अंदर आप विभिन्न प्रकार के टेगों के बारे में पढ़ोगे जो HTML में उपयोग होते हैं। आप वेबसाइट कैसे बनाओगे ? आप वेब पेज बनाने के लिए क्या क्रम (step) होता है के बारे में भी सीखोगे।

2.2 HTML टेग के प्रकार (Types of HTML Tag):

HTML में टेग होते हैं। और प्रत्येक टेग का अपना एट्रिब्यूट और मान होता है। टेग ब्राउजर को सूचना देते हैं कि उनको वेब पेज पर आवश्यकतानुसार दिखाया जाये। उदाहरण : एक इमेज या लिंक दिखाना। प्रत्येक टेग में कोण के अंदर कीवर्ड या अक्षर होता है। उदाहरण: <P> अनुच्छेद के लिए जबकि <B> : टेक्स्ट को बोल्ड करने के लिए। ज्यादातर टेग संभावित एक से ज्यादा एट्रिब्यूट रखते हैं। आप एटिब्यूट में कोई मान स्टोर कर सकते हैं।
उदाहरण :

<BODY Text = "red">

यह <BODY> टेग है, इसका एट्रिब्यूट Text है, और एट्रिब्यूट का मान "red" है। जब आप टेग के साथ एट्रिब्यूट का उपयोग करते हैं तो सभी स्टार्ट टेग के एट्रिब्यूट दो को ठक (< और >) के मध्य लिखे जाते हैं।

टेग दो प्रकार के होते हैं :

- कण्टेनर टेगस् (Container Tags)
- एम्प्टी टेगस् (Empty Tags)

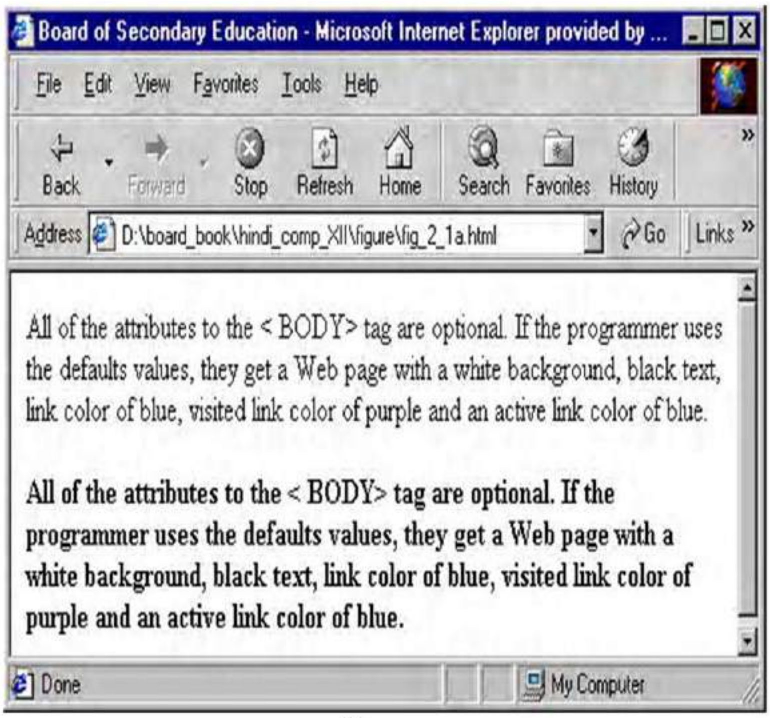
2.2.1 कण्टेनर टेगस् (Container Tags)

कण्टेनर टेग टेक्स्ट के खण्ड पर लगते हैं। कण्टेनर टेग को एक ऑपनिंग टेग तथा एक क्लोजिंग टेग की आवश्यकता होती है जो खण्ड के दोनों तरफ लगते हैं। क्लोजिंग टेग में भी वही कीवर्ड या अक्षर होता है जो कि ऑपनिंग टेग में होता है लेकिन क्लोजिंग टेग में सबसे पहले स्लेश (/) लगाते हैं।

उदाहरण :

<P> और </P>
<B> और </B>

उपरोक्त उदाहरण में, अनुच्छेद टेग (Paragraph tag) <P> तथा बोल्ड फेस टेग (Bold Face tag) <B> का उपयोग किया है। टेग के मध्य बंद सम्पूर्ण टेक्स्ट पर यह लगेगा। अगर हम इसको ब्राउजर में देखगें तो चित्र 2.1 की तरह दिखाई देगा।



चित्र 2.1

2.2.2 एम्प्टी टेग (Empty Tag)

एम्प्टी टेग अकेले टेग होते हैं जो टेक्स्ट पर प्रभाव नहीं डालते हैं। एम्प्टी टेग में ऑपनिंग टेग होता है लेकिन क्लोजिंग टेग की आवश्यकता नहीं होती है। क्षितिज रूल वेब पेज पर क्षितिज लाइन बनाता है।

उदाहरण :

<HR>

2.2.3 एट्रिब्यूटस् (Attributes)

टेग का व्यवहार एट्रिब्यूट का उपयोग कर बदला जा सकता है। एक टेग के कई एट्रिब्यूट हो सकते हैं। एट्रिब्यूट के दिये गये मान डिफॉल्ट (Default) मान को या एट्रिब्यूट के मान की सीमा द्वारा बदला जाता है। मान को डबल कोटेशन मे बंद करना चाहिए।

एट्रिब्यूट का मान टेग की परिभा ा का भाग है तथा ब्राउजर द्वारा इसका मान लिया जायेगा अन्यथा अन्य मान भण्डारित हो जाता है। ज्यादातर एट्रिब्यूट को स्वीकार योग्य सीमा से निश्चित मान या कई मानों में से एक मान दिया जाता है ।

उदाहरण :

<tag_Name attribute = "Value" attribute = "Value>

<HR Size = "1" height = "10" Width = "70%">

उपरोक्त उदाहरण में, क्षितिज टेग को इसके एट्रिब्यूट के साथ लिखा हुआ है।

2.3 वेबपेज बनाना (Creating a Web page)

निम्नलिखित क्रम को वेब पेज बनाने में उपयोग करते हैं इस उदाहरण में डिजाइन करने वाला प्रारम्भिक पेज बनाता है जिसमें वह अपना नाम तथा एड्रस लिखता है।

1. नोटपेड में एक नया दस्तावेज खोलो।
2. टाइटल के साथ शीर्ष खण्ड बनाओ।

   <HTML>

   <HEAD>

   <TITLE> Place the title here </TITLE>

   </HEAD>
3. <Body> टेग और टेक्स्ट को कुछ एट्रिब्यूट के साथ एन्टर करो। उदाहरण : पेज के लिए बेकग्राउण्ड कलर देना।
4. नाम और पता एन्टर करो।
5. <Body> टेग और <HTML> टेग को बंद करो।
6. दस्तावेज को .htm या .html एक्सटेन्शन के साथ (save) सेव करो।
   - फाइल नेम बॉक्स में फाइल का नाम एन्टर करो।
   - फाइल का नाम देते समय यह ध्यान रखना है कि इसका एक्सटेन्शन .htm या .html होगा। इस उदाहरण में firstwebpage.html फाइल नेम बॉक्स में एन्टर करेंगे।
   - save बटन पर क्लिक करो।
7. ब्राउजर खोलो तथा दस्तावेज को देखो।

- विन्डो एक्सप्लोरर को खोलो ।
- फोल्डर को ढूंढो जहाँ वेब पेज रखा है।
- फोल्डर के अंदर उचित आइकॉन पर डबल क्लिक करो ।
- वेब ब्राउजर में पेज खुल जाएगा।

  या
- डेस्कटोप के ऊपर बने आइकॉन पर क्लिक करके वेब ब्राउजर खोलो या स्टार्टमेन्यू में ऑपशन (option) चुनो।
- ऑपन ऑपशन पर माउस लाओ।
- ऑपन बॉक्स के अंदर ब्राउस बटन को चुनो।
- "Look In" बॉक्स में फोल्डर को ढूंढो।
- ऑपन बॉक्स पर डबल क्लिक करो।

8. ब्राउजर में पेज ऑपन करने के लिए "OK" बटन पर क्लिक करो।
   - मैन मेन्यु से फाइल को चुनो।
   - Save As पर माउस लाओ।
   - ड्राप डाउन बॉक्स के द्वारा जगह को चुनो।

2.4 HTML के मूल टेग (Basic Tag of HTML)

वेबपेज पर दो मुख्य खण्ड होते हैं :

- शीर्ष खण्ड (Head Section)
- प्रधान खण्ड (body Section)

2.4.1 शीर्ष खण्ड (Head Section):

जब एक उपयोगकर्ता वेब पेज पर जाता है तब ब्राउजर को लिंक दस्तावेज के बारे में सूचना की आवश्यकता होती है कि वह कैसे दिखायी देने चाहिए यह सूचना वह HTML टेग से प्राप्त करता है।

HTML टेग :

यह एक कन्टेनर टेग है और दस्तावेज के अंत में क्लोजिंग (Closing) टेग की आवश्यकता होती है। यह टेग ब्राउजर को बताता है कि यह फाइल HTML की है।

उदाहरण :

```
<HTML>
</HTML>
```

शीर्ष टेग (Head Tag) :

शीर्ष खण्ड दस्तावेज के बारे में सूचना रखता है उदाहरण के लिए – एक एड्रेस, अन्य पेजों के साथ संबंध, ब्राउजर की प्रायिकता। शीर्ष टेग भी कन्टेनर टेग है टाइटल टेग इसके बाद आता है।

उदाहरण :

<HEAD>
</HEAD>

टाइटल टेग (TITLE TAG) :

यह टेग ब्राउजर के टाइटल बार में टाइटल को प्रकट करने की अनुमति देता है। यह सिफारिश करता है कि टाइटल में साधारण टेक्स्ट होना चाहिए कोई इमेज या लिंक नहीं।

<TITLE> Board of Secondary Education </TITLE>

यह सूचना ब्राउजर के टाइटल बार में प्रकट हो जाएगी वेबपेज के शीर्ष खण्ड में कम से कम निम्न तत्व होने चाहिए –

<HTML>
<HEAD>
<TITLE> Place the title here </TITLE>
</HEAD>
</HTML>

2.4.2 प्रधान खण्ड (Body Section):

बॉडी टेग (Body Tag) :

<BODY> टेग HTML दस्तावेज का हृदय होता है। बॉडी वह है जहाँ वास्तविक मेटीरियल वेबपेज पर प्रदर्शित होने के लिए बनता है। <BODY> टेग <HEAD> टेग के बिल्कुल बाद में आता है ब्राउजर सभी वि ायों को दिखाता है जो कि <BODY> टेग के ओपनिंग और क्लोजिंग के मध्य आते हैं।

उदाहरण :

<Body>
....
....
</Body>

<Body> टेग कई एट्रिब्यूट्स रखता है। एट्रिब्यूट टेग के प्रभाव को बदल देता है। बॉडी टेग के एट्रिब्यूट है : टेक्स्ट, बेकग्राउण्ड, बेकग्राउण्ड इमेज, बेकग्राउण्ड साउण्ड तथा लिंक टेक्स्ट।

टेक्स्ट (Text) :

टेक्स्ट एट्रिब्यूट मुख्य टेक्स्ट के रंग को बदल देता है।

उदाहरण :

<Body text = "red">
या
<Body text = "FF0000">

उपरोक्त उदाहरण में टेक्स्ट का फोरग्राउण्ड रंग बदलकर लाल हो जाएगा।

HTML के अंदर लाल, हरा और नीला रंग का मिश्रण होता है यह मिश्रण एक हेक्साडेसिमल नम्बर बनाता है। कुछ प्रचलित रंग को उनके नाम द्वारा उपयोग किया जा सकता

है उदाहरण के लिए, सफ़ेद, पीला, मेहरून, सलेटी, नीला, लाल, बैंगनी, काला, हरा और सिल्वर के लिए उनके नाम तथा हैक्साडेसिमल कोड दोनों का उपयोग कर सकते हैं।

हैक्स कोड में छः नम्बर होते हैं – 2 अंक लाल रंग के लिए, 2 अंक हरे रंग के लिए और अंतिम 2 अंक नीले रंग के लिए होते हैं। सभी कलर कोड को सारणी 2.1 में दिखाया गया है।

बीजीकलर (bgcolor) :

बीजीकलर एट्रिब्यूट वेब पेज बनाने वाले को बेकग्राउण्ड का रंग बदलने की अनुमति देता है। बेकग्राउण्ड रंग बदलने के लिए निम्न स्टेटमेन्ट लिखते हैं।

उदाहरण :

<Body bgcolor = "blue">

या

<Body bgcolor = "0000FF">

bgcolor एट्रिब्यूट बेकग्राउण्ड का रंग निर्देशित कोड के अनुसार बदल देता है। टेक्स्ट एट्रिब्यूट भी सम्पूर्ण पेज के टेक्स्ट रंग को बदल देता है ना कि केवल पेज के एक विशिष्ट खण्ड को।

सारणी 2.1 रंग कोड

RGB Color	Hexadecimal Value
White	#FFFFFF
Black	#000000
Red	#FF0000
Green	#00FF00
Blue	#0000FF
Magenta	#FF00FF
Cyan	#00FFFF
Yellow	#FFFF00
Aquamarine	#70DB93
Violet	#9F5F9F
Brass	#B5A642
Copper	#B87333
Pink	#FF6EC7
Orange	#FF7F00

बेकग्राउण्ड (Background) :

बेकग्राउण्ड एट्रिब्यूट वेब पेज डिजाइन करने वाले को बेकग्राउण्ड इमेज बदलने की सुविधा देता है।

उदाहरण :

```
<Body background = "images/sunback.gif">
<Body background = "images/tiger.jpg">
```

उपरोक्त GIF और JPG फाइल वेब पेज पर बेकग्राउण्ड इमेज हो जाएगी। यह फाइल जो कि एट्रिब्यूट के मान में लिखा है करंट (Current) डायरेक्ट्री में उपलब्ध होनी चाहिए।

लिंक (Link) :

लिंक एट्रिब्यूट लिंक, विजिटेड लिंक, और एक्टिव लिंक (लिंक पर जब माउस को ले जाते हैं) के रंग बदलने के लिए होता है। निम्न उदाहरण में लिंक कलर लाल, विजिटेड लिंक कलर पीला और एक्टिव लिंक कलर हरा हो जाएगा।

उदाहरण :

```
<Body link = "red" vlink = "yellow" alink = "green">
```

या

```
<Body link = "FF0000" vlink = "FFFF00" alink = "008000">
```

<Body> टेग के सभी एट्रिब्यूट एच्छिक होते हैं यदि प्रोग्रामर डिफाल्ट मानों का उपयोग करता है तो वह वेबपेज का बेकग्राउण्ड रंग सफ़ेद, टेक्स्ट रंग काला, लिंक रंग नीला, विजिटेड लिंक रंग बैंगनी तथा एक्टिव लिंक रंग नीला प्राप्त करते हैं।

उदाहरण : HTML दस्तावेज का निम्नलिखित उदाहरण हैं :

```
<HTML>
<HEAD>
<TITLE> Board of Secondary Education</TITLE>
</HEAD>
<BODY text="yellow" bgcolor="blue"
background= "images/ BOSE.gif ">
This Web site is generated for the Board of Secondary Education.
</BODY>
</HTML>
```

चित्र 2.2 में टाइटल विण्डो के शीर्ष में, बेकग्राउण्ड रंग नीला तथा फोरग्राउण्ड रंग पीला दिखाता है।

2.5 HTML में कमेन्ट (Comment in HTML):

चित्र 2.2

HTML दस्तावेज में व्याख्या (Commenting) करने के लिए कमेन्ट, टेग का उपयोग किया जाता है। यह HTML के कार्य में कोई प्रभाव नहीं डालता है, और टेग और विषय सूची (content) भी ब्राउजर में नहीं दिखाता है।

<!Yashika & Sangeeta-created the Website>

2.6 टेक्स्ट की फॉरमेटिंग (Text Formating) :

2.6.1 बेसफोन्ट टेग (BASEFONT Tag) :

यदि आप डिफाल्ट फोन्ट साइज 3 के अलावा वेब पेज में ज्यादातर टेक्स्ट के लिए अन्य फोन्ट साइज का उपयोग करना चाहते हैं तो आप उस पेज पर विशिष्ट फोन्ट के लिए फोन्ट साइज का निर्धारण बेसफोन्ट से कर सकते हैं। आप इसके लिए <BASEFONT> टेग का उपयोग करते हैं। <BASEFONT> टेग के लिए एण्ड टेग नहीं होता है।

उदाहरण :

<Base Font Size = "4">

उपरोक्त उदाहरण में <Base Font> साईज बढ़कर 4 हो जायेगी।

उदाहरण : <BASE FONT> टेग के लिए HTML दस्तावेज लिखिए।

<HTML>

```
<HEAD>
<TITLE> Board of Secondary Education</TITLE>
</HEAD>
<BODY>
<P align="center"> The application programs are written by the
programmer to solve any real world problem. <BR><BR>
<BASEFONT SIZE="15">These programs are written in ei-
ther Higher Level Languages or Using 4GLs. The application pro-
gram may be altered, modified as per the change in the require-
ment by the programmer. </P>
</BODY>
</HTML>
in either Higher Level Languages or Using 4GLs. The application
program may be altered, modified as per the change in the re-
quirement by the programmer. </P>
</BODY>
</HTML>
```

उपरोक्त उदाहरण में <BASEFONT> टेग के बाद टेक्स्ट की साइज 15 हो जाएगी। यदि हम साइज को नहीं बदलेंगे तो सम्पूर्ण वेबपेज पर यही साइज रहेगी। HTML दस्तावेज का आउटपुट चित्र 2.3 में दिखाया गया है।

चित्र 2.3

2.6.2 फोन्ट टेग (Font Tag):

<FONT> टेग का उपयोग टेक्स्ट की शैली लागू करने में किया जाता है। फोन्ट टेग के बहुत सारे एट्रिब्यूट होते हैं। जैसे – साइज, फेस और रंग।

साइज (Size):

साइज एट्रिब्यूट का उपयोग टेक्स्ट की साइज बदलने में किया जाता है। HTML में टेक्स्ट 7 साइज के होते हैं साइज को रिलेटिव और ऑब्सोल्यूट मानो का उपयोग कर बदल सकते

हैं। ऑब्सोल्यूट मान के लिए साइज का मान 1 से 7 तक एन्टर किया जाता है। टेक्स्ट साइज बिना आसपास के टेक्स्ट को ध्यान में रखते हुए उसके नए मान के अनुसार बदल जाएगी।

उदाहरण :

<Font size = "6"> MBM Engineering College </FONT>

टेक्स्ट की साइज करंट (Current) साइज से कम या ज्यादा रिलेटिव तरीके से बदल सकते हैं। टेक्स्ट की डिफॉल्ट साइज तीन हैं। टेक्स्ट की साइज़ 1 से 7 के मध्य ही रहती है। यदि साइज का मान सीमा से बढ़ जाता है तो यह तब भी टेक्स्ट साइज को 1 से 7 के मध्य रखता है।

उदाहरण :

<Font Size = "+3"> MBM Engineering College </FONT>

<Font Size = "-2"> MBM Engineering College </FONT>

उपरोक्त उदाहरण में, क्रमानुसार टेक्स्ट साइज तीन से बढ़ जाएगी (साइज 6 हो जाएगी क्योंकि डिफाल्ट साइज तीन है।) और 2 से घट जाएगी (साइज 1 हो जाएगी)।

कलर (Color) :

<BODY> टेग में दिये हुए रंग को हटाते हुए color एट्रिब्यूट टेक्स्ट रंग के लिए स्थापित करने की स्वीक ति देता है। रंग का मान या तो हेक्सा डेसिमल कोड में या फिर पूर्व निर्धारित नाम से दिया जाता है।

उदाहरण :

<Font color = "red"> MBM Engineering College </Font>

या

<Font Color = "#FF0000"> MBM Engineering College </Font>

फेस (Face) :

फेस एट्रिब्यूट फोन्ट की शैली (Style) बदलने में उपयोग आता है। फेस एट्रिब्यूट का मान उस टाइपफेस का नाम होता है। यह संभव है कि फॉन्ट की सारणी को कॉमा से अलग–अलग कर एन्टर किया जाए, क्योंकि सारे उपयोगकर्ता सारे फोन्ट को इन्स्टॉल नहीं करते हैं। यदि पहली च्वॉइस उपलब्ध नहीं है तो यह सूची में से अगले फोन्ट का उपयोग करता है। यदि सूची में लिखे हुए फोन्ट में से कोई भी उपलब्ध नहीं है तो यह डिफॉल्ट फेस टाइप का उपयोग करता है। यहां पर कुछ सामान्य फोन्ट के नाम हैं सेरिफ, सान्स सेरिफ कर्सिव, फेनटेसी, और मोनोस्पेस्ड। ब्राउजर भी अपने स्वयं के डिफॉल्ट टाइपफेस रखता है।

उदाहरण :

<Font Face = "times New Roman"> MBM Engineering College </Font>

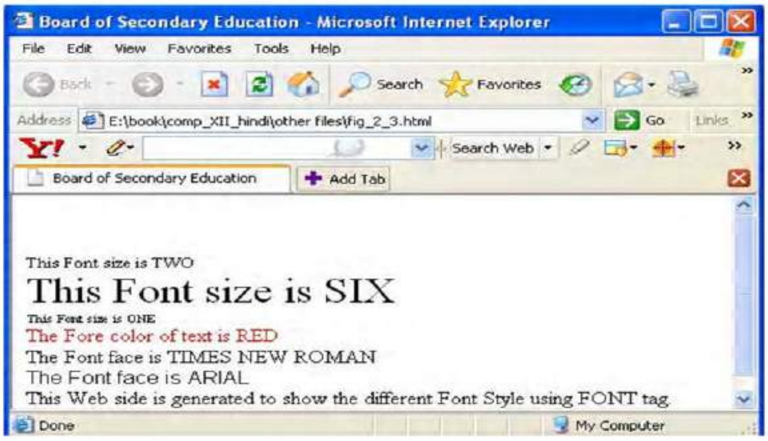
<Font Face = "Arial"> MBM Engineering College </Font>

नोट : <Font> टेग में यह संदेह बारम्बार रहता है कि वह गलत तरीके से नेस्टेड हो जाए। जैसे ही ब्राउजर फोन्ट के अंतिम टेग </Font> पर आता है यह अंतिम टेग को बिना किसी एट्रिब्यूट को बदले बन्द करेगा। यदि अंतिम टेग में एक से ज़्यादा एट्रिब्यूट हैं तो यह पुराने मानो को बंद

कर देगा। नेस्टेड टेग के नियम यह है कि जो पहले टेग खुला है वह अंत में बंद होगा।

उदाहरण : विभिन्न फोन्ट साइज तथा फोन्ट फेस के लिए HTML कोड नीचे दिया है।

```
<HTML>
<HEAD>
<TITLE> Board of Secondary Education</TITLE>
</HEAD>
<BODY>
<BR><BR>
<FONT size="2"> This Font size is TWO </FONT>
<BR>
<FONT size="+3"> This Font size is SIX </FONT>
<BR>
<FONT size="-2"> This Font size is ONE </FONT>
<BR>
<FONT color="red"> The Fore color of text is RED </FONT>
<BR>
<FONT face="Times New Roman">The Font face is TIMES NEW ROMAN
</FONT>
<BR>
<FONT face="Arial">The Font face is ARIAL</FONT><BR>
This Web side is generated to show the different Font Style using FONT tag.
</BODY>
</HTML>
```



चित्र 2.4

उपरोक्त HTML कोड विभिन्न फोन्ट साइज कैरेक्टर ब्राउजर के अंदर बनाता है और यह फोन्ट की रिलेटिव गणना करता है यह डिफॉल्ट फोन्ट साइज तीन लेगा उपरोक्त कोड का आउटपुट चित्र 2.4 में दिखाया गया है।

2.6.3 अनुच्छेद टेग (Paragraph tag):

अनुच्छेद टेग <P> नया अनुच्छेद बनाने की स्वीकृति देता है ब्राउजर की विण्डो साइज के अनुसार लाइन के अंत में, अपने आप लाइन का समायोजन कर लेता है जिसमें यह प्रदर्शित होता है। <P> टेग एक एलाइनमेन्ट एट्रिब्यूट रखता है एलाइन एट्रिब्यूट के तीन मान दाँया, बाँया और मध्य होते हैं लाइन एट्रिब्यूट का डिफॉल्ट मान बाँया है।

उदाहरण :

<P>

The application programs are written by the programmer to solve any real world problem. These programs are written in either Higher Level Languages or Using 4GLs. The application program may be altered, modified as per the change in the requirement by the programmer.

