

CREATING WEB-PAGES

Using HTML5 and CSS3

HTML



CSS



Урок 3

Списки. CSS відступи та поля

ЗМІСТ

1. Списки. CSS-відступи та поля	4
Списки	4
Впорядкований (нумерований) список	4
Маркований список	11
Список визначень	13
Багаторівневі або вкладені списки	17
2. CSS-стилі для списків	22
Використання псевдоелемента <code>::before</code> для стилів списків	30
Що таке псевдоеlementи <code>::before</code> та <code>::after</code> ?	30
Оформлення багаторівневих списків. Вкладені селектори	33

3. CSS-властивості для керування відступами і полями	43
Внутрішні та зовнішні відступи.	
Загальна інформація.	43
Зовнішні відступи — margin.	44
Внутрішні відступи — padding	49
Значення відступів за замовчуванням.	51
4. Домашнє завдання	56
Зовнішній вигляд завдань.	57

Матеріали уроку прикріплені до цього PDF-файлу. Щоб отримати доступ до матеріалів, урок необхідно відкрити у програмі Adobe Acrobat Reader.

1. Списки. CSS-відступи та поля

Списки

На багатьох сторінках в інтернеті використовують перелік будь-яких варіантів послуг або товарів. Для такого переліку чудово підходять списки. У HTML їх існує 3 види: списки визначень, впорядковані (нумеровані) та невлпорядковані (марковані). За допомогою списків можна впорядкувати та систематизувати різні види даних та подати їх у зручному для користувача вигляді.

Впорядкований (нумерований) список

Впорядкований (нумерований) список представляє собою блок з елементів, кожний з яких за замовчуванням має вигляд пронумерованого списку арабськими цифрами:

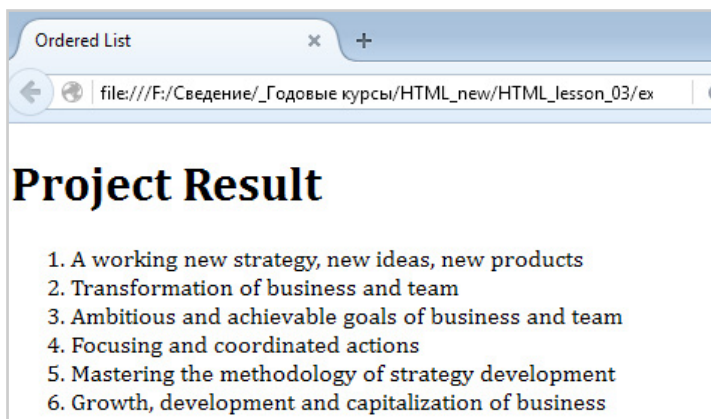


Рисунок 1

В HTML-розмітці нумерований список формується за допомогою тегу `` (від англ. *ordered list*) та вкладених в нього тегів `` (від англ. *list item*)

```
<ol>
  <li> Effective new strategy, new ideas, new
    products</li>
  <li> Business and team transformation</li>
  <li> Ambitious and achievable business and team
    goals</li>
  <li> Focusing and coordination of actions</li>
  <li> Mastering the methodology of strategy
    development</li>
  <li> Growth, development and capitalization of
    business</li>
</ol>
```

Файл прикладу — *ordered_list.html*, додається до уроку і прикріплен до цього PDF-файлу.

Замість арабських цифр ми можемо застосувати нумерацію римськими цифрами або літерами латинського алфавіту, причому як малими, так і великими, а також вказати нумерацію на основі грецької, вірменської або грузинської мов. Для цього в HTML4 існував атрибут *type*, який на даний момент є застарілим. У HTML5 замість нього прийнято вказувати тип списку за допомогою стилів, а саме за допомогою CSS-властивості *list-style-type*:

```
list-style-type: decimal (за замовчуванням, можна не
    вказувати) | decimal-leading-zero |
    armenian | georgian | lower-alpha |
    lower-greek | lower-latin |
    lower-roman | upper-alpha |
    upper-latin | upper-roman;
```

Візуально ці стилі будуть представляти собою таку нумерацію (рис. 2-3):

- арабські цифри (1, 2, 3, ...) — **decimal**;
- арабські цифри з нулем попереду для цифр першого десятку (01, 02, 03, ..., 09) — **decimal-leading-zero**;
- вірменська нумерація — **armenian**;
- грузинська нумерація — **georgian**;
- прописні латинські літери (A, B, C, ...) — **upper-alpha** или **upper-latin**;



Рисунок 2

- малі латинські літери (a, b, c, ...) — **lower-alpha** или **lower-latin**;
- римські цифри верхнього регістра (I, II, III, ...) — **upper-roman**;
- римські цифри нижнього регістра (i, ii, iii, ...) — **lower-roman**;
- малі букви грецького алфавіту — **lower-greek**.

