HTML ਅਤੇ CSS ਨਾਲ ਵੈੱਬ ਡਿਜ਼ਾਈਨਿੰਗ

ਪਾਠ - 1

ਇਸ ਪਾਠ ਦੇ ਉਦੇਸ਼

1.1 ਇੱਕ ਵਧੀਆ ਵੈੱਬ ਡਿਜ਼ਾਈਨ ਕੀ ਹੈ ?
1.2 ਵੈੱਬਸਾਈਟ ਡਿਵੈਲਪ ਕਰਨ ਦੇ ਪੜਾਅ
1.3 HTML ਦੇ Concept
1.4 ਵੈੱਬ ਵਿਚ HTML ਦਾ ਰੋਲ
1.5 HTML ਡਾਕੂਮੈਂਟ ਦੀ ਬਣਤਰ
1.6 HTML ਐਡੀਟਰ
1.7 ਲਿੰਕਸ ਅਤੇ ਇਮੇਜ਼ਿਜ਼ ਨਾਲ ਕੰਮ ਕਰਨਾ
1.8 ਸਟਾਈਲ ਸ਼ੀਟਸ
1.9 ਸਟਾਈਲ ਸ਼ੀਟਸ ਦੀਆਂ ਕਿਸਮਾਂ
1.10 CSS ਸਿਨਟੈਕਸ
1.11 CSS ਸਿਲੈਕਟਰਜ਼
1.12 ਸਟਾਈਲ ਸ਼ੀਟਸ ਦੀਆਂ ਅਲੱਗ-ਅਲੱਗ ਪ੍ਰਾਪਰਟੀਜ਼
1.13 CSS ਪ੍ਰਾਪਰਟੀਜ਼ ਦੇ ਪ੍ਰੋਗਰਾਮ ਦੀ ਉਦਾਹਰਣ

ਜਾਣ-ਪਛਾਣ (Introduction)

ਵੈੱਬ ਡਿਜ਼ਾਈਨ, ਵੈੱਬਸਾਈਟਸ ਬਣਾਉਣ ਦੀ ਪ੍ਰਕਿਰਿਆ ਹੈ। ਇਸ ਵਿੱਚ ਕਈ ਵੱਖੋ ਵੱਖਰੇ ਪਹਿਲੂ ਸ਼ਾਮਿਲ ਹਨ, ਜਿਵੇਂ ਕਿ ਵੈੱਬਪੇਜ ਲੇਆਊਟ, ਸਮੱਗਰੀ (Content), ਉਤਪਾਦਨ ਅਤੇ ਗ੍ਰਾਫਿਕ ਡਿਜ਼ਾਈਨ ਆਦਿ। ਵੈੱਬਸਾਈਟਸ ਨੂੰ ਮਾਰਕਅੱਪ ਭਾਸ਼ਾ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਬਣਾਇਆ ਜਾਂਦਾ ਹੈ ਜਿਸ ਨੂੰ HTML ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਵੈੱਬ ਡਿਜ਼ਾਈਨਰ HTML ਟੈਗਜ਼ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ ਵੈੱਬ ਪੰਨੇ ਬਣਾਉਂਦੇ ਹਨ ਜੋ ਹਰੇਕ ਪੰਨੇ ਦੀ ਸਮੱਗਰੀ ਅਤੇ MetaData ਨੂੰ ਪਰਿਭਾਸ਼ਿਤ ਕਰਦੇ ਹਨ। ਵੈੱਬਪੇਜ ਦੇ ਅੰਦਰ Layout ਅਤੇ Appearance ਆਮ ਤੌਰ 'ਤੇ CSS ਦੀ ਵਰਤੋ ਕਰਕੇ ਪਰਿਭਾਸ਼ਤ ਕੀਤੇ ਜਾਂਦੇ ਹਨ। ਇਸ ਲਈ, ਜ਼ਿਆਦਾਤਰ ਵੈੱਬਸਾਈਟਸ ਵਿੱਚ HTML ਅਤੇ CSS ਦਾ ਸੁਮੇਲ ਸ਼ਾਮਿਲ ਹੁੰਦਾ ਹੈ ਜੋ ਤੈਅ ਕਰਦੇ ਹਨ ਕਿ **Webpage** ਇੱਕ **Web** ਬ੍ਰਾਊਜ਼ਰ ਵਿੱਚ ਕਿਸ ਤਰ੍ਹਾਂ ਦਾ ਦਿਖਾਈ ਦੇਵੇਗਾ।

ਕੁਝ ਵੈੱਬ ਡਿਜ਼ਾਈਨਰ **Hand Code** (ਸ਼ੁਰੂ ਤੋਂ **HTML** ਅਤੇ **CSS** ਟਾਈਪ ਕਰਨਾ) ਕਰਨ ਨੂੰ ਤਰਜੀਹ ਦਿੰਦੇ ਹਨ, ਜਦਕਿ ਹੋਰ ਵੈੱਬ ਡਿਜ਼ਾਈਨਰ **Adobe Dreamweaver** ਵਰਗੇ "What You See Is What You Get" (WYSIWYG) ਐਡੀਟਰ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਨ। ਇਸ ਕਿਸਮ ਦਾ **Editor** (ਐਡੀਟਰ) ਵੈੱਬਪੇਜ ਲੇਆਊਟ ਨੂੰ ਡਿਜ਼ਾਈਨ ਕਰਨ ਲਈ ਇੱਕ ਵਿਜ਼ੂਅਲ ਇੰਟਰਫੇਸ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ ਅਤੇ ਸਾਫਟਵੇਅਰ ਆਪਣੇ ਆਪ ਹੀ **(Automatically) HTML** ਅਤੇ **CSS** ਕੋਡ ਤਿਆਰ ਕਰਦਾ ਹੈ।

ਅਸੀਂ ਪਿਛਲੀ ਕਲਾਸ ਵਿਚ, ਭਾਵ 10ਵੀਂ ਕਲਾਸ ਵਿੱਚ **HTML** ਅਤੇ **CSS** ਦੀਆਂ ਮੁੱਢਲੀਆਂ ਗੱਲਾਂ ਬਾਰੇ ਅਧਿਐਨ ਕੀਤਾ ਹੈ। ਇਸ ਅਧਿਆਇ ਵਿਚ, ਅਸੀਂ ਵੈੱਬ ਡਿਜ਼ਾਈਨਿੰਗ ਬਾਰੇ ਮੁੱਢਲੀਆਂ ਗੱਲਾਂ, ਓਪਨ ਸੋਰਸ ਐਡੀਟਰ- **Notepad++** ਦੇ ਨਾਲ **HTML** ਅਤੇ **CSS** ਦੇ ਕੁਝ **advanced topics** ਬਾਰੇ ਚਰਚਾ ਕਰਨ ਜਾ ਰਹੇ ਹਾਂ।

1.1 ਇੱਕ ਵਧੀਆ ਵੈੱਬ ਡਿਜ਼ਾਈਨ ਕੀ ਹੈ ? (What is a good Web design ?)

ਇੱਕ ਵੈੱਬਸਾਈਟ ਆਮ ਲੋਕਾਂ ਨਾਲ ਜਾਣਕਾਰੀ ਸਾਂਝੀ ਕਰਨ ਦੇ ਉਦੇਸ਼ ਲਈ ਬਣਾਈ ਜਾਂਦੀ ਹੈ। ਇਸ ਲਈ, ਇੱਕ ਵੈੱਬਸਾਈਟ ਦੀ ਸਫਲਤਾ ਇਸਦੇ ਡਿਜ਼ਾਈਨ, ਸਮੱਗਰੀ (Content) ਅਤੇ ਗਤੀ (Speed) 'ਤੇ ਨਿਰਭਰ ਕਰਦੀ ਹੈ। ਵੈੱਬਪੇਜ ਡਿਜ਼ਾਈਨ ਇੱਕ ਬਹੁਤ ਮਹੱਤਵਪੂਰਨ element ਹੈ। ਵੈੱਬਸਾਈਟ ਦੀ ਵਰਤੋਂ ਵਿਸ਼ਵ ਦੇ ਵੱਖ-ਵੱਖ ਹਿੱਸਿਆਂ ਵਿੱਚ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਅਤੇ ਇਹਨਾਂ ਵੱਖ-ਵੱਖ ਹਿੱਸਿਆਂ ਵਿੱਚ ਵੱਖ-ਵੱਖ ਭਾਸ਼ਾਵਾਂ ਲਿਖੀਆਂ, ਬੋਲੀਆਂ ਜਾਂ ਸਮਝੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਇਸ ਲਈ ਵੈੱਬਸਾਈਟ 'ਤੇ ਵਰਤੇ ਜਾਣ ਵਾਲੇ ਸ਼ਬਦ ਅਤੇ ਭਾਸ਼ਾ, ਆਮ ਲੋਕਾਂ ਦੇ ਸਮਝਣ ਲਈ, ਬਹੁਤ ਮਹੱਤਵਪੂਰਨ ਹਨ। ਇੱਕ ਚੰਗਾ ਵੈੱਬਸਾਈਟ ਡਿਜ਼ਾਈਨ ਬਣਾਉਣ ਲਈ ਸਾਨੂੰ ਵੱਖ-ਵੱਖ ਖੇਤਰਾਂ ਦੇ ਪੇਸ਼ੇਵਰਾਂ (Professionals) ਦੀ ਜ਼ਰੂਰਤ ਹੁੰਦੀ ਹੈ। ਇੱਕ ਚੰਗੇ ਵੈੱਬਸਾਈਟ ਡਿਜ਼ਾਈਨ ਲਈ 8 ਮਹੱਤਵਪੂਰਨ ਸਿਧਾਂਤ ਹਨ—

- **ਸਾਦਗੀ (Simplicity) :** ਵੈੱਬਸਾਈਟ ਲਈ ਇੱਕ ਸਾਫ਼/ਸਪੱਸ਼ਟ (Clear) ਅਤੇ ਤਾਜ਼ਾ (Fresh) ਡਿਜ਼ਾਈਨ ਨੂੰ ਤਰਜੀਹ ਦਿੱਤੀ ਜਾਣੀ ਚਾਹੀਦੀ ਹੈ। ਇਹ ਵੈੱਬਸਾਈਟ ਦੇ Load ਹੋਣ ਦੇ ਸਮੇਂ ਨੂੰ ਘਟਾਉਂਦਾ ਹੈ ਅਤੇ ਭਵਿੱਖ ਵਿੱਚ ਵੈੱਬਸਾਈਟ ਨੂੰ ਅਪਡੇਟ ਕਰਨਾ, ਐਡਿਟ ਕਰਨਾ ਸੌਖਾ ਬਣਾਉਂਦਾ ਹੈ। Webpage ਦਾ ਡਿਜ਼ਾਈਨ ਸਾਦਾ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ ਅਤੇ ਇਸਦਾ ਨੈਵੀਗੇਸ਼ਨ ਆਸਾਨ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ।
- **ਇੱਕਸਾਰਤਾ (Consistency) :** ਵੈੱਬਸਾਈਟ ਵਿੱਚ ਹਰੇਕ ਵੈੱਬਪੇਜ ਦਾ ਡਿਜ਼ਾਈਨ ਇੱਕਸਾਰ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ। ਸਾਰੇ ਵੈੱਬਪੇਜਾਂ ਦੇ ਬਟਨ, ਰੰਗ ਅਤੇ ਨੈਵੀਗੇਸ਼ਨ ਦਾ ਡਿਜ਼ਾਈਨ ਇਕੋ ਜਿਹਾ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ ਤਾਂ ਜੋ ਵਰਤਣ ਅਤੇ ਸਮਝਣ ਵਿਚ ਆਸਾਨੀ ਹੋਵੇ।
- **ਟਾਈਪੋਗ੍ਰਾਫੀ ਅਤੇ ਉਪਯੋਗਤਾ ਟੈਕਸਟ (Typography & Usage Text) :** ਵੈੱਬਸਾਈਟ ਦਾ text ਸਾਫ਼/ਸਪੱਸ਼ਟ ਅਤੇ ਪੜ੍ਹਨਯੋਗ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ ਤਾਂ ਕਿ search engine ਵੈੱਬਪੇਜਾਂ ਨੂੰ index ਕਰ ਸਕੇ। ਵੈੱਬਪੇਜਾਂ ਵਿਚ ਟਾਈਟਲਜ਼ (ਹੈਡਿੰਗਜ਼) ਆਦਿ ਇਕੋ ਜਿਹੇ ਹੋਣੇ ਚਾਹੀਦੇ ਹਨ। ਭਾਸ਼ਾ ਦੀ ਵਰਤੋਂ ਵੀ ਇਸ ਤਰੀਕੇ ਨਾਲ ਕੀਤੀ ਜਾਣੀ ਚਾਹੀਦੀ ਹੈ ਕਿ ਆਨਲਾਈਨ ਵੈੱਬਪੇਜ ਟ੍ਰਾਂਸਲੇਟਰਜ਼, ਵੈੱਬਪੇਜ ਨੂੰ ਆਸਾਨੀ ਨਾਲ ਕਿਸੇ ਹੋਰ ਭਾਸ਼ਾ ਵਿੱਚ ਟ੍ਰਾਂਸਲੇਟ ਕਰ ਸਕਣ।
- **ਬਹੁਪੱਖੀ ਡਿਜ਼ਾਈਨ (Multidivisonal Design) :** ਅੱਜ-ਕੱਲ੍ਹ ਸਮਾਰਟਫੋਨਜ਼, ਟੈਬਲੇਟਸ ਅਤੇ ਵੱਖ-ਵੱਖ ਆਕਾਰਾਂ ਲਈ ਡਿਸਪਲੇਅ ਸਕਰੀਨਜ਼ ਦੀ ਵਰਤੋਂ ਵੱਧ ਰਹੀ ਹੈ। ਇਸ ਲਈ, ਵੈੱਬ ਡਿਜ਼ਾਈਨ ਅਜਿਹਾ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ ਤਾਂ ਜੋ Content ਨੂੰ ਸਹੀ ਤਰ੍ਹਾਂ ਪ੍ਰਦਰਸ਼ਿਤ ਕੀਤਾ ਜਾ ਸਕੇ ਅਤੇ ਵੈੱਬਸਾਈਟ ਆਸਾਨੀ ਨਾਲ ਵੱਖ-ਵੱਖ ਆਕਾਰ ਦੇ ਸਕਰੀਨਜ਼ 'ਤੇ ਵਰਤੀ ਜਾ ਸਕੇ।
- **ਇਮੇਜ਼ਿਜ਼, ਵੀਡੀਓ ਅਤੇ ਆਡੀਓ (Images, Video & Audio) :** ਕਿਸੇ ਵੈੱਬਸਾਈਟ ਨੂੰ ਡਿਜ਼ਾਈਨ ਕਰਨ ਵੇਲੇ Image, ਵੀਡੀਓ ਅਤੇ ਆਡੀਓ ਦੀ ਵਰਤੋਂ ਢੁਕਵੀਂ ਜਗ੍ਹਾ 'ਤੇ ਕੀਤੀ ਜਾਣੀ ਚਾਹੀਦੀ ਹੈ। ਆਮ ਤੌਰ 'ਤੇ ਲੋਕ ਟੈਕਸਟ ਨੂੰ ਘੱਟ ਪੜ੍ਹਨਾ ਪਸੰਦ ਕਰਦੇ ਹਨ ਅਤੇ ਆਡੀਓ/ਵੀਡੀਓ ਨੂੰ ਸੁਣਨਾ ਅਤੇ ਵੇਖਣਾ ਵੱਧ ਪਸੰਦ ਕਰਦੇ ਹਨ ਪਰ ਇਹ ਵੀ ਧਿਆਨ ਦੇਣਾ ਚਾਹੀਦਾ ਹੈ ਕਿ Webpage ਦਾ ਸਾਈਜ਼ Heavy ਨਹੀਂ ਹੋਣਾ ਚਾਹੀਦਾ ਅਤੇ ਵੈੱਬਪੇਜ ਨੂੰ Load ਕਰਨ ਦਾ ਸਮਾਂ ਵੀ ਜ਼ਿਆਦਾ ਨਾ ਹੋਵੇ।
- **ਸੰਚਾਰ ਅਤੇ ਪਤਾ (Communication and Address) :** ਵੈੱਬਸਾਈਟ ਨੂੰ ਇਸ ਤਰੀਕੇ ਨਾਲ ਤਿਆਰ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ ਕਿ ਵੱਖਰੀ ਜਾਣਕਾਰੀ ਲਈ ਸਹੀ ਜਗ੍ਹਾ (ਜਿਵੇਂ ਕਿ ਨਵਾਂ ਵੈੱਬਪੇਜ ਜਾਂ ਬਲਾਕ) ਨਿਰਧਾਰਤ ਕੀਤੀ ਜਾ ਸਕੇ। ਵਿਸ਼ੇਸ਼ ਜਾਣਕਾਰੀ ਵਾਲੇ ਵੈੱਬਪੇਜ ਦਾ URL ਸੌਖਾ ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ ਤਾਂ ਜੋ ਇਸਨੂੰ ਯਾਦ ਰੱਖਿਆ ਜਾ ਸਕੇ। ਵੈੱਬਸਾਈਟ ਵਿੱਚ ਯੂਜ਼ਰਜ਼ ਨੂੰ ਅੱਪਡੇਟ ਕੀਤੀ ਗਈ ਜਾਣਕਾਰੀ ਦੀ ਸੂਚਨਾ ਪਹੁੰਚਾਉਣ ਦੀ ਸੁਵਿਧਾ ਵੀ (Feature) ਹੋਣੀ ਚਾਹੀਦੀ ਹੈ ਤਾਂ ਕਿ ਨਵੀਂ ਜਾਣਕਾਰੀ ਲਈ ਉਹ ਦੁਬਾਰਾ ਵੈੱਬਸਾਈਟ 'ਤੇ visit ਕਰ ਸਕਣ।
- **ਸੋਸ਼ਲ ਸ਼ੇਅਰਿੰਗ ਫੀਚਰ (Social Sharing Feature) :** ਅੱਜ ਦਾ ਸਮਾਂ ਸੋਸ਼ਲ ਨੈੱਟਵਰਕਿੰਗ ਦਾ ਹੈ ਅਤੇ ਉਹ ਇਨਫਰਮੇਸ਼ਨ ਜਿਸ ਵਿੱਚ ਯੂਜ਼ਰ ਦਿਲਚਸਪੀ ਲੈਂਦੇ ਹਨ, ਉਸ ਇਨਫਰਮੇਸ਼ਨ ਨੂੰ ਸੋਸ਼ਲ ਮੀਡੀਆ ਦੀ ਵਰਤੋਂ ਕਰਕੇ, ਹੋਰਾਂ ਨਾਲ ਸਾਂਝਾ ਕਰਨਾ ਪਸੰਦ ਕਰਦੇ ਹਨ। ਇਸ ਲਈ ਵੈੱਬਸਾਈਟ ਦੇ ਹਰ ਭਾਗ ਨੂੰ ਇਸ ਤਰ੍ਹਾਂ ਡਿਜ਼ਾਈਨ ਕੀਤਾ ਜਾਣਾ ਚਾਹੀਦਾ ਹੈ ਕਿ ਵੈੱਬਸਾਈਟ ਤੇ ਉਪਲੱਬਧ ਜਾਣਕਾਰੀ ਨੂੰ ਆਸਾਨੀ ਨਾਲ ਸਾਂਝਾ ਕੀਤਾ ਜਾ ਸਕੇ।