</P>

<P align="right">

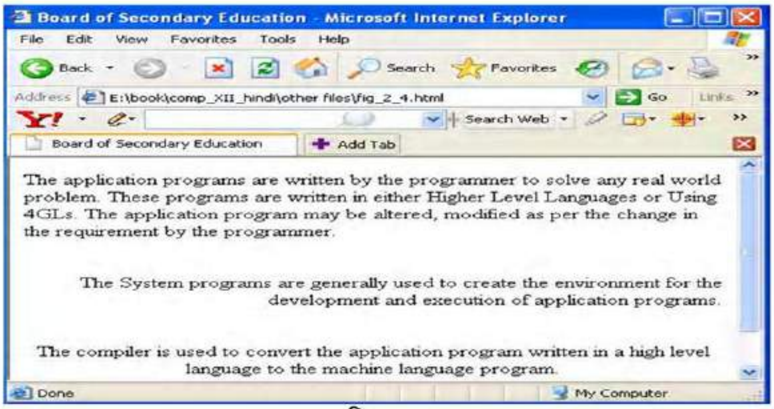
The System programs are generally used to create the environment for the development and execution of application programs.

</P>

<P align="center">

The compiler is used to convert the application program written in a high level language to the machine language program.

</P>



चित्र 2.5

उपरोक्त उदाहरण में सभी तीनों अनुच्छेद नई लाइन से शुरू होंगे। वे विण्डो साइज में अपने आप समायोजित होगे। यदि विण्डो की साइज बदलती है तो द्वितीय अनुच्छेद दाँया एलाइन है, तृतीय अनुच्छेद मध्य एलाइन है, तथा प्रथम अनुच्छेद डिफाल्ट पहले से निर्धारित डिफाल्ट मान रखेगा जो कि बाँया एलाइन है। उपरोक्त उदाहरण का वेब ब्राउजर में आउटपुट चित्र 2.5 में दिखाया गया है।

2.6.4 लाइन ब्रेक टेग (Line Break Tag - BR):

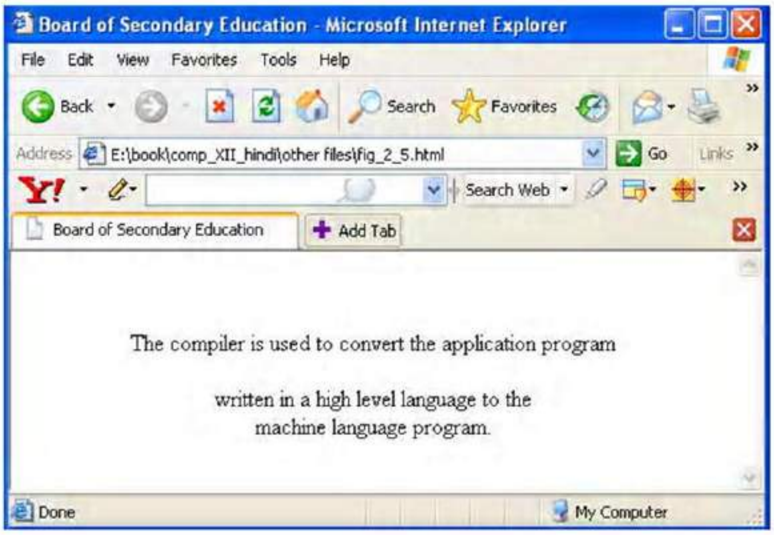
लाइन ब्रेक टेग द्वारा टेक्स्ट को किसी लाइन पर ब्रेक कर देता है इसके बाद आने वाला टेक्स्ट नई लाइन से शुरू होगा। उदाहरण के लिए किसी का पता लिखते समय एक ही अनुच्छेद में अलग–अलग लाइनों में लिखने की आवश्यकता होती है आप अनिवार्य लाइन ब्रेक सम्मिलित कर सकते हैं।

उदाहरण :

<P align="center"> The compiler is used to convert the application program **

**written in a high level language to the **
**machine language program.</P>

उपरोक्त उदाहरण में, अनुच्छेद तीन लाईन में प्रिंट होगा। जब भी
 टेग आयेगा तब टेक्स्ट नई लाईन से शुरू होगा। यदि दो
 टेग एक साथ आते हैं तो एक खाली लाईन बन जायेगी। यह चित्र 2.6 में दिखाया गया है।



चित्र 2.6

 एक खाली तत्व (Tag) है इसका बंद होने वाला टेग नहीं होता है अर्थात् </BR> की आवश्यकता नहीं होती है
 तत्व एक अवयव CLEAR रखता है जिसका मान Left, Right या All हो सकता है। यदि लाइन ब्रेक के साथ CLEAR के साथ LEFT उपयोग में लिया है तो नई लाइन में करसर बांये हाशिये से शुरू होगा। इसी प्रकार Right दाये और ALL बॉये तथा दाँये दोनों और हाशिया छोड़ते हुए नई लाइन शुरू करेगा।

2.7 खण्ड शीर्षक टेग (Section Heading Tag):

<Body> टेग के अंदर, दस्तावेज का विभाजन करने के लिए सामान्यतया शीर्षक टेग H1 से H6 शीर्षक टेग का उपयोग किया जाता है। शीर्षक टेग एच्छिक होते हैं। शीर्षक टेग को किसी भी क्रम में प्रकट होने की अनुमति है। शीर्षक टेग में सबसे ऊपर के स्तर के लिए H1 शीर्षक टेग का उपयोग होना चाहिए। H2 अगला अधिकतम,शीर्षक टेग है तथा H6 सबसे निम्न स्तर का शीर्षक टेग है। शीर्षक टेग का एक एलाइनमेन्ट एट्रिब्यूट होता है। जिसके संभावित मान बायां, दायां और मध्य होते हैं।

उदाहरण :

```
<HTML>
<HEAD>
<TITLE> Board of Secondary Education</TITLE>
</HEAD>
<BODY>
<H1 align="center"> H e a d i n g 1 </H1>
<H2 align="center"> H e a d i n g 2 </H2>
<H3 align="center"> H e a d i n g 3 </H3>
<H4 align="center"> H e a d i n g 4 </H4>
<H5 align="center"> H e a d i n g 5 </H5>
<H6 align="center"> H e a d i n g6 </H6>
</BODY>
</HTML>
```

2.8 टेक्स्ट स्टाइल टेग(Text Style Tag) :

2.8.1 बिग टेग (Big Tag) :

टेक्स्ट साइज को निश्चित सीमा में बदलने के लिए अन्य तकनीक उपलब्ध है। विशेष रूप से फोन्ट साइज या रिलेटिव फोन्ट साइज उपयोग करने के अलावा आप <BIG> टेग से टेक्स्ट की साइज 1 बढ़ा सकते हैं।

उदाहरण :

```
<BIG>
```

</BIG>

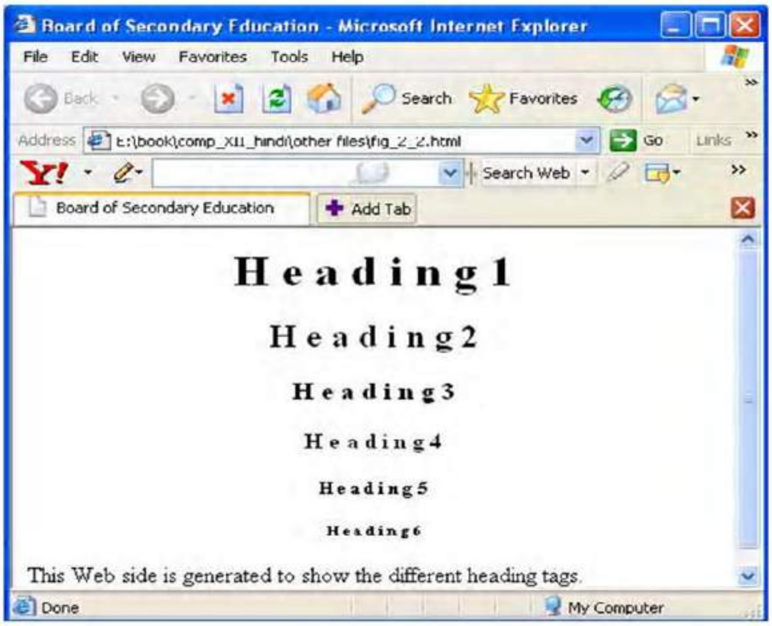
टेग के मध्य लिखे हुए टेक्स्ट की साइज 1 बढ़ जाएगी। यदि आपकी साइट पर डिफॉल्ट टेक्स्ट की साइज 3 है और आप <BIG> टेग एक शब्द पर लगाते हैं तो उस शब्द की साइज 4 हो जाएगी।

2.8.2 स्माल टेग (Small Tag) :

यदि आप कुछ टेक्स्ट की साइज घटाना चाहते हैं तो यह कार्य टेक्स्ट की साइज बढ़ाने जैसा ही सरल है क्योंकि आप स्माल टेग का इस्तेमाल कर सकते हैं।
उदाहरण :

<SMALL>
</SMALL>

उदाहरण : स्माल और बिग टेग को निम्न HTML दस्तावेज में दिखाया गया है।



चित्र 2.7 : शीर्षक टेग

<HTML>
<HEAD>
<TITLE> Board of Secondary Education</TITLE>

```
</HEAD>
<BODY>
<P align="center">
The application programs are written by the programmer to solve any real world problem. <BR><BIG>These programs are written in either Higher Level Languages or Using 4GLs.</BIG>
<BR><SMALL> The application program may be altered, modified as per the change in the requirement by the programmer.
</SMALL>
</P>
</BODY>
</HTML>
```

इस उदाहरण में, दोनों <BIG> और <SMALL> टेग अपना प्रभाव टेक्स्ट पर दिखायेगें। यह चित्र 2.8 में दिखाया गया है।

2.8.3 इटालिक टेक्स्ट (Italic Text <I>) टेग :

यदि आप एक शब्द या लाइन को अपने वेब पेज पर इटालिक करना चाहते हैं तो आप <I> टेग का उपयोग करते हैं।

उदाहरण :

<I>

</I>

<EM> टेग के मध्य लिखा हुआ टेक्स्ट इटालिक (Italic) हो जाएगा। आप समान प्रभाव <I> और </I> टेग से भी प्राप्त कर सकते हैं।

2.8.4 इम्फेसाइजड टेक्स्ट (Emphasized Text <EM>) टेग :

यदि आप एक शब्द या लाइन को सामान्य टेक्स्ट की तुलना में अधिक इम्फेसिस देना चाहते हैं तो आप <EM> टेग का उपयोग करते हैं।

उदाहरण :

<EM>

</EM>

2.8.5 बोल्ड टेक्स्ट (<B>) टेग :

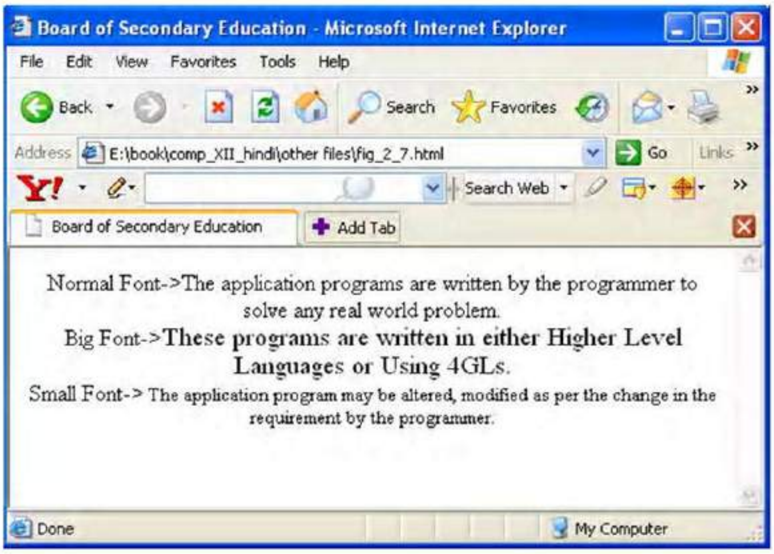
यदि आप एक शब्द या लाइन को अपने वेबपेज पर बोल्ड करना चाहते हैं तो आप <B> टेग का उपयोग करते हैं।

उदाहरण :

<B>

</B>

<B> टेग के मध्य लिखा हुआ टेक्स्ट बोल्ट (BOLD) हो जाएगा।



चित्र 2.8

2.8.6 स्टींग टेक्स्ट (<STORNG>) टेग :

यदि आप एक शब्द या लाइन को **इम्फेसाइजड टेक्स्ट** की तुलना में अधिक इम्फेसीस देना चाहते हैं तो आप <STORNG> टेग का उपयोग करते हैं।

उदाहरण :

<STORNG>

</STORNG>

2.8.7 अण्डरलाइन टेक्सट (Underline Text - <U>) टेग :

यदि आप अपने वेब साइट पर टेक्स्ट को अण्डरलाइन करना चाहते हैं तो आप <U> टेग का उपयोग करते हैं।

उदाहरण :

<U>

</U>

<U> और </U> के मध्य लिखा हुआ टेक्स्ट अण्डरलाइन हो जाएगा।

2.8.8 स्ट्राइक थ्रू टेक्स्ट (<STRIKE>) टेग :

यदि आप वेबपेज पर स्ट्राइक थ्रू टेक्स्ट बनाना चाहते हैं तो <STRIKE> टेग का

उपयोग करते हैं। यह एक क्षैतिज लाइन रखता है और निम्न टेक्स्ट की तरह दिखता है।
उदाहरण :

```
<STRIKE>
</STRIKE>
```

<STRIKE> और </STRIKE> के मध्य लिखा हुआ टेक्स्ट स्ट्राइक थ्रू टेक्स्ट की तरह हो जाएगा।

2.8.9 सुपरस्क्रिप्ट और सबस्क्रिप्ट (<SUP> और <SUB>) टेग :

आप वेब पेज में सुपरस्क्रिप्ट टेक्स्ट (<SUP> टेग से) और सबक्रिप्ट टेक्स्ट (<SUB> टेग से) भी लिख सकते हैं। सुपरक्रिप्ट टेग जयादातर गणितीय समीकरणों में उपयोग आता है। अन्य टेक्स्ट फोरमेटिंग टेगों की तरह <SUP> और <SUB> टेगों को भी एन्ड **(END)** टेग की आवश्यकता होती है।

उदाहरण :

```
X<SUP>4</SUP> +X<SUP>3</SUP> + X<SUP>2</SUP>+X+5=0
C<SUB>2</SUB>H<SUB>4</SUB>O<SUB>3</SUB></P>
```

इसका आउटपुट निम्न होगा।

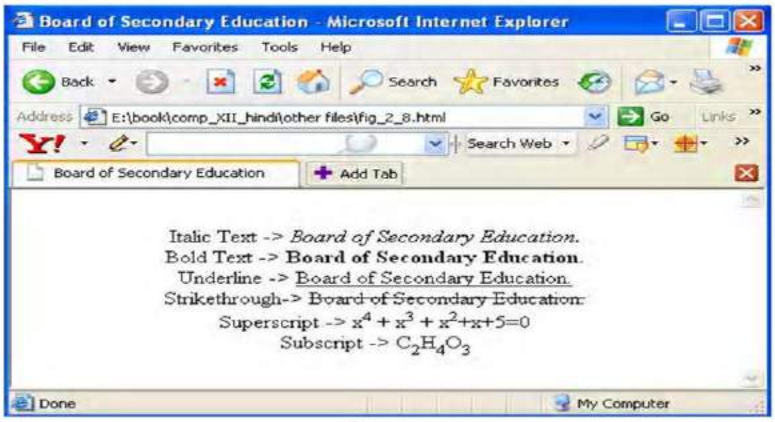
$X^4+X^3+X^2+X+5 = 0$

$C_2H_4O_3$

उदाहरण : यह उदाहरण सभी फॉरमेटिंग टेग को दिखायेगा।

```
<HTML>
<HEAD>
<TITLE> Board of Secondary Education</TITLE>
</HEAD>
<BODY>
<P align="center">
<BR>Italic Text  ->  <EM>Board of Secondary Education.</EM>
<BR>Bold Text    ->  <STRONG>Board of Secondary Education.
                     </STRONG>
<BR>Underline    ->  <U>Board of Secondary Education.</U>
<BR>Strikethrough->  <STRIKE>Board of Secondary Education.
                     </STRIKE>
<BR>Superscript  ->  x<SUP>4</SUP> + x<SUP>3</SUP> +
                     x<SUP>2</SUP>+x+5=0
<BR>Subscript    ->  C<SUB>2</SUB>H<SUB>4</
SUB>O<SUB>3                          </SUB></P>
</BODY>
</HTML>
```

चित्र 2.9 में सभी फोरमेंटिग टेक्स्ट अपने प्रकार के साथ दिखाये गये हैं।



चित्र 2.9

2.8.10 ब्लॉक क्वॉट (Blockquote) टेग :

<Blockquote> टेग का उपयोग लम्बी कोटेशन के लिए होता है। यह खण्ड स्तर का तत्व है। यह टेक्स्ट के खण्ड करने के उपयोग में भी आता है। कुछ ब्राउजरों में ब्लॉकक्वॉट टेक्स्ट इण्डेन्ट (Indent) होता है जबकि कई ब्राउजरों में यह इटेलिक होता है। यह संभव है कि <Blockquote> टेग के अंदर अन्य टेग रख सकते हैं।

उदाहरण :

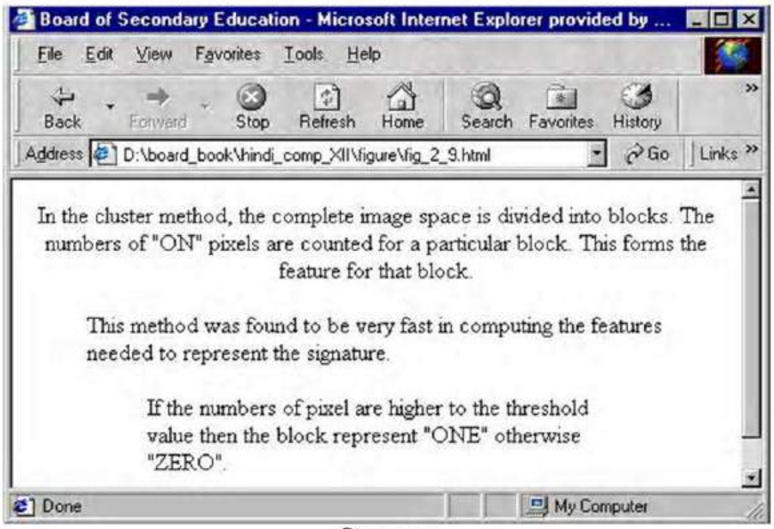
In the cluster method, the complete image space is divided into blocks. The numbers of "ON" pixels are counted for a particular block. This forms the feature for that block. **<BLOCKQUOTE>** This method was found to be very fast in computing the features needed to represent the sig nature. **<BLOCKQUOTE>** If the numbers of pixel are higher to the threshold value then the block represent "ONE" otherwise "ZERO".

