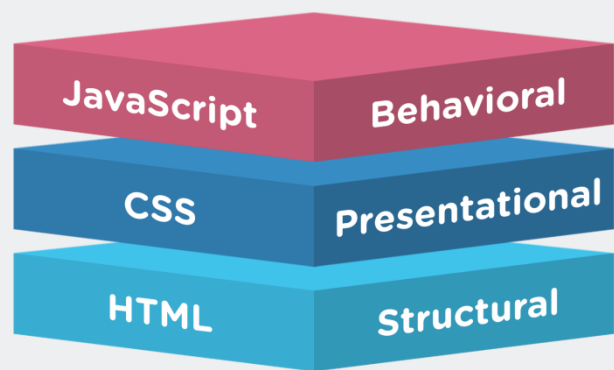


ЗАНЯТТЯ 9

ПРОДОВЖЕННЯ БЛОКІВ ТА ПОЗИЦІОНУВАННЯ. FLOAT.



Курси "Веб-розробка"
за програмою
BrainBasket
"WOW Teachers"

ЗМІСТ ТЕМИ

1. Елементи, що плавають (float).

- a) CSS атрибути **float** та **clear**.
- b) Побудова макету з кількома колонками на базі **float**.
- c) Очистка плаваючих елементів. Метод **Clearfix**.
- d) Використання **display: inline-block** для керування розташуванням блоків.

ЗМІСТ ТЕМИ

2. Глибше про позиціювання.

- a) Абсолютне позиціювання (position: absolute).
- b) Фіксоване позиціювання (position: fixed).
- c) Приклад фіксованого верхнього меню.
- d) Вихід з ситуації, коли це меню накриває контент (відступ контенту згори).

ЗМІСТ ТЕМИ

3. Семантичні теги.

- a) Елемент <header>
- b) Елемент <nav>
- c) Елемент <article>
- d) Елемент <section>
- e) Елемент <aside>
- f) Елемент <footer>
- g) Елемент <address>
- h) Елемент <main>
- i) Елемент <figure>
- j) Елемент <figcaption>
- k) Елемент <time>
- l) Елемент <mark>

1_a CSS атрибут float.

Атрибут **float** визначає, по якій стороні буде вирівнюватися елемент, при цьому інші елементи будуть обтікати його з інших сторін. Коли значення властивості **float** буде *none*, елемент виводиться на сторінці як зазвичай, при цьому допускається, що один рядок оточуючого тексту може бути на тій же лінії, що і сам елемент.

- **Синтаксис**
- float: left | right | none | inherit
- **Значення**
- **left** - вирівнює елемент по лівому краю, а всі інші елементи, на кшталт тексту, обтікають його по правій стороні.
- **right** - вирівнює елемент по правому краю, а всі інші елементи обтікають його по лівій стороні.
- **none** - обтікання елемента не задається.
- **inherit** - наслідуює значення батьківського елемента.

1_a CSS атрибут clear.

Атрибут **clear** встановлює, з якого боку елемента заборонено його обтікання іншими елементами. Якщо задано обтікання елемента за допомогою властивості **float**, то **clear** скасовує його дію для зазначених сторін.

Синтаксис

- `clear: none | left | right | both | inherit`

Значення

- **none** – відміняє дію властивості `clear`, при цьому обтікання елемента відбувається так як задано з допомогою властивості `float` або інших налаштувань.
- **both** - скасовує обтікання елемента одночасно з правого і лівого краю. Це рекомендується встановлювати, коли потрібно зняти обтікання елемента, але невідомо точно з якого боку.
- **left** - скасовує обтікання з лівого краю елемента. При цьому всі інші елементи на цій стороні будуть опущені вниз, і розташовуватися під поточним елементом.
- **right** - скасовує обтікання з правого боку елемента.
- **inherit** - наслідує значення батьківського елемента.

1_b Побудова макету з кількома колонками на базі float.

Кожна колонка – це окремий блок, для якого встановлюється обтікання справа (**float: left;**), а ширина блоку встановлюється пропорційно до кількості колонок (**width: 50%;**).