- **F ਡਿਜ਼ਾਈਨ (F Design) :** ਇਹ ਖੋਜ ਤੋਂ ਪਤਾ ਲੱਗਾ ਹੈ ਕਿ ਆਮ ਯੂਜ਼ਰ, ਵੈੱਬਪੇਜਾਂ (ਕੰਪਿਊਟਰ ਸਕਰੀਨਜ਼) ਨੂੰ ਇੱਕ “F” ਪੈਟਰਨ ਵਿੱਚ ਸਕੈਨ (ਵੇਖਦੇ) ਕਰਦੇ ਹਨ। ਪਹਿਲੀ ਵਾਰ ਸਕਰੀਨ ਦੇ ਸੱਜੇ ਪਾਸੇ ਘੱਟ ਹੀ ਵੇਖਿਆ ਜਾਂਦਾ ਹੈ। ਇਸ ਕਿਸਮ ਦੇ ਡਿਜ਼ਾਈਨ ਦਾ ਅਰਥ Webpage ਨੂੰ ਖੱਬੇ ਤੋਂ ਸੱਜੇ ਅਤੇ ਉੱਪਰ ਤੋਂ ਹੇਠਾਂ ਵੱਲ ਵੇਖਣਾ ਹੈ, ਜੋ ਕਿ ਜ਼ਿਆਦਾਤਰ ਯੂਜ਼ਰ ਪਸੰਦ ਕਰਦੇ ਹਨ। ਇਹ ਇੰਗਲਿਸ਼ ਵਰਣਮਾਲਾ ਦੇ ਅੱਖਰ “F” ਵਾਂਗ ਹੀ ਹੈ।

1.2 ਵੈੱਬਸਾਈਟ ਡਿਵੈਲਪ ਵਿਕਾਸ ਕਰਨ ਦੇ ਪੜਾਅ (The Phases of Website Development)

ਵੈੱਬਸਾਈਟ ਡਿਜ਼ਾਈਨ ਅਤੇ ਵੈੱਬਸਾਈਟ ਵਿਕਾਸ (Development Process) ਵਿਚ 6 ਵੱਡੇ ਕਦਮ (ਮਹੱਤਵਪੂਰਨ ਕਦਮ) ਹਨ। ਇਹ ਕਦਮ ਸ਼ੁਰੂਆਤੀ ਜਾਣਕਾਰੀ ਨੂੰ ਇਕੱਠਾ ਕਰਨ ਤੋਂ ਪੂਰੀ ਤਰ੍ਹਾਂ ਕੰਮ ਕਰਨ ਵਾਲੀ ਵੈੱਬਸਾਈਟ ਨੂੰ ਬਣਾਉਣ ਅਤੇ ਅਖ਼ੀਰ ਵਿਚ ਵੈੱਬਸਾਈਟ ਨੂੰ ਅਪ-ਟੂ-ਡੇਟ ਰੱਖਣ ਲਈ ਰੱਖ-ਰਖਾਅ ਕਰਨ ਤੱਕ ਜਾਂਦੇ ਹਨ।

1. ਜਾਣਕਾਰੀ ਇਕੱਠੀ ਕਰਨੀ (Information Gathering)
2. ਯੋਜਨਾਬੰਦੀ (Planning)
3. ਡਿਜ਼ਾਈਨ (Design)
4. ਵਿਕਾਸ (Development)
5. ਟੈਸਟਿੰਗ ਅਤੇ ਡਿਲਿਵਰੀ (Testing & Delivery)
6. ਰੱਖ-ਰਖਾਅ (Maintenance)

1.3 HTML ਦੇ CONCEPT

ਜਿਵੇਂ ਕਿ ਅਸੀਂ ਜਾਣਦੇ ਹਾਂ ਕਿ HTML (ਹਾਈਪਰਟੈਕਸਟ ਮਾਰਕਅੱਪ ਲੈਂਗੁਏਜ਼) ਵੈੱਬਪੇਜਾਂ ਲਈ ਮੁੱਖ ਮਾਰਕਅੱਪ ਭਾਸ਼ਾ ਹੈ। ਇਹ ਟੈਕਸਟ, ਲਿਸਟਾਂ, ਇਮੇਜ਼ਿਜ਼, ਵੀਡੀਓ, ਆਡੀਓ, ਟੇਬਲ, ਫਰੇਮ, ਹੈਡਿੰਗਜ਼ (ਟਾਈਟਲਜ਼) ਆਦਿ ਨਾਲ ਵੈੱਬ ਪੰਨੇ ਬਣਾਉਣ ਦਾ ਇੱਕ ਸਾਧਨ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ। ਇਹ ਫਾਰਮ ਬਣਾ ਕਿ ਜਾਣਕਾਰੀ ਇਕੱਠੀ ਕਰਨ ਦੀ ਸਹੂਲਤ ਵੀ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ। ਇਹ ਸਾਰਾ ਕੁਝ ਵਿਸ਼ੇਸ਼ ਕਮਾਂਡਾਂ ਦੁਆਰਾ ਕੀਤਾ ਜਾਂਦਾ ਹੈ ਜਿਨ੍ਹਾਂ ਨੂੰ ਟੈਗ ਕਹਿੰਦੇ ਹਨ। HTML, ਹੋਰ ਭਾਸ਼ਾਵਾਂ ਵਿੱਚ Scripts ਨੂੰ ਜੋੜਨ ਜਾਂ Load ਕਰਨ ਦੀ ਯੋਗਤਾ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ (ਜਿਵੇਂ Java Script) ਵੈੱਬਸਾਈਟਸ ਅਤੇ ਵੈੱਬਪੇਜਾਂ ਦਾ ਡਿਜ਼ਾਈਨ ਕਾਸਕੇਡਿੰਗ ਸਟਾਈਲ ਸ਼ੀਟ (CSS) ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਬਹੁਤ ਪ੍ਰਭਾਵਸ਼ਾਲੀ ਢੰਗ ਨਾਲ ਬਣਾਇਆ ਜਾ ਸਕਦਾ ਹੈ। ਇਸ ਲਈ HTML ਬੇਸਿਕ ਅਤੇ ਐਡਵਾਂਸ ਵੈੱਬਪੇਜਾਂ ਨੂੰ ਬਣਾਉਣ ਵਿੱਚ ਸਮਰੱਥ ਹੈ।

HTML ਵਿੱਚ ਆਮ ਤੌਰ ’ਤੇ ਦੋ Types ਦੇ ਟੈਗ ਹੁੰਦੇ ਹਨ :

- **Paired Tags :** ਇਹਨਾਂ ਟੈਗਸ (Tags) ਵਿਚ ਓਪਨਿੰਗ ਟੈਗ < > ਅਤੇ ਕਲੋਜ਼ਿੰਗ ਟੈਗ < / > ਹੁੰਦੇ ਹਨ।
- **Singular Tags :** ਇਹਨਾਂ ਟੈਗਸ (Tags) ਨੂੰ ਬੰਦ ਕਰਨ (ਕਲੋਜ਼ਿੰਗ ਟੈਗਜ਼) ਦੀ ਜ਼ਰੂਰਤ ਨਹੀਂ ਹੈ।

1.4 ਵੈੱਬ ਵਿਚ HTML ਦਾ ਰੋਲ (HTML's Role in the Web)

HTML ਵੈੱਬਸਾਈਟਸ ਨੂੰ ਡਿਜ਼ਾਈਨ ਕਰਨ ਲਈ ਮੂਲ ਭਾਸ਼ਾ ਹੈ। HTML ਕੋਡ ਦੀ ਵਰਤੋਂ static ਵੈੱਬਪੇਜਾਂ ਨੂੰ ਡਿਜ਼ਾਈਨ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਕੋਡ ਵੈੱਬ ਬ੍ਰਾਊਜ਼ਰ ਦੁਆਰਾ ਸਮਝਿਆ ਜਾਂਦਾ ਹੈ ਅਤੇ ਇਸਨੂੰ ਵੇਖਣ ਯੋਗ ਬਣਾ ਕੇ ਡਿਸਲਪੇਅ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। HTML ਦੀ ਵਰਤੋਂ ਟੈਕਸਟ, ਇਮੇਜ਼ਿਜ਼ ਅਤੇ ਹੋਰ ਵੈੱਬਪੇਜ ਐਲੀਮੈਂਟਸ ਨੂੰ ਇੱਕ ਵੈੱਬਪੇਜ ਵਿੱਚ ਵਿਵਸਥਿਤ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਤੁਸੀਂ ਵੈੱਬ ਬ੍ਰਾਊਜ਼ਰ ਵਿਚ ਸੋਰਸ ਕੋਡ (Source code) ਵਿਕਲਪ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹੋਏ, HTML ਕੋਡ ਦੇਖ ਸਕਦੇ ਹੋ। ਇਸ ਕੋਡ ਦੀ ਵਰਤੋਂ ਨਾਲ ਵੈੱਬਪੇਜ ਬਣਾਇਆ ਜਾਂਦਾ ਹੈ। ਇਸ ਲਈ HTML ਦੀ ਵੈੱਬ ਵਿਚ ਬਹੁਤ ਮਹੱਤਵਪੂਰਨ ਭੂਮਿਕਾ ਹੈ ਅਤੇ ਇਸ ਤੋਂ ਬਿਨਾਂ ਵੈੱਬ ਸੰਭਵ ਨਹੀਂ ਹੈ।

1.5 HTML ਡਾਕੂਮੈਂਟ ਦੀ ਬਣਤਰ (Structure of Html Document)

ਹੇਠਾਂ HTML ਪ੍ਰੋਗਰਾਮ ਦੀ ਇੱਕ ਸਧਾਰਣ ਉਦਾਹਰਣ ਦਿੱਤੀ ਗਈ ਹੈ। ਹਰ HTML ਵੈੱਬਪੇਜ ’ਤੇ ਅਜਿਹਾ ਹੀ ਕੋਡ ਹੁੰਦਾ ਹੈ। ਹਰ HTML ਡਾਕੂਮੈਂਟ <html> ਟੈਗ ਨਾਲ ਸ਼ੁਰੂ ਹੁੰਦਾ ਹੈ ਅਤੇ </html> ਟੈਗ ਨਾਲ ਖਤਮ ਹੁੰਦਾ ਹੈ। HTML ਡਾਕੂਮੈਂਟ ਮੁੱਖ ਤੌਰ ਤੇ ਦੋ ਹਿੱਸਿਆਂ ਵਿੱਚ ਵੰਡਿਆ ਜਾਂਦਾ ਹੈ— Head Part ਅਤੇ Body Part, ਆਓ ਹੇਠਾਂ ਦਿੱਤੇ ਚਿੱਤਰ-1.1 ਦੇ ਕੋਡ ਬਾਰੇ ਸਿੱਖੀਏ :

```
<!DOCTYPE html>
<html>
    <head>
    <title>Web Designing</title>
    </head>

    <body>
    Welcome to the world of HTML and CSS
    </body>
</html>
```

ਚਿੱਤਰ 1.1 HTML ਪ੍ਰੋਗਰਾਮ ਦੀ ਸਧਾਰਨ ਉਦਾਹਰਣ

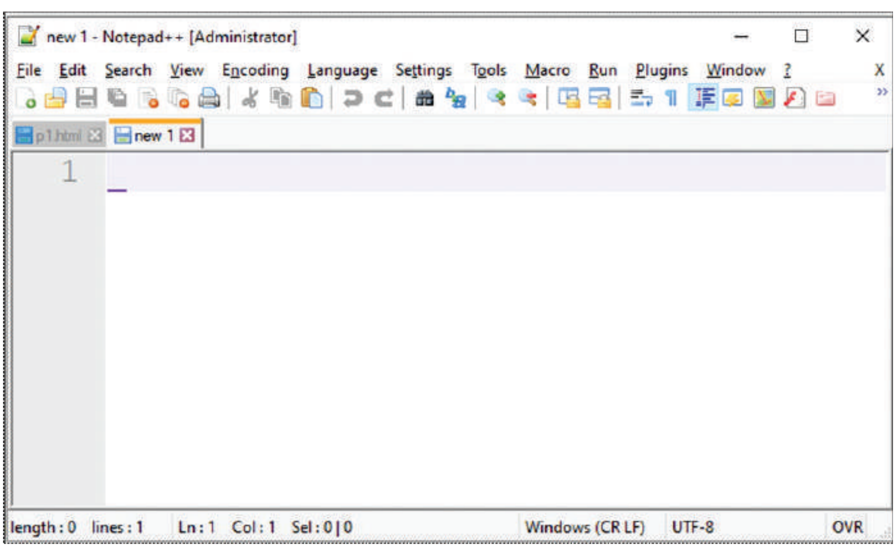
<! DOCTYPE html> : ਇਹ ਟੈਗ html ਦੇ ਵਰਜ਼ਨ (Version) ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਇਹ ਦੱਸਦਾ ਹੈ ਕਿ HTML ਡਾਕੂਮੈਂਟ HTML ਦੇ ਵਰਜ਼ਨ 5 ਦੀ ਵਰਤੋਂ ਕਰਨ ਜਾ ਰਿਹਾ ਹੈ।

Head Part : ਇਹ ਭਾਗ <head> ਟੈਗ ਨਾਲ ਸ਼ੁਰੂ ਅਤੇ </head> ਟੈਗ ਨਾਲ ਖਤਮ ਹੁੰਦਾ ਹੈ। ਇਸ ਹਿੱਸੇ ਵਿੱਚ HTML Document ਦੇ ਸਿਰਲੇਖ (Title) ਅਤੇ Metadata ਬਾਰੇ ਜਾਣਕਾਰੀ ਹੁੰਦੀ ਹੈ। Metadata, HTML ਪੇਜ ਵਿਚ ਉਪਲੱਬਧ ਡਾਟਾ ਨਾਲ ਸੰਬੰਧਿਤ ਹੋਰ (Additional) ਜਾਣਕਾਰੀ ਦਾ ਹਵਾਲਾ ਦਿੰਦਾ ਹੈ।