<Blockquote> टेग का प्रभाव चित्र 2.10 में दिखाया गया है।

2.8.11 पहले से फोरमेंटेड टेक्स्ट (Previously Formatted Text):

साधारणतया ब्राउजर टेक्स्ट में से अतिरिक्त स्पेस हटा देता है। यदि बनाने वाला कई स्पेस तथा कैरिज रिटर्न (Enter Key) देता है तो ब्राउजर उन सब को एक स्पेस की तरह मानता है। <PRE> tag मोनोस्पेस फोन्टफेस का उपयोग करता है तथा यह सभी टेक्स्ट का खाका (Layout) बनाये रखता है।

उदाहरण :



चित्र 2.10

<PRE> An array must declare before it use in the program as same as the other variables, three thing are declared in an array declaration:

(i) Type of an Array

(ii) Name of an array

(iii) Capacity of an array (Number of Elements) **</PRE>**

उदाहरण : **<PRE>** टेग को निम्न HTML दस्तावेज में दिखाया गया है।

```
<HTML>
<HEAD>
<TITLE> Board of Secondary Education</TITLE>
</HEAD>
<BODY>
<P align="center">
<BR>With PRE TAG<BR>
<PRE> An array must declare before it use in the program as same
<BR>as the other variables, three thing are declared in <BR>an array declaration:
        (i)    Type of an Array
        (ii)   Name of an array
        (iii)  Capacity of an array (Number of Elements) </PRE>
<BR>Without PRE TAG<BR>
An array must declare before it use in the program as same as the other
```

variables, three thing are declared in an array declaration:

(i) Types of an Array

(ii) Name of an array

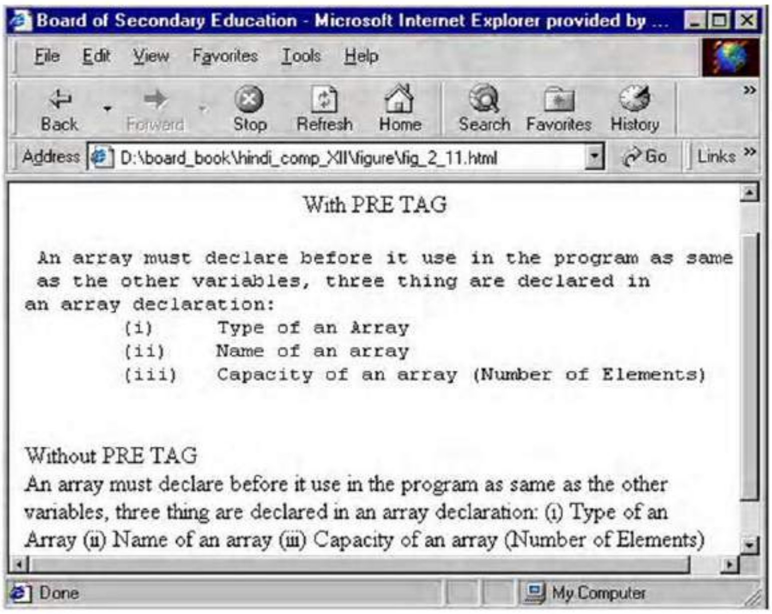
(iii) Capacity of an array (Number of Elements)

</P>

</BODY>

</HTML>

उपरोक्त उदाहरण में जैसा टेक्स्ट लिखा हुआ है वैसा ही ब्राउजर में दिखाई देगा। इसको चित्र 2.11 में दिखाया गया है।



चित्र 2.11

2.9 होरिजोनटल रूल टेग (Horizontal Rule Tag)

<HR> टेग क्षैतीज के समानांतर लाइन (Rule) खिंचता है यह टेग अगर बताया जाए तो पूरे पेज के आर पार भी क्षैतीज के समानांतर लाइन खींच सकता है इस टेग के एट्रीब्यूट निम्न

प्रकार हैं

एट्रीब्यूट	**स्पष्टीकरण**
ALIGN	ब्राउज़र की स्क्रीन पर लाइन को अलाइन करता है जो डिफॉल्ट रूप से स्क्रीन के सेन्टर में होती है। **ALIGN = LEFT** **ALIGN = RIGHT** **ALIGN = CENTER**
SIZE	Rule की साईज परिवर्तित करता है।
WIDTH	Rule की चौड़ाई सेट करता है। इसे हम पिक्सल की संख्या या स्क्रीन की वास्तविक चौड़ाई का प्रतिशत में भी प्रदर्शित कर सकते हैं।

उदाहरण:

Welcome to our Web site.
<HR ALIGN=LEFT WIDTH=10 SIZE=4>

आउटपुट:

महत्वपूर्ण बिन्दु

1. टेग दो प्रकार के होते हैं :
   (अ) कण्टेनर टेग,
   (ब) एम्प्टी टेग
2. टेक्स्ट के चारों तरफ कन्टेनर टेग में ऑपनिंग और क्लोजिंग टेग की आवश्यकता होती है।
3. एम्प्टी टेग अकेले होते हैं।
4. एट्रिब्यूट के द्वारा टेग के व्यवहार को बदला जा सकता है।
5. HTML वेब पेज के दो भाग होते हैं।
   (अ) शीर्ष खण्ड,
   (ब) प्रधान खण्ड
6. शीर्ष टेग में HTML, HEAD और TITLE टेग होते हैं।
7. Body खण्ड Body Tag तथा अन्य सभी टेगों को रखता है।
8. शीर्षक टेग H1 से H6 होते हैं। H1 शीर्षक टेग सबसे ऊपर के स्तर पर तथा H6 टेग सबसे नीचे के स्तर पर उपयोग होता है।
9. <FONT> टेग टेक्स्ट की शैली बदलने के उपयोग आता है। <Font> टेग के Size, Face और color एट्रिब्यूट होते हैं।
10. पैराग्राफ टेग (<P>) नये पैराग्राफ बनाने में उपयोग आता है।
11. लाइन ब्रेक (<BR>) टेग आने पर टेक्स्ट नई लाइन से शुरू होता है।
12. <Basefont>, <BIG>, <SMALL>, <I>, <B> <U> <STRIKE> <SUB> और <SUP> टेक्स्ट फोरमेटिंग में उपयोग आते हैं।

अभ्यासार्थ प्रश्न

बहुचयनात्मक प्रश्न

1. इसमें से कौन सा कन्टेनर टेग नहीं है –
   (अ) <P> (ब) <H1>
   (स) <FONT> (द)
2. HTML दस्तावेज का एक्सटेन्शन क्या होता हैं ?
   (अ) HTML (ब) HML
   (स) HTM (द) अ और स दोनों
3. HTML में कितने शीर्षक टेग उपलब्ध है –
   (अ) 3 (ब) 6
   (स) 5 (द) इसमें से कोई नहीं
4. <FONT> टेग का एट्रिब्यूट कौन सा नहीं है ?
   (अ) face (ब) colour
   (स) size (द) Background color
5. Size = "+3" का <Font> टेग में क्या मतलब है –
   (अ) फोन्ट साइज 3 हो जाएगी
   (ब) फोन्ट साइज 3 घट जाएगी
   (स) फोन्ट साइज 3 बढ़ जाएगी
   (द) डिफाल्ट साइज 3 हो जाएगी।
6.
 टेग लिए कौन सा गलत है ?
   (अ) यह एक खाली टेग है।
   (ब) यह दो लाइनों के मध्य स्पेस देने में उपयोग आता है।
   (स) यह टेक्स्ट को नई लाईन से शुरू करता है।
   (द) यह फोन्ट की साइज को बढ़ता है।
7. <STRONG> टेग का उपयोग होता है –
   (अ) यह <B> टेग के समान है।
   (ब) यह एक ाब्द को बोल्ड बनने में उपयोग आता है।
   (स) यह फोन्ट की साइज एक बनाने में उपयोग आता है।
   (द) अ और ब दोनों ।
8. <EM> टेग का उपयोग होता है
   (अ) एक ाब्द को इटेलिक बनाने में उपयोग
   (ब) यह फोन्ट की साइज घटाने में उपयोग
   (स) यह फोन्ट की साइज बढ़ाने में उपयोग
   (द) इसमें से कोई नहीं

9. निम्न में से कौनसा <Body> टेग का एट्रिब्यूट नहीं है –
   (अ) Text (ब) Colour
   (स) Background (द) Alink

अतिलघुत्तरात्मक प्रश्न

1. HTML में आप टेग को कैसे कमबद्ध (Classify) कर सकते हैं ?
2. एट्रिब्यूट क्या हैं ?
3. दो वेबब्राउजरों के नाम लिखिये।
4. <TITLE> टेग का क्या उद्देश्य है ?
5. ब्रेक ग्राउण्ड रंग बदलने के लिए <Body> टेग का syntax लिखिये।
6. <font> टेग के एट्रिब्यूटों के नाम लिखिये।
7. निम्न टेगों के मतलब लिखिये।
   (अ)
   (ब) <P>
   (स) <Basefont>
   (द) <BIG>
   (य) <Small>

लघुत्तरात्मक प्रश्न

1. कन्टेनर टेग क्या हैं ? उदाहरण की सहायता से समझाइये।
2. खाली टेग क्या हैं ? समझाइये।
3. <Body> टेग के एट्रिब्यूट को समझाइये।
4. शीर्षक टेग के क्या उद्देश्य होता है। समझाइयें।
5. उपरोक्त टेगों को उसके सही रचना (syntax) के साथ समझाइये।
   (अ) Paragraph tag
   (ब) Blockquote tag
   (स) Pre tag
   (द) Font tag

निबन्धात्मक प्रश्न –

1. HTML वेब पेज बनने के पद लिखिये।
2. फोर्मेटिंग टेग दिखाने के लिए एक वेब पेज बनाइये।
3. बेकग्राउण्ड पिक्चर दिखाने के लिए एक वेब पेज बनाइये।
4. पेज में विभिन्न शीर्षक टेग दिखाने के लिए **HTML** कोड लिखिये।
5. <Body> टेग और इसके सभी एट्रिब्यूट को समझाइये।

उत्तरमाला

1 (द) 2 (द) 3 (ब) 4 (द) 5 (स) 6 (द) 7 (द) 8 (अ) 9 (द)

****

3
सूची प्रकार के टेगस्
List Type Tags

3.1 परिचय (Introduction):

लिस्ट तीन प्रकार की होती है जिनको HTML में बना सकते हैं। सभी तीनों लिस्ट केन्टेनर टेग है। इनका क्लोजिंग टेग होना चाहिए अन्यथा बाकी बचा हुआ पेज भी लिस्ट का हिस्सा बन जाएगा। वस्तु या टेक्स्ट का भाग जो कि ऑपनिंग और क्लोजिंग टेग के मध्य होता है लिस्ट बन जाती है। यह दांये हाशिये से दूर होती है। लिस्ट की प्रथम लाइन पिछले टेक्स्ट से अलग करने के लिए खाली लाइन को सम्मिलित करते है।

3.2 ऑरडर्ड सूची (Ordered list) :

<OL> टेग वर्ड प्रोसेसर की नम्बरिंग के समतुल्य है। <LI> लिस्ट आइटम टेग के द्वारा लिस्ट की लाइनों को अलग–अलग करते हैं। तथा लिस्ट आइटम के पहले एक नम्बर दिखाई देगा।

उदाहरण :

```
<OL>
List of Input devices:
<LI> Keyboard </LI>
<LI> Mouse </LI>
<LI> Optical character Reader </LI>
<LI> Scanner </LI>
</OL>
```

<LI> टेग के टाइप (type) एट्रिब्यूट को नम्बरों की शैली बदलने के लिए उपयोग कर सकते हैं। इसकी डिफाल्ट शैली अरेबिक ऑरडिनलस् है। आरडर्ड लिस्ट के टाइप सरणी 3.1 में दिखाये गये हैं।

सारणी 3.1

Type	Char	Example
Capital Roman Numerals	I,II,III	Type="I"
Lower case Roman Numerals	i,ii,iii	Type="i"
Arabic Ordinals	1,2,3	Type="1" (default)
Capital Letters	A,B,C	Type="A"
Lower case Letters	a,b,c	Type="a"

ऑरडर्ड लिस्ट अपना प्रकार मध्य में ही <LI> टेग के टाइप (type) एट्रिब्यूट में उचित मान के द्वारा बदल सकते हैं। पूर्व निश्चित मान से भी आरॅडर्ड लिस्ट को शुरू कर सकते है। शुरूआत का डिफाल्ड मान एक होता है। लिस्ट ऑयटम (item) का मान बदलने के लिए <LI> टेग में वल्यू एट्रिब्यूट का उपयोग करते हैं।

उदाहरण :

```
<OL TYPE="a">
List of Input devices:
<LI> Keyboard </LI>
<LI TYPE="i"> Mouse </LI>
<LI VALUE="10"> Optical character Reader </LI>
<LI> Scanner </LI>
</OL>
```

आप अपनी इच्छानुसार लिस्ट को सामान्य अक्षर के अलावा नम्बर या अक्षर से शुरू कर सकते हैं। जैसे D, ii, या VI इस प्रभाव को प्राप्त करने के लिए आपको <OL> टेग के अंदर START एट्रिब्यूट को जोड़ते हैं।

उदाहरण :

```
<OL START = "D">
```

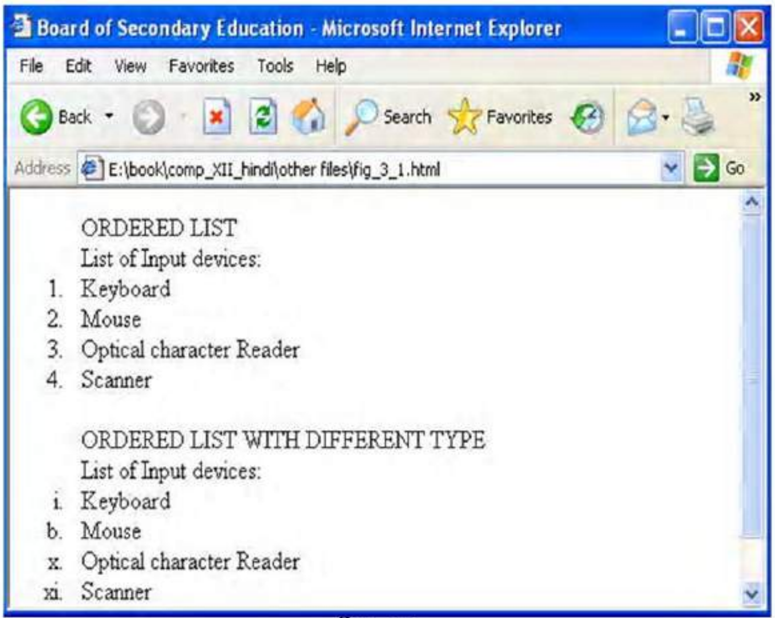
उदाहरण :

ऑरडर्ड लिस्ट के लिए HTML कोड नीचे दिया गया है।

```
<HTML>
<HEAD>
<TITLE> Board of Secondary Education</TITLE>
</HEAD>
<BODY>
<OL>
List of Input devices:
<LI> Keyboard </LI>
<LI> Mouse </LI>
<LI> Optical character Reader </LI>
<LI> Scanner </LI>
</OL>
```

```
<OL TYPE="i">
List of Input devices:
<LI> Keyboard </LI>
<LI TYPE="a"> Mouse </LI>
<LI VALUE="10"> Optical character Reader </LI>
<LI> Scanner </LI>
</OL>
</BODY>
</HTML>
```

उपरोक्त HTML कोड में, यहाँ दो ऑरडर्ड लिस्ट है, प्रथम लिस्ट अरेबिक नम्बर से तथा द्वितीय लिस्ट विभिन्न प्रकार के अक्षरों से शुरू होती है। यह लिस्ट चित्र 3.1 में दिखाया गया है।



चित्र 3.1

3.3 अनऑरडर्ड (Unordered List) :

<UL> टेग वर्ड प्रोसेसर (Word Processor) के बुल्इट (bullet) के साथ तुलना करते हैं। लिस्ट की लाइनों को लिस्ट ऑइटम टेग <LI> अलग–अलग करते हैं। लिस्ट ऑइटम के पहले इसमें बुलेट दिखाई देगा। Type एट्रिब्यूट के द्वारा अनआरडर्ड लिस्ट अपने बुलेट (type bullet) को बदल सकते हैं। इसकी स्वीकार योग्य मान disc, circle और square है। इसकी डिफाल्ट मान

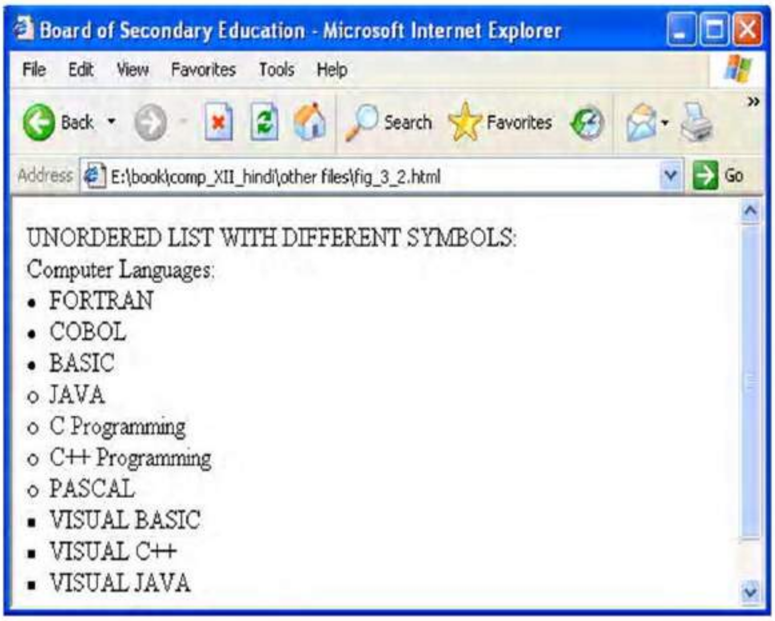
disc है। बुलेट की शैली लिस्ट के मध्य में <LI> टेग के TYPE एट्रिब्यूट को सम्मिलित कर बदल सकते हैं।

उदाहरण :

```
<UL>
Computer Languages:
<LI> JAVA</LI>
<LI> C Programming </LI>
<LI TYPE="square" > C++ Programming </LI>
<LI> Pascal</LI>
<LI TYPE="circle"> COBOL </LI>
<LI> FORTRAN </LI>
</UL>
```

उदाहरण : अनऑरडर्ड लिस्ट HTML कोड नीचे लिखा गया है। अनऑर्डड लिस्ट का ऑउटपुट चित्र 3.2 में दिखाया गया है।

```
<HTML>
<HEAD>
<TITLE> Board of Secondary Education</TITLE>
</HEAD>
<BODY>
UNORDERED LIST WITH DIFFERENT SYMBOLS:
<BR>Computer Languages:
<LI> FORTRAN</LI>
<LI> COBOL </LI>
<LI> BASIC</LI>
<LI TYPE="circle"> JAVA</LI>
<LI TYPE="circle"> C Programming </LI>
<LI TYPE="circle"> C++ Programming </LI>
<LI TYPE="circle"> PASCAL</LI>
<LI TYPE="square"> VISUAL BASIC</LI>
<LI> VISUAL C++</LI>
<LI> VISUAL JAVA </LI>
</UL>
</BODY>
</HTML>
```



चित्र 3.2

3.4 परिभाषा सूची (Definition List) :

परिभाषा सूची तीसरी प्रकार की सूची है। इसके दो भाग होते है।एक शब्द व दूसरा टर्म और व्याख्या में विभक्त होता है। <DT> टेग टर्म को हाईलाइट करने में तथा <DD> टेग व्याख्या को हाईलाइट करने में उपयोग आते हैं।

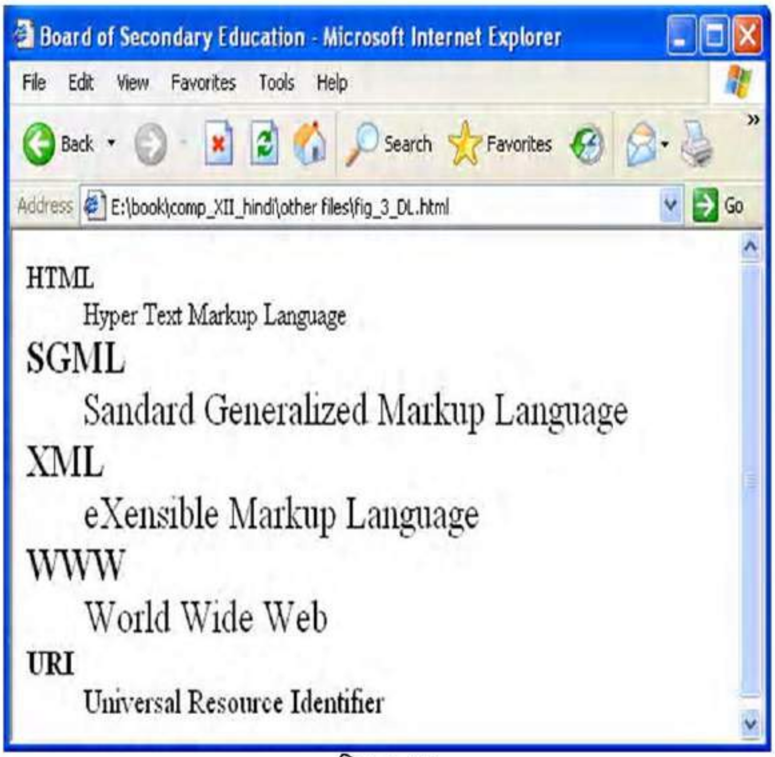
सम्पूर्ण सूची <DL> टेग के अंदर होती है। परिभाषा सूची में भी एण्ड टेग की आवश्कता होती है जो कि सूची को समाप्त करता है। <DD> तथा <DT> टेग अकेली एन्ट्री के लिए होते हैं, इनके एण्ड टेग नहीं होते है।

उदाहरण :

```
<HTML>
<HEAD>
<TITLE> Board of Secondary Education</TITLE>
</HEAD>
<BODY>
<DL>
<DT><B> HTML</B>
<DD>Hyper Text Markup Language </DD>
```

```
<DT><B><FONT size="5">SGML</B>

<DD>Sandard Generalized Markup Language</DD>
<DT><B>XML</B>
<DD>eXensible Markup Language</DD>
<DT><B>WWW</B>
<DD>World Wide Web</DD>
<DT><B><FONT size="4">URI</B>
<DD>Universal Resource Identifier</DD>
</DL>
</BODY>
</HTML>
```

उपरोक्त कोड का ऑउटपुट चित्र 3.3 में दिखाया गया है।



चित्र : 3.3

3.5 एड्रेस टेग (Address Tag) :

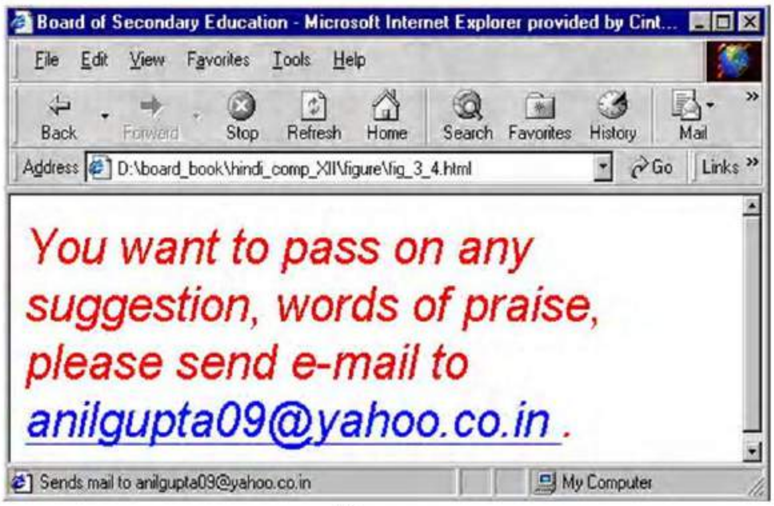
यह वेब पेज पर लेखक और सम्पर्क करने का विवरण दिखाने के लिए उपयोग करते हैं। एड्रेस टेग के अंदर टेक्स्ट इटेलिक होता है। सामान्यतया, <ADDRESS> टेग ई–मेल एड्रेस ही नहीं रखता लेकिन आपके पत्र व्यवहार का पता और टेलिफोन नम्बर भी होते हैं।

उदाहरण :

```
<ADDRESS>
<P>
<FONT COLOR="red" FACE="arial"> You want to pass on any
suggestion, words of praise, please send e-mail to <A
HREF="mailto:anilgupta09@yahoo.co.in"> anilgupta09@yahoo.co.in
</A>.</FONT>
</P>
</ADDRESS>
```

उदाहरण : सम्पूर्ण HTML दस्तावेज एड्रेस टेग का कार्य दिखाता है तथा ब्राउजर में इसका आउटपुट चित्र 3.4 में दिखाया गया है।

```
<HTML>
<HEAD>
<TITLE> Board of Secondary Education</TITLE>
</HEAD>
<BODY>
<ADDRESS>
<P>
<FONT COLOR="red" FACE="arial" size="6"> You want to pass
on any suggestion, words of praise, please send e-mail to <A
HREF="mailto:anilgupta09@yahoo.co.in"> anilgupta09@yahoo.co.in
</A>.</FONT>
</P>
</ADDRESS>
</BODY>
</HTML>
```



चित्र : 3.4

3.6 एन्कर टेग (Anchor Tag) :

