

ចាប់ផ្តើម

CSS

Web Development

អ្នកនឹងសិក្សាពីមេរៀនតំបូង វិធីសាស្ត្រថ្មីៗ
និងវិធីសាស្ត្រប្រើជាញឹកញាប់ក្នុងការបង្កើតវេបសាយ

bayon
hosting

រៀបរៀងដោយ: កុយ ទិព្វតារា

សំរាប់បង្រៀននៅ មជ្ឈមណ្ឌល ITEC និងចែកចាយតាម Bayon Hosting

Module 1: Introduction to Cascading Style Sheets

ការបង្កើត CSS

Internal CSS

ជាការសរសេរនូវ កូដ CSS ក្នុង File ជាមួយគ្នានឹង HTML ។ គេមានវិធីពីរយ៉ាង៖

ការប្រើតាម style tag

Syntax:

```
<style type="text/css">  
    Css_statements;  
</style>
```

ការប្រើនូវ style attribute ក្នុង tag

Syntax:

```
<tag_name style="css_statements"></tag>
```

External CSS

ជាការសរសេរនូវ កូដ CSS ក្នុង File មួយដោយឡែកពី HTML File ។ គេមានវិធីក្នុងការ តភ្ជាប់ទៅកាន់ CSS File ពីរយ៉ាង៖

ការប្រើនូវ link tag

Syntax

```
<link rel="stylesheet" type="text/css" href="css_file_path"/>
```

ការប្រើនូវ @import ក្នុង style tag

Syntax

```
<style type="text/css">  
    @import url(css_path);  
</style>
```

Module 2: CSS Basic

លក្ខណៈទូទៅនៃ CSS

Comments

ជា statements ដែលប្រើសំរាប់ពិពណ៌នាបន្ថែមទៅកាន់ កូដដែលយើងបានសរសេរ សំរាប់ជាការរំលឹកសំរាប់ថ្ងៃក្រោយផង និងងាយស្រួលក្នុងការស្វែងរកទីតាំងនៃកូដផង។ Comments មិនត្រូវបានបកប្រែដោយ Interpreter ទេ។

Syntax

```
/* This is my first css_statement
That wrote to create stylesheets
For my web page*/
```

វិធីក្នុងការកំនត់ Style

កំនត់តាមឈ្មោះរបស់ tag

ជាការកំនត់ដោយយកឈ្មោះរបស់ tag។ ការធ្វើបែបនេះគ្រប់ element ដែលប្រើ tag ដែលបានកំនត់នោះ វានឹងទទួលបានលក្ខណៈដូចគ្នា។

ដើម្បីកំនត់តាមឈ្មោះរបស់យើងគ្រាន់តែដាក់ឈ្មោះរបស់ tag ហើយបន្តដោយ {}

Syntax:

```
Tag_Name{
    Css_statement
}
```

ឧទាហរណ៍៖

យើងនឹងកំនត់ពណ៌របស់គ្រប់ element ដែលប្រើ <h1> tag ទៅជាពណ៌ក្រហម និងតម្រឹមវាមកនៅកណ្តាល។

```
<style type="text/css">
    H1{
        color:red;
        text-align:center;
    }
</style>
.
.
.
<H1>Welcome</H1>
<H2>Heading 2</H2>
<H1>To My Site</H1>
```

កំណត់តាម Class

ជាការកំណត់ដោយបង្កើតជា class (ក្រុម) នៃ style មួយ។ ដើម្បីអោយ element ទទួល style ដែលកំណត់ដោយ class មួយ ក្នុង open tag នៃ element ត្រូវប្រើ attribute មួយឈ្មោះថា class និងកំណត់ value របស់វាទៅតាម class ដែលយើងត្រូវការ។

ដើម្បីបង្កើត class យើងប្រើ សញ្ញា (.) បន្តដោយឈ្មោះ class និង {}

Syntax:

```
.class_Name{
    Css_statement
}
```

ឧទាហរណ៍៖

យើងនឹង class ចំនួនពីរ ទី១ ប្រើ Font Khmer OS ទំហំ 20px ទី២ ប្រើ Font Time New

Roman ទំហំ 14px។

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <style type="text/css">
    .khmer{
      font-family:Khmer OS;
      font-size:20px;
    }
    .english{
      font-family:Time New Roman;
      font-size:14px;
    }
  </style>
</head>
<body>
  <h1 class="khmer">សូស្តី</h1>
  <span class="khmer">សូមស្វាគមន៍មកកាន់</span>
  <span class="english"> Web Page</span>
  <span class="khmer">របស់យើងខ្ញុំ</span>
</body>
```

ចំណាំដើម្បីអោយ Browser អាចបង្ហាញជាអក្សរដែលប្រើ Unicode បានត្រូវកំណត់ នូវ attribute content របស់ meta tag នូវតំលៃថា

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

ការកំណត់តាម id

ជាការកំណត់អោយ element ណាមួយតែម្តង ដោយយក id របស់ element នោះមកប្រើតែម្តង។ គេប្រើសញ្ញា # បន្តដោយ id និង {}

Syntax

```
#element_id{
```

```
Css_statements
}
```

ឧទាហរណ៍៖

យើងនឹងបង្កើតនូវផ្ទាំងអត្ថបទសំរាប់ដាក់អក្សរខ្មែរ មានផ្ទៃខាងក្រោយពណ៌ ប្រេង៖

ទំហំ១៤។

```
<head>
  <meta http-equiv="Content-Type" content="charset=UTF-8"/>
  <style type="text/css">
    #khmer_content{
      Background:#CCCCCC;
      Font-family:Khmer OS System;
      Font-size:14px;
    }
  </style>
</head>
<body>
  <div id="khmer_content">
    Comments ជា statements ដែលប្រើសំរាប់ពិពណ៌នាបន្ថែមទៅកាន់
    កូដដែលយើងបានសរសេរ សំរាប់ជាការរំលឹកសំរាប់ថ្ងៃក្រោយផង
    និងងាយស្រួលក្នុងការស្វែងរកទីតាំងនៃកូដផង។ Comments មិនត្រូវបានបកប្រែដោយ Interpreter ទេ។
  </div>
</body>
```

ចំណាំ៖ យើងអាចកំនត់បានច្រើន tag, class, id ក្នុងពេលតែមួយបានការប្រើ(,)។

- ឧ. H1, H2 {...}
- .khmerTop, .khmerBottom {...}
- #box1, #box2 {...}

លក្ខណៈ: Inheritance (តិគ្គា)

Inheritance រវាង tag និង tag

គេប្រើវាសំរាប់ tag មួយដែលនៅក្នុង tag មួយទៀត។

Syntax

```
tag1Name ag2Name{
  css_statement
}
.
.
.
<tag1>
  <tag2>
  </tag2>
</tag1>
```

ឧទាហរណ៍៖

```
<style>
  div p{
    font-size:16px;
    color:blue;
  }
  span p{
    font-size:18px;
    color:red;
  }
  p{
    font-size:20px;
    color:Green;
  }
</style>
</head>