Body Part : ਇਹ ਭਾਗ <body> ਟੈਗ ਨਾਲ ਸ਼ੁਰੂ ਹੁੰਦਾ ਹੈ ਅਤੇ </body> ਟੈਗ ਨਾਲ ਖਤਮ ਹੁੰਦਾ ਹੈ। ਇਸ ਟੈਗ ਵਿਚ ਇਹ ਸਾਰੀ ਜਾਣਕਾਰੀ ਸ਼ਾਮਿਲ ਹੈ ਜੋ ਵੈੱਬਪੇਜ 'ਤੇ ਪ੍ਰਦਰਸ਼ਿਤ ਕੀਤੀ ਜਾਂਦੀ ਹੈ, ਭਾਵ ਯੂਜ਼ਰ ਨੂੰ ਦਿਖਾਈ ਦੇਵੇਗੀ।

1.6 HTML ਐਡੀਟਰ (HTML Editor)

ਜਿਵੇਂ ਕਿ ਅਸੀਂ ਪਿਛਲੀ ਕਲਾਸ ਵਿਚ ਪੜ੍ਹਿਆ ਸੀ, HTML ਦੀ ਵਰਤੋਂ ਨਾਲ ਵੈੱਬਪੇਜਾਂ ਨੂੰ ਬਣਾਉਣ ਲਈ, ਸਾਨੂੰ ਕਿਸੇ ਸਧਾਰਨ ਟੈਕਸਟ ਐਡੀਟਰ ਦੀ ਲੋੜ ਹੁੰਦੀ ਹੈ, ਜਿਵੇਂ ਕਿ ਨੋਟਪੈਡ (ਵਿੰਡੋਜ ਦਾ ਬਿਲਟ-ਇਨ ਟੈਕਸਟ ਐਡੀਟਰ) ਆਦਿ। ਹਾਲਾਂਕਿ, ਹੋਰ ਬਹੁਤ ਸਾਰੇ ਐਡਵਾਂਸਡ ਟੈਕਸਟ ਐਡੀਟਰਜ਼ ਪ੍ਰੋਗਰਾਮਿੰਗ ਲਈ ਉਪਲੱਬਧ ਹਨ ਜਿਵੇਂਕਿ Notepad++ ਜੋਕਿ ਇੱਕ ਮੁਫਤ ਅਤੇ ਸ਼ਕਤੀਸ਼ਾਲੀ ਕੋਡ ਐਡੀਟਰ ਹੈ। ਇਹ ਬਹੁਤ ਸਾਰੀਆਂ ਪ੍ਰੋਗਰਾਮਿੰਗ ਭਾਸ਼ਾਵਾਂ ਦਾ Source code ਲਿਖਣ ਲਈ ਸਹਾਇਤਾ ਪ੍ਰਦਾਨ ਕਰਦਾ ਹੈ। ਇਸ ਐਡੀਟਰ ਦੀ ਵਰਤੋਂ ਕਰਦਿਆਂ, ਅਸੀਂ HTML ਅਤੇ CSS ਦੀ ਕੋਡਿੰਗ ਅਸਾਨੀ ਨਾਲ ਲਿਖ ਸਕਦੇ ਹਾਂ। ਇਸਨੂੰ ਇੰਟਰਨੈੱਟ ਤੋਂ ਮੁਫਤ Download ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ (ਵੈੱਬਸਾਈਟ : https://notepad-plus-plus.org/)। ਇਸ ਵਿਚ Tabs ਦੀ ਸਹੂਲਤ ਹੈ ਅਤੇ ਅਸੀਂ ਇਕੋ ਸਮੇਂ ਕਈ Files ਦੀ ਐਡੀਟਿੰਗ ਅਤੇ ਉਹਨਾਂ ਵਿੱਚ ਕੰਮ ਕਰ ਸਕਦੇ ਹਾਂ। ਹੇਠਾਂ ਦਿੱਤੀ ਤਸਵੀਰ Notepad++ ਐਡੀਟਰ ਦਾ ਮੇਨ ਇੰਟਰਫੇਸ ਦਰਸਾਉਂਦੀ ਹੈ।



ਚਿੱਤਰ 1.2 Notepad++ ਦੀ ਮੁੱਖ ਵਿੰਡੋ

HTML ਪ੍ਰੋਗਰਾਮ ਦੇ ਆਊਟਪੁੱਟ ਨੂੰ ਵੇਖਣ ਲਈ, ਸਾਨੂੰ ਇਸਨੂੰ ਵੈੱਬ ਬ੍ਰਾਊਜ਼ਰ ਵਿੱਚ ਖੋਲ੍ਹਣਾ ਪੈਂਦਾ ਹੈ। ਜੇ ਅਸੀਂ HTML ਵੈੱਬਪੇਜ ਨੂੰ ਸਧਾਰਨ ਟੈਕਸਟ ਐਡੀਟਰ ਵਿੱਚ ਤਿਆਰ ਕਰਦੇ ਹਾਂ, ਤਾਂ ਸਾਨੂੰ ਵੈੱਬ ਬ੍ਰਾਊਜ਼ਰ ਵਿੱਚ ਵੇਖਣ ਲਈ ਉਹ ਵੈੱਬਪੇਜ਼ Manually ਖੋਲ੍ਹਣਾ ਪਵੇਗਾ ਪਰ Notepad++ ਦੇ ਮਾਮਲੇ ਵਿਚ, ਅਸੀਂ ਇਹ ਕੰਮ Notepad++ ਇੰਟਰਫੇਸ ਵਿਚੋਂ ਹੀ ਕਰ ਸਕਦੇ ਹਾਂ। ਵੈੱਬ ਬ੍ਰਾਊਜ਼ਰ ਵਿਚ HTML ਕੋਡ ਵੇਖਣ ਲਈ, ਅਸੀਂ ਹੇਠਾਂ ਦਿੱਤੇ ਕਦਮਾਂ ਦੀ ਪਾਲਣਾ ਕਰ ਸਕਦੇ ਹਾਂ :

- View Menu 'ਤੇ ਕਲਿੱਕ ਕਰੋ।
- ਸੱਬਮੀਨੂੰ (Submenu) ਖੋਲ੍ਹਣ ਲਈ View Current File in 'ਤੇ ਕਲਿੱਕ ਕਰੋ।
- HTML ਪ੍ਰੋਗਰਾਮ ਦੀ ਆਊਟਪੁੱਟ ਵੇਖਣ ਲਈ ਆਪਣੀ ਇੱਛਾ ਅਨੁਸਾਰ Sub Menu ਵਿਚ ਦਰਜ ਵੈੱਬ ਬ੍ਰਾਊਜ਼ਰਜ਼ ਵਿਚੋਂ ਕਿਸੇ ਇੱਕ ਦੀ ਚੋਣ ਕਰੋ।
- (ਅਸੀਂ HTML ਪ੍ਰੋਗਰਾਮ ਦੇ ਆਊਟਪੁੱਟ ਨੂੰ Notepad++ ਦੀ ਟੈਬ 'ਤੇ ਸੱਜਾ ਕਲਿੱਕ ਕਰਕੇ ਅਤੇ Open in Default Viewer ਦੀ ਚੋਣ ਕਰਕੇ ਵੀ ਵੇਖ ਸਕਦੇ ਹਾਂ।
- ਆਓ ਹੁਣ Notepad++ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਟਾਈਮ ਟੇਬਲ ਦੀ ਬੇਸਿਕ ਉਦਾਹਰਣ ਤੋਂ ਸ਼ੁਰੂਆਤ ਕਰੀਏ। ਉਦਾਹਰਣ ਲਈ ਹੇਠਾਂ ਦਿੱਤੇ ਕਦਮਾਂ ਦੀ ਪਾਲਣਾ ਕਰੋ :
- Notepad++ ਟੈਕਸਟ ਐਡੀਟਰ ਨੂੰ ਖੋਲ੍ਹੋ।
- File → New ਜਾਂ Ctrl+N ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਨਵੀਂ ਫਾਈਲ ਬਣਾਓ।
- ਹੁਣ ਚਿੱਤਰ-1.3 ਵਿਚ ਦਰਸਾਏ ਅਨੁਸਾਰ ਕੋਡ ਟਾਈਪ ਕਰੋ :

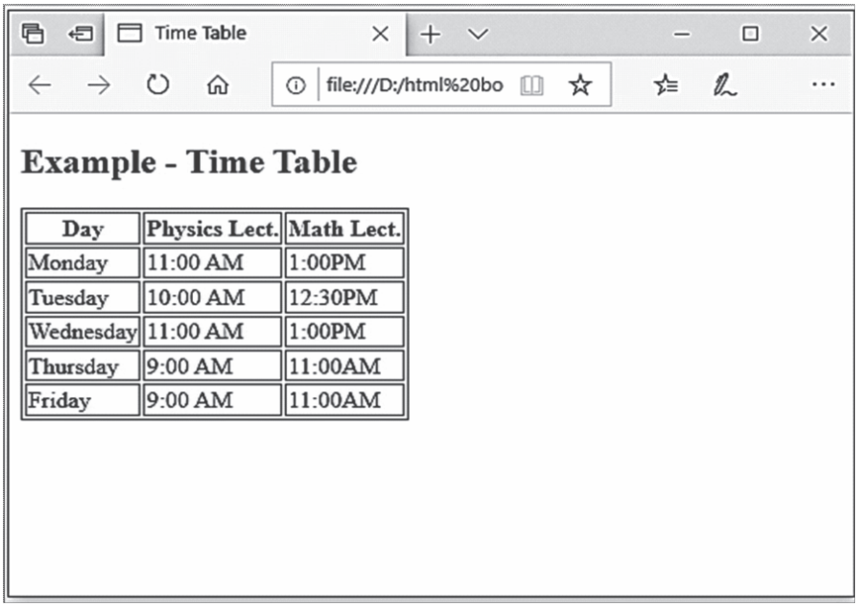
```
<!DOCTYPE html>
<html>
    <head>
    <title>Time  Table</title>
    </head>

    <body>
    <h2>Example - Time Table</h2>
    <table border="1">
    <tr><th>Day</th><th>Physics Lect.</th><th>Math Lect.</th></tr>
    <tr><td>Monday</td><td>11:00 AM</td><td>1:00PM</td></tr>
    <tr><td>Tuesday</td><td>10:00 AM</td><td>12:30PM</td></tr>
    <tr><td>Wednesday</td><td>11:00 AM</td><td>1:00PM</td></tr>
    <tr><td>Thursday</td><td>9:00 AM</td><td>11:00AM</td></tr>
    <tr><td>Friday</td><td>9:00 AM</td><td>11:00AM</td></tr>
    </table>
    </body>
</html>
```

ਚਿੱਤਰ 1.3 Notepad++ ਵਿਚ ਬੇਸਿਕ Time Table ਪ੍ਰੋਗਰਾਮ

ਉਪਰੋਕਤ ਕੋਡ ਦੇ ਪੂਰਾ ਹੋਣ ਤੋਂ ਬਾਅਦ, ਹੁਣ ਇਸ ਨੂੰ File → Save ਜਾਂ ਸ਼ਾਰਟਕੱਟ ਕੀਅ Ctrl + S ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਸੇਵ ਕਰੋ। ਇਹ ਸੁਨਿਸ਼ਚਿਤ ਕਰੋ ਕਿ HTML ਵੈੱਬਪੇਜ਼ ਦੀ ਐਕਸਟੈਂਸ਼ਨ (.html) ਨੂੰ, ਫਾਇਲ ਦੇ ਨਾਮ ਦੇਣ ਤੋਂ ਬਾਅਦ ਟਾਈਪ ਕਰ ਦਿੱਤਾ ਹੈ ਭਾਵ filename.html (ਉਦਾਹਰਣ ਲਈ : p1.html) ਟਾਈਪ ਕਰੋ।

ਉਪਰੋਕਤ ਵੈੱਬਪੇਜ ਦੇ ਨਤੀਜੇ ਨੂੰ ਵੇਖਣ ਲਈ, ਫਾਈਲ ਟੈਬ ਉੱਤੇ ਸੱਜਾ ਕਲਿੱਕ ਕਰੋ ਅਤੇ "Open in default viewer" ਵਿਕਲਪ ਉੱਤੇ ਕਲਿੱਕ ਕਰੋ। ਇਹ ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ ਡਿਫਾਲਟ ਵੈੱਬ ਬ੍ਰਾਊਜ਼ਰ ਵਿੱਚ ਵੈੱਬਪੇਜ ਨੂੰ ਖੋਲ੍ਹ ਦੇਵੇਗਾ :



ਚਿੱਤਰ 1.4 - ਉੱਪਰ ਦਿੱਤੇ ਚਿੱਤਰ 1.3 ਦੇ HTML Code ਦਾ ਨਤੀਜਾ

1.7 ਲਿੰਕਸ ਅਤੇ ਇਮੇਜ਼ਿਜ਼ ਨਾਲ ਕੰਮ ਕਰਨਾ (Working with Links and Images)

Links ਅਤੇ Images, ਵੈੱਬ ਦੀ ਦੁਨੀਆ ਵਿੱਚ ਇੱਕ ਮਹੱਤਵਪੂਰਨ ਭੂਮਿਕਾ ਅਦਾ ਕਰਦੇ ਹਨ। HTML ਵਿੱਚ Links ਨੂੰ Hyperlinks ਕਿਹਾ ਜਾਂਦਾ ਹੈ। Hyperlink ਵਾਲੇ ਟੈਕਸਟ ਨੂੰ ਹਾਈਪਰਟੈਕਸਟ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਮੂਲ ਰੂਪ ਵਿੱਚ ਲਿੰਕ ਦਾ ਰੰਗ ਨੀਲਾ ਹੁੰਦਾ ਹੈ ਅਤੇ ਉਹ ਅੰਡਰਲਾਈਨ ਕੀਤੇ ਹੁੰਦੇ ਹਨ। ਲਿੰਕ ’ਤੇ ਕਲਿੱਕ ਕਰਕੇ ਅਸੀਂ ਕਿਸੇ ਹੋਰ ਵੈੱਬਪੇਜ ’ਤੇ ਜਾ ਸਕਦੇ ਹਾਂ। ਆਮ ਤੌਰ ’ਤੇ ਜਦੋਂ ਮਾਊਸ ਪੁਆਇੰਟਰ ਨੂੰ ਲਿੰਕ ’ਤੇ ਲਿਜਾਇਆ ਜਾਂਦਾ ਹੈ ਤਾਂ ਇਹ Handpointer ਵਿੱਚ ਬਦਲ ਜਾਂਦਾ ਹੈ। ਇਸਤੋਂ ਇਹ ਅੰਦਾਜ਼ਾਂ ਲਗਾਇਆ ਜਾ ਸਕਦਾ ਹੈ ਕਿ ਇਹ ਲਿੰਕ ਹੈ ਅਤੇ ਇਸ ਦੀ ਵਰਤੋਂ ਕਰਨ ਲਈ ਇਸ ਉਤੇ ਕਲਿੱਕ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। ਲਿੰਕ ਸਿਰਫ ਟੈਕਸਟ ’ਤੇ ਹੀ ਨਹੀਂ ਬਲਕਿ ਇਮੇਜ਼ ਜਾਂ ਹੋਰ ਆਬਜੈਕਟ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਵੀ ਬਣਾਏ ਜਾ ਸਕਦੇ ਹਨ। Images ਦੀ ਵਰਤੋਂ ਵੈੱਬਪੇਜਾਂ ਨੂੰ ਸੁੰਦਰ ਅਤੇ ਵਧੇਰੇ ਵਰਣਨਸ਼ੀਲ ਬਣਾਉਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਆਓ ਸਮਝੀਏ ਕਿ ਵੈੱਬਪੇਜਾਂ ਵਿਚ HTML ਦੀ ਵਰਤੋਂ ਕਰਦਿਆਂ Link ਅਤੇ Images ਨਾਲ ਕਿਵੇਂ ਕੰਮ ਕਰਨਾ ਹੈ।

1.7.1 ਲਿੰਕਸ ਨਾਲ ਕੰਮ ਕਰਨਾ (Working with Links)

ਹਾਈਪਰਲਿੰਕਸ HTML ਵਿਚ <a> ਟੈਗ ਨਾਲ ਬਣਾਏ ਜਾਂਦੇ ਹਨ। ਟੈਗ <a> ਨੂੰ ਐਂਕਰ ਟੈਗ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਅਸੀਂ href ਐਟਰੀਬਿਊਟ ਦੇ ਨਾਲ ਐਂਕਰ ਟੈਗ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ, ਐਟਰੀਬਿਊਟ href ਹਾਈਪਰਟੈਕਸਟ ਦੀ reference ਹੈ। ਇਸਦੀ URL ਦੀ Value/Linked ਫਾਈਲ/ਪੇਜ ਦੇ ਪਤੇ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ। ਬੇਸਿਕ ਲਿੰਕ ਟੈਗ ਦੀ ਉਦਾਹਰਣ ਹੈ :

linktext

ਲਿੰਕ ਟੈਗ ਦੇ ਦੋ ਮੁੱਖ ਹਿੱਸੇ ਹਨ : “ਲਿੰਕ ਟੈਕਸਟ”/link test ਜਿਸ ਨੂੰ ਯੂਜ਼ਰ ਲਿੰਕ ਕੀਤੇ ਪੇਜ ਨੂੰ ਖੋਲ੍ਹਣ ਲਈ ਕਲਿੱਕ ਕਰਦਾ ਹੈ ਅਤੇ “URL/Address” ਉਹ ਹਿੱਸਾ ਹੈ ਜੋ ਲਿੰਕ ’ਤੇ ਜਾਣ ਵਾਲੇ ਪੇਜ ਦੇ ਪਤੇ ਨੂੰ ਦਰਸਾਉਂਦਾ ਹੈ। ਉਪਰੋਕਤ ਕੋਡ ਵਿੱਚ, href = “url” ਭਾਗ ਐਂਕਰ ਟੈਗ ਦਾ Address Part ਹੈ। ਵੱਖੋ-ਵੱਖਰੀਆਂ ਪ੍ਰਾਪਰਟੀਜ਼ ਜੋ ਐਂਕਰ ਟੈਗ ਨਾਲ ਵਰਤੀਆਂ ਜਾ ਸਕਦੀਆਂ ਹਨ, ਹੇਠਾਂ ਦਿੱਤੀਆਂ ਗਈਆਂ ਹਨ :