वेब पर पढ़ने वाले को एक फाइल से दूसरी फाइल (भौगोलिक रूप से कहीं पर रखी हो) को लिंक करता है। लिंक में टेग के साथ अंदर आइटम (Item) होता, सामान्यतया यह टेक्स्ट होता है। ऑपनिंग टेग फाइल एड्रेस रखता है जहां पर फाइल स्थापन होती है। बंद टेक्स्ट वह टेक्स्ट होता है जो वेब पेज पर दिखाई देता हैं। लिंक टेग <A> के दो एट्रिब्यूट होते है – NAME और HREF। NAME वेब पेज पर चिन्ह बनाता है जहां पर लिंक करना है जबकि HREF लिंक पेज का URL बताता है।

यहां पर तीन तरीके से लिंक कर सकते है :

- ❑ इन्टरनेट पर बाह्य पेज को लिंक करना।
- ❑ साइट के अंदर पेज को लिंक करना।
- ❑ वेब पेज में किसी बिन्दु पर लिंक करना।

उदाहरण :

<A HREF="location/filename.extension"> Text to appear on the web page </A>< A HREF ="body.html"> Link to a previous Chapter </ A>

< A HREF ="www.microsoft.com"> Microsoft Home Page </A>

यह सम्भव है कि <A> टेग के अंदर अन्य तत्वों को रख सकते हैं।

उदाहरण : शीर्षक के लिए लिंक बनाने के लिए निम्न का उपयोग करेंगे।

<A HREF = "filename"> <HI> TEXT </H1> </A>

3.7 लिंक बनाना (Create links) :

एक पेज से दूसरे पेज पर लिंक करना इस भाग में देखेंगे। एक जगह से दूसरी जगह पर जाने की योग्यता, वेबे का उपयोगी होना एक मुख्य कारण है। हाइपरलिंक का गन्तव्य स्थान संसार में अन्य वेब साइट पर या केवल यह अपनी वेब साइट के अंदर कोई विशेष जगह हो सकता है। आप एक शब्द का ग्राफिक्स पर क्लिक जो HTML के अंदर लिखा है और ब्राउजर आपके URL के अनुसार वेब पर ले जाएगा।

3.7.1 HTML दस्तावेज के अन्य स्थानों पर लिंक करना :

आप सोचते हैं कि आप अपनी वेब साइट पर किसी विशेष खण्ड को ढूंढने मे विजिटर्स (ढूढने वाला) की सहायता करना चाहते हैं। यदि आपने उस पेज के लिए कोई लिंक नहीं बनाया है तो उनको उस पेज को नीचे करते रहना पड़ेगा जब तक कि आप अपने रूचि का खण्ड नहीं ढूंढ लेते है। इसके अलावा, वे जो चाहते है उस पेज पर आप उनको शीघ्र ही भेज सकते हैं। यह एन्कर टेग कि सहायता से प्राप्त कर सकते हैं।

उदाहरण :

यह स्टेटमेन्ट लिंक बनाने के लिए लिखा जाता है।

<A HREF = "Filename#Text"> Text </A>

यह स्टेटमेन्ट वहां लिखा जाता है जहाँ आप पहुंचना चाहते हैं।

<A NAME = "TEXT"> TEXT

यदि आप उपरोक्त दोनों स्टेटमेन्ट को वेबपेज पर रखते हैं तो वेबपेज पर प्रथम स्टेटमेन्ट पर क्लिक करने पर शीघ्र ही दूसरे स्टेटमेन्ट पर पहुँच जाएंगे। टेक्स्ट शब्द के स्थान पर कोई टेक्स्ट लिखना पड़ता है। जहां पर आप HTML कोड लिखते हैं वहीं फाइल का नाम होगा।

उदाहरण :

<A HREF = "Index.html# List of Input devices"> List of input Devies </A>

और

<A NAME = "List of Input devices"> List of input devise </A>

उदाहरण : यह HTML दस्तावेज दो लिंक बनाता है।

1. अन्तिम पेज पर पहुंचने के लिए।
2. प्रथम पेज पर पहुंचने के लिए।

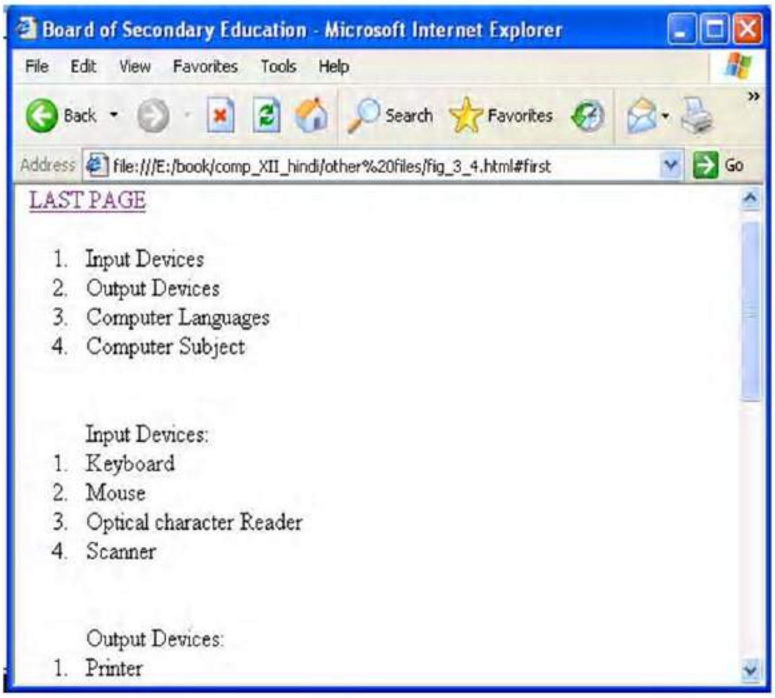
HTML दस्तावेज की फाइल का नाम index.html है।

```
<HTML>
<HEAD>
<TITLE> Board of Secondary Education</TITLE>
</HEAD>
<BODY>
<A HREF="index.html#last">LAST PAGE</A>
<A Name="first"></A>
<OL>
```

```
<LI>Input Devices</LI>
<LI>Output Devices</LI>
<LI>Computer Languages</LI>
<LI>Computer Subject</LI>
<BR>
</OL>
<OL>
<BR>
Input Devices:
<LI> Keyboard </LI>
<LI> Mouse </LI>
<LI> Optical character Reader </LI>
<LI> Scanner </LI>
</OL>
<OL>
<BR>Output Devices:
<LI> Printer </LI>
<LI> Plotter </LI>
<LI> Monitor </LI>
<LI> Floppy </LI>
</OL>
<OL>
<BR>
Computer Languages:
<LI> JAVA</LI>
<LI> C Programming </LI>
<LI> C++ Programming </LI>
<LI> Pascal</LI>
<LI> COBOL </LI>
<LI> FORTRAN </LI>
</OL>
<OL>
<BR>
Computer Subject
<LI> Data Analysis and Design</LI>
<LI> Data Structure </LI>
<LI> Data Mining </LI>
<LI> HTML </LI>
<LI> Visual Basic </LI>
```

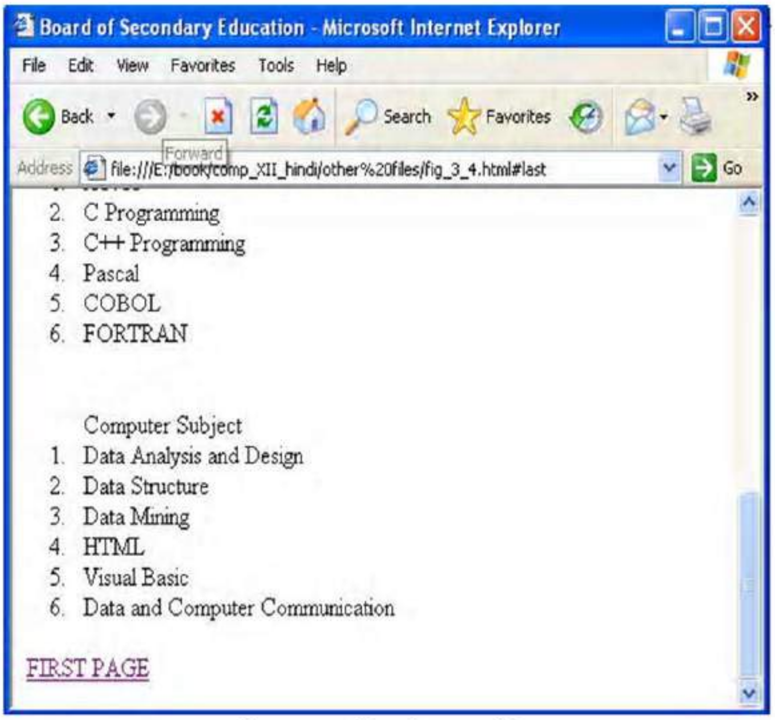
```
<LI> Data and Computer Communication </LI>
</OL>
<A NAME="last"></A>
<A HREF="index.html#first">FIRST PAGE</A>
</BODY>
</HTML>
```

उपरोक्त **HTML** फाइल में, हमने इसमें दो लिंक प्रथम पेज से अन्तिम पेज पर जाने के लिए तथा अन्तिम पेज से प्रथम पेज पर आने के लिए बनाये है। इनकी आउटपुट स्किन चित्र 3.5 और 3.6 में दिखाई गई है।



चित्र 3.5: प्रथम पेज पर लिंक

चित्र 3.5 में, वेब पर लास्ट पेज **(LAST PAGE)** हाइपर लिंक दिखाया है। जब आप लास्ट पेज लिंक पर क्लिक करेगें तब यह वेब पेज का अन्तिम पेज दिखायेगा। समानरूप से, चित्र 3.6 में, वेब पर फस्टपेज **(First Page)** हाइपरलिंक दिखाया गया है। जब आप फस्ट पेज लिंक क्लिक करेगें तब यह वेबपेज का प्रथम पेज दिखायेगा।



चित्र 3.6: अन्तिम पेज पर लिंक

3.7.2 अन्य HTML दस्तावेज के लिए लिंक :

आप सोचते हैं कि आप अपनी वेबसाइट पर अन्य HTML दस्तावेजों के किसी विशेष खण्ड को ढूंढने के लिए विजिट्र्स की सहायता करना चाहते हैं। आपको उन पेजों को लिंक करने की आवश्यकता होगी। यह एन्कर टेग की सहायता से प्राप्त कर सकते हैं।

उदाहरण : यह स्टेटमेन्ट लिंक बनाने के लिए मुख्य फाइल में लिखेंगे।

<A HREF = "Filename#Text"> Text</A>

यह स्टेटमेन्ट वहाँ लिखेंगे जहाँ हम पहुंचना चाहेंगे।

<A NAME = "Text"> Text</A>

यदि आप उपरोक्त दोनों स्टेटमेन्ट को वेब पेज पर रखते हैं तो वेबपेज पर प्रथम स्टेटमेंट पर क्लिक करने पर शीघ्र ही दूसरे स्टेटमेन्ट पर पहुँच जाएंगे। टेक्स्ट शब्द के स्थान पर कोई टेक्स्ट लिखना पड़ता है।

उदाहरण :

<A HREF = "link.html # List of Input Devices"> List of Input Devices </A>

और

< A NAME = "List of Input Devices </A>

यहां हम दो HTML फाइल का उपयोग करेंगे। एक index.html और दूसरी link.html है। दोनों फाइलें नीचे दिखाई गई है।

1. Index.html HTML file:

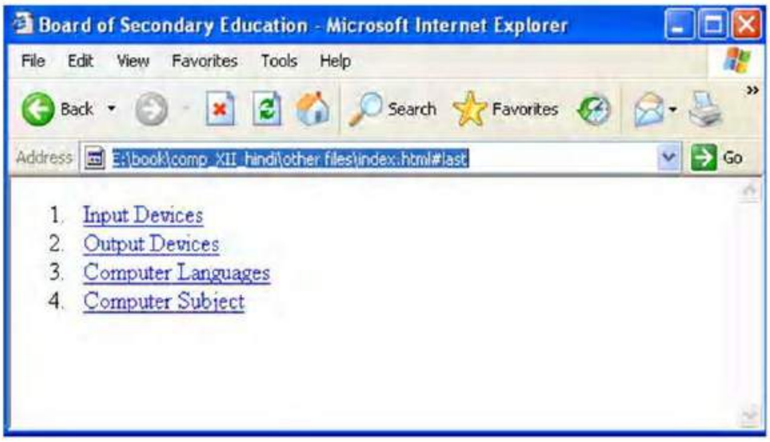
```
<HTML>
<HEAD>
<TITLE> Board of Secondary Education</TITLE>
</HEAD>
<BODY>
<OL>
<LI><A HREF="link.html#Input Devices">Input Devices</A></
LI>
<LI><A HREF="link.html#Output Devices">Output Devices</
A></LI>
<LI><A HREF="link.html#Computer Languages">Computer
Languages</A></LI>
<LI><A HREF="link.html#Computer Subject">Computer Sub-
ject</A>
</LI>
<BR>
</OL>
</BODY>
</HTML>
```

2. Link.html HTML file:

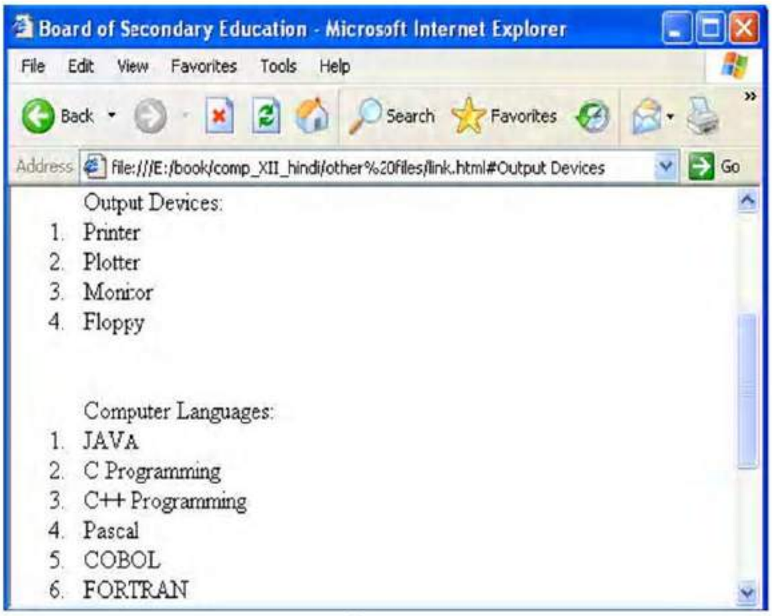
```
<HTML>
<HEAD>
<TITLE> Board of Secondary Education</TITLE>
</HEAD>
<BODY>
<OL>
<BR><A NAME= "Input Devices">Input Devices:</A>
```

```
<LI> Keyboard </LI>
<LI> Mouse </LI>
<LI> Optical character Reader </LI>
<LI> Scanner </LI>
</OL>
<OL>
<BR><A NAME= "Output Devices">Output Devices:</A>
<LI> Printer </LI>
<LI> Plotter </LI>
<LI> Monitor </LI>
<LI> Floppy </LI>
</OL>
<OL>
<BR>
<A NAME= "Computer Languages">Computer Languages:</A>
<LI> JAVA</LI>
<LI> C Programming </LI>
<LI> C++ Programming </LI>
<LI> Pascal</LI>
<LI> COBOL </LI>
<LI> FORTRAN </LI>
</OL>
<OL>
<BR><A NAME= "Computer Subject">Computer Subject</A>
<LI> Data Analysis and Design</LI>
<LI> Data Structure </LI>
<LI> Data Mining </LI>
<LI> HTML </LI>
<LI> Visual Basic </LI>
<LI> Data and Computer Communication </LI>
</OL>
</BODY>
</HTML>
```

उपरोक्त उदाहरण में, यहाँ दो HTML फाइल है। एक index.html और दूसरी link.html हमने index.html हाइपर लिंक बनाये जो शीघ्र ही link.html फाइल के विशेष पेज को खोल देते हैं। उनको चित्र 3.7 और 3.8 में दिखाया गया है।


Board of Secondary Education - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address E:\book\comp_XII_hindi\other files\index.html#last
Go
1. Input Devices
2. Output Devices
3. Computer Languages
4. Computer Subject

चित्र 3.7


Board of Secondary Education - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address file:///E:/book/comp_XII_hindi/other%20files/link.html#Output Devices
Go
Output Devices:
1. Printer
2. Plotter
3. Monitor
4. Floppy
Computer Languages:
1. JAVA
2. C Programming
3. C++ Programming
4. Pascal
5. COBOL
6. FORTRAN

चित्र 3.8

3.7.3 अन्य साइट और डाटा ऑब्जेक्ट लिंक करना :
(Link to other Website or Date Objects) :

वेबसाइट में हाइपर लिंक एक बहुत सामान्य तरीका है जो विजिटर्स को अन्य वेबसाइटस् से लिंक करना है। आप अपनी वेबसाइट को भी लिंक कर सकते हो।

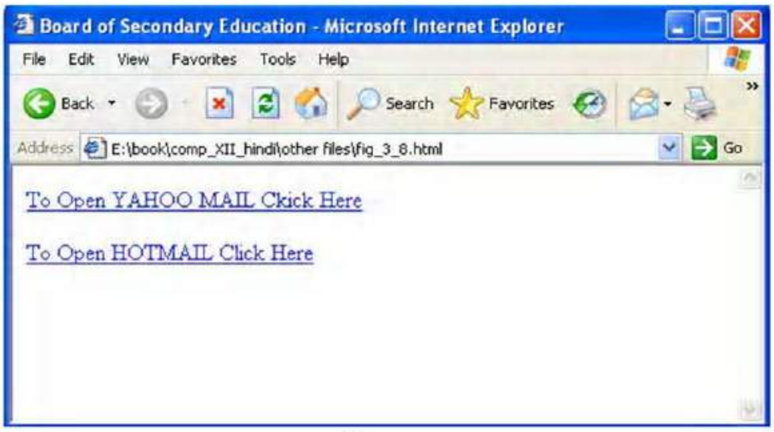
उदाहरण :

```
< A HREF = "अन्य साइट का नाम"> </A>
< A HREF = "http://www.yahoo.com"> </A>
```

उदाहरण :यह HTML फाइल यह दर्शाती है कि आप अन्य वेबसाइट से कैसे लिंक बनाते हो।

```
<HTML>
 <HEAD>
 <TITLE> Board of Secondary Education</TITLE>
 </HEAD>
 <BODY>
<P><A HREF="http://www.yahoo.com"></A> YAHOO MAIL
<BR>
<A HREF="http://www.hotmail.com"></A> HOTMAIL</P>
<BR>
</BODY>
</HTML>
```

उपरोक्त उदाहरण में, हमने दो हाइपर लिंक सम्मिलित किये हैं। यह अलग–अलग वेबसाइट के बाह्य लिंक है। इस स्थिति में, बाह्य लिंक के लिए URL "http://www.yahoo.com" और "http://www.hotmail.com" है। यह वेबपेज चित्र 3.9 व 3.10 में दर्शाये गये हैं।



चित्र 3.9

चित्र 3.10

3.8 छाया (Image) :

वेब पेज का प्रभाव बढ़ाने के लिये इमेज़ (छाया) का उपयोग किया जा सकता है। इसका एक बड़ी हानि है कि वे डाउनलोड होने में समय लेती है। यह सर्वमान्य है कि वेबपेज 30k साइज से कम होना चाहिए अन्यथा यह दूसरे क्रम (एक पेज जो लिंक होता है) का पेज है और पढ़ने वाला डाउनलोड समय के प्रति जागरूक होता है। यह भी सिफारिश करता है कि 256 रंगों वाला समूह रखें। इमेज HTML के वास्तविक पेज के अंदर नहीं रखी जाती हैं, इसको डाउनलोड करते हैं। इसकी आवश्यकता दिखाने के लिए होती है तब ही यह लाइन लोड होती है।
उदाहरण : वेब पेज पर इमेज को दर्शाने के लिए इमेज टेग का उपयोग करते हैं।

< IMG Src = "location / filename.extension">

इमेज टेग के कई एट्रिब्यूट होते हैं। यह निम्न प्रकार हैं :

सारणी 3.2

align	इमेज को अमुक स्थान रखने के लिए उपयोग होता है। एलाइनमेन्ट के मान ऊपर (top), बीच में (middle) नीचे (Bottom), दाँया (right), बाँया (Left) होते हैं।
alt	इमेज का विवरण स्क्रीन पर पढ़ने वालों के लिए रखने के या जब इमेज डाउनलोड़ होती है तो उस जगह पर दर्शाने के लिए या यदि ब्राउजर इमेज

	को प्राप्त नहीं करता है तो दर्शाने के लिए उपयोग आता है।
height	इमेज की ऊँचाई बताने के लिए उपयोग होता है।
hspace	इमेज के चारों तरफ क्षितिज अतिरिक्त स्थान बताने के उपयोग में आता है।
src	इमेज जिस जगह स्टोर उसका स्थान बताने के उपयोग में आता है।
vspace	इमेज के चारों तरफ ऊध्व अतिरिक्त स्थान बताने के उपयोग में आता है।
width	इमेज की चौड़ाई बताने में उपयोग करते है।

उदाहरण :

```
< IMG  Src = "glope.gif">
< IMG src="tiger.jpg" height="200" width="200">
< IMG src="glope.gif" align="right">
```

जब इमेज वेब पेज पर बाँये या दाँये रखी जाती है। तब इसके बाद वाला टेक्स्ट इसके दाँये या बाँये आ जाएगा। टेक्स्ट को दो इमेजों के मध्य रखने के लिए इमेज तत्वों को टेक्स्ट से तत्क्षण पूर्व रखना होता है ताकि यह उनके मध्य दिखाई दे। प्रथम इमेज को दाँये रखनी चाहिये और align="right" इमेज टेग में सम्मिलित करना आवश्यक है। द्वितीय इमेज तत्व के इमेज टेग में align="left" इमेज टेग में सम्मिलित करना आवश्यक है। तब दो इमेज के मध्य हाशिया दिखाई देता है।

इमेज के चारों तरफ इमेज को रोकने के लिए लाइनब्रेक के साथ एट्रिब्यूट उपयोग करना चाहिये। इसके निम्न मान हो सकते हैं जिनको सारणी 3.3 में दर्शाया गया है।

सारणी 3.3

< BR Clear = "left">	यह प्रकट होते हैं जब बाँयें हाशिये (Margin) में कोई इमेज नहीं होती है।
< BR Clear = "right">	यह प्रकट होते हैं जब दाँयें हाशिये (Margin) में कोई इमेज नहीं होते हैं।
< BR Clear = "all">	यह प्रकट होते हैं जब दोनों तरफ कोई इमेज नहीं होती है।

टेक्स्ट को hspace और vspace एट्रिब्यूट का उपयोग करके चारों तरफ से दूर रख सकते है। यह इमेज के ऊपर नीचे के तत्क्षण दाँये और बाँये जगह (स्पेस) रखेगा।

उदाहरण :

```
<IMG Src = "globe.gif hspace = "10" vspace = "15">
```

लिंक इमेज (Link Image) :

अन्य वेबपेज के लिए इमेज लिंक बन सकती है। <A> टेग <IMG> के साथ <A> टेग के ऑपनिंग और क्लोजिंग के मध्य लिंक बनाने में उपयोग किया जाता है।