<body>
  <div> <p>Welcome </p></div>
  <span><p>Welcome </p></span>
  <p>Welcome </p>
</body>
```

លទ្ធផល៖

Welcome

Welcome

Welcome

Inheritance នៃ id និង tag

គេប្រើវាពេលណាដែលមាន ប្លុកពីរផ្សេងគ្នា ប៉ុន្តែ ប្រើដោយ tag តែមួយហើយគ្រប់ធាតុខាងក្នុងដូចគ្នាទាំងអស់។ ឧបមាអ្នកចង់ធ្វើកំណត់ style នៃធាតុក្នុងប្រអប់នោះផ្សេងគ្នា នោះអ្នកត្រូវកំណត់ id របស់ element ដាក់ផ្នែកប្លុកទាំងពីរផ្សេងគ្នា ហើយអ្នកអាចប្រើលក្ខណៈ inheritance សំរាប់កំណត់ style នៃ tag នីមួយៗ ក្នុងប្លុកទាំងពីរផ្សេងគ្នាបាន។

Syntax:

```
#element_id tagName{
  Css_statement;
}
.
.
.
<tagName id="element_id">
  <tagName>
  </tagName>
</tagName>
```

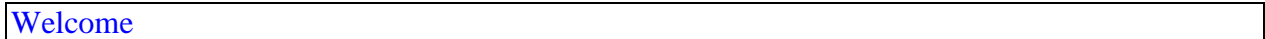
ឧទាហរណ៍៖

```
<style>
  #borderBox p{
    border:solid 1px;
    color:blue;
  }
  #greyBox p{
```

```
        background:#CCCCCC;
    }
</style>
</head>

<body>
    <div id="greyBox"><p>Welcome</p></div>
    <div id="borderBox"><p>Welcome</p></div>
</body>
```

លទ្ធផល៖



លក្ខណៈ: Inheritance នៃ class និង tag

.className tagName

មានលក្ខណៈដូចនឹង Inheritance នៃ id និង tag ដែរ។ style វានឹងកំណត់ដោយ tag ណាដែល បិតក្នុង element ដែលមាន class ដែលបានបញ្ជាក់ប៉ុណ្ណោះ។

Syntax:

```
.className tagName{
    Css_statements
}
.
.
.
<tag1 class="className">
    <tagName>
    </tagName>
</tag1>
```

tagName.className

គេប្រើវាពេលយើងចង់បង្កើត class មួយសំរាប់ប្រើតែក្នុង tag ណាមួយប៉ុណ្ណោះ។

Syntax:

```
tagName.className{
    css_statement
}
```

ឧទាហរណ៍៖

```
<style>
    div.redText{
        color:red;
        font-size:20px;
    }
    span.redText{
        color:red;
        font-size:12px;
    }
</style>
</head>
```

```
<body>  
  <div class="redText">Welcome</div>  
  <span class="redText">Welcome</span>  
</body>
```

លទ្ធផល៖

Welcome

Welcome

Formatting

Universal Resource Location

Syntax:

```
url (http://www.example.com/style/banner.css);
```

ឧទាហរណ៍៖

```
@import url (http://www.style.com/style/box.css);  
Body{  
  Background:url (http://www.gallery.com/images/2435.jpg);  
}
```

ឬ

```
Body{  
  Background:url (../images/my Image.jpg);  
}
```

Color

RGB Format

Syntax:

```
rgb (red, green, blue)
```

red, blue, green មានតំលៃពី 0 ដល់ ២៥៥

Hexa Format

Syntax:

```
#rrggbb
```

rr,gg,bb មានតំលៃពី 00-FF

Name Format

Syntax:

```
colorName
```

ដូចជា blue, red, black, white, gray, green, purple, pink...

Link Behavior

a:link

មិនធ្លាប់ចុច

a:visited

ចុចម្តងហើយ

a:active

window ដែលបាន link ទៅកំពុងបើក

a:hover

ពេលយក mouse រំកិលលើ

Syntax

```
a:link{
    color:green;
    textdecoration=none;
}
a:visited{
    color:red;
    textdecoration=none;
}
a:active{
    color:blue;
    textdecoration=none;
}
a:hover{
    color:black;
    textdecoration=overline;
}
```

យើងអាចប្រើលក្ខណៈ Descending បានឧទាហរណ៍៖

```
a.footer:link{color:blue}
a.header:link{color:red}
#leftBlock a:link{color:green}
```

Applied CSS

Formatting Text

font-family:

```
font-family: Arial, Helvetica, sans-serif;
```

ប្រើសំរាប់ប្តូរពុម្ពអក្សរ។ គេអាចដាក់ពុម្ពអក្សរបានច្រើនការពារករណីដែលម៉ាស៊ីនអ្នកប្រើមិនមានពុម្ពអក្សរនោះ។ ចំពោះពុម្ពអក្សរណាដែលមានដកឃ្លាត្រូវតែដាក់ក្នុង “ ”។

ក្រុមអក្សរដែលមានរូបរាងស្រដៀងគ្នាមានដូចជា៖

- Arial, Helvetica, sans-serif
- "Times New Roman", Times, serif
- "Courier New", Courier, monospace
- Georgia, "Times New Roman", Times, serif
- Verdana, Arial, Helvetica, sans-serif
- Geneva, Arial, Helvetica, sans-serif
- Tahoma, "Lucida Grande", Arial, sans-serif
- "Lucida Console", Monaco, monospace
- "Marker Felt", "Comic Sans MS", fantasy
- "Century Gothic", "Gill Sans", Arial, sans-serif

color:

```
color:#F1F1F1;
color:rgb(241,241,241);
color:gray;
```

ប្រើសំរាប់ប្តូរពណ៌អក្សរ។

font-size:

```
font-size:[16px|1em|100%|...];
```

ប្រើសំរាប់ប្តូរទំហំរបស់អក្សរ។ ជាធម្មតាអក្សរត្រូវបានកំណត់យកទំហំ 16px, 1em, 100%។

font-style:

```
font-style: [italic|oblique|normal];
```

ប្រើសំរាប់ប្តូររាងអក្សរអោយមាន លក្ខណៈ: ទ្រេតទៅមុខ។ ចំពោះ italic និង oblique មានលក្ខណៈ: ដូចគ្នា។ normal ប្រើសំរាប់ប្តូររាមកទំរង់ដើមវិញ

font-weight:

font-weight: [100-900|bold|bolder|lighter|normal]

ប្រើសំរាប់ប្តូរកំរាស់របស់អក្សរ។ ១០០ ដល់ ៩០០ គឺពីក្រាស់បំផុត ដល់ ស្តើងបំផុត។ bold គឺការកំនត់យកអក្សរដិតធំមួយ។

text-tranform:

text-tranform: [capitalize|lowercase|uppercase|none];

ប្រើសំរាប់ប្តូរ case របស់អក្សរ។ none សំរាប់កំនត់កុំអោយយកលក្ខណៈ Inherit ពី parent style។

font-variant:

font-variant: [small-caps|none];

ប្រើសំរាប់ប្តូររាល់អក្សរ lowercase អោយទៅជា uppercase ដែលមានទំហំតូចជាង uppercase ដើម។
ឧទាហរណ៍ HELLO WORLD! ។

text-decoration:

text-decoration: [underline|overline|line-through|none]

សំរាប់គូរសបន្ទាត់ពីក្រោម កណ្តាល និងពីលើអក្សរ។

letter-spacing:

letter-spacing: [-#px|#px|normal];

ប្រើសំរាប់ប្តូរគំលាតរបស់អក្សរ។ បើមានសញ្ញា ដក វានឹងធ្វើអោយអក្សររួមចូលគ្នា។

word-spacing:

word-spacing: [-#px|#px|normal];

ប្រើសំរាប់ប្តូរគំលាតរបស់ពាក្យនីមួយៗ។

line-height:

line-height: [30px|150%|1.5em|1.5];

ប្រើសំរាប់ប្តូរគំលាតពីបន្ទាត់មួយ ទៅបន្ទាត់មួយទៀត។

text-align:

text-align: [left|center|right|justify|normal]

ប្រើសំរាប់តម្រឹមអក្សរមកឆ្វេង កណ្តាល ស្តាំ រឺទាំងសងខាង។

text-indent:

```
text-indent: [#px|#em];
```

ប្រើសំរាប់ចូលបន្ទាត់រាល់បន្ទាត់តំបូងគេ នៃ tag ដែលវាស្ថិតនៅដូចជា p tag ជាដើម។

tagName:first-letter

```
p:first-letter{
  declaration;
}
```

ប្រើសំរាប់ប្តូរតែអក្សរដែលនៅតំបូងគេបង្អស់នៃ tag នោះ។

tagName:first-line

```
p:first-line{
  declaration;
}
```

ប្រើសំរាប់ប្តូរតែបន្ទាត់តំបូងប៉ុន្មាននោះនៃ tag នោះ។

Styling List

list-style-type:

```
list-style-type:[...|none];
```

ប្រើជាមួយនឹង ,, tag ដែលគេប្រើសំរាប់បង្កើតជា List ។ យើងអាចកំណត់នូវ style របស់វាដូចខាងក្រោម៖

<p>disc</p> <ul style="list-style-type: none"> item 1 item 2 item 3 	<p>circle</p> <ul style="list-style-type: none"> item 1 item 2 item 3 	<p>square</p> <ul style="list-style-type: none"> item 1 item 2 item 3
<p>decimal</p> <ol style="list-style-type: none"> item 1 item 2 item 3 	<p>decimal-leading-zero</p> <ol style="list-style-type: none"> 01. item 1 02. item 2 03. item 3 	<p>upper-alpha</p> <ol style="list-style-type: none"> A. item 1 B. item 2 C. item 3
<p>lower-alpha</p> <ol style="list-style-type: none"> a. item 1 b. item 2 c. item 3 	<p>upper-roman</p> <ol style="list-style-type: none"> I. item 1 II. item 2 III. item 3 	<p>lower-roman</p> <ol style="list-style-type: none"> i. item 1 ii. item 2 iii. item 3

list-style-position:

```
list-style-position: [inside|outside];
```

ប្រើសំរាប់កំណត់ថា តើ style របស់ list ត្រូវប្រតិបត្តិនៅខាងក្រៅ រឺនៅខាងក្នុង ដូចខាងក្រោម៖

list-style-position: outside

- iLorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
 - Ut wisi enim ad minim veniam, quis nostrud exerci tatio tincidunt ut laoreet dolore magna aliquam erat volutpat.
 - Tincidunt ut laoreet dolore magna aliquam erat volutpat nonummy nibh euismod tincid.
1. iLorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
 2. Ut wisi enim ad minim veniam, quis nostrud exerci tatio tincidunt ut laoreet dolore magna aliquam erat volutpat.
 3. Tincidunt ut laoreet dolore magna aliquam erat volutpat nonummy nibh euismod tincid.

list-style-position: inside

- iLorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
 - Ut wisi enim ad minim veniam, quis nostrud exerci tatio tincidunt ut laoreet dolore magna aliquam erat volutpat.
 - Tincidunt ut laoreet dolore magna aliquam erat volutpat nonummy nibh euismod tincid.
1. iLorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
 2. Ut wisi enim ad minim veniam, quis nostrud exerci tatio tincidunt ut laoreet dolore magna aliquam erat volutpat.
 3. Tincidunt ut laoreet dolore magna aliquam erat volutpat nonummy nibh euismod tincid.

Remove indent of list item

```
padding-left: 0px;
margin-left: 0px;
```

ប្រើសំរាប់លុបរាល់គំលាតរបស់ list item ។

list-style-image:

```
list-style-image: url (images/bullet.jpg);
```

ប្រើសំរាប់ដាក់នូវរូបភាពជំនួស style ចាស់។

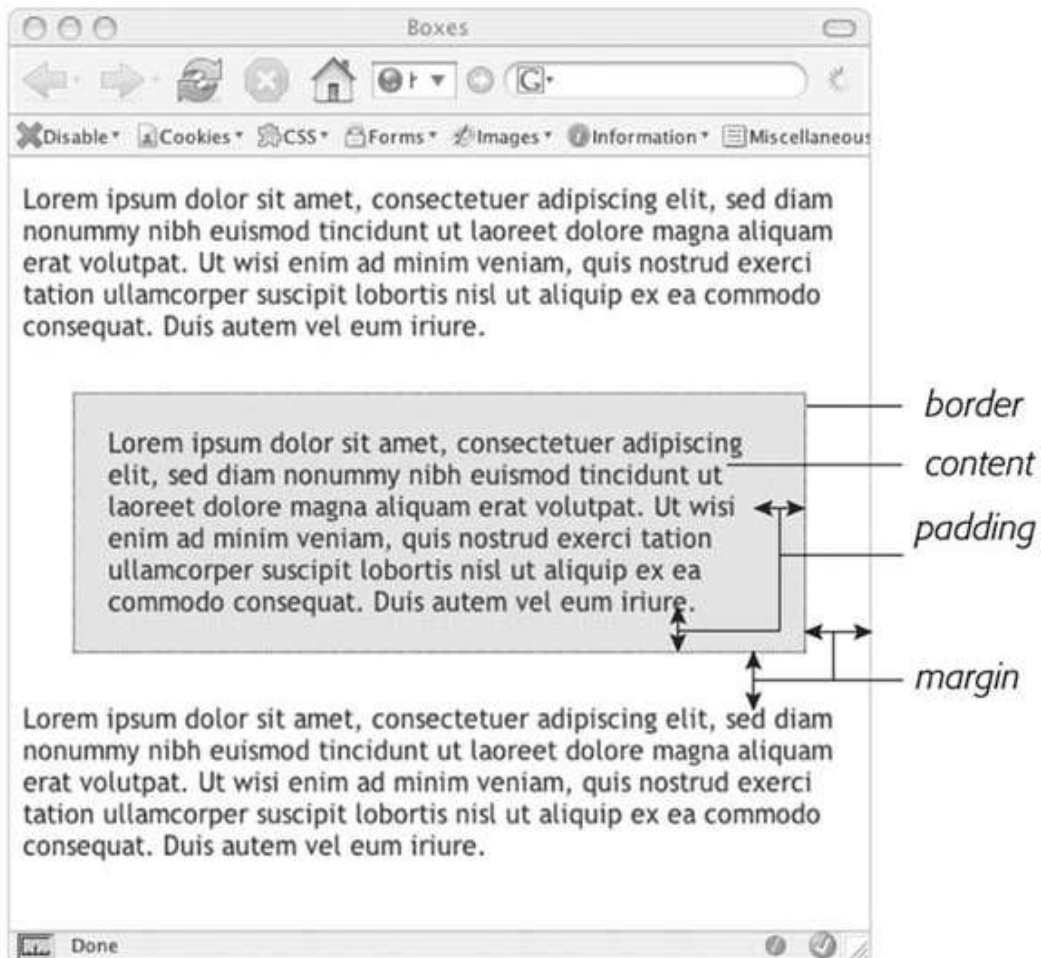
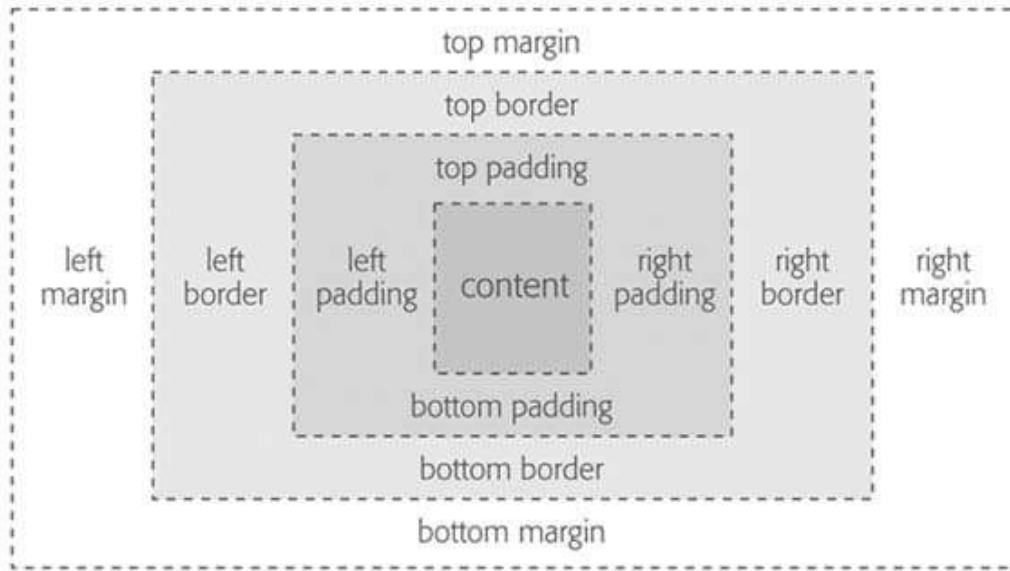
Margins, Padding, and Borders

Understanding the Box Model

Box មាននូវ properties សំខាន់ៗមួយចំនួនដូចជា៖

- Padding : ជាគំលាតរវាង content និង border របស់ content ។

- Border: គឺជាបន្ទាត់ដែលគូសជុំវិញជ្រុងរបស់ box ។ យើងអាចកំណត់ជ្រុងទាំងបួន និងជ្រុងនីមួយៗបាន។
- Background-color: គឺផ្ទៃដែលបិទក្នុង border រួមទាំង padding ផងដែរ។
- Margin: ជាគំលាតរវាង tag មួយទៅ tag មួយទៀត ដូចគំលាតខាងលើ និងខាងក្រោមរបស់ p tag ។



Define margin and padding

All Edge

```
margin:#px|#em|#%;
padding: #px|#em|#%;
```

Specified Edge

```
margin-left:#px|#em|#%;
margin-right:#px|#em|#%;
margin-top: #px|#em|#%;
margin-bottom:#px|#em|#%;

padding-left:#px|#em|#%;
padding-right:#px|#em|#%;
padding-top: #px|#em|#%;
padding-bottom:#px|#em|#%;
```

Shorthand

```
margin: top right bottom left; //TRouBLLe
padding: top right bottom left;
```

ឧទាហរណ៍៖

```
margin: 0 10px 10px 20px;
padding: 10px 0 10px 0;
```

Display with inline and block-level block

រាល់ tag របស់ HTML គឺសុទ្ធតែមានលក្ខណៈជា box ។ ប៉ុន្តែ box មាន 2 ប្រភេទគឺ inline និង block box ។

block box បង្កើតនូវការចុះបន្ទាត់ពីមុនវា និង ពីក្រោយវា ដូចជា p, div, table, ul, ol, li ... ។

inline box មិនបង្កើតនូវការចុះបន្ទាត់អ្វីទាំងអស់ ហើយវាបិតនៅជាមួយបន្ទាត់តែមួយនឹង tag ដទៃទៀត។ inline box tag មានដូចជា , , <a>, ។ ចំពោះ tag ប្រភេទនេះគេអាចកំណត់ padding, margin ឆ្លងស្តាំបាន ប៉ុន្តែមិនអាចកំណត់ ខាងលើ និងក្រោមបានទេ។ ដើម្បីកែតម្រូវប្រើនូវ line-height ជំនួសវិញ។

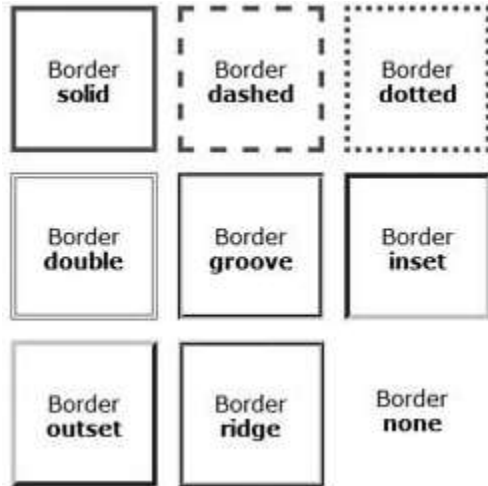
ដើម្បីធ្វើការប្តូរនូវប្រភេទ box របស់វាត្រូវកំណត់នូវ display របស់វាក្នុង CSS ដូចខាងក្រោម៖

display:

```
display:[inline|block|none]
```

Defining Border

Border មាននូវលក្ខណៈបីយ៉ាងដែលត្រូវកំណត់គឺ ពណ៌ ទំហំ និង style របស់វា។ ចំពោះ style មានដូចខាងក្រោម៖



border:

```
border: color width style;  
border: #AABB44 1px solid;
```

សំរាប់កំនត់លក្ខណៈ: របស់ជ្រុងទាំងបួន

Each Border Edge:

```
border-left: color width style;  
border-top: color width style;  
border-right: color width style;  
border-bottom: color width style;
```

Each Border Property

```
border-width: width;  
border-color: color;  
border-style: style;
```

Each Border Edge and Property

```
border-left-color: color;  
border-top-style: style;  
...
```

background-color:

```
background-color: color;
```

ប្រើសំរាប់កំនត់ពណ៌របស់ផ្ទៃខាងក្រោយរបស់ box។

overflow:

```
overflow: visible|scroll|auto|hidden;
```


គេប្រើវានៅពេលអក្សរនៅក្នុងលើសទំហំដែលត្រូវផ្ទុក។ ជាធម្មតាក្នុងIE វានឹងពង្រីកទំហំរបស់ប្រអប់ អោយត្រូវនឹងបរិមាណទំហំអក្សរ។ តែក្នុង browser ដទៃទៀតវានឹងធ្វើអោយអក្សរក្នុង ហៀរចេញមក ក្រៅ box។ ហេតុដូច្នេះនេះគេប្រើនូវ overflow ដើម្បីដោះស្រាយបញ្ហានេះ។

- visible ៖ គឺដូចធម្មតា គឺវានឹងបង្ហាញរាល់អក្សរដែលមាន
- scroll ៖ គឺវាបង្កើតនូវ scrollbar ដើម្បីរំកិល។ វាបង្ហាញជានិច្ចបើទោះជាទំហំគ្រប់គ្រាន់ ក៏ដោយ។
- auto ៖ គឺដូច scroll តែវាមាន scrollbar តែពេលវាលើសទំហំប៉ុននោះ។
- hidden ៖ គឺវាបង្ហាញតែអក្សរដែលបិតក្នុងដែនកំណត់ របស់វាប៉ុននោះ។

Wrap Content with Floating Elements

float:

float: left|right|none;

- left : ប្រើសំរាប់ទាញអោយ Element ទៅផ្នែកខាងឆ្វេង និងអ្វីដែលនៅក្រោម នឹងមកបិតជាប់ខាងស្តាំនៃ element នោះ។
- right : ប្រើសំរាប់ទាញអោយ Element ទៅផ្នែកខាងស្តាំ និងអ្វីដែលនៅក្រោម នឹងមកបិតជាប់ខាងឆ្វេងនៃ element នោះ។
- none : ប្រើសំរាប់រំកិលមកទីតាំងដើមវិញ។

Background, Border with float Problem

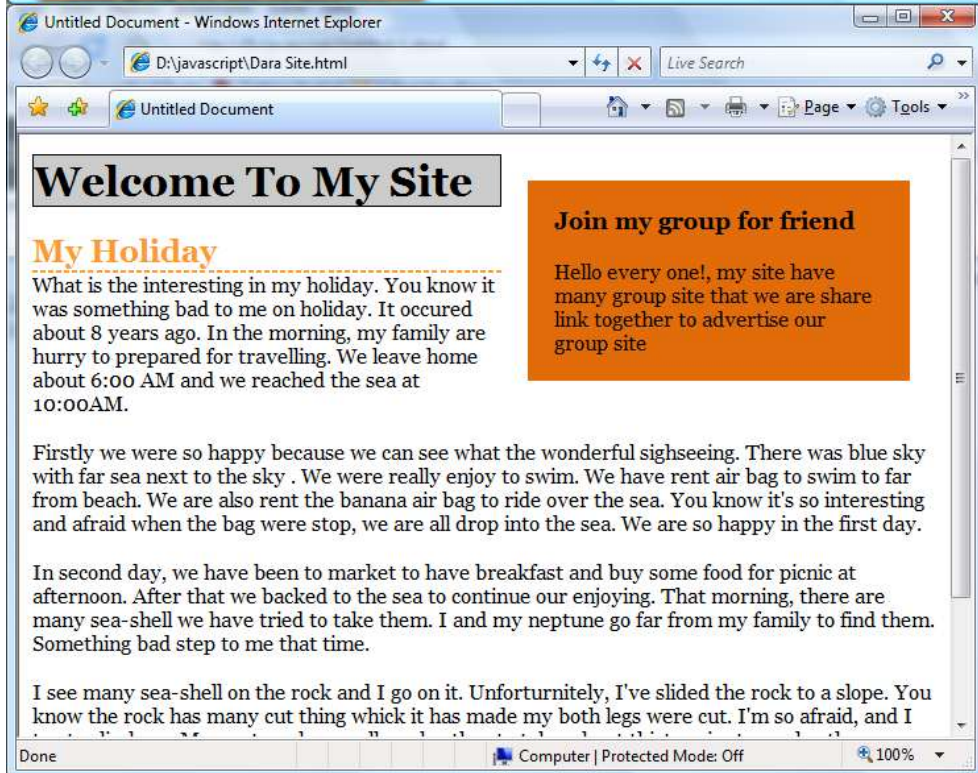
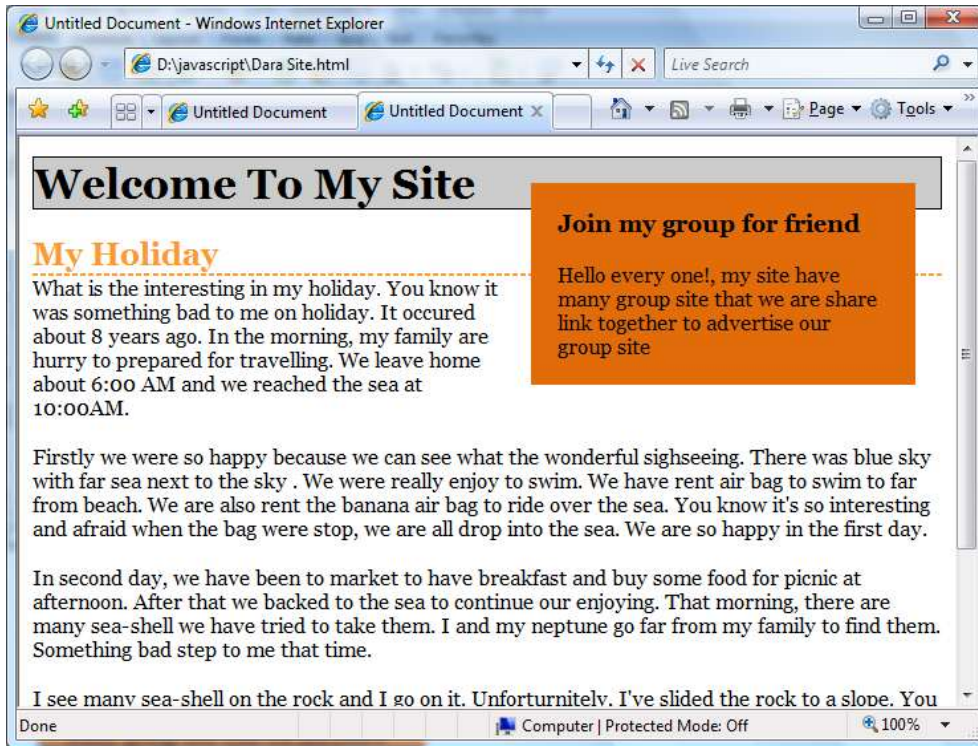
ជាធម្មតាពេលយើងកំណត់នូវ style របស់ box អោយ float ទៅខាងស្តាំ តែក្នុងករណីដែល យើងមាននូវ Element ដែលនៅពីក្រោម ដែលតម្រូវអោយបំពេញតែផ្នែកខាងឆ្វេង វាមិនមាន ប្រសិទ្ធភាពទៅលើ background និង border ទេដូចបង្ហាញនៅរូបតំបូង។ ដើម្បីដោះស្រាយនូវ បញ្ហានេះ ក្នុង style របស់ element ដែលមាន background, border នោះត្រូវបន្ថែមនូវ overflow:hidden ទើបបាន ដូចកូដខាងក្រោម៖