ਐਟਰੀਬਿਊਟ	ਵਿਆਖਿਆ
href	ਇਹ ਲਿੰਕ ਦਾ ਡੈਸਟੀਨੇਸ਼ਨ (Destination) ਐਡਰੈਸ ਹੁੰਦਾ ਹੈ ਭਾਵ ਉਸ ਵੈੱਬਪੇਜ ਜਾਂ ਆਬਜੈਕਟ ਦਾ ਪਤਾ ਜੋ ਲਿੰਕ ’ਤੇ ਕਲਿੱਕ ਕਰਕੇ ਓਪਨ ਹੁੰਦਾ ਹੈ। ਇਹ ਡੈਸਟੀਨੇਸ਼ਨ ਐਡਰੈਸ (Address) ਕਿਸੇ ਤਰ੍ਹਾਂ ਦੀ ਫਾਈਲ ਹੋ ਸਕਦੀ ਹੈ।

target	ਇਹ ਦਰਸਾਉਂਦਾ ਹੈ ਕਿ ਨਵਾਂ ਆਬਜੈਕਟ Current ਵਿੰਡੋ ਵਿਚ ਹੀ ਓਪਨ ਹੋਵੇਗਾ ਜਾਂ ਫੇਰ ਨਵੀਂ ਬ੍ਰਾਉਜ਼ਰ ਵਿੰਡੋ ਵਿਚ ਓਪਨ ਹੋਵੇਗਾ। ਇਸ ਦੀਆਂ values ਹਨ : _blank, _self, _top, _parent
Link Text	ਇਹ ਲਿੰਕ ਦਾ "ਟਾਈਟਲ ਟੈਕਸਟ" ਦਰਸਾਉਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਓਪਨਿੰਗ ਟੈਗ <a> ਅਤੇ ਕਲੋਜ਼ਿੰਗ ਟੈਗ ਵਿਚਕਾਰ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ।
title	ਇਹ ਲਿੰਕ ਦੀ ਟੂਲਟਿਪ ਦਿਖਾਉਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਜਦੋਂ ਮਾਊਸ ਦਾ ਪੁਆਇੰਟਰ ਲਿੰਕ 'ਤੇ ਕੁਝ ਸਕਿੰਟਾਂ ਲਈ ਸਥਿਰ ਰੱਖਿਆ ਜਾਂਦਾ ਹੈ ਤਾਂ ਇਹ ਟੈਕਸਟ ਦਿਖਾਈ ਦਿੰਦਾ ਹੈ।

ਟੇਬਲ 1.1 - ਐਂਕਰ ਟੈਗ ਦੇ ਐਟਰੀਬਿਊਟ

ਹੇਠਾਂ ਦਿਤਾ ਪ੍ਰੋਗਰਾਮ ਦਰਸਾਉਂਦਾ ਹੈ ਕਿ ਕਿਸ ਤਰ੍ਹਾਂ ਵੈੱਬਪੇਜਾਂ ਵਿਚ ਐਂਕਰ ਟੈਗ ਦੀ ਵਰਤੋਂ ਹੁੰਦੀ ਹੈ :

ਚਿੱਤਰ 1.5 ਹਾਈਪਰਲਿੰਕਸ ਦਾ ਕੋਡ ਅਤੇ ਇਸਦੀ ਆਉਟਪੁੱਟ ਵਿੰਡੋ

1.7.2 ਇਮੇਜ਼ਿਜ਼ ਨਾਲ ਕੰਮ ਕਰਨਾ (Working with Images)

ਇਮੇਜ਼ ਨੂੰ ਵੈੱਬਪੇਜਾਂ ਦੀ ਬੈਕਗ੍ਰਾਊਂਡ ਦੇ ਤੌਰ 'ਤੇ ਵੀ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ ਜਾਂ ਵੈੱਬਪੇਜਾਂ ਵਿੱਚ ਇੱਕ object ਦੇ ਤੌਰ 'ਤੇ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ। Images ਦੀ ਸਹੀ ਵਰਤੋਂ ਵੈੱਬਸਾਈਟਸ ਨੂੰ ਵਧੇਰੇ ਆਕਰਸ਼ਕ ਬਣਾਉਂਦੀ ਹੈ। <img> ਟੈਗ ਨੂੰ ਇਮੇਜ਼ ਐਲੀਮੈਂਟ ਦੇ ਰੂਪ ਵਿਚ HTML ਵੈੱਬਪੇਜਾਂ ਵਿੱਚ ਸ਼ਾਮਿਲ ਕਰਨ ਲਈ ਇਸਤੇਮਾਲ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਇੱਕ ਸਿੰਗਲ ਟੈਗ ਹੈ। ਇਸ ਵਿੱਚ Closing (ਬੰਦ ਕਰਨ ਵਾਲਾ) ਟੈਗ ਸ਼ਾਮਿਲ ਨਹੀਂ ਹੁੰਦਾ। ਹੇਠ ਲਿਖੀਆਂ ਪ੍ਰਾਪਰਟੀਜ਼ ਇਸ ਟੈਗ ਵਿੱਚ ਵਰਤੀਆਂ ਜਾ ਸਕਦੀਆਂ ਹਨ :

ਪ੍ਰਾਪਰਟੀ	ਵਿਆਖਿਆ
src	ਵੈੱਬਪੇਜ ਵਿਚ ਜੋ ਇਮੇਜ਼ ਦਾਖਲ ਕਰਨੀ ਹੈ, src ਉਸਦਾ ਸੋਰਸ (source) ਐਡਰੈਸ ਜਾਂ URL ਦਰਸਾਉਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਜੇ ਵੈੱਬਪੇਜ ਤੇ ਇਮੇਜ ਇਕੋ ਫੋਲਡਰ ਵਿਚ ਹਨ ਤਾਂ ਇਮੇਜ਼ ਦਾ ਸਿਰਫ਼ ਪੂਰਾ ਨਾਮ ਹੀ ਲਿਖਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਇਸ ਐਟਰੀਬਿਊਟ ਵਿਚ ਕਿਸੇ External ਵੈੱਬਸਾਈਟ ਦੀ ਇਮੇਜ਼ ਦਾ ਐਡਰੈਸ (URL) ਵੀ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ।
alt	ਇਸਨੂੰ ਅਲਟਰਨੇਟ (alternate) ਟੈਕਸਟ ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਇਹ ਸਿਰਫ਼ ਉਸ ਸਮੇਂ ਹੀ ਦਿਖਾਈ ਦਿੰਦਾ ਹੈ, ਜਦੋਂ ਇਮੇਜ਼ Load ਹੋਣ ਵਿਚ ਜ਼ਿਆਦਾ ਸਮਾਂ ਲੈ ਰਹੀ ਹੋਵੇ ਜਾਂ ਫਿਰ ਇਮੇਜ਼ Missing ਹੋਵੇ ਭਾਵ ਇਮੇਜ਼ ਦਾ ਪਤਾ (URL) ਸਹੀ ਨਾ ਹੋਵੇ।
title	ਇਹ ਇਮੇਜ਼ ਲਈ ਟੂਲਟਿਪ ਦਿਖਾਉਣ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਜਦੋਂ ਮਾਊਸ ਪੁਆਇੰਟਰ ਇਮੇਜ਼ 'ਤੇ ਕੁਝ ਦੇਰ ਲਈ ਸਥਿਰ ਰੱਖਿਆ ਜਾਂਦਾ ਹੈ ਤਾਂ Title ਵਿਚ ਲਿਖਿਆ ਟੈਕਸਟ ਟੂਲਟਿਪ ਵਜੋਂ ਦਿਖਾਈ ਦਿੰਦਾ ਹੈ।
Width, height	ਇਹ ਐਟਰੀਬਿਊਟਸ ਇਮੇਜ਼ ਦੀ ਚੌੜਾਈ ਅਤੇ ਉਚਾਈ ਦਰਸਾਉਣ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ।

ਹੇਠਾਂ ਦਿੱਤਾ ਪ੍ਰੋਗਰਾਮ ਦਰਸਾਉਂਦਾ ਹੈ ਕਿ ਕਿਸ ਤਰ੍ਹਾਂ ਵੈੱਬਪੇਜਾਂ ਵਿਚ ਇਮੇਜ਼ਿਜ਼ ਨੂੰ ਦਾਖਲ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ :

```
<!DOCTYPE html>
<html>
<head>
<title> </title>
</head>
<body>
<img src="flower.jpg" alt="Image of Flower" width="400" height="500">
</body>
</html>
```

ਚਿੱਤਰ 1.6

ਇਮੇਜ਼, ਲਿੰਕ ਬਟਨ ਵਜੋਂ ਵੀ ਵਰਤੀ ਜਾ ਸਕਦੀ ਹੈ। ਜੇ ਅਸੀਂ Image ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਲਿੰਕ ਬਣਾਉਣਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ <img> ਟੈਗ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ। ਇਸ ਲਈ ਇਮੇਜ਼ ਟੈਗ ਨੂੰ <a> ਅਤੇ ਟੈਗ ਦੇ ਵਿਚਕਾਰ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ। ਜਿਵੇਂ ਕਿ ਹੇਠਾਂ ਦਿੱਤੀ ਉਦਾਹਰਣ ਵਿੱਚ ਦਿਖਾਇਆ ਗਿਆ ਹੈ :

```
<a href= "http://pseb.ac.in"><img src= "pseb.jpg" width= "40" height= "20"></a>
```

1.8 ਸਟਾਈਲ ਸ਼ੀਟਸ (Style Sheets)

ਕੈਸਕੇਡਿੰਗ ਸਟਾਈਲ ਸ਼ੀਟ, ਜਿਸ ਨੂੰ CSS (Cascading Style Sheets) ਵੀ ਕਿਹਾ ਜਾਂਦਾ ਹੈ, ਇੱਕ ਸਧਾਰਨ ਡਿਜ਼ਾਈਨ ਭਾਸ਼ਾ ਹੈ ਜੋ ਵੈੱਬਪੇਜਾਂ ਦੇ ਡਿਜ਼ਾਈਨ ਨੂੰ ਸਧਾਰਨ ਅਤੇ ਪ੍ਰਭਾਵਸ਼ਾਲੀ ਬਣਾਉਣ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਜਿਥੇ CSS ਸਿੱਖਣਾ ਅਤੇ ਸਮਝਣਾ ਆਸਾਨ ਹੈ, ਉਥੇ ਇਹ ਮਜ਼ੇਦਾਰ ਵੀ ਹੈ। ਇਹ ਆਮ ਤੌਰ 'ਤੇ HTML ਨਾਲ ਵੈੱਬਸਾਈਟਸ ਬਣਾਉਣ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। CSS ਨਿਰਧਾਰਤ ਕਰਦਾ ਹੈ ਕਿ HTML Element ਕਿਵੇਂ ਪ੍ਰਦਰਸ਼ਿਤ ਕੀਤੇ ਜਾਣੇ ਚਾਹੀਦੇ ਹਨ।

CSS ਦੀ ਵਰਤੋਂ ਕਿਸੇ ਵੈੱਬਪੇਜ ਦੀ ਦਿੱਖ ਨੂੰ ਬਿਹਤਰ ਬਣਾਉਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਟੈਕਸਟ ਦਾ ਰੰਗ, ਫੌਂਟ ਸ਼੍ਰੇਣੀਆਂ, ਪੈਰਾਡਿਜ਼ਮ, ਕਾਲਮ, ਇਮੇਜ਼, ਲੇਆਉਟ ਡਿਜ਼ਾਈਨ, ਵੱਖ-ਵੱਖ ਡਿਵਾਈਸਾਂ ਅਤੇ ਸਕਰੀਨਜ਼ ਲਈ ਡਿਸਪਲੇਅ ਸਾਈਜ਼ ਅਤੇ ਹੋਰ ਬਹੁਤ ਸਾਰੇ ਡਿਜ਼ਾਈਨ ਅਤੇ ਡਿਜ਼ਾਈਨ Elements ਦੇ ਪ੍ਰਦਰਸ਼ਨ ਵਿਚ ਤਬਦੀਲੀਆਂ ਕਰਨ ਲਈ CSS ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

1.8.1 CSS ਦੇ ਫਾਇਦੇ (Benefits of CSS)

- **ਸਮੇਂ ਦੀ ਬਚਤ-** CSS ਕੋਡ ਨੂੰ ਜ਼ਰੂਰਤ ਅਨੁਸਾਰ ਵਾਰ-ਵਾਰ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ ਅਤੇ ਫਿਰ ਉਸੇ ਸ਼ੀਟ ਨੂੰ ਕਈ HTML ਪੰਨਿਆਂ ਵਿਚ ਦੁਬਾਰਾ ਇਸਤੇਮਾਲ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ।
- **ਸਪੀਡ (ਤੇਜ਼ੀ ਨਾਲ ਵੈੱਬਪੇਜ ਲੋਡ ਹੋਣਾ)-** CSS ਨੂੰ html ਵਿਚ ਫਾਰਮੈਟਿੰਗ ਕਰਨ ਲਈ ਘੱਟ ਕੋਡ ਦੀ ਜ਼ਰੂਰਤ ਹੁੰਦੀ ਹੈ, ਜਿਸ ਨਾਲ ਵੈੱਬਪੇਜ ਬਣਾਉਣ ਅਤੇ ਵੈੱਬ ਬ੍ਰਾਊਜ਼ਰ ਵਿਚ Load ਹੋਣ ਦੀ ਸਪੀਡ ਵੀ ਵੱਧ ਜਾਂਦੀ ਹੈ। ਇਸ ਨਾਲ ਵੈੱਬਪੇਜ ਦਾ ਆਕਾਰ ਵੀ ਘੱਟਦਾ ਹੈ ਅਤੇ ਵੈੱਬਪੇਜ ਹਲਕਾ (Light Design) ਦਾ ਬਣਦਾ ਹੈ।
- **ਆਸਾਨ ਰੱਖ-ਰਖਾਅ-** ਇੱਕ ਜਗ੍ਹਾ 'ਤੇ CSS ਕੋਡਿੰਗ ਵਿਚ ਕੀਤੀਆਂ ਗਈਆਂ ਤਬਦੀਲੀਆਂ ਆਪਣੇ ਆਪ ਹੀ ਹਰ ਜਗ੍ਹਾ 'ਤੇ ਲਾਗੂ ਹੋ ਜਾਂਦੀਆਂ ਹਨ, ਮਤਲਬ ਕਿ ਪੂਰੇ ਵੈੱਬਸਾਈਟ/ਵੈੱਬਪੇਜ ਦੇ ਭਾਗਾਂ (ਬਣਤਰ ਅਤੇ ਡਿਜ਼ਾਈਨ) ਨੂੰ ਬਦਲਿਆ ਜਾ ਸਕਦਾ ਹੈ ਅਤੇ ਇਸ ਤਰ੍ਹਾਂ ਮੁੱਖ ਵੈੱਬ ਕੋਡਿੰਗ ਨੂੰ ਬਦਲਣ ਦੀ ਜ਼ਰੂਰਤ ਨਹੀਂ ਹੈ ਜਿਸ ਨਾਲ ਗਲਤੀਆਂ ਹੋਣ ਦੀ ਸੰਭਾਵਨਾ ਵੀ ਘੱਟ ਜਾਂਦੀ ਹੈ।
- **ਵਧੇਰੇ ਸ਼ਕਤੀਸ਼ਾਲੀ-** ਵੈੱਬਪੇਜਾਂ ਨੂੰ CSS ਦੀ ਵਰਤੋਂ ਨਾਲ ਵਧੇਰੇ ਬਿਹਤਰ ਅਤੇ ਆਸਾਨ ਤਰੀਕੇ ਨਾਲ ਤਿਆਰ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। CSS ਵਿਚ HTML ਨਾਲੋਂ ਜ਼ਿਆਦਾ ਪ੍ਰਾਪਰਟੀਜ਼ ਹਨ।
- **ਮਲਟੀਪਲ ਡਿਵਾਈਸ ਸਪੋਰਟ-** CSS ਦੀ ਵਰਤੋਂ ਵੱਖ-ਵੱਖ ਸਕਰੀਨ ਸਾਈਜ਼ ਡਿਵਾਈਸਿਸ ਤੇ ਵੈੱਬਪੇਜਾਂ ਨੂੰ ਸਹੀ ਢੰਗ ਨਾਲ ਡਿਸਪਲੇਅ (ਦਿਖਾਉਣ) ਕਰਨ ਲਈ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ ਅਤੇ CSS ਦੀ ਵਰਤੋਂ ਕਰਦਿਆਂ, ਵੱਖੋ-ਵੱਖਰੇ ਡਿਵਾਈਸਿਸ ਲਈ ਵੈੱਬਸਾਈਟ ਦੇ ਵੱਖ-ਵੱਖ version ਬਣਾਏ ਜਾ ਸਕਦੇ ਹਨ।