उदाहरण :

```
<A href="location/filename.extension">
<IMG src="location/filename.extension">
</A>
```

एक ही इमेज पर दो लिंकों को अलग–अलग करने के लिए बार्डर का उपयोग करते हैं। बार्डर को हटाने के लिए <IMG> टेग में बोर्डर एट्रिब्यूट को सम्मिलित करना होता है।

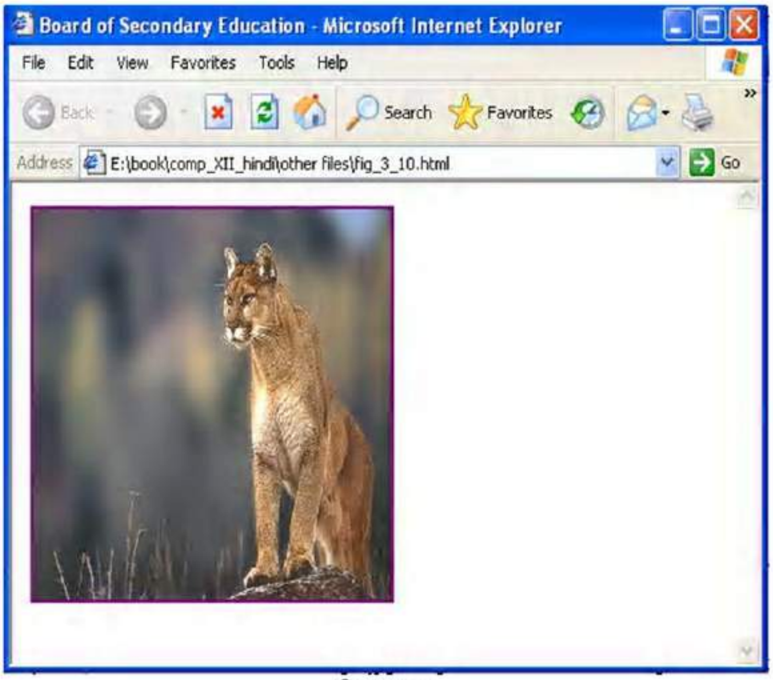
उदाहरण :

```
<A href="link.htm">
<IMG src="next1.gif" height= "50" width="50" align="right" border=0>
</A>
```

उदाहरण : इमेज के साथ हाइपरलिंक बनाने के लिए **HTML** का कोड लिखिये।

```
<HTML>
<HEAD>
<TITLE> Board of Secondary Education</TITLE>
</HEAD>
<BODY>
<A HREF="link.html">
<IMG src="tiger.jpg" height="250" width="250" align="left" border=2>
</A>
</BODY>
</HTML>
```

उपरोक्त HTML कोड में, टेग के अंदर टेग का उपयोग किया है। यदि आप इमेज पर क्लिक करेंगे तो ब्राउजर में link.html फाइल खुल जाएगी। यदि आप **HTML** कोड में से टेग हटा देगें तो आप पिक्चर के साथ हाइपरलिंक नहीं बना सकते। इसको चित्र 3.11में दिखाया गया है।



चित्र 3.11

3.9 क्षैतिज (Horizontal) टेग :

यह टेग पेज पर एक आरपार लाइन बनाने के लिए उपयोग में आता है। क्षैतिज टेग सामान्यताया विषय में बदलाव को इंगित करने के उपयोग में आता है। <HR> के कई एट्रिब्यूट होते है। इनको लगाने पर इसकी दिखावट बदल जाती है।

align एलाइनमेन्ट के सामान्य मान दॉया, बॉया ओर मध्य है। इसका डिफाल्ट मान मध्य है।

size साइन लाइन की मोटाई बताता है। साइज को पिक्सल में लिखते हैं।

width चौड़ाई लाइन की लम्बाई को बताता है। यह मान पिक्सल और प्रतिशत दोनों में दे सकते हैं। इसका डिफाल्ट मान 100 प्रतिशत है।

noshade यह एट्रिब्यूट लाइन को दो रंग के गुण से समतल बनाती है।

उदाहरण :

<HR>

<HR size="3" width="70%" noshade >

नोट : – माइक्रोसोफट इन्टरनेट एक्सप्लोरर टेग में एक अतिरिक्त एट्रिब्यूट कलर को अनुमति देता है। रंग को हेक्साडेसिमल कोड या पूर्वघोषित नाम से बताया जा सकता है।

उदाहरण :

<HR size="3" width="70%" color="blue" >

3.10 HTML कोड लिखने के नियम :

HTML एक फोरमेंटिंग भाषा की तरह मानी जाती है। यह अन्य प्रोग्रामिंग भाषाओं से अलग होती है। इसके लिए कुछ नियमों का अनुकरण करने की आवश्यकता होती है। नियम अन्य भाषा से अलग हो सकते हैं।

1. लैस–देन और ग्रेटर देन के चिन्ह के बाहरी भाग में उपयोग करो।
   उदाहरण : जैसा <BODY> टेग में है।
2. यदि टेग में एण्ड टेग की आवश्यकता हो तो इसको उपयोग करना चाहिए। एण्ड टेग स्टार्ट टेग के समान होता है।
   उदाहरण : जैसा </BODY> टेग में है।
3. आपको अपनी HTML फाइल को <HTML> टेग के साथ शुरू करना चाहिए और समाप्त </HTML> टेग के साथ इसके इलावा सब कुछ इन दो टेग के मध्य होना चाहिए।
4. जब आप एट्रिब्यूट में मान लिखते हो तो बराबर चिन्ह का उपयोग करो।
   उदाहरण : जैसा TEXT= में है।
5. जब आप एट्रिब्यूट लिखते हैं तो बराबर चिन्ह के दोनों तरफ स्पेस नहीं रखनी चाहिए।
6. एट्रिब्यूट के मान के दोनों तरफ कोटेशन चिन्ह लगाना होता है।
   उदाहरण : size="20"
7. हैश चिन्ह (#) का उपयोग करना चाहिएजब रंग को उनके हैक्साडेसीमल कोड में लिखे।

उदाहरण : जैसा TEXT = "#871E11"

8. आपको ज़्यादातर यह सुनिश्चित करने की आवश्यकता होती है कि आपने एण्ड टेग को क्रम से लगाया है या नहीं । जब आप टेग के अंदर टेग रखते है, तो स्टार्ट टेग तथा एण्ड टेग को देखते रहना होता है ताकि आप उनको एक के ऊपर एक नहीं लगा दें। अन्य शब्दों में, यदि एक वाक्य को अन्डर लाइन करना चाहते हैं और तब प्रथम टेग के दोनों तरफ अन्य टेग के युग्म को लगाते हैं। जैसा निम्न उदाहरण में है।
उदाहरण :
<U><I>This sentence appears underline and in italic type.</I></U>
इस उदाहरण में गलत तरीके से टेगों को एक के ऊपर एक लगाना दिखाया गया है।
<U> <I> This sentence appears underline and in italic type. The tags are placed incorrectly </U> </I>

9. टेग और एट्रिब्यूट के लिए बड़े ाब्दों का उपयोग करना चाहिये जैसे <BODY> और <TEXT>। आप छोटे शब्दों का उपयोग कर सकते है, लेकिन जब आप HTML फाइल में कुछ बदलाव करना चाहेंगे तो टेगों और एट्रिब्यूटो को पहचानना कठिन बन जाता है।

10. आप अपने **HTML** कोड को टेब और रिटर्न का उपयोग कर पढ़ने योग्य बना सकते हैं। HTMLकोड को प्रत्येक लाइन के बाद एन्टर कुंजी (Enter key) दबाओ। कोड की लाइनों को पृष्ठ प्रान्त से हटा के लिखने से कोड की लाइनों के मध्य संबंध दिखता है। निम्न उदाहरण से यह मालूम होता है कि HTML में कौन से टेग किस टेग का सहायक है। इन्डेन्ट आपको बताता है कि सम्पूर्ण सूची अन्य सूची की सहायक है।

```
<UL>
<LI>
<LI>
    <UL>
    <LI>
    <LI>
    </UL>
<LI>
<LI>
</UL>
```

11. आप अपने कार्य को जाँचते जाये, ताकि आपको घंटों इस कार्य में नहीं लगे कि वेब पेज वैसा नहीं कर रहा है जैसा आपको आशा थी।

महत्वपूर्ण बिन्दु

1. HTML में तीन प्रकार की सूची सम्मिलित होती है : –
   (अ) ऑरडर्ड सूची (ब) अनऑरडर्ड सूची (स) परिभाषा सूची
2. एड्रेस लेख या मिलने वाले व्यक्ति के बारे में सूचना दिखाने के उपयोग में आता है।
3. एन्कर टेग अन्य दस्तावेज या साइट को लिंक करने के उपयोग में आता है।
4. एन्कर टेग की वाक्य रचना (syntax) है :– <A HREF="filename"> text </A>
5. इमेज टेग वेबपेज पर इमेज दिखाने के लिए उपयोग आती है।
6. इमेज लिंग बनाने के लिए <A> टेग के अंदर <IMG> टेग का उपयोग करते हैं।
7. पेज के आरपार क्षैतिज लाइन खिंचने के लिए क्षैतिज क्रम टेग क्रम का उपयोग करते हैं।

अभ्यासार्थ प्रश्न

बहुचयनात्मक प्रश्न

1. ऑरडर्ड सूची के लिए उपयोग होने वाला टेग है :
   (अ) <LI> (ब) <OL>
   (स) <UL> (द) <DL>
2. इनमें से कौन सा ऑरडर्ड सूची टेग का एट्रिब्यूट नहीं है ?
   (अ) type (ब) start
   (स) Value (द) end
3. <DT> टेग का उपयोग होता है –
   (अ) टर्म का हाईलाइट करने के (ब) विवरण को हाईलाइट करने के
   (स) फोन्ट की साइज बदलने के (द) उपरोक्त में से कोई नहीं
4. इनमें से कौनसा मान अनऑरडर्ड सूची के टाइप एट्रिब्यूट का नहीं है।
   (अ) Square (ब) Circle
   (स) disc (द) rectangle
5. एन्कर टेग का उपयोग होता है –
   (अ) इन्टरनेट पर बाह्य पेज को लिंक करने के (ब) साइट के अंदर पेजों को लिंक करने के
   (स) वेबपेज के किसी बिन्दु को लिंक करने के (द) उपरोक्त सभी
6. कौन सा एन्कर टेग का एट्रिब्यूट है ?
   (अ) HREF (ब) NAME
   (स) अ और ब दोनों (द) Size
7. <IMG> टेग का कौनसा एट्रिब्यूट नहीं हैं ?
   (अ) height (ब) width
   (स) align (द) उपरोक्त में से कोई नहीं
8. <IMG> टेग के SRC एट्रिब्यूट की सही मान कौन सा हैं ?
   (अ) Glope.gif (ब) tiger.jpg
   (स) अ व ब दोनों (द) उपरोक्त में से कोई नहीं

9. <HR> टेग का कौन सा एट्रिब्यूट हैं ?
   (अ) Align (ब) Size
   (स) Width (द) Font

अति लघुत्तरात्मक प्रश्न :

1. कौन सा टेग ऑरर्डड सूची बनाने के उपयोग में आता हैं ?
2. कौन सा टेग अनऑरर्डड सूची बनाने के उपयोग में आता है ?
3. <OL> टेग के टाइप एट्रिब्यूट के क्या मान होते हैं ?
4. <UL> टेग का क्या उपयोग हैं ?
5. एन्कर टेग की क्या वाक्य रचना हैं ?
6. वेबपेज के अंदर लिंक बनाने के लिए कौन से टेग उपयोग में आते हैं। उनकी सूची लिखिये।

लघुत्तरात्मक प्रश्न :

1. <DL> टेग में स्टार्ट एट्रिब्यूट का क्या मतलब होता हैं ?
2. एड्रेस टेग को उदाहरण की सहायता से समझाइये।
3. वेब पेज के ब्रेकग्राउण्ड में पिक्चर दिखाने के लिए <BODY> टेग की वाक्य रचना (Syntax) लिखिये।
4. क्षैतिज क्र

म टेग क्या हैं ? समझाइये।

5. इमेज के अंदर लिंक बनाने के लिए <A> और <IMG> टेग की वाक्य रचना लिखिये।
6. उसी HTML पेज पर हाइपरलिंक बनाने के लिए वाक्य रचना लिखिये।
7. लिंक के प्रकार लिखिये।
8. निम्न टेगों को समझाइये।
   (अ) <A> (ब) <HR> (स) <LI>

निबंधात्मक प्रश्न :

1. उसी HTML पेज पर हाइपरलिंक बनाने के लिए HTML का कोड लिखिये।
2. अन्य HTML पेज के लिए हाइपरलिंक बनाने हेतु HTML का कोड लिखिये।
3. उचित टेग का उपयोग करते हुए वेब पेज बनाइये। क्लिक करने पर विषय सूची दिखाये।
4. ऑरडर्ड सूची का उपयोग करते हुए सभी इनपुट और आउटपुट डिवाइसेज को दिखाने के लिए HTML कोड लिखिये।

उत्तरमाला

1 (ब) 2 (अ) 3 (द) 4 (ब) 5 (द) 6 (स) 7 (द) 8 (स) 9 (द)

4

अग्रिम एचटीएमएल
Advance HTML

4.1 टेबल (Table)

आधिकारिक तौर पर टेबलस् को HTML 3.2 में सन् 1995 में जोड़ा गया था। इसके पश्चात् टेबलस ने वेब पेज डिजाइन के क्षेत्र में क्रांति ला दी। टेबलस् का उपयोग वेब पेज पर सूचनाओं को सारणी (Table) के रूप में प्रदर्शित करने के अतिरिक्त वेब पेज के ले आउट (layout) को कंट्रोल करने के लिए भी उपयोग लिया जाने लगा।

किसी वेब पेज में टेबल बनाने के लिए मुख्यतः निम्न टेगस् का उपयोग किया जाता है।

Table → सारणी को परिभाषित करने के लिए
TR → सारणी की पंक्ति (ROW) को परिभाषित करने के लिए
TD → सारणी के डाटा कोश (Data Cell)
TH → सारणी के शीर्षक कोश (Header Cell)
CAPTION → सारणी का केप्शन (Caption) देने के लिए
THEAD → सारणी के शीर्ष (Head) के लिए
TBODY → सारणी की बाडी (Body) के लिए
TFOOT → सारणी के फुटर (Footer) के लिए
COL → स्तम्भ (Column) परिभाषित करने के लिए
COLGROUP → स्तम्भ समुह (Column Group) परिभाषित करने के लिए

4.2 <Table> टेग (<Table> Tag)

किसी वेब पेज में सारणी बनाने के लिए इस टेग का उपयोग किया जाता है।

Syntax–

<Table> .... </Table>

उपरोक्त टेग एक सारणी के प्रारम्भ तथा अंत को प्रदर्शित करता है।

किसी वेब पेज में पंक्ति / सेल (Row/Cell) मेट्रिक्स (Matrix) बनाने में लिए <Table> टेग

का उपयोग किया जाता है। सारणी के कोश में विभिन्न प्रकार के आब्जेक्ट (object) जैसे– संख्या (Numbers), टेक्स्ट (Text), चित्र (Image), अथवा एंकर (Anchor) आदि को रखा जा सकता है। सारणियों से सम्बन्धित सभी टेगस् को <Table> तथा </Table> टेग युग्म के मध्य में उपयोग में लिया जाता है।

एक सामान्य सारणी की HTML संरचना निम्न प्रकार होती है–

```
<HTML>
<TITLE> ................ </TITLE>
<TABLE>                <–! Table start hare>
<TR>                   <–! First table row start here>
<TD> Content </TD>     <–! First table data in first row of table>
</TR>                  <–! end of first row>
<TR>                   <–! Second table row start here>
<TD> ....................</TD> <–! First table data in second row of table>
<TD> .................... </TD><–! Second table data in second row of table>
</TR>                  <–! End of second row>
</Table>               <–! End of table>
</HTML>
```

4.2.1 <Table> टेग के एट्रीब्यूट (Attibutes of <Table> Tag)

Border= "number of pixels"

इस एट्रीब्यूट का उपयोग सारणी तथा कोश के चारों ओर के बाह्य किनारों (Border) की चौड़ाई (width) को पिक्सल संख्या के रूप में परिभाषित करने में किया जाता है। यदि border = 0 है तो सारणी तथा कोश के चारो ओर का बार्डर हट जाता है।बार्डर (border) एट्रीब्यूट का डिफालट (default) मान 1 होता है।

Cellpadding = "number of pixels or %"

इस एट्रीब्यूट का उपयोग कोश के बार्डर तथा सेल में उपस्थित कन्टेट (content) के मध्य पिक्सल अथवा टेबल की कुल चौड़ाई के प्रतिशत के रूप में खाली स्थान/दूरी तय करने के काम में लिया जाता है।इस एट्रीब्यूट का डिफाल्ट (default) मान 1 पिक्सल होता है।

Cellspacing = "number of pixels or %"

इस एट्रीब्यूट का उपयोग एक सारणी के विभिन्न कोशों (cells) के मध्य पिक्सल अथवा सारणी की कुल चौडाई के प्रतिशत के रूप में खाली स्थान अथवा दूरी तय करने के काम में लिया जाता है। इस एट्रीब्यूट का डिफाल्ट (default) मान 2 पिक्सल होता है।

frame = "void|above|below|hsides|vsides|lhs|rhs|box|border"

यह एट्रीब्यूट वेब पेज ब्राउजर को यह बताता है कि सारणी के चारों ओर कहाँ पर बार्डर प्रदर्शित करना है।

void = सारणी के चारों ओर फ्रेम (frame) नहीं आता है।

above = केवल सबसे ऊपर (top) का बार्डर (border)
below = केवल सबसे नीचे (bottom) का बार्डर (border)
hside = सबसे ऊपर (top) तथा सबसे नीचे (bottom) का बार्डर (border)
vside = दाँये तथा बाँये साइड का बार्डर (border)
lhs = बाँये हाथ (Left Hand Side) की तरफ का बार्डर (border)
rhs = दाँये हाथ (Right Hand Side) की तरफ का बार्डर (border)
box = टेबल के चारों ओर का बार्डर (border)
border = सभी तरफ का बार्डर (border)

Rules = "all|cols|grups|none|rows|"

यह एट्रीब्यूट यह तय करता है कि किसी सारणी में कोश को अलग करने वाली लाईन कहाँ पर बनानी है।

all = सभी कोशों (cells) के मध्य
cols = प्रत्येक पंक्ति के मध्य
group = लॉजिकल (logical groups) जैसे TBody तथा HEAD के मध्य
none = कोई लाईन नहीं
rows = प्रत्येक पंक्ति के मध्य

width = "number or percentaged %"

इस एट्रीब्यूट का उपयोग सारणी की चौड़ाई पिक्सल अथवा पृष्ठ की कुल चौड़ाई के प्रतिशत के रूप में तय करने के काम में लिया जाता है।

4.3 <TR> टेग (<TR> Tag)

इस टेग का उपयोग एक सारणी में एक पंक्ति (row) को परिभाषित करने के लिए किया जाता है। इस टेग का उपयोग सदैव <table> तथा </table> टेग युग्म के मध्य किया जाता है।

Syntax–

<TR> ....<–! पंक्ति का प्रारम्भ >
</TR>.................. <–! पंक्ति का अंत>

4.3.1 एट्रीब्यूट (Attribute)

align = "left|center|right|justify|char"

इस एट्रीब्यूट का उपयोग एक कोश में उपस्थित सूचनाओं के क्षैतिज एलाइनमेंट (horizontal alignment) के लिए किया जाता है।

left = सूचनाएं बाँये (left) हाथ की तरफ एलायन (align) होती हैं।
center = सूचनाएं सेल के मध्य (center) में एलायन (align) होती हैं।
right = सूचनाएं दाँये (right) हाथ की तरफ एलायन (align) होती हैं।
justify = सूचनाएं जस्टिफाइड (justified) होती हैं।

Charoff = "Length"

यह एट्रीब्यूट प्रत्येक लाइन के प्रथम अलायनमेंट करेक्ट (aligement Character) से दूरी तय करता है।

Char = "Char"

यह एट्रीब्यूट एक करेक्टर (Character) को परिभाषित करता है। जिसके सापेक्ष कोश के कन्टेंट एलायन किये जाते है। इस एट्रीब्यूट का डिफाल्ट (default) मान दशमलव बिन्दु (.) होता है।

Valign = "top/middle/bottom/baseline"

यह एट्रीब्यूट सेल के कन्टेंट का उर्ध्वाधर (Vertical) एलाइनमेंट (alignment) परिभाषित करने के काम आता है।

4.4 <TD> टेग (<TD> Tag)

इस टेग का उपयोग एक सारणी की किसी पंक्ति में स्तम्भ / सारणी डाटा बनाने के लिए किया जाता है। इस टेग का उपयोग <TR> तथा </TR> टेग युग्म के मध्य में किया जाता है एक <TR> तथा </TR> टेग युग्म के मध्य जितनी संख्या में <TD> तथा </TD> टेग युग्म उपस्थित होते हैं। वह पंक्ति (Row) उतने स्तम्भों (Column) में विभिक्त हो जाती है।

4.4.1 एट्रीब्यूट (Attribute)

Align = "Left/Right/Center"

इस एट्रीब्यूट का उपयोग एक कोश में उपस्थित सूचनाओं को क्षैतिज एलाइनमेंट (Horizontal alignemnt) देने में किया जाता है। इस टेग का डिफाल्ट (Default) मान left होता है।

Charoff = "Number"

यह एट्रीब्यूट प्रत्येक लाइन में प्रथम एलायन में करेक्टर से दूरी तय करता है।

Character = "Character"

यह एट्रीब्यूट एक कर्रक्टर (Character) को निश्चित करता है। जिसके सापेक्ष सेल के कन्टेट एलायन (align) किये जाते हैं। इस एट्रीब्यूट का डिफालट मान दशमलव बिन्दु (.) होता है।

Header = "id references"

इस एट्रीब्यूट हेडर सेलस् (Headercell) की list को प्रदर्शित करता है जो वर्तमान डाटा सेल की हेडर से सम्बन्धित सूचना उपलब्ध कराता है। इस एट्रीब्यूट का मान एक खाली स्थान (space) से पृथक सेलस् के नामो की लिस्ट होती है।

Colspan = "Number"
यह एट्रीब्यूट यह प्रदर्शित करता है कि वर्तमान सेल का फैलाव (span) कितने स्तम्भ तक होगा। इस एट्रीब्यूट का डिफालट (defaut) मान "1" होता है।

Rowspan = "Number"
यह एट्रीब्यूट यह तय करता है कि वर्तमान सेल का फैलाव (span) कितनी पक्तियों तक होगा। इस एट्रीब्यूट का डिफालट (default) मान "1" होता है। Rowspan = 0 का अर्थ है कि सेल का फैलाव (Cell span) वर्तमान पक्ति से अंतिम पंक्ति तक होगा।

Scope = "row/col/rowgroup/colgroup"
यह एट्रीब्यूट यह तय करता है कि वर्तमान हेडर सेल (Leader Cell) किन डाटा सेलस् (Data Cells) की हेडर की सूचना उपलब्ध करायेगा।