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Дві колонки</title>
<style>
* {
  box-sizing: border-box;
}
p {
  width: 50%;
  float: left;
  padding: 10px;
}
section {
  clear: left;
}
</style>
</head>
<body>
<h1>Дві колонки</h1>
<div>
  <p id="p1">Еней .....</p>
  <p id="p2">був парубок моторний ....</p>
  <section>Це звичайне наповнення після блоків параграфа.</section>
</div>
</body>
</html>
```

1_с Очистка плаваючих елементів. Метод Clearfix.

Clearfix - допоміжний псевдоклас допомагає вирішувати ситуації зі злипаючою висотою у батьківського div'у коли всередині нього розташований плаваючі елементи наприклад з float: left.

```
.clearfix:before,  
.clearfix:after {  
  content: "";  
  display: table;  
  clear: both; }
```

```
<div class="container clearfix">
```


1_d Використання `display: inline-block` для керування розташуванням блоків.

У специфікації HTML представлені теги, які є спочатку або рядковими (властивість **`display:inline`**), або блоковими (**`display: block`**). Однак з виходом стандарту CSS3 з'явилася можливість змінювати відображення елемента на рядково-блочний (**`display: inline-block`**).

По суті рядково-блокові елементи є чимось середнім між рядковими і блоковими. Тобто вони мають властивості і тих, і інших елементів, комбінуючи їх. Яким чином? - У тексті вони поведуться, як рядкові елементи, тобто займають рівно стільки місця, скільки тексту міститься в них (якщо не задані ширина або відступи), однак при призначенні властивостей `box-моделі`, інтерпретують їх, як блокові елементи.

Приклади застосування:

- для виділення фрагментів тексту (рамкою, кольоровим тлом тощо) в середині текстового контенту;
- для розподілу контенту на колонки;
- для формування об'єктів - кнопок з текстовими надписами.

2_a Абсолютне позицювання (position: absolute).

Стиль **position: absolute** вказує, що елемент абсолютно позиціонується, при цьому інші елементи відображаються на веб-сторінці немов абсолютно позиційованого елемента і немає.

Положення елемента задається властивостями **left, top, right і bottom**.

Також на положення впливає значення властивості **position** батьківського елемента. Так, якщо у батька значення **position** встановлено як **static** або батька немає, то відлік координат ведеться від краю вікна браузера. Якщо у батька значення **position** задано як **fixed, relative** або **absolute**, то відлік координат ведеться від краю батьківського елемента.

```
position: absolute; /*  
bottom: 15px;  
right: 15px;
```

2_b Фіксоване позиціювання (position: fixed).

Стиль **position: absolute** за своєю дією близький до **absolute**, але на відміну від нього прив'язується до зазначеної властивостями **left**, **top**, **right** і **bottom** точки на екрані і не змінює свого положення при прокручуванні веб-сторінки.

Браузер Firefox взагалі не відображає смуги прокрутки, якщо положення елемента задано фіксованим, і воно не поміщається цілком у вікно браузера. У браузері Opera хоча і показуються смуги прокрутки, але вони ніяк не впливають на позицію елемента.

```
position: fixed;  
top: 10px;  
left: 0;  
height: 30px;  
width: 100%;  
margin: 0;
```

2_с Приклад фіксованого верхнього меню.

```
<div id="menu-top-almost-fixed">
  <ul>
    <li>
      <a href="#">Головна</a>
    </li>
    <li>
      <a href="#">Новини</a>
    </li>
    <li>
      <a href="#">Контакти</a>
    </li>
    <li>
      <a href="#">Пошук</a>
    </li>
  </ul>
</div>
```