1.9 ਸਟਾਈਲ ਸ਼ੀਟਸ ਦੀਆਂ ਕਿਸਮਾਂ (Types of Style Sheets)

CSS ਨੂੰ ਵਰਤਣ ਦੇ 3 ਤਰੀਕੇ ਹਨ :

ੳ. Inline style : ਇਹ HTML ਦੇ <BODY> ਸੈਕਸ਼ਨ ਵਿੱਚ ਟੈਗਜ਼ ਵਿਚ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਉਦਾਹਰਣ :

```
<p style = "color: red; border: 2px solid;"> This is an inline css example. </p>
```

ਅ. Internal style sheet : ਇਹ ਪੂਰੇ ਵੈੱਬਪੇਜ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ ਅਤੇ ਇਸਦਾ ਕੋਡ <head> ਟੈਗ ਵਿੱਚ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ, ਉਦਾਹਰਣ :

```
<head>
    <style> p {color: red; border: 2px solid; } </style>
</head>
```

ੲ. External style sheet : ਇਸਦੀ ਵਰਤੋਂ ਵੱਖਰੀ ਫਾਈਲ ਬਣਾ ਕੇ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਸਾਰੀ ਕੋਡਿੰਗ ਇਸ ਵੱਖਰੀ ਫਾਈਲ ਵਿੱਚ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਅਤੇ ਇਸ ਫਾਈਲ ਦੀ ਐਕਸਟੈਂਸ਼ਨ .css ਹੁੰਦੀ ਹੈ। ਇਹ ਫਾਈਲ ਦੇ ਰੂਪ ਵਿੱਚ ਬਣਾਈ ਜਾਂਦੀ ਹੈ ਅਤੇ ਇਸ ਫਾਈਲ ਨੂੰ ਅਲੱਗ-ਅਲੱਗ ਵੈੱਬਪੇਜਿਜ਼ ਨਾਲ ਲਿੰਕ ਕਰਕੇ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਜਦੋਂ ਇਸ ਫਾਈਲ ਵਿਚ ਤਬਦੀਲੀ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਤਾਂ ਸਾਰੇ ਲਿੰਕ ਕੀਤੇ ਵੈੱਬਪੇਜਾਂ 'ਤੇ ਇਹ ਤਬਦੀਲੀ ਦਿਖਾਈ ਦਿੰਦੀ ਹੈ। ਅਸੀਂ ਇਹ ਐਕਸਟਰਨਲ ਸਟਾਈਲ ਸ਼ੀਟ ਫਾਈਲ ਨੂੰ ਲਿੰਕ ਕਰਨ ਲਈ, ਵੈੱਬਪੇਜ ਦੇ <head> ਸੈਕਸ਼ਨ ਵਿਚ <link> ਟੈਗ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ। ਉਦਾਹਰਣ :

```
<head>
    <link rel = "stylesheet" type = "text / css" href = "mystyle.css">
</head>
```

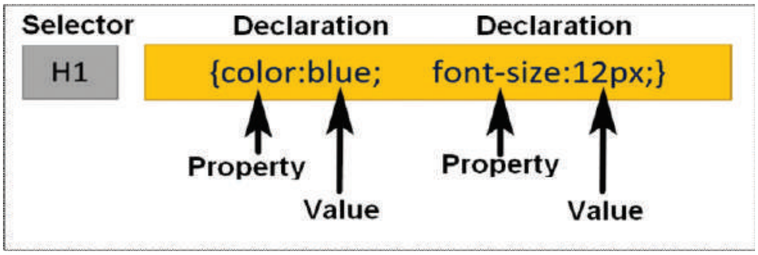
External CSS ਫਾਈਲਜ਼ ਕਿਸੇ ਵੀ ਟੈਕਸਟ ਐਡੀਟਰ ਵਿੱਚ ਬਣਾਈਆਂ ਜਾ ਸਕਦੀਆਂ ਹਨ। CSS ਫਾਈਲ ਵਿੱਚ ਕੋਈ ਵੀ html ਟੈਗ ਨਹੀਂ ਹੋਣੇ ਚਾਹੀਦੇ। ਸਟਾਈਲ ਸ਼ੀਟ ਫਾਈਲ ਦੀ ਉਦਾਹਰਣ :

CSS Filename: "myStyle.css":

```
p {
color: red; border:2px solid;
}
```

1.10 CSS SYNTAX

CSS rule-set ਵਿੱਚ ਇੱਕ selector ਅਤੇ declaration block ਹੁੰਦਾ ਹੈ :



ਚਿੱਤਰ 1.7

- **CSS Selector :** ਇਹ ਉਸ ਐਲੀਮੈਂਟ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ਜਿਸਦਾ style ਅਸੀਂ ਫਾਰਮੈਟ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ।
- **CSS Declaration :** ਇਹ ਐਲੀਮੈਂਟ ਦੀਆਂ ਪ੍ਰਾਪਰਟੀਜ਼ ਨੂੰ ਨਿਰਧਾਰਤ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ ਅਤੇ ਹਰੇਕ ਪ੍ਰਾਪਰਟੀ ਨੂੰ ਸੈਮੀਕਾਲਨ (;) ਨਾਲ End ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਇੱਥੇ ਬਹੁਤ ਸਾਰੀਆਂ ਪ੍ਰਾਪਰਟੀਜ਼ ਹੁੰਦੀਆਂ ਹਨ ਜੋ ਡਿਕਲੇਰੇਸ਼ਨ ਬਲਾਕ ਵਿੱਚ ਵਰਤੀਆਂ ਜਾ ਸਕਦੀਆਂ ਹਨ। ਇਸਨੂੰ HTML ਟੈਗ ਐਟਰੀਬਿਊਟ ਦੇ ਤੌਰ 'ਤੇ ਇੰਟਰਪ੍ਰੈਟ (Interpret) ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ।

1.11 CSS ਸਿਲੈਕਟਰਜ਼

ਸਿਲੈਕਟਰਜ਼ HTML ਐਲੀਮੈਂਟਸ ਨੂੰ ਚੁਣਨ ਅਤੇ ਫਾਰਮੈਟ ਕਰਨ ਵਿੱਚ ਸਾਡੀ ਸਹਾਇਤਾ ਕਰਦੇ ਹਨ ਅਤੇ id, classes, types, attributes ਆਦਿ Selectors ਵੱਜੋਂ ਵਰਤੇ ਜਾ ਸਕਦੇ ਹਨ।

1.11.1 ਐਲੀਮੈਂਟ ਸਿਲੈਕਟਰ

ਐਲੀਮੈਂਟ ਸਿਲੈਕਟਰ, html ਐਲੀਮੈਂਟ ਨੂੰ ਚੁਣਨ ਜਾਂ ਫਾਰਮੈਟ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ ਜਿਵੇਂ ਕਿ ਪੈਰਾਗ੍ਰਾਫ। ਇੱਕ ਪੈਰਾਗ੍ਰਾਫ ਲਈ html ਵੈੱਬਪੇਜ ਵਿੱਚ ਹੇਠ ਲਿਖੀ ਉਦਾਹਰਣ ਨੂੰ ਸਮਝੋ :

```
<p> This is an example of paragraph in web page</p>
```

ਜੇ ਅਸੀਂ ਇਸ ਨੂੰ Element Selector ਦੀ ਸਹਾਇਤਾ ਨਾਲ CSS ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਫਾਰਮੈਟ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ, ਤਾਂ ਸਾਨੂੰ ਇਸਨੂੰ ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ ਲਿਖਣਾ ਪਏਗਾ (ਜਾਂ ਤਾਂ Internal CSS ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਜਾਂ External ਦੀ ਵਰਤੋਂ ਕਰਕੇ) :

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

1.11.2 ID ਸਿਲੈਕਟਰ

ਇਹ HTML ਦੇ id ਐਟਰੀਬਿਊਟ ਨੂੰ ਫਾਰਮੈਟ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਅਸੀਂ ਇਸ ਸਿਲੈਕਟਰ ਦੀ ਵਰਤੋਂ ਕਿਸੇ ਵਿਸ਼ੇਸ਼ ਐਲੀਮੈਂਟ ਨੂੰ ਫਾਰਮੈਟ ਕਰਨ ਲਈ ਕਰਦੇ ਹਾਂ। ਇਹ # (ਹੈਸ਼) ਚਿੰਨ੍ਹ ਅਤੇ ਐਲੀਮੈਂਟ ਦੇ ਨਾਮ ਨਾਲ ਸ਼ੁਰੂ ਹੁੰਦਾ ਹੈ। ਇੱਕ ਪੈਰਾਗ੍ਰਾਫ ਲਈ html ਵੈੱਬਪੇਜ ਵਿੱਚ ਹੇਠ ਲਿਖੀ ਉਦਾਹਰਣ ਵੇਖੋ :

```
<p id="#para1"> This is an example of paragraph in web page</p>
```

ਜੇ ਅਸੀਂ ਇਸ ਨੂੰ id ਸਿਲੈਕਟਰ ਦੀ ਸਹਾਇਤਾ ਨਾਲ CSS ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਫਾਰਮੈਟ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ, ਤਾਂ ਸਾਨੂੰ ਇਸਨੂੰ ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ ਲਿਖਣਾ ਪਏਗਾ (ਜਾਂ ਤਾਂ Internal CSS ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਜਾਂ External CSS ਦੀ ਵਰਤੋਂ ਕਰਕੇ) :

```
# para1 {
      text-align: center;
      color: red;
}
```

1.11.3 ਕਲਾਸ ਸਿਲੈਕਟਰ

ਇਹ ਐਲੀਮੈਂਟ ਦੇ ਕਲਾਸ ਐਟਰੀਬਿਊਟ ਨੂੰ ਫਾਰਮੈਟ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਬਿੰਦੀ (.) ਅਤੇ CLASS ਦੇ ਨਾਮ ਨਾਲ ਸ਼ੁਰੂ ਹੁੰਦਾ ਹੈ। Html ਵੈੱਬਪੇਜ ਵਿੱਚ ਹੇਠ ਲਿਖੀ ਉਦਾਹਰਣ ਨੂੰ ਸਮਝੋ :

```
<p class= "center"> This is an example of paragraph in web page</p>
<h1class= "center"> This is an example of Heading Level 1</h1>
<p class= "center"> This is another example of paragraph in web page</p>
```

ਜੇ ਅਸੀਂ CLASS ਸਿਲੈਕਟਰ ਦੀ ਸਹਾਇਤਾ ਨਾਲ ਇਨ੍ਹਾਂ ਦੋਵਾਂ ਐਲੀਮੈਂਟਸ ਨੂੰ ਇਕੋ ਜਿਹੇ ਫਾਰਮੈਟ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ, ਤਾਂ ਸਾਨੂੰ ਇਸਨੂੰ ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ ਲਿਖਣਾ ਪਵੇਗਾ (ਜਾਂ ਤਾਂ Internal CSS ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਜਾਂ External CSS ਦੀ ਵਰਤੋਂ ਕਰਕੇ)

```
.center {
      text-align: center;
      color: red;
}
```

1.11.4 ਗਰੁੱਪਿੰਗ ਸਿਲੈਕਟਰਜ਼

ਇਹ ਕੋਡਿੰਗ ਨੂੰ ਘਟਾਉਣ ਅਤੇ ਸਮਾਂ ਬਚਾਉਣ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਕਈ ਐਲੀਮੈਂਟਸ ਨੂੰ ਇਕੋ ਸਮੂਹ ਵਿਚ ਇਕੱਠਿਆਂ ਕਰਕੇ ਇਕੋ ਜਿਹਾ ਫਾਰਮੈਟ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ। HTML ਵੈੱਬਪੇਜ ਵਿੱਚ ਹੇਠ ਲਿਖੀ ਉਦਾਹਰਣ 'ਤੇ ਵਿਚਾਰ ਕਰੋ :

```
<p> This is an example of paragraph in web page</p>
<h1> This is an example of Heading Level 1</h1>
<h2> This is an example of Heading Level 2</h2>
```

ਜੇ ਅਸੀਂ Grouping Selector ਦੀ ਸਹਾਇਤਾ ਨਾਲ ਇਨ੍ਹਾਂ ਸਾਰੇ ਐਲੀਮੈਂਟਸ ਨੂੰ ਇਕੋ ਜਿਹਾ ਫਾਰਮੈਟ ਕਰਨਾ ਚਾਹੁੰਦੇ ਹਾਂ ਤਾਂ ਸਾਨੂੰ ਇਸਨੂੰ ਹੇਠਾਂ ਦਿੱਤੇ ਅਨੁਸਾਰ ਲਿਖਣਾ ਪਵੇਗਾ (ਜਾਂ ਤਾਂ Internal CSS ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਜਾਂ External CSS ਦੀ ਵਰਤੋਂ ਕਰਕੇ) :

```
h1, h2, p {
        text-align: center;
        color: red;
}
```

1.12 CSS ਦੀਆਂ ਅਲੱਗ-ਅਲੱਗ ਪ੍ਰਾਪਰਟੀਜ਼ (Different Style Sheet Properties)

ਹੁਣ ਸਾਨੂੰ ਪਤਾ ਲੱਗ ਗਿਆ ਹੈ ਕਿ ਵੈੱਬਪੇਜਾਂ ਵਿਚ CSS ਦੀ ਵਰਤੋਂ ਕਿਵੇਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਹੁਣ ਅਸੀਂ CSS ਦੀਆਂ ਵੱਖੋ-ਵੱਖਰੀਆਂ Properties ਬਾਰੇ ਜਾਣਕਾਰੀ ਹਾਸਿਲ ਕਰਾਂਗੇ। ਇਹ ਪ੍ਰਾਪਰਟੀਜ਼ ਵੈੱਬਪੇਜਾਂ ਦੇ ਵੱਖ-ਵੱਖ ਐਲੀਮੈਂਟਸ ਨੂੰ ਫਾਰਮੈਟ ਕਰਨ ਲਈ ਵਰਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਵੈੱਬਪੇਜਾਂ ਵਿੱਚ ਵਰਤੋਂ ਕਰਨ ਲਈ ਬਹੁਤ ਸਾਰੀਆਂ CSS ਪ੍ਰਾਪਰਟੀਜ਼ ਹਨ। ਅਸੀਂ ਉਨ੍ਹਾਂ ਨੂੰ ਵੱਖ-ਵੱਖ ਸ਼੍ਰੇਣੀਆਂ ਵਿੱਚ ਸ਼੍ਰੇਣੀਬੱਧ ਕਰਕੇ ਸਮਝਾਂਗੇ।

1.12.1 CSS Background

ਬੈਕਗ੍ਰਾਊਂਡ ਪ੍ਰਾਪਰਟੀਜ਼ ਦੀ ਵਰਤੋਂ ਵੈੱਬਪੇਜਾਂ ਜਾਂ ਕਈ HTML ਐਲੀਮੈਂਟਸ ਦੇ ਬੈਕਗ੍ਰਾਊਂਡ ਨੂੰ ਫਾਰਮੈਟ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਹੇਠ ਦਿੱਤੀ ਸਾਰਣੀ ਬੈਕਗ੍ਰਾਊਂਡ ਦੀਆਂ ਸਾਂਝੀਆਂ ਪ੍ਰਾਪਰਟੀਜ਼ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ :

ਪ੍ਰਾਪਰਟੀ	ਵਿਆਖਿਆ
background	ਇਹ ਪ੍ਰਾਪਰਟੀ ਇੱਕ ਡਿਕਲੇਰੇਸ਼ਨ ਵਿਚ ਬੈਕਗ੍ਰਾਊਂਡ ਦੀਆਂ ਸਾਰੀਆਂ ਪ੍ਰਾਪਰਟੀਜ਼ ਨੂੰ ਪ੍ਰਭਾਸ਼ਿਤ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
background-color	ਇਹ ਪ੍ਰਾਪਰਟੀ ਐਲੀਮੈਂਟ ਦਾ ਬੈਕਗ੍ਰਾਊਂਡ ਕਲਰ (ਰੰਗ) ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
background-image	ਇਹ ਪ੍ਰਾਪਰਟੀ ਐਲੀਮੈਂਟ ਦੀ ਬੈਕਗਰਾਊਂਡ ਇਮੇਜ਼ ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
background-position	ਇਹ ਪ੍ਰਾਪਰਟੀ ਐਲੀਮੈਂਟ ਦੀ ਬੈਕਗ੍ਰਾਊਂਡ ਇਮੇਜ਼ ਦੀ ਪੁਜ਼ੀਸ਼ਨ ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
background-repeat	ਇਹ ਪ੍ਰਾਪਰਟੀ ਐਲੀਮੈਂਟ ਦੀ ਬੈਕਗ੍ਰਾਊਂਡ ਇਮੇਜ਼ ਦੀ ਰੀਪੀਟ (Repeat) ਪ੍ਰਾਪਰਟੀ ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।

ਉਦਾਹਰਣਾਂ :

```
p{
background-color: #b0c4de;
}
body {
background-image: url("paper.gif");
```

```
background-repeat: repeat-x;
background-position: right top;
}
```

ਵੱਖਰੀਆਂ ਬੈਕਗ੍ਰਾਊਂਡ ਪ੍ਰਾਪਰਟੀਜ਼ ਇਕੱਠੀਆਂ ਵੀ ਲਿਖੀਆਂ ਜਾ ਸਕਦੀਆਂ ਹਨ :

```
body {
background: #ffffff url ("plant.png") no-repeat right top;
}
```

1.12.2 CSS ਟੈਕਸਟ (CSS Text)

ਟੈਕਸਟ ਪ੍ਰਾਪਰਟੀਜ਼ ਦੀ ਵਰਤੋਂ ਵੈੱਬਪੇਜ 'ਤੇ ਟੈਕਸਟ ਨੂੰ ਫਾਰਮੈਟ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਹੇਠਾਂ ਦਿੱਤੀ ਸਾਰਣੀ ਟੈਕਸਟ ਦੀਆਂ ਸਾਂਝੀਆਂ ਪ੍ਰਾਪਰਟੀਜ਼ ਨੂੰ ਦਰਸਾਉਂਦੀ ਹੈ :

ਪ੍ਰਾਪਰਟੀ	ਵਿਆਖਿਆ
color	ਇਹ ਪ੍ਰਾਪਰਟੀ ਟੈਕਸਟ ਦਾ ਰੰਗ ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
text-align	ਇਹ ਪ੍ਰਾਪਰਟੀ ਟੈਕਸਟ ਦੀ ਅਲਾਈਨਮੈਂਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
Text-decoration	ਇਹ ਪ੍ਰਾਪਰਟੀ ਟੈਕਸਟ ਡੈਕੋਰੇਸ਼ਨ (decoration) ਨੂੰ ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਇਸ ਵਿਚ ਵਰਤੀਆਂ ਜਾਣ ਵਾਲੀਆਂ ਆਮ values ਹਨ : overline, underline, line-through

ਅਸੀਂ ਰੰਗ ਦੀ value ਅਨੇਕਾਂ ਤਰ੍ਹਾਂ ਲਿਖ ਸਕਦੇ ਹਾਂ, ਉਦਾਹਰਣ :

HEX value HEX value : ਉਦਾਹਰਣ : #ff0000
Using RGB value ਵਰਤ ਕੇ : ਉਦਾਹਰਣ : rgb(255,0,0)
Using color name ਵਰਤ ਕੇ : ਉਦਾਹਰਣ : red

ਵੈੱਬਪੇਜਾਂ ਵਿਚਲੇ ਟੈਕਸਟ ਨੂੰ By default left ਅਲਾਈਨ ਕੀਤਾ ਜਾਂਦਾ ਹੈ। ਅਸੀਂ ਆਪਣੀ ਜ਼ਰੂਰਤ ਦੇ ਅਨੁਸਾਰ ਟੈਕਸਟ ਨੂੰ ਅਲਾਈਨ ਵੀ ਕਰ ਸਕਦੇ ਹਾਂ। ਟੈਕਸਟ ਨੂੰ horizontally ਰੂਪ ਵਿੱਚ ਬਦਲਣ ਲਈ ਇੱਥੇ ਚਾਰ ਵਿਕਲਪ ਹਨ : center, left, right, justified. ਹੇਠ ਦਿੱਤੀ ਉਦਾਹਰਣ ਦਰਸਾਉਂਦੀ ਹੈ ਕਿ ਵੈੱਬਪੇਜਾਂ ਵਿੱਚ ਟੈਕਸਟ Properties ਨੂੰ ਕਿਵੇਂ ਵਰਤਣਾ ਹੈ :

ਉਦਾਹਰਣ:

```
h1 {
color: red;
text-align: center;
text-decoration: underline;
}
```

1.12.3 CSS ਫੌਂਟਸ (CSS Fonts)

ਇਸ ਪ੍ਰਾਪਰਟੀ ਦੀ ਮਦਦ ਨਾਲ ਅਸੀਂ ਫੌਂਟ ਦਾ ਸਟਾਈਲ ਅਤੇ ਸਾਈਜ਼ ਸੈੱਟ ਕਰ ਸਕਦੇ ਹਾਂ।

ਪ੍ਰਾਪਰਟੀ	ਵਿਆਖਿਆ
Font	ਇਹ ਪ੍ਰਾਪਰਟੀ ਫੌਂਟ ਦੀਆਂ ਸਾਰੀਆਂ ਪ੍ਰਾਪਰਟੀਜ਼ ਨੂੰ ਇੱਕ ਡਿਕਲੇਰੇਸ਼ਨ ਵਿਚ ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
Font-family	ਇਹ ਪ੍ਰਾਪਰਟੀ ਫੌਂਟ-ਫੈਮਲੀ ਨੂੰ ਸੈਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।

Font-size	ਇਹ ਪ੍ਰਾਪਰਟੀ ਫੌਂਟ ਦਾ ਸਾਈਜ਼ ਸੈਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਇਸ ਵਿਚ px ਜਾਂ em unit ਦੀ ਵਰਤੋਂ ਹੋ ਸਕਦੀ ਹੈ। World Wide Web Consortium (W3C) ਦੁਆਰਾ ਫੌਂਟ ਦਾ ਸਾਈਜ਼ em ਯੂਨਿਟਸ ਵਿਚ ਸੈਟ ਕਰਨ ਦੀ ਸਿਫਾਰਸ਼ ਕੀਤੀ ਗਈ ਹੈ। ਇਸ ਯੂਨਿਟ ਅਨੁਸਾਰ 1 em ਦਾ ਮਤਲਬ 16 px ਹੈ ਅਤੇ ਇਹ ਸਾਈਜ਼ ਵੈੱਬ ਬ੍ਰਾਊਜ਼ਰ ਦਾ ਵੀ ਡਿਫਾਲਟ ਫੌਂਟ ਸਾਈਜ਼ ਹੁੰਦਾ ਹੈ।
Front-style	ਇਹ ਪ੍ਰਾਪਰਟੀ ਫੌਂਟ ਦੇ ਸਟਾਈਲ ਨੂੰ ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ ਜਿਵੇਂ normal, italic ਜਾਂ oblique
Font-weight	ਇਹ ਪ੍ਰਾਪਰਟੀ ਫੌਂਟ ਦਾ Weight ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਇਸ ਵਿਚ ਵਰਤੀਆਂ ਜਾਣ ਵਾਲੀਆਂ ਆਮ values ਹਨ : normal, bold, bolder, lighter, number (ਕੋਈ ਵੀ ਅੰਕਾਂ ਵਾਲੀ ਕੀਮਤ ਜਿਵੇਂ 900)
Font-variant	ਇਹ ਪ੍ਰਾਪਰਟੀ ਫੌਂਟ ਦਾ ਵੈਰੀਏਂਟ (variant) ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ ਜਿਵੇਂ normal, small-caps

ਉਦਾਹਰਣ :

```
p {
font-style: italic;
font-weight: bold;
font-size: 12px;
}
```

ਹੇਠਾਂ ਦਿੱਤੀ ਉਦਾਹਰਣ ਦਰਸਾਉਂਦੀ ਹੈ ਕਿ h1 ਹੈਡਿੰਗ ਤੇ Font Property ਦੀ Shorthand ਡਿਕਲੇਰੇਸ਼ਨ ਕਿਵੇਂ ਵਰਤੀ ਜਾਵੇ। ਇੱਥੇ, ਫੌਂਟ ਨੂੰ ਇਟੈਲਿਕ ਅਤੇ ਬੋਲਡ ਸੈੱਟ ਕੀਤਾ ਗਿਆ ਹੈ, ਫੌਂਟ ਦਾ ਅਕਾਰ 12 ਪਿਕਸਲ ਸੈੱਟ ਕੀਤਾ ਗਿਆ ਹੈ, ਲਾਈਨ 30 ਪਿਕਸਲ ਸੈੱਟ ਕੀਤੀ ਗਈ ਹੈ ਅਤੇ Font Family Georgia ਵਿੱਚ ਸੈੱਟ ਕੀਤੀ ਗਈ ਹੈ।

```
h1 {
font: italic bold 12px/30px Georgia, serif;
}
```

1.12.4 CSS ਲਿੰਕਸ (CSS Links)

ਇਹ CSS ਪ੍ਰਾਪਰਟੀਜ਼ ਹਾਈਪਰਲਿੰਕਸ ਨੂੰ Format ਕਰਨ ਲਈ ਵਰਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਵੈੱਬਪੇਜ਼ ਵਿੱਚ ਆਮ ਕਰਕੇ ਚਾਰ ਟਾਈਪਸ ਦੇ ਲਿੰਕ ਹਨ :

a: link – ਇਹ ਆਮ ਅਤੇ unvisited ਲਿੰਕ ਲਈ ਹੈ।

a: visited – ਇਹ ਯੂਜ਼ਰ ਦੁਆਰਾ ਵਿਜ਼ਟ ਕੀਤੇ ਜਾ ਚੁੱਕੇ ਲਿੰਕ ਲਈ ਹੈ।

a: hover – ਇਹ ਯੂਜ਼ਰ ਦੁਆਰਾ ਜਦੋਂ ਮਾਊਸ ਪੁਆਇੰਟਰ ਲਿੰਕ ’ਤੇ ਲੈਕੇ ਜਾਇਆ ਜਾਂਦਾ ਹੈ, ਲਈ ਹੈ।

a: active – ਇਹ ਉਸ ਸਮੇਂ ਲਈ ਹੈ ਜਦੋਂ ਲਿੰਕ ਕਲਿੱਕ ਕੀਤਾ ਜਾਂਦਾ ਹੈ।

ਉਦਾਹਰਣ :

```
a: link {color: # FF0000; text-decoration: none;}
a: visited {color: # 00FF00; text-decoration: none;}
a: hover {color: # FF00FF; text-decoration: underline}
a: active {color: # 0000FF; text-decoration: underline}
```

1.12.5 CSS ਲਿਸਟਾਂ (CSS Lists)

ਇਹ ਪ੍ਰਾਪਰਟੀ, ਲਿਸਟਾਂ ਨੂੰ ਫਾਰਮੈਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਲਿਸਟ ਵਿੱਚ ਆਈਟਮ ਦੇ ਚਿੰਨ੍ਹ (Symbol) ਜਾਂ ਨੰਬਰ ਜਾਂ Character ਨੂੰ ਆਈਟਮ ਮਾਰਕਰ ਵਜੋਂ ਦਰਸਾਇਆ ਗਿਆ ਹੈ। Item ਮਾਰਕਰ ਦੀਆਂ ਉਦਾਹਰਣਾਂ Circle, Square, Disc, Upper-Roman, Lower-Roman, Lower-Alpha, Upper-Alpha ਹਨ।

ਉਦਾਹਰਣ :

```
ul {
list-style-type: circle;
}
```

List-Style-Image ਪ੍ਰਾਪਰਟੀ ਇਮੇਜ਼ ਨੂੰ Item Marker ਵਜੋਂ ਇਸਤੇਮਾਲ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾ ਸਕਦੀ ਹੈ।

ਉਦਾਹਰਣ :

```
ul {
list-style-image: url ('arrow.gif');
}
```

1.12.6 CSS ਬਾਰਡਰ (CSS Border)

ਇਹ ਪ੍ਰਾਪਰਟੀ, ਐਲੀਮੈਂਟਸ ਦੇ ਦੁਆਲੇ ਬਾਰਡਰ ਸੈੱਟ ਕਰਨ ਅਤੇ ਫਾਰਮੈਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਅਸੀਂ ਇਸ ਪ੍ਰਾਪਰਟੀ ਦੀ ਵਰਤੋਂ ਕਰਦਿਆਂ html ਐਲੀਮੈਂਟਸ ਲਈ ਬਾਰਡਰ ਦੀ style, width ਅਤੇ color ਨਿਰਧਾਰਿਤ ਕਰ ਸਕਦੇ ਹਾਂ। ਇਹ ਤਿੰਨ ਪ੍ਰਾਪਰਟੀਜ਼ ਬਾਰਡਰ ਨੂੰ ਫਾਰਮੈਟ ਕਰਨ ਲਈ ਵਰਤੀਆਂ ਜਾਂਦੀਆਂ ਹਨ :

ਪ੍ਰਾਪਰਟੀ	ਵਿਆਖਿਆ
border-style	ਇਹ ਪ੍ਰਾਪਰਟੀ ਬਾਰਡਰ ਦਾ ਸਟਾਈਲ ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਇਸ ਦੀਆਂ ਆਮ ਵਰਤੀਆਂ ਜਾਣ ਵਾਲੀਆਂ values ਹਨ : none, dotted, dashed, solid, double, groove, ridge, inset ਅਤੇ outset
border-width	ਇਹ ਪ੍ਰਾਪਰਟੀ ਬਾਰਡਰ ਦੀ ਚੌੜਾਈ ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
border-color	ਇਹ ਪ੍ਰਾਪਰਟੀ ਬਾਰਡਰ ਦਾ ਰੰਗ ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
border	ਇਹ ਪ੍ਰਾਪਰਟੀ ਬਾਰਡਰ ਦੀਆਂ ਸਾਰੀਆਂ ਪ੍ਰਾਪਰਟੀਜ਼ ਨੂੰ ਇੱਕ ਡਿਕਲੇਰੇਸ਼ਨ ਵਿਚ ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।

ਉਦਾਹਰਣ :

```
p {
border-style: solid;
border-width: 5px;
border-color: blue;
}
```

ਹੇਠਾਂ ਦਿੱਤੀ ਉਦਾਹਰਣ ਇੱਕ ਡਿਕਲੇਰੇਸ਼ਨ ਵਿੱਚ ਸਾਰੀਆਂ ਬਾਰਡਰ ਪ੍ਰਾਪਰਟੀਜ਼ ਦੀ ਵਰਤੋਂ ਦਰਸਾਉਂਦੀ ਹੈ। ਹੇਠਾਂ ਦਿੱਤੀ ਉਦਾਹਰਣ ਵਿੱਚ, 5 px ਬਾਰਡਰ ਦੀ width ਹੈ, ਬਾਰਡਰ ਦਾ ਸਟਾਈਲ solid ਹੈ ਅਤੇ ਬਾਰਡਰ ਦਾ ਰੰਗ ਲਾਲ ਹੈ:

```
h1 {
border: 5px solid red;
}
```

ਬਾਰਡਰ ਦੀਆਂ ਚਾਰੇ ਸਾਈਡਜ਼ ਵੀ CSS ਵਿੱਚ ਵੱਖਰੇ ਢੰਗ ਨਾਲ ਫਾਰਮੈਟ ਕੀਤੀਆਂ ਜਾ ਸਕਦੀਆਂ ਹਨ ਉਦਾਹਰਣ:

```
p {
   border-top-style: dotted;
   border-right-style: solid;
   border-bottom-style: dashed;
   border-left-style: dotted;
}
```

1.12.7 CSS ਮਾਰਜ਼ਨ (CSS Margin)

ਇਹ CSS ਪ੍ਰਾਪਰਟੀ ਐਲੀਮੈਂਟ ਦੇ ਦੁਆਲੇ (around) ਦੀ ਜਗ੍ਹਾ ਨਿਰਧਾਰਿਤ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। ਇਸ ਜਗ੍ਹਾ ਨੂੰ ਮਾਰਜਨ (ਹਾਸ਼ੀਆ) ਕਿਹਾ ਜਾਂਦਾ ਹੈ। ਹਾਸ਼ੀਏ (Margin) ਦਾ ਕੋਈ ਰੰਗ ਨਹੀਂ ਹੁੰਦਾ, ਇਹ ਪਾਰਦਰਸ਼ੀ ਹੁੰਦੇ ਹਨ। ਇਹ ਬਾਰਡਰ ਦੇ ਬਾਹਰਲੇ ਪਾਸੇ ਹੁੰਦੇ ਹਨ। ਹਾਸ਼ੀਏ (Margin) ਨੂੰ ਨਿਰਧਾਰਤ ਕਰਨ ਲਈ ਇੱਥੇ ਚਾਰ ਨਿਰਦੇਸ਼ ਹਨ : ਖੱਬੇ, ਉੱਪਰ, ਸੱਜੇ, ਹੇਠਾਂ :