Row = वर्तमान सेल की पंक्ति को छोड़कर अन्य सभी पंक्तियों (Rows) से सम्बंधित हेडर की सूचना (Header Information) को वर्तमान सेल उपलब्ध कराता है।
Col = वर्तमान सेल के स्तम्भ (column) को छोड़कर अन्य स्तम्भों (Rows) से सम्बंधित हेडर की सूचना (Header Information) को वर्तमान सेल उपलब्ध कराता है।
Rowgroup = वर्तमान कोश (cell) की पंक्ति को छोड़कर अन्य सभी पंक्ति समुह (Rowgroup) से सम्बन्धित हेडर की सूचना (Header Information) को वर्तमान सेल उपलब्ध कराता है।
Colgroup = वर्तमान कोश (Cell) के स्तम्भ (Column) को छोड़कर अन्य सभी स्तम्भ समुह (Column group) से सम्बन्धित शीर्षक सूचना (Header Information) को वर्तमान कोश (Cell) उपलब्ध कराता है।

4.5 <TH> टेग (<TH> Tag)

<TH> टेग एक सारणी के शीर्षक कोश (Header Cell) को परिभाषित करता है। <TH> टेग के एट्रीब्यूट <TD> टेग के एट्रीब्यूट के समान होते हैं।

Syntax–
<TH>........</TH>

4.6 <Caption> टेग (Caption Tag)

<Caption> टेग एक सारणी के कन्टेट (Content) तथा उद्देश्य का संक्षिप्त सार (Brief Summary) उपलब्ध कराता है।

Syntax–
<Caption> Brief Summary of Table</Caption>

4.7 <Col> टेग (Col tag)

एक पंक्ति समुह (Column group) के अन्दर पंक्तियो के स्वभाव / लक्षण (properties) को तय करता है।

4.7.1 एट्रीब्यूट (Attribute)

Align = "left/right/center"
यह स्तम्भ (Colum) के कोशों में उपस्थित टेक्स्ट (Text) का एलायनमेंट (alignment) करता है।

Char = "Character"
यह एट्रीब्यूट एक करेक्ट को निश्चित करता है जिसके सापेक्ष सेल के कन्टेट सलाया (align) किये जाते है। इस एट्रीब्यूट का डिफाल्ट मान दशमलव बिन्दु (.) होता है।

Charoff = "Number"
यह एट्रीब्यूट प्रत्येक लाइन में प्रथम एलायनमेंट करेक्टर (alignment character) से दूरी को तय करता है।

Span = "Number"
यह एट्रीब्यूट यह तय करता है कि <Col> एलीमेंट (element) का फैलाव कितने स्तम्भों (Colums) तक है।

Valign = "top/middle/bottom/baseline"
यह स्तम्भ के कोश में उपस्थित टेक्स्ट (Text) के उर्ध्वाधर एलायमेंट (Vertical alignment) को तय करता है।

4.8 <Colgroup> टेग (<Colgroup> Tag)

Colgroup tag स्तम्भो के समुह को परिभाषित करता है।
इस टेग के एट्रीब्यूट <Col> टेग के एट्रीब्यूट के समान होते हैं।

4.9 <THEAD> टेग (<THEAD> Tag)

<THEAD> .... </THEAD>

<THEAD> टेग एक सारणी के शीर्ष को परिभाषित करता है।

<THEAD> तथा </THEAD> टेग युग्म के मध्य कम से कम एक <TR>

तथा </TR> टेग युग्म होना चाहिए।

4.9.1 एट्रीब्यूट (Attribute)

Align = "left/right/center"
एक स्तम्भ (Colum) के कोशों में उपस्थित टेक्स्ट (Text) का एलायनमेंट (alignment) तक करता है।

Char = "Character"
एक एट्रीब्यूट एक करेक्टर को निश्चित करता है जिसके सापेक्ष सेल के कन्टेट सलाया (align) किये जाते हैं। इस एट्रीब्यूट का डिफाल्ट मान दशमलव बिन्दु (.) होता है।

Charoff = "Number"
यह एट्रीब्यूट प्रत्येक लाइन में प्रथम अलायन्यमेंट करेक्ट (alignment character) से दूरी को तय करता है।

Valign = "top/middle/bottom/baseline"
एक स्तम्भ के कोश में उपस्थित टेक्स्ट (Text) के उर्ध्वाधर एलायमेंट (Vertical alignment) को तय करता है।

4.10 <TBODY> टेग (<TBODY> Tag)

<TBODY> टेग एक पंक्ति या पंक्तियों के समुह को एक सारणी की Body के रूप में परिभाषित करता है। <TBODY> तथा </TBODY> टेग युग्म के मध्य कम से कम एक <TR> तथा </TR> युग्म होना चाहिए।

<TBODY> टेग के एट्रीब्यूट <THEAD> टेग के समान होते हैं।

4.11 <TFOOT> टेग (<TFOOT> Tag)

<TFOOT> टेग सारणी के फुट (Foot) को परिभाषित करता है। यह टेग सारणी की पंक्तियों से सम्बन्धित सूचना रखता है।

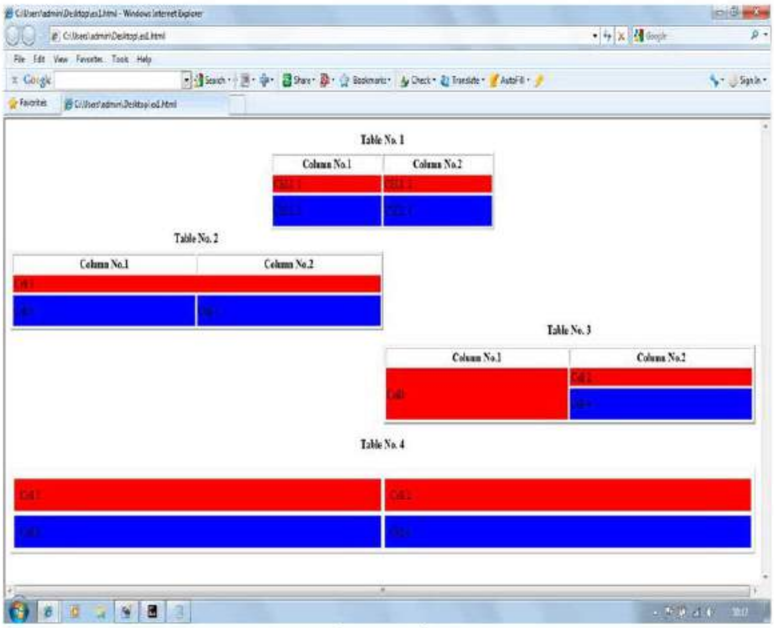
<TFOOT> टेग के एट्रीब्यूट </THEAD> व <TBODY> टेग के समान होते हैं।

4.12 उदाहरण (Examples)

```
<HTML>

<TABLE border=1 align=center width= 30%>
 <THEAD>
  <CAPTION align= Centre><B>Table No. 1</B>
  <TH align=Center><B>Column No.1</B></TH>
  <TH align=Center><B>Column No.2</B></TH>
 </THEAD>
  <TR bgcolor=RED height=20>
<TD> CELL 1 </TD>
    <TD> CELL 2 </TD>
  </TR>
  <TR bgcolor =Blue height=40>
<TD> CELL 3 </TD>
    <TD> CELL 4 </TD>
  </TR>
</TABLE>

<TABLE border=2 align=Left width= 50% >
 <THEAD>
  <CAPTION align= Centre><B>Table No. 2</B>
  <TH align=Center><B>Column No.1</B></TH>
  <TH align=Center><B>Column No.2</B></TH>
 </THEAD>
 <TR bgcolor=RED  height=20>
   <TD Colspan = 2 > Cell 1 </TD>
 </TR>
 <TR  bgcolor =Blue  height=40>
   <TD> Cell 3 </TD>
   <TD> Cell 4 </TD>
 </TR>
</TABLE>
<BR><BR><BR><BR><BR><BR>

<TABLE border=3 align=Right width= 50% >
 <THEAD>
  <CAPTION align= Centre><B>Table No. 3</B>
  <TH align=Center><B>Column No.1</B></TH>
  <TH align=Center><B>Column No.2</B></TH>
 </THEAD>
 <TR bgcolor=RED  height=20>
  <TD  ROWSPAN = 2>Cell1 </TD>
  <TD> Cell 2</TD>
 </TR>
 <TR  bgcolor =Blue  height=40>
  <TD> Cell 4 </TD>
 </TR>
```

```
</TABLE>

<BR><BR><BR><BR><BR><BR><BR>
<TABLE border=1 align=Right width= 100% cellspacing=5 Cellpadding=10>
 <THEAD>
  <TR><CAPTION align= Centre><B>Table No. 4</B></CAPTION></TR>
 </THEAD>
 <TR bgcolor=RED height=20>
  <TD>Cell 1 </TD>
  <TD> Cell 2</TD>
 </TR>
 <TR bgcolor =Blue height=40>
  <TD> Cell 3 </TD>
  <TD> Cell 4 </TD>
 </TR>
</TABLE>

</HTML>
```

उपरोक्त उदाहरण का इंटरनेट एक्सपलोरर ब्राउजर में आउटपुट चित्र 4.1 में दिखाया गया है -



चित्र 4.1 टेबल

4.13 फ्रेम (Frames)

फ्रेम ब्राउजर विंडो (Browser Window) को छोटी सब बिंडोज (Sub window) में विभिक्त करने का एक तरीका है। प्रत्येक सब विंडो (Sub window) एक अलग HTML डाक्यूमेंट को प्रदर्शित कर सकती है।

किसी वेब पेज में फ्रेम (Frame) बनाने के लिये मुख्यतः निम्न टेगस का उपयोग किया जाता है।

Frame – एक फ्रेम बनाने के लिए।
Framesct – एक फ्रेम की संरचना या अन्य फ्रेम समुध्ययों की संरचना तय करता है।
Noframe – वेब पेज के Content प्रदर्शित किये जाते हैं यदि वेब ब्राउजर फ्रेम property को support नहीं करते है।

4.14 <Frameset> टेग (<Frameset> Tag)

जब एक वेबपेज फ्रेमस (Frames) में विभाजित किया जाता है तो उन सभी फ्रेमस को एक टॉप लेवल फ्रेमसेट (Top level Frameset) में रखा जाता है।

Syntax–

```
<FRAMESET>
<FRAME>      </FRAME>
<FRAME>      </FRAME>
</FRAMESET>
```

4.14.1 एट्रीब्युट (Attribute)

Cols ="फ्रेम की लम्बाई की लिस्ट, % or * "

यह एट्रीब्युट एक फ्रेमसेट में स्तम्भो की संख्या तथा उनकी साइज को तय करता है। फ्रेम सेट में स्तम्भो की संख्या लिस्ट में उपस्थित मानों की संख्या पर निर्भर करता है तथा प्रत्येक स्तम्भ का साइज लिस्ट में उपस्थित मान (Value) के बराबर होता हैं।

Rows = " लिस्ट no. % or * "

यह एट्रीब्युट एक फ्रेम सेट में पक्तियों की संख्या तथा उनके साइज को तय करता है फ्रेम सेट में पक्तियों की संख्या लिस्ट में उपस्थित मानों की संख्या पर निर्भर करता है तथा प्रत्येक का साइज उस मान पर निर्भर करता है।

4.15 <Frame> टेग (</Frame> Tag)

यह Frameset में एक फ्रेम को परिभाषित करता है।

4.15.1 एट्रीब्युट (Attribute)

frameborder="1|0"

इस एट्रीब्यूट का उपयोग किसी फ्रेम के चारों ओर के बाह्य किनारों (Border) को दिखाने के लिए किया जाता है। यदि frameborder = 0 है तो फ्रेम के चारो ओर का बार्डर हट जाता है।यदि frameborder = 1 है तो फ्रेम के चारों ओर का बार्डर दिखायी देगा ।

marginwidth="*number*"

इस एट्रीब्यूट का उपयोग किसी फ्रेम के left व right margin को परिभाषित करने में किया जाता है।

marginheight="*number*"

इस एट्रीब्यूट का उपयोग किसी फ्रेम के top व bottom margin को परिभाषित करने में किया जाता है।

name="*text*"

इस एट्रीब्यूट के द्वारा फ्रेम का नाम परिभाषित किया जाता है।

noresize="noresize"

यदि noresize="noresize" है तो फ्रेम के साइज को परिवर्तित नहीं किया जा सकता है।

scrolling="yes|no|auto"

इस एट्रीब्यूट का उपयोग किसी फ्रेम में स्क्रोलबार (Scrollbar) प्रदर्शित करने के लिए किया जाता है।यदि scrolling = yes है तो फ्रेम में स्क्रोलबार (Scrollbar) प्रदर्शित होंगी ।यदि scrolling = no है तो फ्रेम में स्क्रोलबार (Scrollbar) प्रदर्शित नहीं होंगीं ।यदि scrolling = auto है तो फ्रेम में आवश्यकता पड़ने पर स्क्रोलबार (Scrollbar) प्रदर्शित होंगीं ।

src="*URL*"

इस एट्रीब्यूट का उपयोग किसी फ्रेम के अंदर खुलने वाले आलेख (document) के पते (address) को परिभाषित करने के लिए किया जाता है।

4.16 <Noframe> टेग (<Noframe> Tag)

```
<noframes>
Sorry, your browser does not handle frames!
</noframes>
```

किसी वेब पेज में < Noframe> टेग का उपयोग उन वेब ब्राउजर को मदद करने के लिए किया जाता है जो फ्रेम टेग का सर्मथन (Support) नहीं करते है।< Noframe> </ Noframe> टेग के मध्य वेब पेज का फ्रेम टेग अनआधारित कोड रखा जाता है।

4.17 उदाहरण (Examples)

main.html का HTML कोड निम्न है–

```
<html>
<frameset rows="20%,80%">
 <frame src="Top.html" ></frame>
 <frame src="Bottom.html"></frame>
</frameset>
</html>
```

उपरोक्त उदाहरण में हमने वेब पेज को top तथा bottm नाम के दो पंक्ति फ्रेमों में विभक्त किया है। top नाम के पंक्ति फ्रेम में top.html वेब पेज खुलेगा तथा bottom नाम के पंक्ति फ्रेम में bottom.html वेब पेज खुलेगा। top.html तथा bottom.html का HTML कोड निम्न है–

top.html का HTML कोड निम्न है–

```
<HTML>
<center><H1>Board of Secondary Education,Ajmer</H1></Center>
</HTML>
```

bottom.html का HTML कोड निम्न है–

```
<html>
<frameset cols="20%,80%">
 <frame name=left src="left.html" ></frame>
 <frame name=right src="http://rajeduboard.nic.in/" ></frame>
</frameset>
</html>
```

bottom.html में हमने वेब पेज को left तथा right नाम के दो स्तम्भ फ्रेमों में विभक्त किया है। left नाम के स्तम्भ फ्रेम में left.html वेब पेज खुलेगा तथा right नाम के स्तम्भ फ्रेम में राजस्थान बोर्ड का वेब पेज खुलेगा। left.html का HTML कोड निम्न है–

left.html का HTML कोड निम्न है–

```
<HTML>
<LI><A href="http://rajeduboard.nic.in" target=right />HOME</A></LI>
<LI><A href="http://cbse.nic.in" target=right />CBSE</A></LI>
</HTML>
```

left.html में हमने दो हाइपर लिंक सम्मिलित किये है इस स्थिति में, ब्राह लिंक के लिए URL "http://rajeduboard.nic.in" और "http://cbse.nic.in" है। दोनों हाइपर लिंक का target right फ्रेम है।

उपरोक्त उदाहरण का इंटरनेट एक्सपलोरर ब्राउजर में आउटपुट चित्र 4.2 में दिखाया गया है -

चित्र 4.1 फ्रेम

महत्वपूर्ण बिन्दु

1. किसी वेब पेज में पंक्ति / सेल (Row/Cell) मेट्रिक्स (Matrix) बनाने में लिए <Table> टेग का उपयोग किया जाता है। सारणी के कोश में विभिन्न प्रकार के आब्जेक्ट (object) जैसे– संख्या (Numbers), टेक्स्ट (Text), चित्र (Image), अथवा एंकर (Anchor) आदि को रखा जा सकता है। सारणियों से सम्बन्धित सभी टेगस् को <Table> **तथा** </Table> टेग युग्म के मध्य में उपयोग में लिया जाता है।
2. <TR> टेग इस टेग का उपयोग एक सारणी में एक पंक्ति (row) को परिभाषित करने के लिए किया जाता है।
3. <TD> टेग का उपयोग एक सारणी की किसी पंक्ति में स्तम्भ / सारणी डाटा बनाने के लिए किया जाता है। इस टेग का उपयोग <TR> तथा </TR> टेग युग्म के मध्य में किया जाता है एक <TR> तथा </TR> टेग युग्म के मध्य जितनी संख्या में <TD> तथा </TD>

टेग युग्म उपस्थित होते हैं। वह पंक्ति (Row) उतने स्तम्भों (Column) में विभिक्त हो जाती है।

4. <TH> टेग एक सारणी का शीर्षक कोश (Header Cell) को परिभाषित करता है।
5. <Caption> टेग एक सारणी के कन्टेट (Content) तथा उद्देश्य का संक्षिप्त सार (Brief Summary) उपलब्ध कराता है।
6. <Col> टेग एक पंक्ति समुह (Column group) के अन्दर पक्तियो के स्वभाव / लक्षण (properties) को तय करता है।
7. <THEAD> टेग एक सारणी के शीर्ष को परिभाषित करता है। <THEAD> तथा </THEAD> टेग युग्म के मध्य कम से कम एक <TR> तथा </TR> टेग युग्म होना चाहिए।
8. <TBODY> टेग एक पंक्ति या पंक्तियों के समुह को एक सारणी की Body के रूप में परिभाषित करता है। <TBODY> तथा </TBODY> टेग युग्म के मध्य कम से कम एक <TR> तथा </TR> युग्म होना चाहिए।
9. <TFOOT> टेग सारणी के फुट (Foot) को परिभाषित करता है। यह टेग सारणी की पंक्तियों से सम्बन्धित सूचना रखता है।
10. फ्रेम ब्राउजर विंडो (Browser Window) को छोटी सब बिंडोज (Sub window) में विभिक्त करने का एक तरीका है। प्रत्येक सब विंडो (Sub window) एक अलग HTML डाक्यूमेंट को प्रदर्शित कर सकती है।
11. जब एक वेबपेज हो फ्रेमस (Frames) में विभाजित किया जाता है तो उन सभी फ्रेमस को एक टॉप लेवल फ्रेमसेट (Top level Frameset) में रखा जाता है।
12. फ्रेम ब्राउजर विंडो (Browser Window) को छोटी सब बिंडोज (Sub window) में विभिक्त करने का एक तरीका है।
13. जब एक वेबपेज हो फ्रेमस (Frames) में विभाजित किया जाता है तो उन सभी फ्रेमस को एक टॉप लेवल फ्रेमसेट (Top level Frameset) में रखा जाता है।
14. किसी वेब पेज में < Noframe> टेग का उपयोग उन वेब ब्राउजर को मदद करने के लिए किया जाता है जो फ्रेम टेग का सर्मथन (Support) नहीं करते हैं।

अभ्यासार्थ प्रश्न

बहुचयनात्मक प्रश्न

1. एक टेबल में शीर्षक कोश को (Head Cell) परिभाषित करने के लिए किस टेग युग्म का उपयोग किया जाता है –

   (अ) <TR>..</TR> (ब) <TD>..</TD>

   (स) <TH>..</TH> (द) <THEAD>..</THEAD>

2. एक टेबल में कन्टेट (Content) तथा उद्देश्य का संक्षिप्त सार परिभाषित करने के लिए किस टेग युग्म का उपयोग किया जाता है –
   (अ) <CAPTION>..</CAPTION> (ब) <THEAD>..</THEAD>
   (स) <TBODY>..</TBODY> (द) <TFOOT>..</TFOOT>
3. एक टेबल में विभिन्न कोशों (cells) के मध्य पिक्सल अथवा सारणी की कुल चौड़ाई के प्रतिशत के रूप में खाली स्थान अथवा दूरी तय करने के लिए किस एट्रीब्यूट का उपयोग किया जाता है –
   (अ) cellpadding (ब) cellspacing
   (स) frame (द) valign
4. एक टेबल में TD टेग में colspan एट्रीब्यूट का मान 3 है तो कोश का फैलाव होगा –
   (अ) 3 स्तम्भों तक (ब) 3 पक्तियों तक
   (स) 3 टेबल तक (द) उपरोक्त में से कोई नहीं
5. किसी वेब पेज को तीन पक्तियों में विभक्त करने के लिए निम्नाकित में से किस एट्रीब्यूट का उपयोग करेगें –
   (अ) col="20%,60%,20%" (ब) row="20%,60%,20%"
   (स) rows="20%,60%,20%" (द) cols="20%,60%,20%"

लघुत्तरात्मक प्रश्न

1. <Noframe> टेग की क्या आवश्यकता है?
2. फ्रेम (<FRAME>) टेग के scrolling एट्रीब्यूट के कौन से मान संभव होते हैं।
3. <Col> टेग के उपयोग को समझाइए ।
4. <TH> टेग के उपयोग को समझाइए ।
5. <FRAMESET> टेग के उपयोग को समझाइए ।
6. <THEAD> तथा <TFOOT> टेग के उपयोग को समझाइए ।

निबन्धात्मक प्रश्न –

1. फ्रेम (frame) टेग की विवेचना कीजिए ।
2. टेबल (<Table>) टेग के विभिन्न एट्रीब्यूटों पर संक्षिप्त टिप्पणी लिखिए ।
3. <TR> टेग के विभिन्न एट्रीब्यूटों पर संक्षिप्त टिप्पणी लिखिए ।
4. <TD> टेग के विभिन्न एट्रीब्यूटों पर संक्षिप्त टिप्पणी लिखिए ।

उत्तरमाला

1 (स) 2 (अ) 3 (ब) 4 (अ) 5 (द)

5

डोक्यूमेंट आब्जेक्ट मॉडल (Document Obect Model)

5.1 डोक्यूमेंट आब्जेक्ट मॉडल (Document Obect Model)

DOM एक W3C के द्वारा दिया गया मानक है। XML तथा एच.टी.एम.एल. जैसे डोक्यूमेंट के लिए DOM एक मानक को परिभाषित करता है। DOM एक प्लेटफार्म एवं लैग्वेंज न्यूट्रल इण्टरफैस है जो एक्स.एम.एल. डोक्यूमेंट को सरल रूप से एक्सेस (access) करने की अनुमति देता है।

DOM को तीन विभिन्न स्तरों पर विभाजित किया जाता है –

a) Core DOM (कोर डोम) – किसी भी संरचनात्मक डोक्यूमेंट के लिए एक स्तरीय मॉडल है।

b) XML DOM (एक्स.एम.एल. डोम) – एक्स.एम.एल. डोक्यूमेंट के लिए स्तरीय मॉडल है।

c) HTML DOM (एच.टी.एम.एल. डोम) – एच.टी.एम.एल. डोक्यूमेंट के लिए स्तरीय मॉडल है। DOM डोक्यूमेंटस तत्वों के विषय वस्तु और सामग्री को परिभाषित करता है तथा उन तक पहुँचने की विधियाँ (इण्टरफैस) को भी परिभाषित करता है।