```

body{
    font-family:Georgia, "Times New Roman", Times, serif;
}
h1{
    background-color:#CCCCCC;
    border:1px solid;
    overflow:hidden;
}
h2{
    margin-bottom:0px;
    border-bottom:2px dashed;
    color:#FF9933;
    overflow:hidden;
    zoom:1;//ប្រើចំពោះតែ IE 5 និង 6 ប៉ុណ្ណោះចំពោះ IE 7 មិនចាំបាច់ទេ
}
p{
    margin-top:0;
}

```

```
#sidebar{
    background:rgb(227,108,10);
    border:#CC3300;
    float:right;
    width:250px;
    padding:20px;
    margin:20px;
}
* html #sidebar{ display:inline;} //ប្រើចំពោះតែ IE 5 និង 6
```



clear:

```
clear: left|right|both|none;
```

ប្រើសំរាប់បង្គាប់អោយ element កុំអោយ wrap បំពេញ element ដែលបាន float។ ការធ្វើបែបនេះវាធ្វើអោយ element ធ្លាក់មកពីក្រោម element ដែល float។

- left ៖ វានឹងធ្លាក់មកក្រោម element ដែល float ខាងឆ្វេង តែនៅបំពេញ element right float
- right ៖ វានឹងធ្លាក់មកក្រោម element ដែល float ខាងស្តាំ តែនៅបំពេញ element left float
- both ៖ វានឹងធ្លាក់មកក្រោម element ដែល float
- none ៖ វានឹងទៅបំពេញ element ដែល float។

Adding Graphics to Web Page

background-image:

```
background-image: url(image/bg.jpg);
```

ប្រើសំរាប់ដាក់ background ទៅអោយ tag ណាមួយ។

background-repeat:

```
background-repeat: repeat|repeat-x|repeat-y|no-repeat;
```

ប្រើសំរាប់កំណត់ការ repeat របស់ background image ។

background-position:

```
background-position: [left|center|right|#px|%] [bottom|center|top|#px|%];
```

ប្រើសំរាប់កំណត់ទីតាំងរបស់ background image។

html { height: 100%; }

មានប្រយោជន៍សំរាប់ពេល ដែលមានបញ្ហាក្នុង firefox browser ដែលកំពស់របស់ body មិនស្មើនឹងកំពស់របស់ page ទាំងមូល

background-attachment:

```
background-attachment: scroll|fixed;
```

fixed: កំណត់កុំអោយ background-image រំកិលតាមការដែលយើងបាន scroll

scroll: កំណត់អោយ background-image រំកិលតាមការដែលយើងបាន scroll

background:

```
background: url(...) #FFF center center fixed ...;
```

សំរាប់កំណត់គ្រប់ property របស់ background

Navigation Technique

Requirement

ជាធម្មតាក្នុងការបង្កើតនូវ Link គេតែងតែប្រើ `<a/>`,``,`` tag ដើម្បីដាក់ link ជាក្រុមៗ។ ការធ្វើបែបនេះវាអាច អោយរចនានូវប្រភេទនៃ navigation option បានច្រើន ដូចជា Vertical Navigation, Horizontal Navigation។

Vertical Navigation Sample

Preloading Techniques

Pixy Techniques

ជាវិធីសាស្ត្រដែលគេប្រើក្នុង preloading-rollover។ ក្នុងវិធីសាស្ត្រនេះមានដំណាក់កាលដូចខាងក្រោម៖

- ត្រូវបង្កើតនូវរូបភាពមួយដែលមានទាំង normal state, rollover state
- ត្រូវកំណត់គំលាតពី state នីមួយៗអោយបានជាក់លាក់
- ត្រូវកំណត់ style អោយ link ដូចខាងក្រោម៖(ឧបមាថាគំលាត 20px)
- `a{ background: url(state.jpg) 0px 0px;`
- `a:hover{ background: url(state.jpg) 0px -20px;`

Using Hidden Loading

រូបភាពត្រូវបាន download ទៅកាន់ client តែម្តងគត់បើទោះជា រូបភាពមួយនោះត្រូវបានប្រើច្រើនដងក្នុង page ក៏ដោយ។ ដូចនេះយើងគ្រាន់តែបង្កើតនូវ div tag មួយរួចដាក់នូវរូបភាពណាដែលត្រូវ load ខាងក្នុងវា បន្ទាប់មកកំណត់ visibility:hidden, position:absolute ជាការស្រេច។

Sliding Door Technique

ជាវិធីសាស្ត្រដែលគេប្រើសំរាប់បង្កើតនូវប៊ូតុងមួយដែលអាច រីករួមទំហំបានដែលយើងបានកំណត់។

- បង្កើតនូវប៊ូតុងមួយដែលមានទំហំអតិបរមា
 - បន្ទាប់ត្រូវបំបែករូបភាពជាពីរចំរៀក។ ចំរៀកទី១ដាក់ តែម្ខាងដែលប្រែប្រួលខាងឆ្វេងគេ បង្អស់ និងចំរៀកទី២ ដាក់រូបភាពដែលនៅសល់។
 - ក្នុងកូដ HTML គ្រប់ link ត្រូវដាក់ក្នុង li ។
 - ក្នុង style sheet ត្រូវកំណត់ background របស់ li ដោយរូបភាពទី២
- ```
li{ background: url(btn_right.png) right top;
```
- ក្នុង a ត្រូវកំណត់
- ```
a{ background: url(btn_left.png) left top;
```

Formatting Table and Form

border-collapse:

```
border-collapse: collapse|separate;
```

ប្រើសំរាប់កំណត់នូវ ការដាច់គ្នា រឺជាប់គ្នារបស់ border នៃ table ។

CSS Page Layout

CSS Page Layout មានពីរប្រភេទគឺ absolute positioning និងfloat។ Absolute Positioning គឺជាការដាក់ element មួយ អោយនៅទីតាំងណាមួយនៃ Page តាមរយៈការកំណត់ចំនួន pixel។

Type of Web Page Layout

Fixed Width

គឺជាការកំណត់ទទឹងជាក់លាក់របស់ Page ទោះបីជាទំហំរបស់ browser ប្រែប្រួលយ៉ាងណាក៏ដោយដែលមានក្នុង Page មិនរីករួមទៅតាមការប្រែប្រួលនោះទេ

ចំពោះទំហំ Screen 800x600 គេតែងតែជ្រើសយកទំហំរបស់ Page 760px ដើម្បីបន្ទុលកន្លែងទំនេរខ្លះសំរាប់ scroll bar និងផ្នែកផ្សេងៗនៃ Browser

ចំពោះទំហំ Screen 1024x768 គេតែងជ្រើសយក 950px។

Liquid

គឺជាការរចនាអោយប្រែប្រួលទៅតាម ទំហំរបស់ browser វានឹងចង្អៀត រឺទូលាយអាស្រ័យទៅតាមទំហំរបស់ browser។

Elastic

គឺជាការរចនាដែល កំណត់ទំហំរបស់ page ទៅតាមទំហំរបស់អក្សរដោយប្រើប្រាស់ em សំរាប់កំណត់។ ដូច្នោះពេលដែល user ពង្រីកអក្សរនៅលើ browser នោះទំហំ container នឹងប្រែប្រួលទៅតាមទំហំរបស់អក្សរបានយ៉ាងល្អ។



Fixed



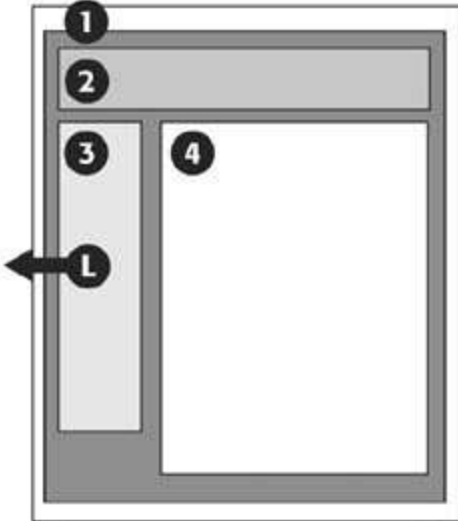
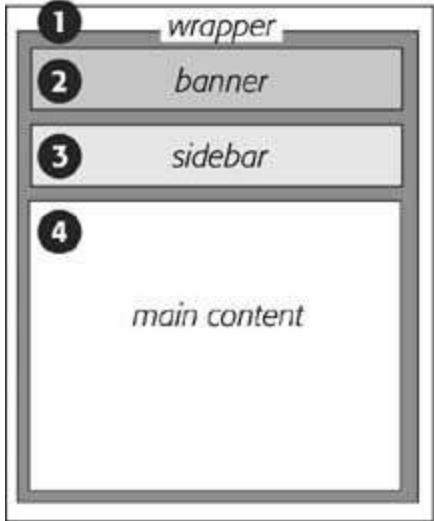
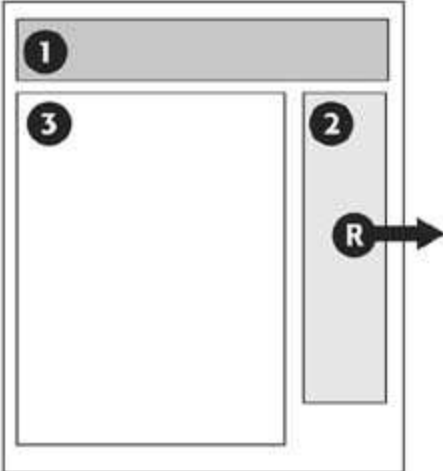
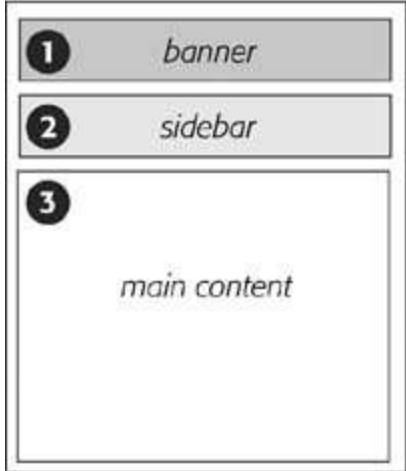
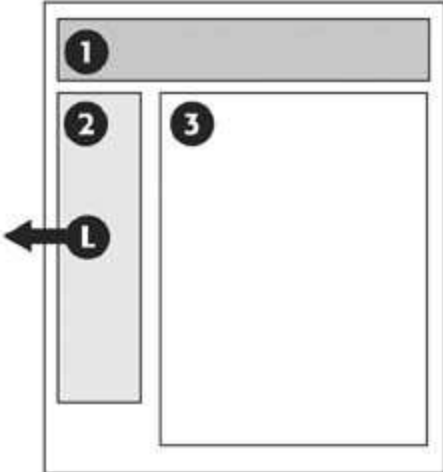
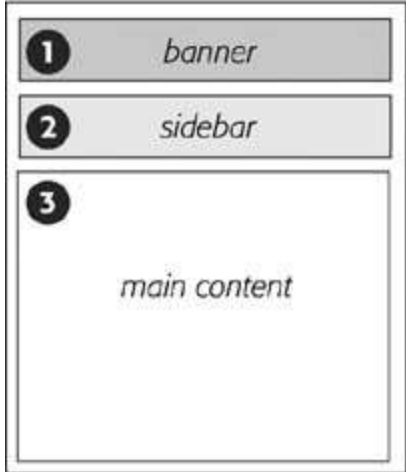
Liquid



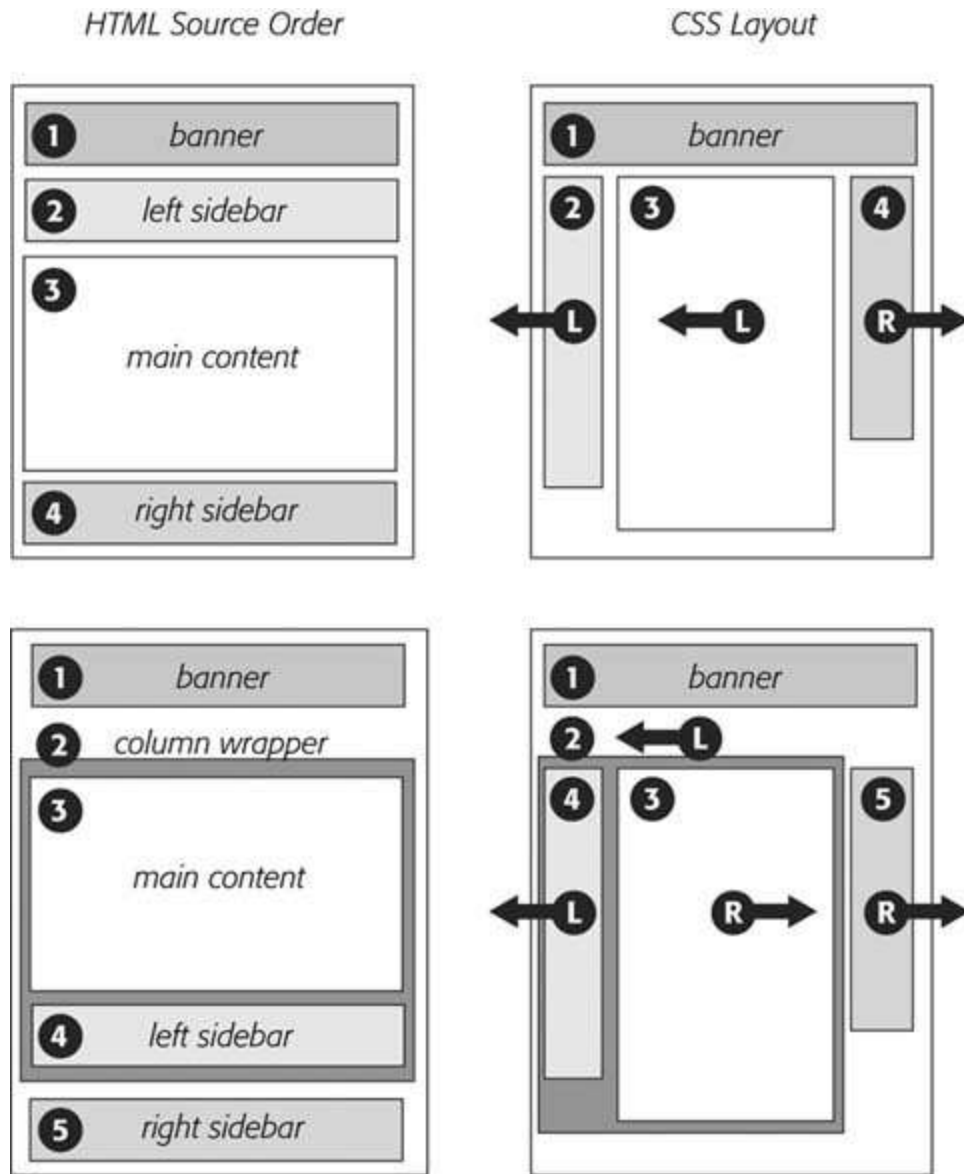