```
#menu-top-almost-fixed{
  position: fixed; margin: 0; left: 0; top: 150px;
  height: 30px;
}
#menu-top-almost-fixed ul, #menu-top-almost-
fixed li{
  list-style: none; margin: 0; padding: 0;
}
#menu-top-almost-fixed ul{
  display: block; text-align: center; width: 100%;
  float: left;
}
#menu-top-almost-fixed ul li{
  display: inline; line-height: 30px; width: 120px;
  padding: 0 5px; text-align: center;
}
```

2_d Вихід з ситуації, коли це меню накриває контент (відступ контенту згори).

- **Логотип сайту**

Слоган сайту

Банер

[Головна](#) [Новини](#) [Контакти](#) [Пошук](#)

[вміст сторінки]

[підвал сайту]

3_а Семантична верстка.

Логотип, назва сайту

Головний контент

Меню

Новини

Статті

Авторські права і інша
додаткова інформація

`<header>`

`</header>`

`<nav>`

`<div>`

`<section>`

`<section>`

...

...

`<article>`
`</article>`

`<article>`
`</article>`

...

...

`</section>`

`</section>`

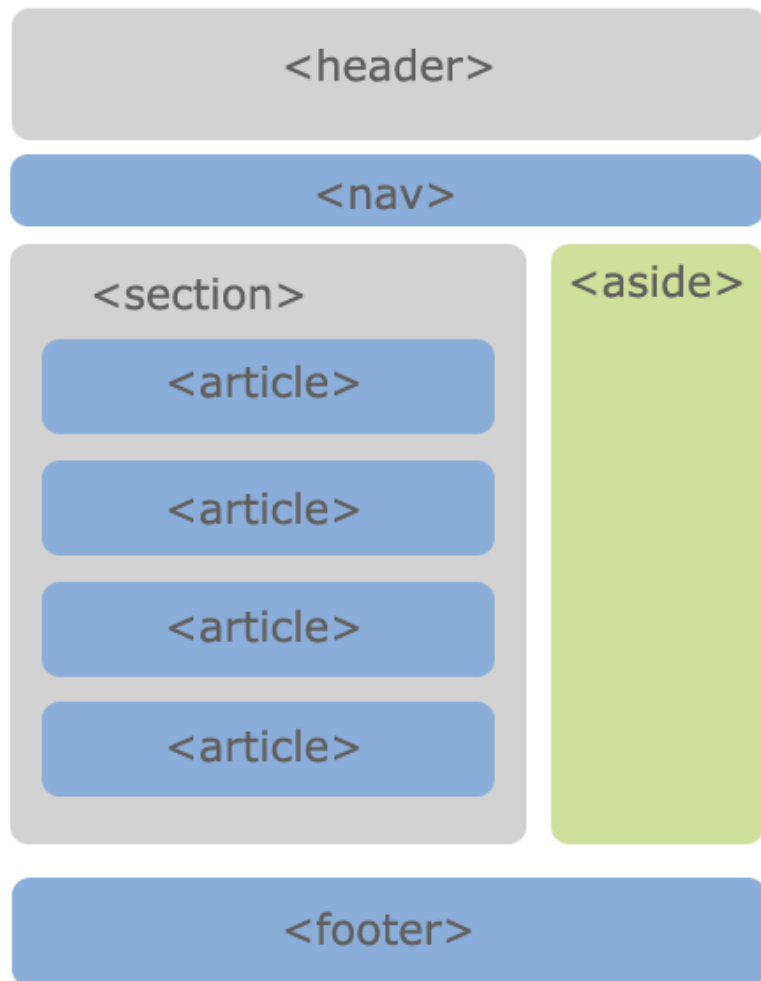
`</div>`

`</nav>`

`<footer>`

`</footer>`

3_а Семантична верстка.