ਪ੍ਰਾਪਰਟੀ	ਵਿਆਖਿਆ
Margin	ਇਹ ਪ੍ਰਾਪਰਟੀ ਸਾਰੀਆਂ ਮਾਰਜ਼ਨ ਪ੍ਰਾਪਰਟੀਜ਼ ਨੂੰ ਇੱਕ ਡਿਕਲੇਰੇਸ਼ਨ ਵਿਚ ਸੈਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
Margin-left	ਇਹ ਪ੍ਰਾਪਰਟੀ ਐਲੀਮੈਂਟ ਦੇ ਲੈਫ਼ਟ ਮਾਰਜ਼ਨ ਨੂੰ ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
Margin-right	ਇਹ ਪ੍ਰਾਪਰਟੀ ਐਲੀਮੈਂਟ ਦੇ ਸੱਜੇ (Right) ਮਾਰਜ਼ਨ ਨੂੰ ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
Margin-top	ਇਹ ਪ੍ਰਾਪਰਟੀ ਐਲੀਮੈਂਟ ਦੇ ਟਾਪ ਮਾਰਜ਼ਨ ਨੂੰ ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
Margin-bottom	ਇਹ ਪ੍ਰਾਪਰਟੀ ਐਲੀਮੈਂਟ ਦੇ ਹੇਠਲੇ (Bottom) ਮਾਰਜ਼ਨ ਨੂੰ ਸੈੱਟ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।

ਉਦਾਹਰਣ :

```
p {
margin-left: 40px; margin-right: 60px;
margin-bottom: 50px; margin-top: 70px;
}
```

ਉਦਾਹਰਣ : (Margin - Shorthand property) :

```
margin: 50px 60px 70px 80px;     (margins for top, right, bottom, and left side)
margin: 50px 60px 70px;          (margin for top, left and right, bottom side)
margin: 50px 60px;               (margin for top and bottom, left and right side)
margin: 50px;                    (margin for all four sides)
```

1.12.8 CSS ਪੈਡਿਗ (CSS Padding)

ਇਹ ਪ੍ਰਾਪਰਟੀ, ਕੰਟੈਂਟ ਅਤੇ ਐਲੀਮੈਂਟ ਦੇ ਬਾਰਡਰ ਵਿਚਕਾਰ ਜਗ੍ਹਾ ਨਿਰਧਾਰਿਤ ਕਰਨ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ। Padding ਸੈੱਟ ਕਰਨ ਲਈ ਇੱਥੇ ਚਾਰ ਨਿਰਦੇਸ਼ ਹਨ : ਖੱਬੇ, ਉੱਪਰ, ਸੱਜੇ, ਹੇਠਾਂ :

ਪ੍ਰਾਪਰਟੀ	ਵਿਆਖਿਆ
Padding	ਇਹ ਪ੍ਰਾਪਰਟੀ ਇੱਕ ਡਿਕਲੇਰੇਸ਼ਨ ਵਿਚ Padding ਦੀਆਂ ਪ੍ਰਾਪਰਟੀਜ਼ ਲਿਖਣ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
Padding-left	ਇਹ ਪ੍ਰਾਪਰਟੀ HTML ਐਲੀਮੈਂਟ ਦੇ ਖੱਬੇ ਪਾਸੇ Padding ਦਰਸਾਉਣ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।

Padding-right	ਇਹ ਪ੍ਰਾਪਰਟੀ HTML ਐਲੀਮੈਂਟ ਦੇ ਸੱਜੇ ਪਾਸੇ Padding ਦਰਸਾਉਣ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
Padding-top	ਇਹ ਪ੍ਰਾਪਰਟੀ HTML ਐਲੀਮੈਂਟ ਦੇ ਉੱਪਰਲੇ (Top) ਪਾਸੇ Padding ਦਰਸਾਉਣ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।
Padding-bottom	ਇਹ ਪ੍ਰਾਪਰਟੀ HTML ਐਲੀਮੈਂਟ ਦੇ ਹੇਠਲੇ (Bottom) ਪਾਸੇ Padding ਦਰਸਾਉਣ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ।

ਉਦਾਹਰਣ :

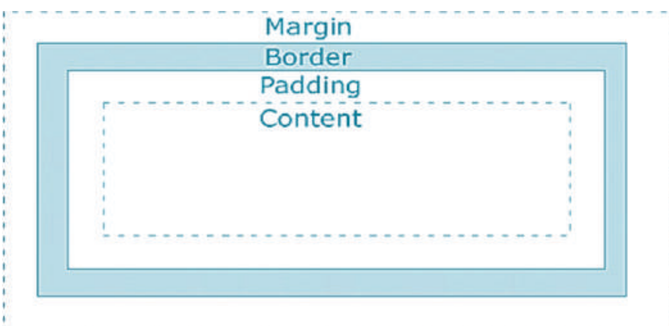
```
p {
padding-top: 25px; padding-bottom: 25px;
padding-right: 50px; padding-left: 50px;
}
```

ਉਦਾਹਰਣ (Padding-Shorthand Property) :

p {padding: 50px 60px 70px 80px;}	("ਉੱਪਰਲੀ", "ਸੱਜੀ", "ਹੇਠਲੀ" ਅਤੇ "ਖੱਬੀ" ਸਾਈਡ ਲਈ Padding)
p {padding: 50px 60px 70px;}	(ਉੱਪਰਲੀ, "ਖੱਬੀ ਅਤੇ ਸੱਜੀ", "ਹੇਠਲੀ" ਸਾਈਡ ਲਈ Padding)
p {padding: 50px 60px;}	("ਉੱਪਰਲੀ ਅਤੇ ਹੇਠਲੀ", "ਖੱਬੀ ਅਤੇ ਸੱਜੀ" ਸਾਈਡ ਲਈ Padding)
p {padding: 50px;}	(ਚਾਰੋਂ ਸਾਈਡਾਂ ਲਈ Padding)

1.12.9 CSS ਬਾਕਸ ਮਾਡਲ (CSS Box Model)

ਟਰਮ Box Model ਦੀ ਵਰਤੋਂ ਵੈੱਬਪੇਜਾਂ ਦੇ ਡਿਜ਼ਾਈਨ ਅਤੇ ਲੇਆਊਟ ਦੇ ਸੰਬੰਧ ਵਿੱਚ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। HTML ਦਾ ਹਰੇਕ Element/Part ਇੱਕ ਤਰ੍ਹਾਂ ਨਾਲ ਇੱਕ Box ਹੁੰਦਾ ਹੈ। Web ਬ੍ਰਾਊਜ਼ਰ ਹਰ Element ਨੂੰ CSS Box Model ਦੇ ਅਨੁਸਾਰ ਇੱਕ Rectangle Box ਵਜੋਂ ਪੇਸ਼ ਕਰਦਾ ਹੈ। Box Model ਵਿੱਚ ਕਈ ਐਟਰੀਬਿਊਟ ਹੁੰਦੇ ਹਨ। ਇਸ ਵਿੱਚ ਸ਼ਾਮਿਲ ਹਨ : ਹਾਸ਼ੀਏ (Margin), Border, Padding, ਅਤੇ Actual Content, ਇਨਾਂ ਨੂੰ ਬਾਕਸ ਮਾਡਲ ਰੂਪ ਵਿਚ ਚਿੱਤਰ-1.8 ਵਿਚ ਦਰਸਾਇਆ ਗਿਆ ਹੈ :



ਚਿੱਤਰ 1.8

ਇਹਨਾਂ Parts (ਭਾਗਾਂ) ਦੀ ਵਿਆਖਿਆ ਇਸ ਪ੍ਰਕਾਰ ਹੈ :

ਭਾਗ	ਵਿਆਖਿਆ
Content	ਇਸ ਏਰੀਏ ਵਿਚ ਬਾਕਸ ਦੇ ਕੰਟੈਂਟ ਨੂੰ ਦਿਖਾਇਆ ਜਾਂਦਾ ਹੈ। ਇਹ ਕੰਟੈਂਟ ਟੈਕਸਟ, ਤਸਵੀਰ ਜਾਂ ਫੇਰ ਕੋਈ ਹੋਰ ਮੀਡੀਆ (Object) ਹੋ ਸਕਦਾ ਹੈ।
Padding	ਇਹ Box ਦੇ ਬਾਰਡਰ ਅਤੇ ਕੰਟੈਂਟ ਦੇ ਵਿਚਕਾਰ ਦੀ ਜਗ੍ਹਾ (Space) ਹੁੰਦੀ ਹੈ।

Border	ਇਹ Box ਦੇ ਮਾਰਜਨ ਅਤੇ Padding ਵਿਚਕਾਰ ਦੀ ਜਗ੍ਹਾ (Space) ਹੁੰਦੀ ਹੈ।
Margin	ਇਹ ਬਾਰਡਰ ਤੋਂ ਬਾਹਰਲੇ ਪਾਸੇ ਵਾਲੀ ਜਗ੍ਹਾ ਹੁੰਦੀ ਹੈ। ਮਾਰਜਨ ਟਰਾਂਸਪੇਰੇਂਟ (ਪਾਰਦਰਸ਼ੀ) ਹੁੰਦਾ ਹੈ।

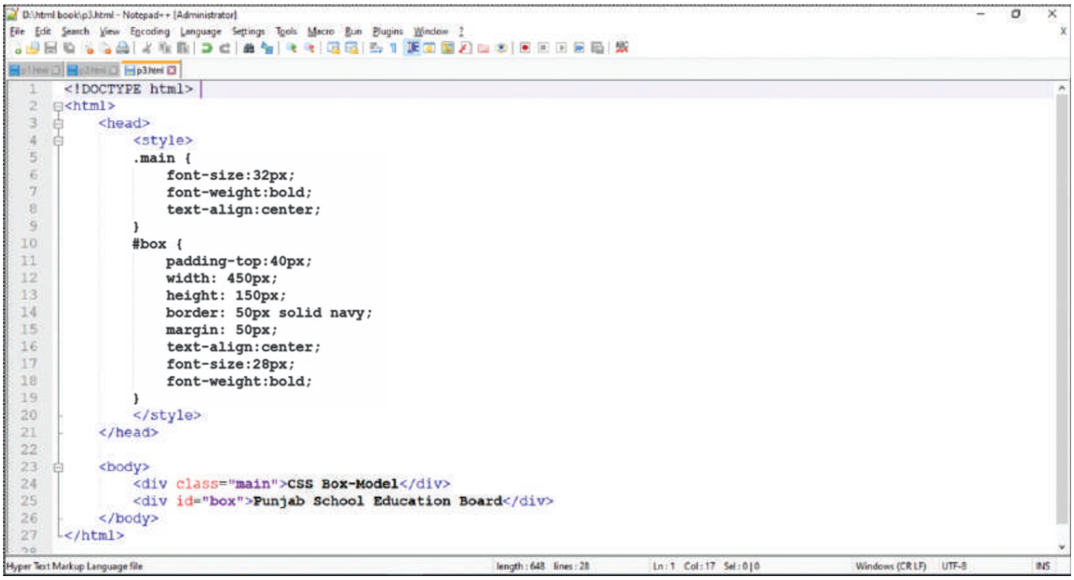
CSS ਦੇ Box Model ਦੀ ਵਰਤੋਂ ਕਰਦਿਆਂ ਵੈੱਬਪੇਜਾਂ ਦੇ Layout ਨੂੰ ਡਿਜ਼ਾਈਨ ਕਰਨ ਲਈ, HTML ਵਿੱਚ ਆਮ ਤੌਰ 'ਤੇ ਦੋ ਐਲੀਮੈਂਟਸ <div> ਅਤੇ <span> ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਇਸ ਐਲੀਮੈਂਟ ਦੀ ਵਿਆਖਿਆ ਹੇਠਾਂ ਕੀਤੀ ਗਈ ਹੈ :

ਦੋਵੇਂ <div> ਅਤੇ <span> ਐਲੀਮੈਂਟ, ਵੈੱਬਪੇਜ ਦੇ ਭਾਗਾਂ ਨੂੰ ਪ੍ਰਭਾਸ਼ਿਤ ਕਰਨ ਲਈ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਇੱਕ ਭਾਗ (<div>) ਇੱਕ ਬਲਾਕ-ਪੱਧਰ ਦਾ ਐਲੀਮੈਂਟ ਹੁੰਦਾ ਹੈ ਅਤੇ ਸਪੈਨ ਇੱਕ inline ਐਲੀਮੈਂਟ ਹੁੰਦਾ ਹੈ। <div> ਦੀ ਵਰਤੋਂ ਡਾਕੂਮੈਂਟ ਦੇ ਭਾਗਾਂ ਨੂੰ wrap ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਣੀ ਚਾਹੀਦੀ ਹੈ, ਜਦੋਂ ਕਿ ਟੈਕਸਟ, Images, ਆਦਿ ਦੇ ਛੋਟੇ ਹਿੱਸਿਆਂ ਨੂੰ wrap ਕਰਨ ਲਈ ਸਪੈਨ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

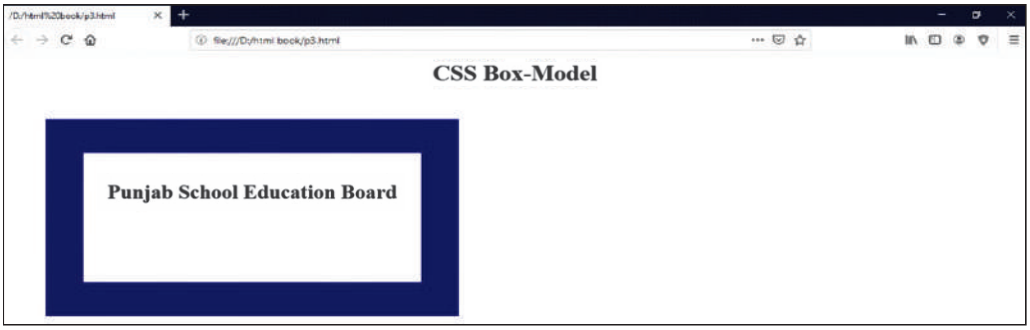
ਇੱਥੇ ਇੱਕ ਉਦਾਹਰਣ ਹੈ :

```
<div>Demo Text, with <span>some other</span> text.</div>
```

<div> ਐਲੀਮੈਂਟ ਨੂੰ HTML ਵਿੱਚ CSS ਅਧਾਰਿਤ ਲੇਆਉਟ ਬਣਾਉਣ ਸਮੇਂ ਇਸਤੇਮਾਲ ਕੀਤਾ ਜਾਂਦਾ ਹੈ, ਜਦੋਂ ਕਿ <span> ਐਲੀਮੈਂਟ, ਟੈਕਸਟ ਨੂੰ stylish ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। <div> ਅਤੇ <span> ਐਲੀਮੈਂਟਸ ਵਿੱਚ ਲੋੜੀਂਦੇ ਐਟਰੀਬਿਊਟ ਨਹੀਂ ਹੁੰਦੇ, ਪਰ style, class ਅਤੇ id ਆਮ ਐਟਰੀਬਿਊਟ ਹਨ ਜੋ ਇਨ੍ਹਾਂ ਦੋਵਾਂ Elements ਨਾਲ ਵਰਤੇ ਜਾਂਦੇ ਹਨ। ਹੇਠਾਂ ਦਿੱਤੀ ਉਦਾਹਰਣ ਦਰਸਾਉਂਦੀ ਹੈ ਕਿ ਵੈੱਬਪੇਜਾਂ ਵਿੱਚ Box Model ਨੂੰ ਕਿਵੇਂ ਪਰਿਭਾਸ਼ਿਤ ਅਤੇ ਉਪਯੋਗ ਕਰਨਾ ਹੈ :