Elastic

HTML Source Order

CSS Layout



width: 760px



Step to Create Page Layout

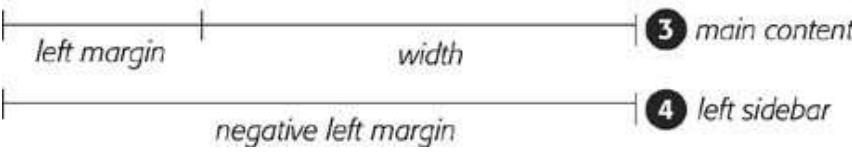
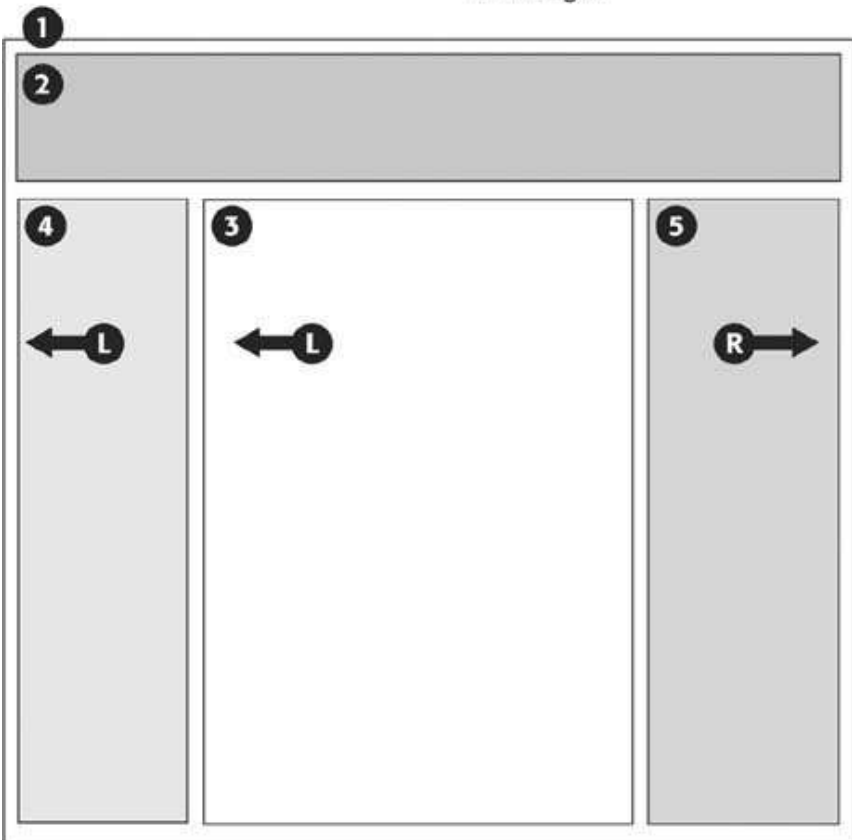
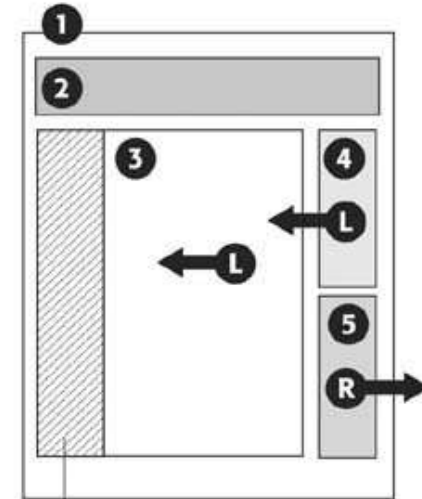
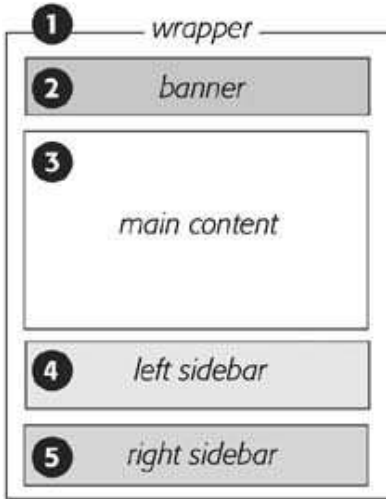
1. ដាក់រាល់ Page Content ក្នុង wrapper <div>
បង្កើតនូវ container សំរាប់ការកំណត់ទទឹងជាក់លាក់មួយ សំរាប់ដាក់រាល់ content នៅលើ page និងផ្តល់ភាពងាយស្រួល មិនចាំបាច់កំណត់ទំហំ banner, column, footer ។
2. កំណត់ទទឹងនៃ wrapper <div>
បង្កើត style សំរាប់ wrapper <div> ដើម្បីកំណត់ទទឹងរបស់វា
3. ដាក់ column នីមួយៗក្នុង <div> tag ដោយកំណត់ id និង class attribute:
4. កំណត់Float នៃ divs សំរាប់ column នីមួយៗ
ដាក់ left sidebar និង main content ក្នុង div មួយ ហើយ float left, ក្នុង div នោះ div របស់ left sidebar ត្រូវ float left និង main content ត្រូវ float right ។ ចំនែក right side bar ត្រូវ float right.
5. កំណត់ width នៃ column នីមួយៗ

ជាទូទៅ យើងគួរតែកំណត់ width នៃ float element ដោយពឹងផ្អែកលើការចនាសម្រួលអ្នកអ្នកអាចបន្ថែម padding, margin និង border ។

ចាំថា width សរុបរបស់ column នីមួយៗគឺបានមកពីការបូកបញ្ចូលគ្នារវាង left and right margin, padding, និង border

HTML Source Order

CSS Layout



6. បន្ថែម left margin ទៅកាន់ main column
7. កំនត់ left margin របស់ left_sidebar : -587px;