3_а Приклад семантичної верстки.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Приклад блочної верстки</title>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>

<body>

<div id="container">
  <div id="header">
    <h2>Шапка</h2>
  </div>

  <div id="navigation">
    <h2>Блок навігації</h2>
  </div>

  <div id="menu">
    <h2>Меню</h2>
  </div>

  <div id="content">
    <h2>Контент</h2>
  </div>

  <div id="clear">

  </div>

  <div id="footer">
    <h2>Підвал сайту</h2>
  </div>
</div>

</body>
</html>
```

```
body {
  background: #f3f2f3;
  color: #000000;
  font-family: Trebuchet MS, Arial, Times New Roman;
  font-size: 12px;
}

#container {
  background: #99CC99;
  margin: 30px auto;
  width: 900px;
  height: 600px;
}

#header {
  background: #66CCCC;
  height: 100px;
  width: 900px;
}

#navigation {
  background: #FF9999;
  width: 900px;
  height: 20px;
}

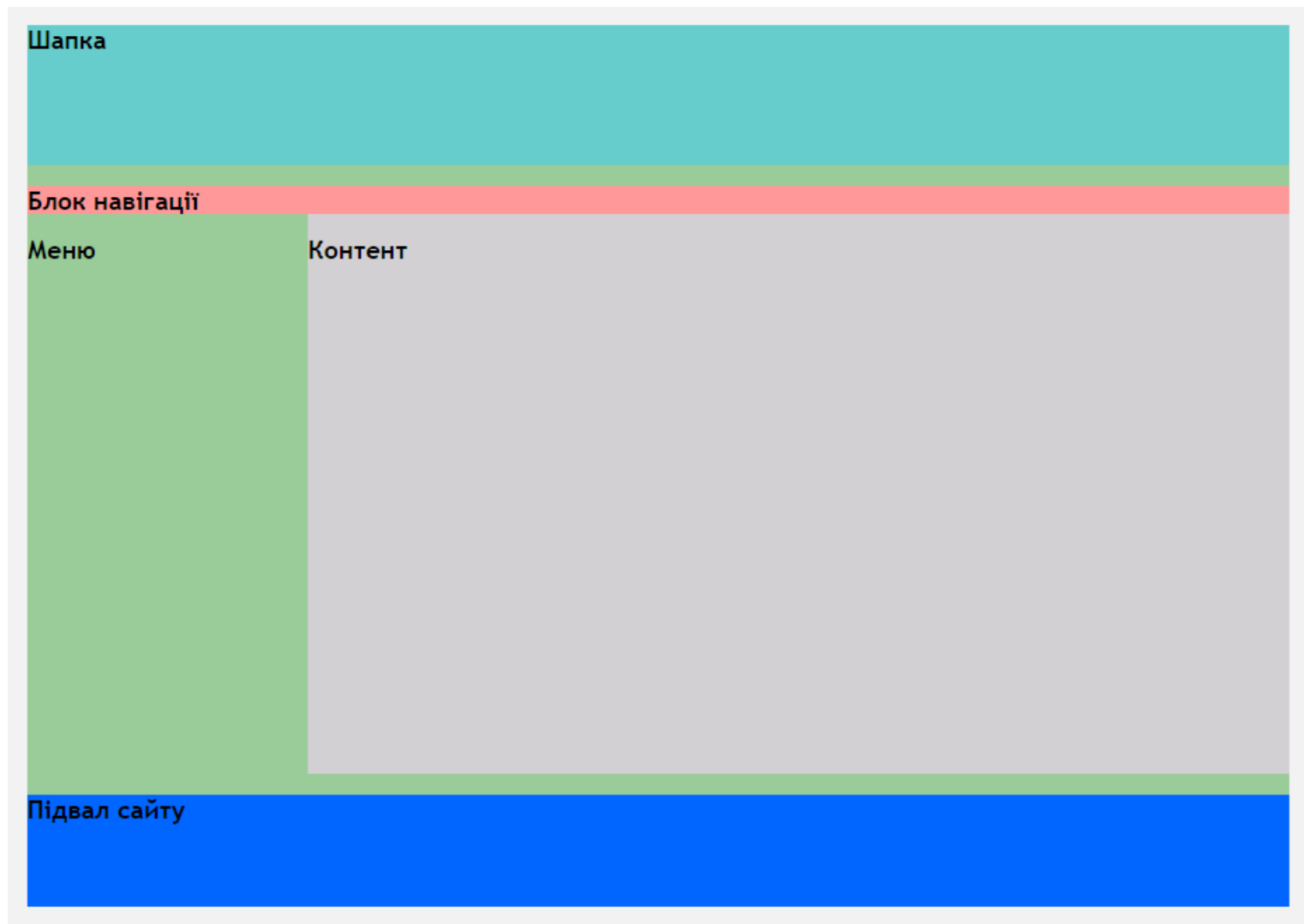
#menu {
  background: #99CC99;
  float: left;
  width: 200px;
  height: 400px;
}

#content {
  background: #d2d0d2;
  float: right;
  width: 700px;
  height: 400px;
}

#clear {
  clear: both;
}

#footer {
  background: #0066FF;
  height: 80px;
  width: 900px;
}
```