5.1.2 एच.टी.एम.एल. डोम (HTML DOM)

एच.टी.एम.एल. DOM सभी एच.टी.एम.एल. ऐलीमेंट की विशेषताओं और आब्जेक्ट को परिभाषित करता है तथा उन तक पहुँचने की विधियाँ (इण्टरफेस) को भी को भी परिभाषित करता है।

5.1.3 एक्स.एम.एल. डोम (XML DOM)

एक्स.एम.एल. के लिए यह एक स्टैण्डर्ड आब्जेक्ट मॉडल है। यह एक्स.एम.एल. के लिए एक स्टैण्डर्ड प्रोग्रामिंग इण्टरफेस प्रदान करता है। यह एक **W3C** के द्वारा दिया गया मानक है।

5.2 केसकेडिड स्टाइल शीट (cascaded style sheet)

एच.टी.एम.एल. केवल सरल मार्कअप भाषा है, यदि आप कुछ अलग वेबशीट में करना चाहते हैं तो स्टाइल शीट का प्रयोग किया जाता है। स्टाइल शीट आकर्षक पेज बनाने के लिए एक बेहतर उपागम है। स्टाइल शीट फॉरमेटिंग स्टाइल का वह समूह है जिसे हम वेब पेज पर प्रयोग कर सकते हैं। एच.टी.एम.एल. स्टाइल शीट का उपयोग करने का सबसे बड़ा फायदा वेब पेजों को आकर्षक बनाना है।

5.2.1 स्टाइल शीट का सिंटेक्स (Syntax of Style Sheet)

स्टाइल शीट के syntax के दो मुख्य भाग होते हैं – पहला सलेक्टर और दूसरा डिक्लरेशन।

```
Selector        Declaration                 Declaration
h1              {color  :  blue  ;  font – size : 12 px ;}
                  ↑          ↑          ↑              ↑
                property   value   property        value
```

(a)सलेक्टर (Selector) – सलेक्टर एक एच.टी.एम.एल. का ऐलीमेंट है जिसे हम स्टाइल करना चाहते हैं।

(b)डिक्लेरेशन (Declaration) – colon डिक्लेरेशन को दो भागों में विभाजित करता है। colon से पहले वाले भाग में property तथा दूसरे वाले भाग में उसकी value दी जाती है।

उदाहरण के लिए :–

स्टाइल शीट के डिक्लरेशन का अंत हमेशा semi colon(;) के द्वारा होता है। डिक्लरेशन समूह curly bracket से घिरा हुआ होता है –

p {color : red ; text – align : center ;}

यदि स्टाइल शीट को ज्यादा पढ़ने योग्य बनाना है तो हम प्रत्येक डिक्लरेशन को अलग–अलग पंक्ति में लिख सकते है – जैसे –

```
p
{
color : red ;
text – align : center ;
}
```

टाइल रूल उदाहरण – किसी भी दो प्रकार की स्टाइल प्रोपर्टीज को समूह के रूप में लागू करने को समूह स्टाइल रूल कहते हैं। दोनों को अलग प्रदर्शित करने के लिए सलेक्टर के बीच में कोमा लगाया जाता है।

```
<HTML>
    <HEAD>
        <STYLE>
            H1, H2 {color : red : font-family : Arial}
        </STYLE>
    </HEAD>
    <BODY>
        <H1> This is the H1 element </H1>
        <H2> This is the H2 element </H2>
        <H3> This is the H3 element with its default
            style as displayed in the browser </H3>
    </BODY>
</HTML>
```

5.2.2 एच.टी.एम.एल. डोक्यूमेंट को एक स्टाइल शीट से जोड़ना (Linking stylesheet with HTML Document)

एच.टी.एम.एल. डोक्यूमेंट फाइल से स्टाइल शीट जोड़ने के चार तरीके हैं :–

1. लिंक (Link) :– एच.टी.एम.एल. फाइल को अन्य फाइल के स्टाइल शीट से लिंक कर सकते हैं।
2. एम्बेडिंग (Embedding) :– एच.टी.एम.एल. फाइल के अन्दर स्टाइल शीट को एम्बेडेड कर सकते हैं।
3. इन लाइन (Inline) :– इसका उपयोग सामान्यतः टेग लेवल पर किया जाता है।
4. इम्पोर्ट (Import) :– एच.टी.एम.एल. फाइल में एक स्टाइल शीट का आयात करने के लिए @ इम्पोर्ट की वर्ड का उपयोग कर सकते हैं।

5.2.2.1 स्टाइल शीट को लिंक करना (Linking Stylesheet)

एक फाइल में स्टाइल शीट को बना कर तथा इसी स्टाइल शीट को अपने वेबसाइट के सभी पेजों पर लागू कर सकते हैं। CSS जो कि दूसरी फाइल में स्टोर है उसको एचटीएमएल फाइल में लिंक करते हैं तथा इसे निम्नलिखित उदाहरण से समझ सकते हैं-

5.2.2.2 स्टाइल शीट को एमबेडेड (embedded) करना (Embedding Style Sheet)

इस विधि में स्टाइल शीट को अलग फाइल में स्टोर करने की जरूरत नहीं है, आप स्टाइल शीट को प्रत्येक एच.टी.एम.एल. फाइल के अन्दर ही एम्बेडेड कर सकते हैं। एच.टी.एम.एल. फाइल में एक स्टाइल शीट को एम्बेडेड करने के लिए <STYLE> कंटेनर का उपयोग करते हैं। इसे निम्नलिखित उदाहरण से समझ सकते हैं-

```
<HTML
    <HEAD>
```

```
            <STYLE>
            P {font – family : Arial}
            </STYLE>
        </HEAD>
    <BODY>
    ........
    ........
    </BODY>
```

<STYLE> is included within the <HEAD> tag.

5.2.2.3 स्टाइल इन लाइन को परिभाषित करना (Defining style inline)

किसी विशेष टेग के लिए स्टाइल को ओवरराइड करने के लिए भी आप इन लाइन स्टाइल का उपयोग होता है। इन लाइन स्टाइल को परिभाषित करने के लिए आप जिस टेग का स्टाइल बदलना चाहते हैं, उस टेग में स्टाइल गुण को जोड़े और इसके मूल्य को स्ट्रिंग पर सेट करें जो स्टाइल परिभाषा का प्रतिनिधित्व करता है।

उदाहरण :– H1 टेग पर स्टाइल को प्रयोग किया गया है –

```
<HTML>
    <BODY>
    <H1 Style = "Color : linegreen"> This is a style applied to an H1
    element </H1>
    <H1> This is the default display of an H1 element </H1>
    </BODY>
</HTML>
```

उपरोक्त उदाहरण में सबसे पहले H1 ऐलीमेंट की कलर प्रोपर्टी लाइम ग्रीन कलर में सेट की गयी है और दूसरे में कोई भी स्टाइल एट्रीब्यूट नहीं हैं।

यदि सम्पूर्ण ब्लाक के लिए स्टाइल सेट करना हो तो इन लाइन स्टाइल के साथ <DIV> टेग को भी परिभाषित कर सकते हैं।

उदाहरण के लिए अगर आप एक सम्पूर्ण टेगों के ब्लाकों के रंग को नीला करना चाहते हैं तो आप इन टेगों को <DIV> टेग के साथ प्रयोग में ला सकते हैं।

उदाहरण :–

```
<DIV STYLE="Color : blue">
<H1> This is a heading </H1>
<P> This is a paragraph. It will look blue in the user's browser </P>
```

5.2.2.4 स्टाइल शीट का आयात (Importing Stylesheet)

एचटीएमएल डोक्यूमेंट में स्टाइल शीट को आयात करने के लिए @ इम्पोर्टकी–वर्ड का भी उपयोग कर सकते हैं। इसका syntax निम्नलिखित हैं–

Syntax –

```
<HTML>
    <HEAD>
    <STYLE TYPE="TEXT/CSS">
    @IMPORT URL (The path);
    </STYLE>
    </HEAD>
    <BODY>
    ......................
    </BODY>
</HTML>
```

5.2.3 स्टाईल शीट की प्रोपर्टीज (Properties of Stylesheet)

स्टाइल शीट में 50 से अधिक प्रोपर्टीज तथा वेल्यू को समाहित किया गया है। इस सेक्शन में केवल उन्हीं विशिष्ट प्रोपर्टीज को लिया गया है जो लगातार प्रयोग में आती रहती हैं।

5.2.3.1 बेकग्राउंड (Background)– एक ऐलीमेंट के बैकग्राउंड में परिवर्तन करने के लिए इसका प्रयोग करते है।

(a) बैकग्राउंड कलर
(b) बैकग्राउंड इमेज
(c) बैकग्राउंड रिपीट
(d) बैकग्राउंड पोजीशन

a) बैकग्राउंड कलर (Background Color) – बैकग्राउंड कलर प्रोपर्टी के द्वारा किसी भी ऐलीमेंट के कलर में परिवर्तन किया जा सकता है।

उदाहरण – body {background–color: #boc4de;}

b) बैकग्राउंड इमेज (Background Image) – किसी भी बेकग्राउंड में यदि इमेज का प्रयोग करना चाहते है तो बैकग्राउंड इमेज का प्रयोग करते हैं।

उदाहरण –

body {background–image; url('paper.gif');}

c) **बैकग्राउंड इमेज को होरीजेन्टली तथा वर्टीकली (आड़ा या सीधा) रिपीट करना (Repeating image horizontally and Vertically)**

बैकग्राउंड इमेज को आड़ा या सीधा दोनों ही तरह से बार–बार रिपीट कर सकते हैं। होरीजोन्टली के लिए repeat x का प्रयोग तथा वर्टीकल के लिए repeat y का प्रयोग करते हैं।

d) **बैकग्राउंड पोजीशन (Background Position)**— बैकग्राउंड की निम्न पोजीशन को परिवर्तित कर सकते है –

लेफ्ट सेन्टर
लेफ्ट बॉटम
राइट बॉटम
राइट टॉप
राइट सेन्टर
राइट बॉटम
सेन्टर टॉप
सेन्टर सेन्टर
सेन्टर बॉटम

5.2.3.2 फोन्ट प्रोपर्टीज (Font Properties)

a) फोन्ट फैमिली – फोन्ट को इंगित करता है।

b) फोन्ट साइज – टेक्स्ट के साइज को इंगित करता है।

c) फोन्ट स्टाइल – फोन्ट के स्टाइल को इंगित करता है जैसे – नोरमल, बोल्ड, इटैलिक आदि।

d) फोन्ट वेट (weight) – फोन्ट के भार तथा डार्कनेस को इंगित करता है। इसकी वेल्यू की रेन्ज 100 से 900 तक हो सकती है।

5.2.3.3 टेक्स्ट की प्रोपर्टीज (Text Properties)

a) लेटर स्पेसिंग (Letter Spacing) . प्रत्येक शब्दों के बीच की जगह को सेट करने के लिए लेटर स्पेसिंग का प्रयोग किया जाता है। स्पेस के लिए विभिन्न प्रकार के measurement हो सकते हैं।

जैसे – inches(in), centimeters(cm), millimeter (mm), points (pt) etc.

b) वर्ड स्पेसिंग (Word Spacing) . प्रत्येक अक्षर के बीच की जगह को सेट करने के लिए वर्ड स्पेसिंग का प्रयोग किया जाता है। वर्ड स्पेसिंग के measurements लेटर स्पेसिंग जैसे ही होते हैं।

c) वर्टीकल एलाइन (Vwrtical Align) . बेसलाइन के रिफरेन्स में टेक्स्ट तथा इमेज की पोजीशन को वर्टीकल करने के लिए वर्टीकल एलाइमेंट का प्रयोग करते हैं। जैसे – बेसलाइन, सब, सुपर टॉप, टेक्स्ट–टॉप आदि।

d) टेक्स्ट एलाइन (Text Align) . किसी टेक्स्ट के एलाइन को सेट करता है। जैसे जसटीफाई, सेन्टर आदि।

e) टेक्स्ट इन्टेड (Text Intend) . यह मार्जिन को इंगित करता है। किसी भी ऐलीमेंट की पहली लाइन के टेक्स्ट को इन्टेंड किया जा सकता है।

f) टेक्स्ट ट्रांसफोर्म (Text Transform) .– टेक्स्ट के रूप को परिवर्तित करने के लिए टेक्स्ट ट्रांसफोर्म का प्रयोग किया जा सकता है। जैसे – अपरकेस (capital letter) लोअर केस (small letters) आदि.

g) टेक्स्ट डेकोरेट (Text Decorate) . टेक्स्ट के रूप को आकर्षित बनाने के लिए टेक्स्ट डेकोरेशन का प्रयोग किया जाता है जैसे – ब्लिंक, अण्डरलाइन आदि। फोन्ट तथा टेक्स्ट प्रोपर्टीज का उदाहरण निम्नतिखित है।

टेक्स्ट एलाइन का उदाहरण

```
h1{text-align:center;}
p.date{text-align:right;}
p.main {text-align:justify;}
```

टेक्स्ट डेकोरेट का उदाहरण

```
h1{text-decoration:overline;}
h2{text-decoration:line-through;}
h3{text-decoration:underline;}
h4 {text-decoration:blink;}
```

फोन्ट स्टाइल का उदाहरण

```
p.normal{font-style:normal;}
p.italic{font-style:italic;}
p.oblique{font-style:oblique;}
```

फोन्ट साइज का उदाहरण

```
h1{font-size:40px;}
h2{font-size:30px;}
p {font-size:14px;}
```

5.2.3.4 मार्जिन, बोर्डर तथा पेडिंग प्रोपर्टीज

(Margin, Border and Padding Properties)

सभी प्रोपर्टीज को बोक्स मॉडल के अन्तर्गत उपयोग में लाया जाता है। सबसे पहले इन प्रोपर्टीज को समझने के लिए निम्न चित्र को देख सकते हैं –

(a) **मार्जिन प्रोपर्टीज (Margin Properties)** - मार्जिन वेल्यू की लेन्थ में परिभाषित होते हैं। ब्लॉक ऐलीमेंट का मार्जिन विभिन्न प्रकार से सेट कर सकते हैं। जैसे – मार्जिन –टॉप, मार्जिन– बॉटम, या मार्जिन– लेफ्ट।

(b) **बोर्डर प्रोपर्टीज (Border Properties)** -

(i) **बॉर्डर स्टाइल (Border Style)** - बॉर्डर की दिखावट को सेट करने के लिए बॉर्डर स्टाइल का प्रयोग किया जाता है। इसमें विभिन्न स्टाइल हो सकती है जैसे सोलिड, डबल, ग्रूव आदि।

(ii) **बॉर्डर विथ (Border width)** -बॉर्डर की मोटाई को सेट करने के लिए बॉर्डर विड्थ का प्रयोग किया जाता है। इसमें विभिन्न मोटाई हो सकती है जैसे – बॉर्डर राइट विथ, बॉर्डर बोटम विड्थ, आदि।

(iii) **बॉर्डर कलर (Border Color)** -– बॉर्डर को किसी भी रंग में परिवर्तित किया जा सकता है।

(c) **पेडिंग प्रोपर्टीज (Padding Properties)** - ऐलीमेंट के बॉर्डर तथा content के बीच की जगह को पेडिंग कहा जाता है। चार प्रकार से पेडिंग प्रयोग में लायी जा सकती है। पेडिंग– टॉप, पैंडिग –राइट, पैडिंग– बॉटम, और पैडिंग –लेफ्ट।

5.3. लेयर तथा आई लेयर (Layer and ILayer)

5.3.1 लेयर (LAYER)

लेयर एच.टी.एम.एल. में एक टेग होता है जिसके उपयोग से हम विषयवस्तु के ब्लॉक को वेब पेज के विभिन्न स्थानों पर लगा सकते हैं। विषय वस्तु के ब्लॉकों को एक दूसरे के उपर पारदर्शी तथा अपारदर्शी दोनों ही रूपों में आवश्यकतानुसार लगा सकते है।

लेयर सिंटेक्स (Layer Syntax)

```
<LAYER>
        ID = "layername"
        LEFT = "pixel position"
        TOP = "pixelposition"
        PAGEX = "pagex"
        PAGE = "pagey"
</LAYER>
```

उदाहरण –

```
<LAYER ID = layer1 TOP = 250 LEFT = 50 WIDTH = 200 HEIGHT = 200
BGCOLOR = RED>
<P> Layer 1 </P>
<LAYER ID = layr2 TOP = 350 LEFT = 150 WIDTH = 200 HEIGHT = 200
BGCOLOR = BLUE>
<P> Layer 2 </P>
```

उपरोक्त उदाहरण में दो लेयर हैं पहली वाली लेयर जो नीचे की तरफ उसका रंग लाल है तथा दूसरी लेयर जो उस से ओवरलेप कर रही है उसका रंग नीला है।

आई लेयर (ILAYER)

आई लेयर डोक्यूमेंट के फ्लो के साथ तैरती हुई नजर आती है।

सिंटेक्स (Syntax)

```
<ILAYER>
attributes*
</ILAYER>
```

* यहाँ ILAYER के एट्रीब्यूट तथा लेयर के एट्रीब्यूट्स एक ही प्रकार के होते हैं।

महत्वपूर्ण बिन्दु

1. डोक्यूमेंट आब्जेक्ट मॉडल एच.टी.एम.एल. तथा एक्स.एम.एल. डोक्यूमेंट जैसे फॉरमेट के लिए प्लेटफार्म एवं लैंग्वेंज इण्डिपेन्डेंट स्टैण्डर्ड आब्जेक्ट की तरह कार्य करता है।
2. कासकेडेड स्टाइल शीट डाटा को आकर्षक बनाने के लिए प्रयोग में लायी जाती है।
3. कासकेडेड स्टाइल शीट को चार तरीके से डोक्यूमेंट के साथ जोड़ा जाता है।
4. कासकेडेड स्टाइल शीट के द्वारा हम बाक्स प्रोपर्टी रंग इत्यादि को परिभाषित कर सकते है।

5. कासकेडेड स्टाइल शीट को एच.टी.एम.एल. एवं एक्स.एम.एल. के साथ प्रयोग में ला सकते है।
6. लेयर एच.टी.एम.एल. में एक टेग होता है जिसके उपयोग से हम विषयवस्तु के ब्लॉक को वेब पेज के विभिन्न स्थानों पर लगा सकते हैं। विषय वस्तु के ब्लॉकों को एक दूसरे के ऊपर पारदर्शी तथा अपारदर्शी दोनों ही रूपों में आवश्यकतानुसार लगा सकते हैं।
7. आई लेयर डोक्यूमेंट के फ्लो के साथ तैरती हुई नजर आती हैं।

अभ्यासार्थ प्रश्न

बहुचयनात्मक प्रश्न

1. सी.एस.एस. का पूरा नाम क्या है –
   (अ) cascaded style sheet (ब) computer style sheet
   (स) C language style sheet (द) उपरोक्त में से कोई नहीं
2. DOM है।
   (अ) प्लेटफार्म इण्डिपेडेंट (ब) लैंग्वेंज इण्डिपेडेंट
   (स) इण्टरफेस इण्डिपेडेंट (द) उपरोक्त सभी
3. CSS के अनुसार निम्नलिखित में से कौन सा syntax सही है –
   (अ) body {color : black} (ब) body : color = block
   (स) {body : color + black <body} (द) <body : color : black>
4. सी.एस.एस. की कौन सी प्रोपर्टीज टेक्स्ट साइज को नियंत्रित करती है –
   (अ) text-size (ब) font-style
   (स) text-style (द) font-size
5. निम्नलिखित में से कौन सी बॉक्स प्रोपर्टी नहीं है –
   (अ) मार्जिन प्रोपर्टीज (ब) बार्डर प्रोपर्टीज
   (स) पैडिंग प्रोपर्टीज (द) लेयर प्रोपर्टीज
6. DOM का पूरा नाम है –
   (अ) डोक्यूमेंट आब्जेक्टिव मॉडल (ब) डोक्यूमेंट आब्जेक्ट मॉडल
   (स) डोक्यूमेंट ऑपरेशन मॉडल (द) कोई नहीं
7. निम्नलिखित में से कौन सी प्रोपर्टी बैकग्राउण्ड कलर को परिवर्तित करती है
   (अ) color : (ब) background-color :
   (स) bgcolor : (द) उपरोक्त सभी

लघुत्तरात्मक प्रश्न

1. सी.एस.एस. की क्या आवश्यकता है?
2. सी.एस.एस. बनाने की तरीकों का संक्षेप में वर्णन कीजिए।
3. DOM को परिभाषित कीजिए।

4. बाक्स की प्रोपर्टीज क्या होती है? बताइये।
5. C.S.S. का Syntax लिखिए?
6. C.S.S. को एक वेब पेज से लिंक करने की विधि को बताइये?