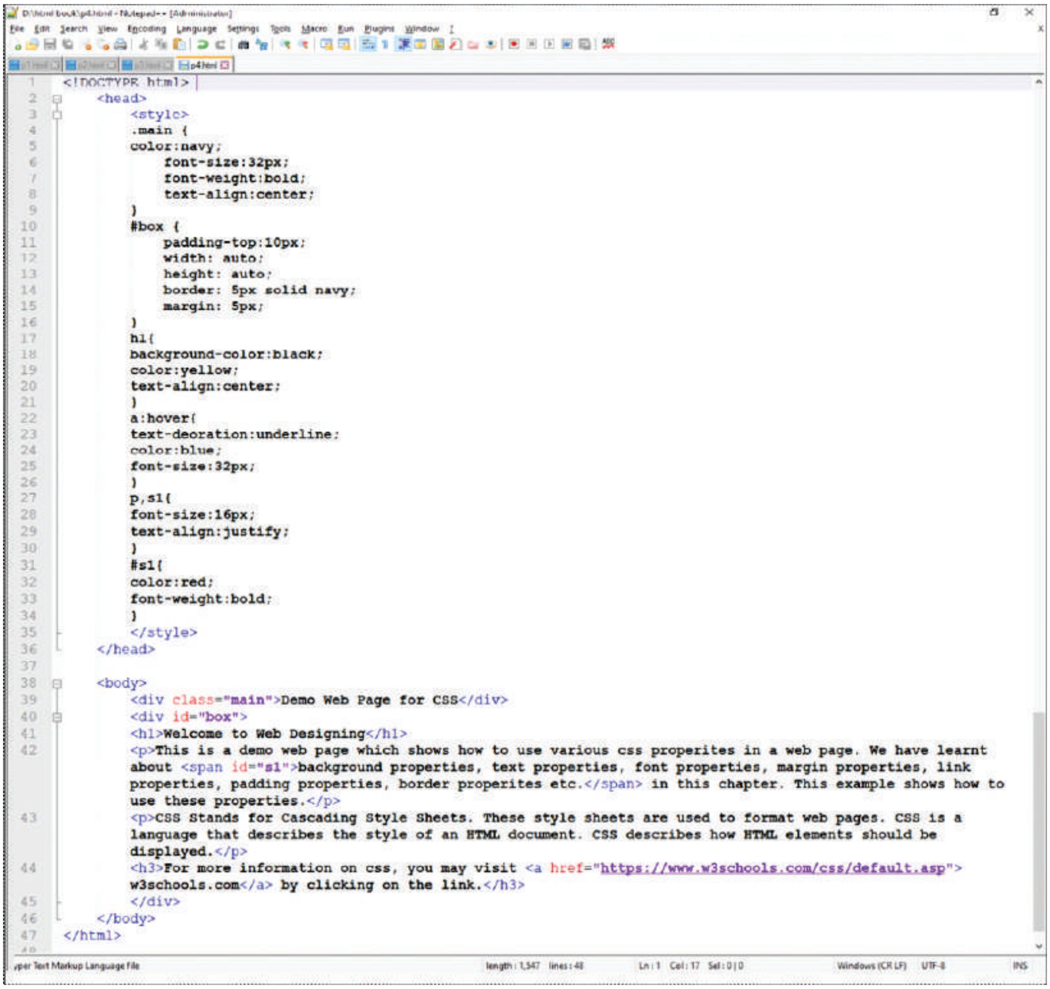
ਚਿੱਤਰ 1.9 Box Model ਦੀ ਉਦਾਹਰਣ ਵਾਲਾ ਪ੍ਰੋਗਰਾਮ



ਚਿੱਤਰ 1.10 Box Model ਉਦਾਹਰਣ ਪ੍ਰੋਗਰਾਮ ਦਾ ਆਉਟਪੁੱਟ

1.13 CSS ਪ੍ਰਾਪਰਟੀਜ਼ ਦੇ ਪ੍ਰੋਗਰਾਮ ਦੀ ਉਦਾਹਰਣ (Example Program for CSS Properties)

ਹੁਣ ਤੱਕ ਜੋ ਅਸੀਂ ਪੜ੍ਹਿਆ ਹੈ, ਇੱਕ ਸਾਧਾਰਨ ਪ੍ਰੋਗਰਾਮ ਦੀ ਵਰਤੋਂ ਨਾਲ ਦਿਖਾਇਆ ਗਿਆ ਹੈ। ਆਓ ਇੱਕ Demo ਪ੍ਰੋਗਰਾਮ ਦੇ ਜ਼ਰੀਏ Internal style sheet ਦੀ ਵਰਤੋਂ ਕਰਦਿਆਂ ਸਮਝੀਏ ਕਿ CSS ਐਟਰੀਬਿਊਟਸ/ ਪ੍ਰਾਪਰਟੀਜ਼ ਦੀ ਵਰਤੋਂ ਕਿਵੇਂ ਕਰਨੀ ਹੈ। ਹੇਠਾਂ ਦਿੱਤਾ ਪ੍ਰੋਗਰਾਮ Notepad++ ਐਡੀਟਰ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਬਣਾਇਆ ਗਿਆ ਹੈ। ਇਸਦੀ ਆਉਟਪੁੱਟ ਵੇਖਣ ਲਈ ਗੂਗਲ ਕਰੋਮ ਜਾਂ ਕਿਸੇ ਹੋਰ ਵੈੱਬ ਬ੍ਰਾਉਜ਼ਰ ਦੀ ਵਰਤੋਂ ਕਰੋ।



```
<!DOCTYPE html>
    <head>
        <style>
        .main {
        color:navy;
            font-size:32px;
            font-weight:bold;
            text-align:center;
        }
        #box {
            padding-top:10px;
            width: auto;
            height: auto;
            border: 5px solid navy;
            margin: 5px;
        }
        h1{
        background-color:black;
        color:yellow;
        text-align:center;
        }
        a:hover{
        text-deoration:underline;
        color:blue;
        font-size:32px;
        }
        p,s1{
        font-size:16px;
        text-align:justify;
        }
        #s1{
        color:red;
        font-weight:bold;
        }
        </style>
    </head>

    <body>
        <div class="main">Demo Web Page for CSS</div>
        <div id="box">
        <h1>Welcome to Web Designing</h1>
        <p>This is a demo web page which shows how to use various css properites in a web page. We have learnt about <span id="s1">background properties, text properties, font properties, margin properties, link properties, padding properties, border properites etc.</span> in this chapter. This example shows how to use these properties.</p>
        <p>CSS Stands for Cascading Style Sheets. These style sheets are used to format web pages. CSS is a language that describes the style of an HTML document. CSS describes how HTML elements should be displayed.</p>
        <h3>For more information on css, you may visit <a href="https://www.w3schools.com/css/default.asp">w3schools.com</a> by clicking on the link.</h3>
        </div>
    </body>
</html>
```

ਚਿੱਤਰ 1.11 ਉਦਾਹਰਣ Program for using CSS

ਚਿੱਤਰ 1.12 ਪ੍ਰੋਗਰਾਮ ਦੀ ਆਉਟਪੁੱਟ

ਯਾਦ ਰੱਖਣ ਯੋਗ ਗੱਲਾਂ

1. ਵੈੱਬਸਾਈਟ ਦਾ ਡਿਜ਼ਾਈਨ ਸਧਾਰਨ ਅਤੇ ਇੱਕਸਾਰ (Simple & Uniform) ਹੋਣਾ ਚਾਹੀਦਾ ਹੈ।
2. ਵੈੱਬਪੇਜ ਤੇ ਟੈਕਸਟ, Images, ਵੀਡੀਓ ਅਤੇ ਆਡੀਓ ਦੇ ਸੁਮੇਲ ਨੂੰ ਇਸ ਤਰੀਕੇ ਨਾਲ ਇਸਤੇਮਾਲ ਕਰਨਾ ਚਾਹੀਦਾ ਹੈ ਕਿ ਵੈੱਬਪੇਜ ਦਾ Loading Time (ਲੋਡ ਹੋਣ ਦਾ ਸਮਾਂ) ਜ਼ਿਆਦਾ ਨਾ ਹੋਵੇ ਅਤੇ ਵੈੱਬਪੇਜ Light Weight (ਹਲਕੇ ਵਜ਼ਨ) ਦਾ ਹੋਵੇ।
3. HTML (ਹਾਈਪਰਟੈਕਸਟ ਮਾਰਕਅੱਪ ਲੈਂਗੁਏਜ਼) ਵੈੱਬਪੇਜਾਂ ਲਈ ਮੁੱਖ ਮਾਰਕਅੱਪ ਭਾਸ਼ਾ ਹੈ।
4. ਟੈਕਸਟ, ਲਿਸਟਾਂ, ਇਮੇਜ਼ਿਜ਼, ਵੀਡੀਓ, ਆਡੀਓ, ਟੇਬਲ, ਫਰੇਮ, ਸਿਰਲੇਖ, ਫਾਰਮ ਅਤੇ ਹੋਰ ਬਹੁਤ ਸਾਰੇ ਐਲੀਮੈਂਟਸ ਅਤੇ ਆਬਜੈਕਟਸ ਨਾਲ, HTML ਵੈੱਬਪੇਜ ਤਿਆਰ ਕਰਨ ਦਾ ਇੱਕ ਸਾਧਨ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ।
5. HTML ਸਾਨੂੰ Java ਸਕ੍ਰਿਪਟ ਵਰਗੀਆਂ ਭਾਸ਼ਾਵਾਂ ਵਿੱਚ ਲਿਖੀਆਂ ਸਕ੍ਰਿਪਟਾਂ ਨੂੰ ਸ਼ਾਮਿਲ ਕਰਨ ਭਾਵ Load ਕਰਨ ਦੀ ਸੁਵਿਧਾ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ।
6. HTML ਵਿੱਚ Link ਨੂੰ Hyperlink ਕਿਹਾ ਜਾਂਦਾ ਹੈ ਅਤੇ <a> (ਐਂਕਰ) ਟੈਗ ਨਾਲ ਬਣੇ Link 'ਤੇ ਕਲਿੱਕ ਕਰਕੇ ਕਿਸੇ ਹੋਰ ਵੈੱਬਪੇਜ ਜਾਂ ਕਿਸੇ ਵੀ ਕਿਸਮ ਦੀ ਫਾਈਲ ਵਿੱਚ Navigate ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ।
7. ਕਾਸਕੇਡਿੰਗ ਸਟਾਈਲ ਸ਼ੀਟਾਂ (CSS) ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਅਸੀਂ ਵੈੱਬਸਾਈਟਸ ਅਤੇ ਵੈੱਬਪੇਜਾਂ ਦੇ ਡਿਜ਼ਾਈਨ ਨੂੰ ਬਹੁਤ ਆਸਾਨ ਅਤੇ ਪ੍ਰਭਾਵਸ਼ਾਲੀ ਬਣਾ ਸਕਦੇ ਹਾਂ।
8. ਤਿੰਨ ਟਾਈਪਸ ਦੇ CSS ਹਨ : Inline, Internal & External (ਇਨਲਾਈਨ, ਇੰਟਰਨਲ ਅਤੇ ਐਕਸਟਰਨਲ)।
9. CSS Padding (ਪੈਡਿੰਗ) ਪ੍ਰਾਪਰਟੀ, ਕੰਟੈਂਟ ਅਤੇ ਬਾਰਡਰ ਦੇ ਵਿਚਕਾਰ ਦੂਰੀ ਨਿਰਧਾਰਤ ਕਰਦੀ ਹੈ।
10. CSS ਬਾਕਸ ਮਾਡਲ ਦੀ ਵਰਤੋਂ ਵੈੱਬਪੇਜਾਂ ਲਈ ਡਿਜ਼ਾਈਨ ਅਤੇ ਲੇਆਊਟ ਦੇ ਸੰਬੰਧ ਵਿੱਚ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। HTML ਦਾ ਹਰੇਕ ਐਲੀਮੈਂਟ, ਇੱਕ ਤਰ੍ਹਾਂ ਨਾਲ ਇੱਕ Box ਹੀ ਹੁੰਦਾ ਹੈ।

ਅਭਿਆਸ

ਭਾਗ-ੳ

ਪ੍ਰ 1. ਖਾਲੀ ਥਾਵਾਂ ਭਰੋ:

I. ਵੈੱਬਸਾਈਟ clear ਅਤੇ fresh ਬਣਾਉਣ ਲਈ .... ਨੂੰ ਤਰਜੀਹ ਦਿੱਤੀ ਜਾਣੀ ਚਾਹੀਦੀ ਹੈ।

II. ਇਹ ਇੱਕ ਖੋਜ ਤੋਂ ਸਿੱਖਿਆ ਗਿਆ ਹੈ ਕਿ ਆਮ ਯੂਜ਼ਰ .... ਪੈਟਰਨ ਵਿੱਚ ਵੈੱਬਪੇਜਾਂ (ਕੰਪਿਊਟਰ ਸਕਰੀਨਜ਼) ਨੂੰ ਸਕੈਨ ਕਰਦੇ ਹਨ।

III. HTML ਦੀ ਵਰਤੋਂ ਟੈਕਸਟ, Images ਅਤੇ ਹੋਰ ਵੈੱਬਪੇਜਾਂ ਦੇ .... ਨੂੰ ਵੈੱਬਪੇਜਾਂ 'ਤੇ ਵਿਵਸਥਿਤ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।

IV. .... ਦੀ ਵਰਤੋਂ ਨਾਲ ਵੈੱਬਸਾਈਟਸ ਅਤੇ ਵੈੱਬਪੇਜਾਂ ਦਾ ਡਿਜ਼ਾਈਨ ਬਹੁਤ ਹੀ ਸਧਾਰਨ ਅਤੇ ਪ੍ਰਭਾਵਸ਼ਾਲੀ ਢੰਗ ਨਾਲ ਤਿਆਰ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ।

V. ...., HTML ਐਲੀਮੈਂਟ ਚੁਣਨ ਅਤੇ ਨਿਰਧਾਰਿਤ ਕਰਨ ਵਿੱਚ ਸਹਾਇਤਾ ਕਰਦਾ ਹੈ।

ਪ੍ਰ 2. ਬਹੁਤ ਛੋਟੇ ਉੱਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ:

I. W3C ਸ਼ਬਦ ਦਾ ਸੰਖੇਪ ਰੂਪ ਕੀ ਹੈ ?

II. CSS ਸ਼ਬਦ ਦਾ ਸੰਖੇਪ ਰੂਪ ਕੀ ਹੈ ?

III. External CSS ਫਾਈਲ ਦੀ ਐਕਸਟੈਂਸ਼ਨ ਕੀ ਹੈ ?

IV. ਕਿਸ ਟੈਗ ਵਿਚ ਇੰਟਰਨਲ CSS ਨੂੰ HTML ਵੈੱਬਪੇਜ ਵਿਚ ਪਰਿਭਾਸ਼ਿਤ ਕੀਤਾ ਗਿਆ ਹੈ ?

V. WYSIWYG ਦਾ ਪੂਰਾ ਨਾਮ ਲਿਖੋ।

ਭਾਗ-ਅ

ਪ੍ਰ : 3 ਛੋਟੇ ਉੱਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ (4-5 ਲਾਈਨਾਂ ਵਿਚ ਉੱਤਰ ਲਿਖੋ)

I. HTML ਡਾਕੂਮੈਂਟ ਦੀ ਬਣਤਰ ਲਿਖੋ।

II. ਵੈੱਬਸਾਈਟ ਡਿਵੈਲਪਮੈਂਟ ਦੇ ਵੱਖ-ਵੱਖ ਪੜਾਅ ਕੀ ਹਨ ?

III. ਵੈੱਬ ਵਿਚ HTML ਦੀ ਭੂਮਿਕਾ ਕੀ ਹੁੰਦੀ ਹੈ ?

IV. CSS ਕਿਸ ਲਈ ਵਰਤੀ ਜਾਂਦੀ ਹੈ ?

ਭਾਗ- ੲ

ਪ੍ਰ : 4 ਵੱਡੇ ਉੱਤਰਾਂ ਵਾਲੇ ਪ੍ਰਸ਼ਨ (10-15 ਲਾਈਨਾਂ ਵਿਚ ਉੱਤਰ ਲਿਖੋ)

I. ਚੰਗੇ ਵੈੱਬ ਡਿਜ਼ਾਈਨ ਦੇ ਸਿਧਾਂਤਾਂ ਦੀ ਵਿਆਖਿਆ ਕਰੋ।

II. ਵੈੱਬ ਡਿਜ਼ਾਈਨ ਲਈ CSS ਦੀ ਵਰਤੋਂ ਕਿਵੇਂ ਕੀਤੀ ਜਾ ਸਕਦੀ ਹੈ।

III. CSS ਦੇ ਕੀ ਫਾਇਦੇ ਹਨ ?

IV. CSS BOX ਮਾਡਲ ਦੀ ਵਿਆਖਿਆ ਕਰੋ।

V. ਕੋਡਿੰਗ ਦੇ ਨਾਲ CSS ਵਿੱਚ ਬਾਰਡਰ ਪ੍ਰਾਪਰਟੀ ਦੀ ਵਿਆਖਿਆ ਕਰੋ।

ਲੈਬ ਐਕਟੀਵਿਟੀ

- ਆਪਣੇ ਸਕੂਲ ਸੰਬੰਧੀ ਜਾਣਕਾਰੀ ਨੂੰ ਇਕ ਵੈਬਪੇਜ ਰਾਹੀਂ ਦਰਸਾਓ ਅਤੇ ਇਸ ਪਾਠ ਵਿਚ ਦਿੱਤੀਆਂ CSS ਪ੍ਰਾਪਰਟੀਜ਼ ਦੀ ਵਰਤੋਂ ਨਾਲ ਉਸਨੂੰ ਫਾਰਮੈਟ ਕਰੋ।