3_а Приклад семантичної верстки.



3_b Семантичні теги.

Елемент **<header>**

Утворює вміст верхньої (шапки) частини сторінки або її секції. Об'єднує вступну інформацію і навігаційні елементи, може розташовуватися в будь-якій частині сторінки. У HTML-документі може міститися одночасно кілька елементів **<header>**. Не є обов'язковим.

<header>

<h1>...</h1>

</header>

Елемент може містити основний заголовок, або групу заголовків, які в свою чергу можна помістити в елементах **<hgroup>**.

Парний тег **<header>** не можна поміщати всередину елементів **<footer>**, **<address>** або іншого елемента **<header>**.

3_b Семантичні теги.

Елемент **<hgroup>**

Використовується для угруповання елементів **<h1>** - **<h6>** в разі, коли заголовок має складну структуру, наприклад, має уточнюючі підзаголовки, альтернативні заголовки і т.д.

```
<header>
```

```
<hgroup>
```

```
<h1>...</h1>
```

```
<h2>...</h2>
```

```
</hgroup>
```

```
</header>
```

3_b Семантичні теги.

Елемент <nav>

Призначений для створення блоку навігації веб-сторінки або всього веб-сайту, при цьому не обов'язково повинен знаходитися усередині <header>. На сторінці може бути кілька елементів <nav>. Не замінює теги або , він просто їх обрамляє:

```
<nav>
<ul>
<li><a>...</a></li>
<li><a>...</a></li>
<li><a>...</a></li>
</ul>
</nav>
```

Як елементи панелі навігації можна використовувати не тільки елементи списків:

```
<nav>
<p><a href="">...</a></p>
<p><a href="">...</a></p>
</nav>
```

Також можна додавати заголовки всередину елемента:

```
<nav>
<h2>...</h2>
<ul>
<li><a>...</a></li>
<li><a>...</a></li>
<li><a>...</a></li>
</ul>
</nav>
```

Елемент **<article>**

Використовується для групування записів - публікацій, статей. Являє собою незалежний, відокремлений блок, призначений для багаторазового використання. Як правило, починається з заголовка.

`<article>`

`<header>`

`<h2>...</h2>`

`</header>`

`<p>...</p>`

`</article>`

Елемент **<section>**

Є способом поділу сторінки або статті на тематичні розділи. Не використовується багаторазово, зазвичай містить заголовок. Не є блоком-обгорткою, для цих цілей доречніше використовувати елемент `<div>`.

`<article>`

`<h1>...</h1>`

`<section>`

`<h2>...</h2>`

`<p>...</p>`

`</section>`

`</article>`

3_b Семантичні теги.

Елемент **<aside>**

Групує вміст, пов'язаний з навколишнім його контентом безпосередньо, але яке можна вважати окремим (тобто, видалення цього блоку не вплине на розуміння основного вмісту). Найчастіше елемент позиціонується як бічна колонка (як в книгах) і включає в себе групи елементів `<nav>`, цифрові дані і витягнуті цитати:

`<aside>`

`<h2>...</h2>`

`<p>...</p>`

`</aside>`

3_b Семантичні теги.

Елемент **<footer>**

Формує вміст нижньої частини (підвалу) сторінки або її секції. Призначений для розміщення інформації про веб-ресурс, наприклад, відомості про авторські права, посилання на умови використання, контактну інформацію, посилання на пов'язаний вміст і т.д. В одному веб-документі може бути кілька елементів `<footer>`. Як кожна сторінка, так і кожна стаття може мати свій елемент `<footer>`, більш того, `<footer>` можна помістити в елемент `<blockquote>`, щоб вказати джерело цитування.

`<footer>`

`<address>...</address>`

`<small>...</small>`

`</footer>`

Елемент **<address>**

Використовується для визначення контактної інформації автора/власника документа або статті. Для позначення автора документа тег розміщують всередині елемента **<body>**, для відображення автора статті - всередині тега **<article>**. У браузері зазвичай відображається курсивом.