निबन्धात्मक प्रश्न –

1. डोक्यूमेंट आब्जेक्ट मॉडल की विवेचना कीजिए ।
2. एच.टी.एम.एल. डोक्यूमेंट को एक स्टाइल शीट से जोड़ने के तरीकों की विवेचना कीजिए ।
3. स्टाईल शीट की प्रोपर्टीज पर निबंध लिखिए।
4. लेयर तथा आई लेयर टेग की विवेचना कीजिए ।

उत्तरमाला

1 (अ) 2 (द) 3 (अ) 4 (द) 5 (द) 6 (ब) 7 (ब)

6

एक्सटेंसिबल मार्कअप लैंग्वेंज
(eXtensible Markup Language)

6.1 परिचय (Introduction)

एक्स.एम.एल., एच.टी.एम.एल. की तरह ही मार्क अप भाषा है। X.M.L. का पूरा नाम एक्सटेक्सिबल मार्क अप लैंग्वेंज (Extensible Markup language) है। X.M.L. को वर्ल्ड वाइड वेब कनसोरटियम (World Wide Web Consortium) (W3C) के द्वारा बनाया गया। ये संस्था संसार भर की वेबसाइट के नियम निर्धारित करती है। एक्स.एम.एल. वास्तव में एस.जी.एम.एल. ही है। एक्स.एम.एल. से मिलने वाली सभी सुविधायें एस.जी.एम.एल. में भी होती हैं। एस.जी.एम.एल. तथ एक्स.एम.एल. केवल एक ही अन्तर है कि एक्स.एम.एल. प्रयोग करने में सरल होती है।

6.2 XML के उद्देश्य (Aim of XML)

1. XML इंटरनेट में प्रयोग के लिए सरल होनी चाहिए।
2. XML ,SGML के अनुकूल होनी चाहिए।
3. XML डोक्यूमेंट पढ़ने में सरल होने चाहिए।
4. XML के द्वारा विभिन्न प्रकार के प्रोग्रामों को लिखा जा सके।

इस प्रकार संक्षेप में कह सकते हैं कि XML बनाने का उद्देश्य एक ऐसी भाषा बनाना था जिसमें प्रोग्राम को लिखना, पढ़ना सरल हो तथा यह भाषा SGML के अनुकूल हो।

6.3 XML की विशेषतायें/लाभ (Advantage of XML)

XML का प्रयोग विस्तृत रूप में हुआ है। कुछ विशेषतायें निम्न हैं :–

a) सरल डाटा विनिमय – XML के द्वारा विभिन्न एप्लीकेशन के मध्य डाटा को स्थान्तरित किया जा सकता है, चाहे एप्लीकेशन किसी भी भाषा के हों।

b) XML डोक्यूमेंट को देखने पर पता लग जाता है कि क्या कार्य किया जा रहा है।

c) स्वयं की भाषा निर्माण – XML अपने विशिष्ट कार्य के लिए स्वयं की मार्क अप लैंग्वेंज का निर्माण करता है। XML भाषा का प्रयोग – बैंक इंटरनेट पेमेन्ट सिस्टम, फाइनेन्शियल एक्सचेंज और अन्य कार्य के लिए किया जाता है।

d) XML, HTML के तरह की मार्कअप लैंग्वेंज है।

e) XML का प्रयोग डाटा के वर्णन के लिए किया जाता है जबकि HTML का उपयोग डाटा को प्रदर्शित करने के लिय होता है।

f) XML टेग को यूजर स्वयं के अनुसार परिभाषित कर सकता है जबकि HTML में ऐसा नहीं कर सकते हैं।

g) XML को W3C के द्वारा निर्मित किया गया है।

6.4 XML डोक्यूमेंट के उदाहरण (Example of XML Document)

XML डोक्यूमेंट की सरल सरंचना को निम्न उदाहरण के द्वारा समझा जा सकता है –

```
<?xml version = "1.0" encoding = "ISO-8859-1"?>
<note>
        <to>ramesh/to>
        <from>monica</from>
        <heading>Reminder</heading>
        <body>Don't forget me this weekend!</body>
</note>
```

पहली पंक्ति XML Declaration की है। इसके अन्दर Xml version (1.0) तथा जिसमें एनकोडिंग (ISO-8859-1 = Lation-1/West Europian Character set) प्रयोग में लायी गयी है।

दूसरी पंक्ति डोक्यूमेंट के (Root element) रूट ऐलीमेंट को वर्णित कर रही है। यहाँ पर note एक रूट ऐलीमेंट हैं।

अगली चार पंक्तियाँ रूट के चार चाइल्ड ऐलीमेंट को वर्णित कर रही है।

(to, from, heading and body) :

<to> ramesh</to>

<from>monica</from>

<heading> Reminder</heading>

<body> don't forget me this weekend!</body>

और अन्तिम पंक्ति रूट ऐलीमेंट के अन्त को वर्णित कर रही है :

</note>

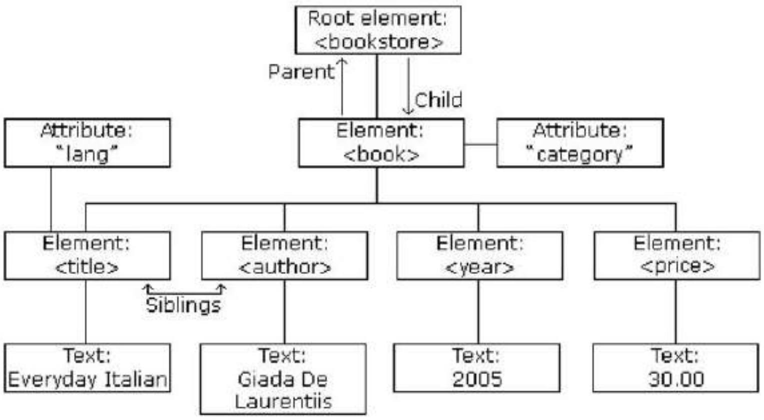
आप अंदाजा लगा सकते हैं कि इस उदाहरण में XML डोक्यूमेंट में एक नोट है जो मोनिका ने रमेश के लिए लिखा है।।

इस प्रकार हम देख सकते है कि XML से डाटा का पूरा वर्णन किया जा सकता है।

6.5 XML डोक्यूमेंट की ट्री संरचना (Tree Structure of XML Document)

XML डोक्यमेंट में Tree की तरह ही Root element (रूट ऐलीमेंट) होता है। ये रूट ऐलीमेंट सभी ऐलीमेंट का जनक होता है। सभी एलीमेंट में पेड की शाखाओं की तरह सब एलीमेंट (Sub element) हो सकते हैं। जिसे हम child element चाइल्ड एलीमेंट कहते हैं।

उदाहरण –



उपरोक्त चित्र में दर्शायी गयी किताब को XML में देखते हैं –

```
<?xml version = "1.0" encoding = "ISO-8859-1"?>
<bookstore>
 <book category = "COOKING">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
 </book>
 <book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J. K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
 </book>
```

```
 <book category="WEB">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
 </book>
</bookstore>
```

उपरोक्त उदाहरण में रूट ऐलीमेंट <bookstore> है। <book> एलीमेंट के चार Children है।

<title>, <author>, <year>, <price>

6.6 XML का ढांचा (Structure of XML)

XML की संरचना में विभिन्न तत्व शामिल होते है। इन सभी तत्वों की विस्तृत संरचना निम्नलिखित है :–

a) कमेन्ट्स (Comments) – कमेन्ट्स की सहायता से किसी भी programme को हम अच्छी तरह से समझ सकते है। XML में comment लिखने का निम्न syntax है –

```
<!— This is a comment —>
Example
<!— god is great —>
```

कभी कभी XML का version दिखाने के लिए निम्नलिखित प्रोसेसिंग इन्सट्रक्शन का प्रयोग किया जाता हैं।

```
<?xml version = "1.0" encoding = "ISO-8859-1"?>
```

b) XML के ऐलीमेन्ट (Elements of XML) – XML के प्रारम्भिक टेग (tag) से लेकर XML के अन्तिम टेग (tag) के मध्य जो भी सूचनायें होती है वह XML का ऐलीमेंट कहलाती है। XML के ऐलीमेंट में दूसरे ऐलीमेंट को समाहित कर सकते है, चाहे वह सरल टेक्सट में हो या दोनों का मिश्रण हो। ऐलीमेंटस में एट्रीब्यूट्स (Attributes) भी हो सकते हैं।

ऐलीमेंट का नाम देने के नियम (Rules for Declaring Name of Element)

XML ऐलीमेन्टस के नाम देने के निम्नलिखित नियम है –

1. ऐलीमेन्ट नाम का निर्माण अक्षरों (letters), संख्या (numbers), (underscore (_), colons (:), hyphens (-) or periods (.) से होता है।
2. ऐलीमेन्ट्स के नाम का प्रारम्भ संख्या से नहीं हो सकता है।
3. ऐलीमेन्टस के नाम xml कीवर्ड नहीं हो सकता है।
4. एलीमेन्ट्स नाम में space को नहीं रख सकते है।

   उदाहरण – from , order ।

c) XML के ऐट्रीब्यूटस (Attributes of XML)

एट्रीब्यूट्स के द्वारा हम ऐलीमेन्ट्स की सूचनाओं को प्राप्त कर सकते हैं। एट्रीब्यूट्स ऐलीमेन्ट्स के प्रारम्भिक टेग से परिभाषित होते हैं। ऐट्रीब्यूट्स हमेशा नाम और इसकी Value के समूह के रूप में होते हैं।

उदाहरण के लिए

Name Value pair

```
<order>
<charge_card card type="credit">
 <card_number> 12345678901234 </card_number>
 <expired_date> 01/03/2009 </expired_date)
</charge_card>
</order>
```

d) XML एन्टिटी रिफरेन्सेज (XML Entity References)

कुछ characters का XML में विशिष्ट अर्थ होता है। XML में पाँच, एन्टिटी रिफरेन्सेज है जिन्हें निम्न तालिका में प्रदर्शित किया गया है।

XML entities references

Entity	Encodes
&	& (ampersand)
<	<(less than)
>	> (greater than)
"	"(quotation mark)
'	'(apostrophe)

उदाहरण के लिए यदि किसी XML ऐलीमेन्ट के अन्दर "<" character को लेंगे तो XML पारसर (parser) त्रुटि उत्पन्न करता है।

जैसे कि –

<message> if salary < [erros (त्रुटि)] 1000 then</message>

उपरोक्त पंक्ति में '<' character एक त्रुटि है इसे निम्न तरीके से ठीक किया जा सकता है –

<message> if salary & lt ; 1000 then </message>

उपरोक्त वाक्य में विशिष्ट character & lt एक entity reference है।

6.7 एन्टिटी डिक्लरेशन (Entity Decleration)– एन्टिटी डिक्लरेशन का syntax निम्नलिखित है –

Syntax –

```
<!ENTITY name "replacement text">
```

उपरोक्त syntax में name की जगह replacement text को लिखा जाएगा।

6.8 एलीमेन्ट डिक्लरेशन (Element Decleration)– XML में एलीमेन्ट डिक्लरेशन के बहुत से तरीके हैं।

एक सरल तरीके का syntax निम्न है –

<!ELEMENT name content model>

प्रस्तुत syntax में नाम को XML के स्टैण्डर्ड नेमिंग रूल के अनुसार लिखा जाएगा।

a) एमप्टी एलीमेंट (Empty Element) –

एम्प्टी एलीमेंट में किसी भी प्रकार की विषय वस्तु नहीं होती है। एम्प्टी एलीमेंट डिक्लेयर करने का syntax निम्न प्रकार है :–

<!ELEMENT element_name EMPTY>

यहां पर element_name यूजर डिफाइन वेलिड ऐलीमेंट नेम है। उदाहरण –

<!ELEMENT book EMPTY>

b) अनरिसटिक्टेड एलीमेंट (Unrestricted element) –

सामान्यतः अनरिस्टिक्टेड एलीमेंट एम्पटी एलीमेंट का उल्टा होता है। अनरिसटिक्टेड एलीमेंट को निम्न प्रकार से डिक्लेयर किया जाता है।

<!ELEMENT element_name ANY>

उदाहरण – <!ELEMENT book ANY>

6.9 एलीमेंट कन्टेंट मॉडल (element content model)

एलीमेंट कन्टेन्ट मॉडल एलीमेंट के अन्दर उपस्थित डाटा के बारे में जानकारी प्रदान करताहै।

(a) एलीमेंट सिक्वेन्स (element sequence) –

यह एलीमेन्ट कन्टेंट मॉडल का सबसे सरल प्रारूप है। सामान्यतः यह सम्भावित एलीमेंट की सूची को दर्शाता है, जहाँ कोमा प्रत्येक एलीमेंट को अलग अलग करता है। सिक्वेंस द्वारा अलग–अलग होती है।

उदाहरण –

<!ELEMENT counting <first, second, third, fourth)>

उपरोक्त उदाहरण में काउन्टिंग एलीमेंट चार भिन्न प्रकार के एलीमेंट को दर्शाता है जिसमें पहले एलीमेंट का दूसरा, दूसरे का तीसरा तथा तीसरे का चौथा एलीमेंट अनुसरण कर रहे हैं।

b) एलीमेंट चॉइस (Element choice) –

एलीमेंट चॉइस एलीमेंट कन्टेंट मॉडल का ही एक प्रकार है। इसमें बहुत से विकल्पों को अलग से इंगित करने के लिए वर्टीकल लाइन (।) का प्रयोग किया जाता है।

उदाहरण 1

```
<!ELEMENT choose (this one / that.one>
```

उदाहरण 2

```
<!ELEMENT choose (this one | that one | the other. one | another. one)>
```

c) कम्बाइंड सीक्वेन्स एण्ड चॉइस (combined sequence and choices) –

आप एलीमेंट के समूह बनाकर कम्बाइंड सीक्वेंन्स तथा च्वाइस को इकट्ठा कर सकते हैं।

उदाहरण :

```
<!ELEMENT lots.of.choice (may be / could be), (this.one, that.one)>
```

6.10 एलीमेन्टस अकरेन्स इंडीकेटरस (Element Occurrence Indicators) –

एलीमेन्टस को विभिन्न प्रकार से इंगित करने के लिए एलीमेन्टस अकरेन्स इन्डीकेटर का प्रयोग किया जाता है, ये तीन प्रकार के हैं –

a) Question Mark (?) - ? अक्षर यह इंगित करता है कि ? अक्षर के साथ लिए गए एक ऐलीमेंट या ऐलीमेंट के ग्रुप को एक बार ले सकते हैं या छोड़ भी सकते हैं।

```
<!ELEMENT Testing (one, two, three?)>
```

b) Asterisk sign (*) – * साइन यह इंगित करता है कि एक ऐलीमेंट या ऐलीमेंट के समूह को छोड़ भी सकते है, शून्य भी प्रदर्शित हो सकता है तथा एक से अधिक बार भी ले सकते है।

```
<!ELEMENT nice (mmm, mmm *)>
```

c) Plus Sign (+) : + साइन यह इंगित करता है कि एक एलीमेंट या ऐलीमेंट के समूह को कम से कम एक बार और सुविधानुसार एक से अधिक बार भी प्रयोग में ला सकते है।

```
<!ELEMENT funny (ha, haha+)>
```

6.11 करेक्टर कन्टेन्ट (Character Content)

जब किसी ऐलीमेंट के अन्दर केवल और केवल टेक्स्ट की ही अनुमति दी जाती है तब करेक्टर कन्टेन्ट का प्रयोग किया जाता है। इस कार्य के लिए

XML में एक स्टैण्डर्ड की वर्ड PCDATAC (Parseable Character data) का प्रयोग किया जाता है। उदाहरण के लिए डिक्लरेशन निम्न है –

```
<!ELEMENT para (title,text)
<!ELEMENT title (#PCDATA)
<!ELEMENT text (#PCDATA)
```

6.12 डोक्यूमेंट टाइप डिक्लरेशन एण्ड वेलीडेशन

(Document Type Decleration and Validation)

डोक्यूमेंट टाइप डिक्लरेशन **(DTD)** XML डोक्यूमेंट की विषयवस्तु की संरचना का वर्णन करता है। मॉडल के द्वारा क्या उपस्थित होता है, कौन ऐच्छिक है क्या उनके एट्रीब्यूट्स है और वह एक दूसरे से संबंधित होते हुए कैसे संरचित है, बताता है। **XML** आपको स्वयं का डी. टी.डी. बनाने की अनुमति प्रदान करता है। डी.टी.डी. **XML** डोक्यूमेंट की संरचना पर और विषय वस्तु की जांच की प्रक्रिया पर आपको पूरी तरह से नियंत्रण प्रदान करता है। जांच की प्रक्रिया को वेलिडेशन (validation) कहा जाता है।

डोक्यूमेंट टाइप डिक्लरेशन का syntax निम्न है –

<!DOCTYPE DTD. name [internal.subset]>

6.12.1 डी.टी.डी. का विस्तार (Development of DTD) – डी.टी.डी. को बनाने का कार्य इस बात पर निर्भर है कि आप सूचनाओं के साथ क्या करना चाहते हैं।

डी.टी.डी. को बनाने के लिए दो तरीकों का उपयोग किया जाता है। –

(i) Modify an exiting SGML DTD.

(ii) XML डोक्यूमेंट से मैन्यूली या ऑटोमेटिकली डी.टी.डी. का निर्माण करना।

हम दूसरे प्रकार को लेते हैं –

डी.टी.डी. को XML Code से बनाना – सबसे पहले XML डी.टी.डी. को बनाने का बेहतरीन तरीका है एक्स.एम.एल. डोक्यूमेंट को बनाना –

```
<?xml version = "1.0"?>
<page>
<head>
<title> My Home Page </title>
</head>
<body>
<title> welcome to my home page </title>
<para>

sorry page is under construction
</para>
</body>
</page>
```

अब इसे डी.टी.डी. syntax में देखते हैं।

```
<!DOCTYPE page [
<!ELEMENT page (head, body)>
<!ELEMENT head (title)
<!ELEMENT body (title, para)]
```

डी.टी.डी. का विश्लेषण करने पर हम देखते हैं कि

- पेज रूट ऐलीमेंट है।

- पेज ऐलीमेंट एक head रखता है।
- Head ऐलीमेंट एक title ऐलीमेंट को रखता है।
- Body ऐलीमेंट title ऐलीमेंट को रखता है जो कि para ऐलीमेंट का अनुसरण करता है।

6.13 एक्स.एम.एल. फाइल को इण्टरनेट एक्सप्लोरर में देखना (Viewing XML in Internet Explorer)

एक्स.एम.एल. फाइल को इण्टरनेट एक्सप्लोरर पर देखने का एक सरल तरीका यह है कि आप एक्स.एल.एल. फाइल को सबसे पहले **.xml** एक्सटेंशन से स्टोर करेंगे। उदाहरण के लिये निम्नलिखित **singer.xml** file को देखते हैं।

```
<?xml version = "1.0"?>
<singers>
 <singer>
  <name> Ashwin </name>
  <instrument> drums </instrument>
   <Nr_of_Recordings>/</Nr_of_Recordings>
 </singer>
 <singer>
  <name> Ram </name>
  <instrument> guitar </instrument>
   <Nr_of_Recordings>3</Nr_of_Recordings>
 </singer>
 <singer>
  <name> Ashish </name>
  <instrument> clarinet </instrument>
   <Nr_of_Recordings>2</Nr_of_Recordings>
 </singer>
 <singer>
  <name> Himanshu </name>
  <instrument> trumpet </instrument>
   <Nr_of_Recordings>2</Nr_of_Recordings>
 </singer>
</singers>
```

उपरोक्त फाईल को **singer.xml** फाइल नाम से स्टोर करेंगे। अब उपरोक्त फाइल **singer.xml** को इंटरनेट एक्सप्लोरर पर रन करवा देंगे।

6.14 XSL (Extensible Style Sheet Language)

CSS (Cascadle Style Sheet) को हम एच.टी.एम.एल. की तरह एक्स.एम.एल. के साथ भी प्रयोग कर सकते हैं। लेकिन **CSS** की कुछ सीमायें हैं उदाहरण के लिये हम **CSS** से ऐलीमेंट को क्रम में नहीं लगा सकते हैं। इसलिये इस तरह की कमी को पूरा करने के लिए दूसरी स्टाइल शीट लैग्वेंज का प्रयोग करते हैं जिसे एक्स.एम.एल. लैग्वेंज कहा जाता है।

एक्स.एस.एल. निम्नलिखित भागों से मिलकर बनी है –

(i) एक्स.एस.एल. ट्रान्सफॉरमेशन (XSLT) – यह एक एक्स.एम.एल. आधारित भाषा होती है जो कि एक्स.एम.एल डोक्यूमेंट को दूसरे एक्स.एम.एल. डोक्यूमेंट में बदलने के काम आती है।

(ii) एक्स.एम.एल. पाथ (XPATH) – इस भाषा का प्रयोग करके एक्स.एम.एल. डोक्यूमेंट के विभिन्न भागों को एसेस (access) कर सकते है।

इसके अलावा एक्स.एम.एल. में कुछ फोरमैंटिंग ऑब्जेक्ट होते हैं जिससे हम डाटा को विभिन्न रूपों में प्रदर्शित कर सकते हैं। एक्स.एम.एल. में डाटा को चुनने तथा विभिन्न रूपों में प्रदर्शित करने के लिए बहुत सारे एलीमेंट होते हैं। ये ऐलीमेंट निम्नलिखित हैं –

1. stylesheet
2. value of
3. for-each
4. sort
5. text

महत्वपूर्ण बिन्दु

1. एक्स.एम.एल. मार्क अप लैग्वेंज डाटा के बारे में बताता है जबकि एच.टी.एम.एल. से हम डाटा को विभिन्न प्रारूपों में प्रदर्शित कर सकते हैं।
2. एक्स.एम.एल. में यूजर अने टेग को परिभाषित कर सकता है।
3. एक्स.एम.एल. डोक्यूमेंट में एक रूट ऐलीमेंट होता है और उस रूट एलीमेंट के अन्दर बहुत सारे चाइल्ड ऐलीमेंट हो सकते है।
4. डोक्यूमेंट टाइप डेफिनेशन (डी.टी.डी.) का प्रयोग XML डोक्यूमेंट में उपस्थित डाटा की संरचना के लिए किया जाता है।

अभ्यासार्थ प्रश्न

बहुचयनात्मक प्रश्न

1. एक्स.एम.एल. वास्तव में है –

(अ) एच.टी.एम.एल. (ब) एस.जी.एम.एल.
(स) डी.एच.टी.एम.एल. (द) उपरोक्त में से कोई नहीं

2. एक्स.एम.एल. का पूरा नाम है –
   (अ) एक्सटेन्सिव मार्कअप लैंग्वेंज (ब) जिरोक्स मार्कअप लैंग्वेज
   (स) एक्सटेन्सिबल मेटा लैंग्वेज (द) उपरोक्त सभी
3. एक्स.एम.एल. में कमेन्ट देने का सही तरीका है –
   (अ) <? - - > (ब) <! - - !> (स) <! - - > 4 (द) </ - - >
4. कौन सा वाक्य सत्य है –
   (अ) सभी एक्स.एम.एल. डोक्यूमेंट के पास डी.टी.डी. होता है।
   (ब) सभी ऐलीमेंट के पास क्लोजिंग टेग होते है।
   (स) सभी एक्स.एम.एल. के ऐलीमेंट छोटे अक्षरों से लिखे जाते है।
   (द) सभी स्टेटमेन्ट सत्य हैं।
5. निम्नलिखित में से एक्स.एम.एल. के ऐलीमेंट के नाम देने के नियमों के अनुसार है –
   (अ) - my element (ब) my element
   (स) # my element (द) उपरोक्त में से कोई नहीं
6. DTD का पूरा नाम है –
   (अ) डोक्यूमेंट टाइप डेफिनेशन (ब) डोक्यूमेंट टू डोक्यूमेंट
   (स) डोमेन टू डोक्यूमेंट (द) डाटा टाइप डोक्यूमेंट
7. ऐलीमेंट अकरेन्स इडिकेण्टर निम्न में से कौन सा नहीं है –
   (अ) + (ब) * (स) ? (द) –

लघुत्तरात्मक प्रश्न –

1. एक्स.एम.एल. क्या है?
2. मार्कअप लैग्वेंज को परिभाषित कीजिए?
3. एक्स.एम.एल. तथा एच.टी.एम.एल. में क्या अन्तर होता है?
4. DTD को परिभाषित कीजिए?
5. एक्स.एम.एल. डोक्यूमेंट की ट्री संरचना को एक उदाहरण से समझाइये?
6. एक्स.एम.एल. के कन्टेन्ट मॉडल का वर्णन कीजिए?
7. एक्स.एम.एल. में एलीमेंट [element] के नाम देने के नियमों को लिखिए?

निबन्धात्मक प्रश्न –

1. XML के उद्देश्य एंव विशेषताओं की विवेचना कीजिए ।
2. XML तत्वों की विस्तृत संरचना को समझाए।
3. एलीमेंट कन्टेंट मॉडल से आप क्या समझते हैं।
4. डोक्यूमेंट टाइप डिक्लरेशन एण्ड वेलीडेशन पर संक्षिप्त टिप्पणी लिखिए ।

उत्तरमाला

1 (ब) 2 (अ) 3 (अ) 4 (ब) 5 (द) 6 (स) 7 (द)