3_b Семантичні теги.

Елемент **<main>**

Групує основний контент веб-сторінки. Вміст елемента має бути унікальним на сторінці і не повинен відображатися де-небудь ще на сайті. Тому що повторюваний на сайті контент, наприклад навігація по сайту, пошук по сайту, логотип, контактні дані та інше, не можна поміщати в елемент **<main>**.

Основне призначення елемента в тому, щоб прив'язати значення **main** атрибута **role** ARIA до певного елемента HTML. Це допоможе скрінрідерам визначити, де починається основний зміст документа.

3_b Семантичні теги.

Елементи **<figure>** і **<figcaption>**

Елемент `<figure>` представляє автономний контент, можливо з заголовком `<figcaption>`, який є самостійним елементом потоку. Елемент може бути переміщений з основного контенту документа в бічну колонку або додаток, не зачіпаючи потік документа. За допомогою елемента `<figure>` можна додавати замітки до ілюстрацій, фотографій, діаграм, фрагментів коду, даючи підписи або коментарі.

`<figure>`

``

`<figcaption>Осінній ліс</figcaption>`

`</figure>`

Елемент **<time>**

Визначає час (24 години) або дату за григоріанським календарем з можливим зазначенням часу і зміщення часового поясу. Текст, укладений в даний тег, не має стильового оформлення браузером.

Для тега доступний атрибут **datetime**, вміст якого вказує на те, що буде бачити користувач на екрані свого комп'ютера.

Елемент **<bdi>**

Відокремлює фрагмент тексту, який повинен бути ізольований від решти тексту для двонаправленого форматування тексту. Використовується для текстів, написаних одночасно на мовах, що читаються зліва направо і справа наліво. Атрибут `dir` за замовчуванням має значення `auto`.

Елемент **<mark>**

Текст, поміщений всередину тега **<mark>**, виділяється за замовчуванням жовтим кольором (колір фону і колір шрифту у виділеному блоці можна змінити, задавши певні css-стилі). За допомогою даного тега можна відзначати важливий вміст, а також ключові слова.

Елемент **<wbr>**

Показує браузеру місце, де можна додати розрив довжини рядка в разі потреби.

КЛАСНА РОБОТА (ПРАКТИКА)

Створити стилізовану веб-сторінку, на якій розмістити блоки з різними способами обтікання.

Плаваючі елементи

width: 250px; height: 50px; border: 1px solid black;

`background-color: #A52A2A; float: left; height: 150px;`

`background-color: #5F9EA0; clear: both;`

`background-color: #FF7F50;`

`background-color: #DEB887; float: right; height: 100px;`

Це регулярний вміст, що продовжується після коробок абзацу.

КЛАСНА РОБОТА (ПРАКТИКА)

	CSS
<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <title>Плаваючі елементи</title> </head> <body> <h1>Плаваючі елементи</h1> <div> width: 250px; height: 50px; border: 1px solid black; <p id="p1"> background-color: #A52A2A; float: left; height: 150px;</p> <p id="p2"> background-color: #DEB887; float: right; height: 100px;</p> <p id="p3"> background-color: #5F9EA0; clear: both;</p> <p id="p4"> background-color: #FF7F50;</p> </div> Це регулярний вміст, що продовжується після коробок абзацу. </section> </div> </body> </html></pre>	<pre>div { background-color: #00FFFF; } p { width: 250px; height: 50px; border: 1px solid black; color: yellow; } #p1 { background-color: #A52A2A; float: left; height: 150px; } #p2 { background-color: #DEB887; float: right; height: 100px; } #p3 { background-color: #5F9EA0; clear: both; } #p4 { background-color: #FF7F50; } section { clear: left; }</pre>

ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)

- <http://htmlbook.ru/css/display>
- <http://htmlbook.ru/css/float>
- <http://htmlbook.ru/css/clear>
- <https://vivasart.com/lab/kak-ispolzovat-clearfix-v-css>
- <https://webformyself.com/ochistka-plavayushhix-elementov-obzor-razlichnyx-metodov-clearfix/>
- <https://webref.ru/layout/after-before/pseudo-elements>
- [http://menushka.at.ua/load/veb majstru/sajt svojimi rukami/fiksovane gorizontalne menju/19-1-0-32](http://menushka.at.ua/load/veb_majstru/sajt_svojimi_rukami/fiksovane_gorizontalne_menju/19-1-0-32)
- [https://puzzleweb.ru/css/15 navbar.php](https://puzzleweb.ru/css/15_navbar.php)
- <https://html5book.ru/html5-semantic-elements/#part1>

ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)

Варіант 1 (за зразком гіпертексту)

Створити стилізовану веб-сторінку, на якій розмістити блоками текст у три колонки.

Три колонки

Еней був парубок моторний
І хлопець хоть куди козак,
Удався на всеє зле проворний,
Завзятий од всіх бурлак.
Но греки, як спаливши Трою,
Зробили з неї ширту гною,
Він, взявши торбу, тягу дав;
Забравши деяких троянців,
Осмалених, як гиря, ланців,
П'ятами з Трої наживав.

Еней, попливши синім морем,
На Карфагену оглядався;
Боровсь з своїм, сердега, горем,
Сльзьми, бідняжка, обливався.
Хоть од Дидони плив поспішно,
Та плакав гірко, неутішно.
Почувши ж, що в огні спеклась,
Сказав: "Нехай їй вічне царство,
Мені же довголітнє панство,
І щоб друга вдова найшлась!"

Еней-сподар, посумовавши,
Насиду трохи вгамовався;
Поплакавши і поридавши,
Сивушкою почастовався;
Но все-таки його мутило
І коло серденька крутило,
Небіжчик часто щось вздыхав;
Він моря так уже боявся,
Що на богів не покладався
І батькові не довіряв.

Це звичайне наповнення після боксів параграфа.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Три колонки</title>
<style>
* {
  box-sizing: border-box;
}
div {
  background-color: #00FFFF;
}
p {
  width: 33%;
  float: left;
  padding: 10px;
}
#p1 {
  background-color: #FFFF00;
}
#p2 {
  background-color: #DEB887;
}
#p3 {
  background-color: #00BFFF;
}
section {
  clear: left;
}
</style>
</head>
<body>
<h1>Три колонки</h1>
<div>
  <p id="p1">Еней був парубок ...</p>
  <p id="p2">Еней, попливши синім морем...</p>
  <p id="p3">Еней-сподар, посумовавши...</p>
  <section>Це звичайне наповнення після боксів
  параграфа.</section>
</div>
</body>
</html>
```

ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)

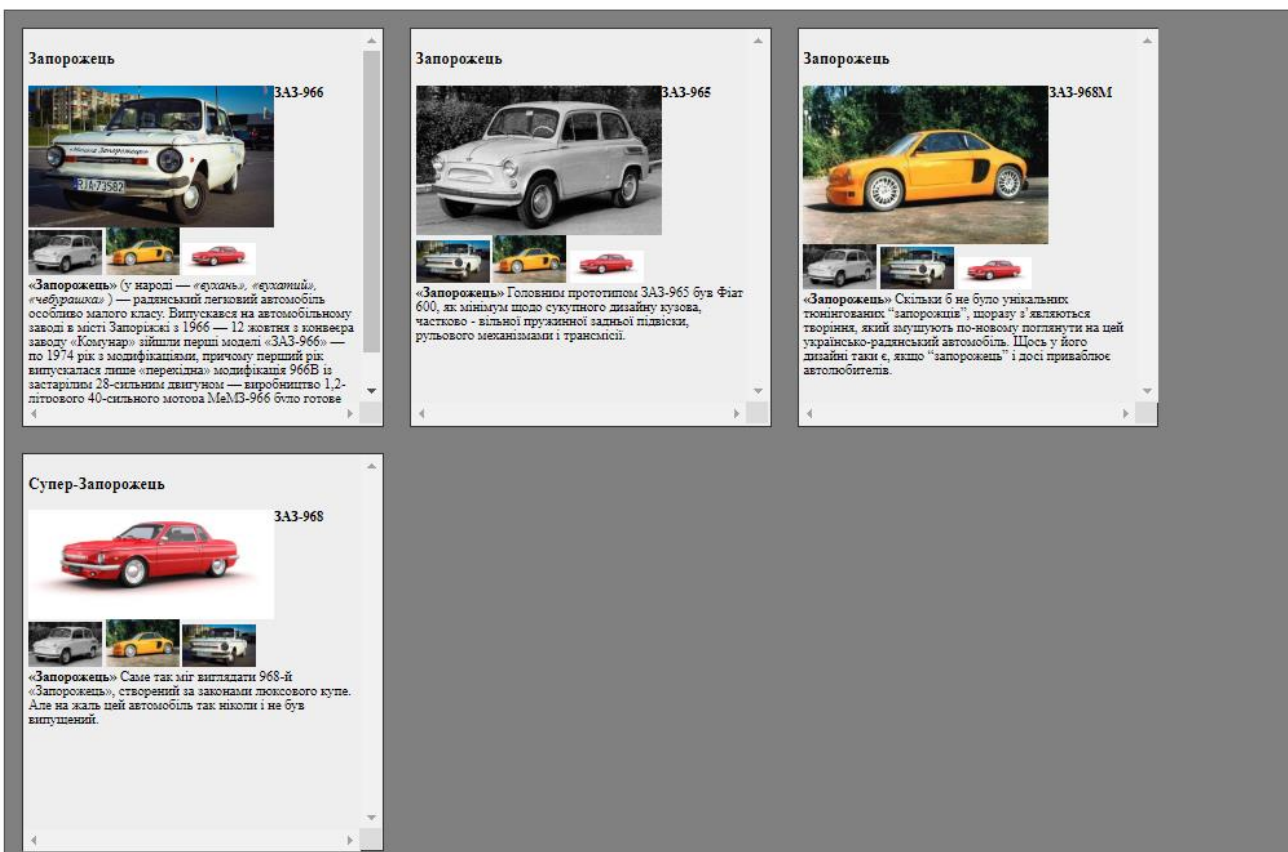
Варіант 1 (за зразком гіпертексту) Медіа каталог стрічкою.

HTML	CSS
<pre><html> <head> <meta charset="utf-8"> <title>Шафа</title> </head> <body> <h1>Media Box</h1> <div id="box"> <div class="content1"> </div> <div class="content2"> </div> <div class="content3"> </div> <div id="box"> <div class="content1"> </div> <div class="content2"> </div> <div class="content3"> </div> </BODY> </html></pre>	<pre>.body { background-color: pink; } #box { display:inline-block; background-color: blue; padding: 10px; border: 5px solid black; width: 100%; height: 150pt; margin-top: 30px; overflow: auto; } #box:hover {background: red;} .content1 { display:inline-block; width: 25%; height: 97%; overflow: auto; background-color: #00FA9A; } .content2 { display:inline-block; width: 50%; height: 97%; overflow: auto; background-color: #90EE90; /* green */ } .content3 { display:inline-block; width: 23%; height: 97%; overflow: auto; background-color: #FFFF00; } h1 { margin-bottom: 30px; } img {display:inline-block;}</pre>

ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)

Варіант 2 (за зразком веб-сторінки)

Створити стилізовану веб-сторінку, на якій розмістити блоки «плиткою» з обтіканням за такою структурою: на «стенді» (.stend) розміщено чотири «плитки» (.plytka). У кожній плитці розміщено наступні блоки: заголовок, контейнер з фото (.photo), всередині якого одне велике (img) і три маленьких (logo_2) зображення в ряд та текстовий контейнер з описом відображеного на фото.



```
.stend {  
  width: 100%;  
  border: 1px #555 solid ;  
  padding: 5px;  
  margin: 15px;  
  float: left;  
  background-color: gray;  
}  
  
.plytka{  
  width: 25%;  
  height: 450px;  
  border: 2px #333 solid;  
  background: #eee;  
  float: left;  
  padding: 5px;  
  margin: 15px;  
  overflow: scroll;  
}  
  
img {  
  width: 100%;  
}  
img.logo_2 {  
  width: 30%;  
}  
  
.photo{  
  width: 75%;  
  float: left;  
}  
}
```