Подивитися на зовнішній вигляд списків з різними стилями можна у файлі *ordered-list-style.html*.

Why you should travel with us in Georgia

```
list-style-type: georgian;
```

- ა. Ready tours developed by Georgian experts
- ბ. We have been working in the field of tourism for over 10 years
- გ. Georgian-Russian team
- დ. Own fleet and drivers
- ე. Own team of trained guides
- ვ. Friendly and flexible staff
- ზ. We are always online (24/7)

Cities and popular resorts in Armenia

```
list-style-type: armenian;
```

- Ա. Yerevan: the capital of Armenia
- Բ. Gyumri: the cultural capital of Armenia
- Գ. Tsaghkadzor" the ski resort of Armenia
- Դ. Shushi: the cultural center of Artsakh
- Ե. Jermuk: high mountain resort

Why it is worth buying real estate at Dionis Real Estate

```
list-style-type: lower-greek;
```

- α. More than 20 years in the real estate market. These years have allowed us to gain a reputation as a reliable company and extensive business ties.
- β. Our company is a developer with a huge experience in the construction industry. That's why we always thoroughly check the quality of the construction of foreign objects and their legality.
- γ. Wide choice of real estate in Greece in the segment of economy class.
- δ. All the staff of the company speaks English.
- ε. Support of the client at all stages of the real estate acquisition, both in Russia and in Greece.

Рисунок 3

Нумеровані списки мають також додаткові атрибути: **start** для тегу `` і **value** для тегу ``. У цьому випадку список починається не з цифри 1, а з тієї, що вказана в атрибуті **start**. Як правило, цей атрибут буває необхідним, якщо список потрібно розірвати будь-яким поясненням або зображенням. Наприклад (рис. 4):

Requirements to a candidate for a position in the company IT-Software

At an interview with a candidate for a vacant position, first of all, the following indicators are evaluated:

- Candidate's knowledge and skills
- Education
- IT experience
- Team work skills

After a successful interview, the candidate will be offered the following tests:

- English proficiency test
- External certification
- Knowledge assessment
- Evaluating performance in a time-limited environment

Рисунок 4

```
<p> At an interview with a candidate for a vacant
position, first of all, the following indicators
are evaluated:</p>
<ol>
  <li>Candidate's knowledge and skills</li>
  <li>Education</li>
  <li>IT experience</li>
  <li>Team work skills</li>
</ol>
<p>After a successful interview, the candidate
will be offered the following tests:</p>
```



```

<ol start="5">
  <li>English proficiency test</li>
  <li>External certification</li>
  <li>Knowledge assessment </li>
  <li>Evaluating performance in a time-limited
    environment</li>
</ol>

```

Слід зазначити, що він використовується вкрай рідко. Також рідко використовується атрибут **value** для елемента **li**, який змінює порядок нумерації, починаючи з тієї цифри, яка вказана як значення атрибута. Цей атрибут має пріоритет над атрибутом **start**, якщо вони вказані в тому самому списку. Крім того, ці 2 атрибути застосовуються до будь-якої нумерації (латинськими літерами, римськими цифрами тощо). У прикладі виділені червоним кольором ті елементи **li**, які мають атрибут **value** (рис. 5):

Promotions, bonus programs, events

1. early booking (EB) system;
2. last minute tours at special offers (S01-S04);
3. partner's special offers (PS0);
10. discount system for regular and corporate clients;
11. help in preparing a bank loan for travel;
12. promotions and bonus programs are regularly held;
30. refer-a-friend - We offer you loyalty and various preferential terms when purchasing services from our partners;
31. regular events and open days dedicated to tourist innovations and tourist season opening!
32. encouraging clients with gift certificates.

Рисунок 5

```

<ol>
  <li>Early booking (EB) system;</li>
  <li>last minute tours at special offers (S01-S04);</li>
  <li>partner's special offers (PSO);</li>
  <li value="10">discount system for regular and
    corporate clients;</li>
  <li>help in preparing a bank loan for travel;</li>
  <li>promotions and bonus programs are regularly
    held;</li>
  <li value="30">refer-a-friend - we offer loyalty...;</li>
  <li>regular events and open days dedicated to
    tourist innovations and tourist season
    opening!</li>
  <li>encouraging clients with gift certificates.</li>
</ol>

```

Ще один атрибут для тегу ``, який рідко використо-
вується — **reversed** — змінює порядок нумерації списку на
зворотній. Тобто замість 1, 2, 3, 4 буде 4, 3, 2, 1. Наприклад,
ось так (рис. 6):

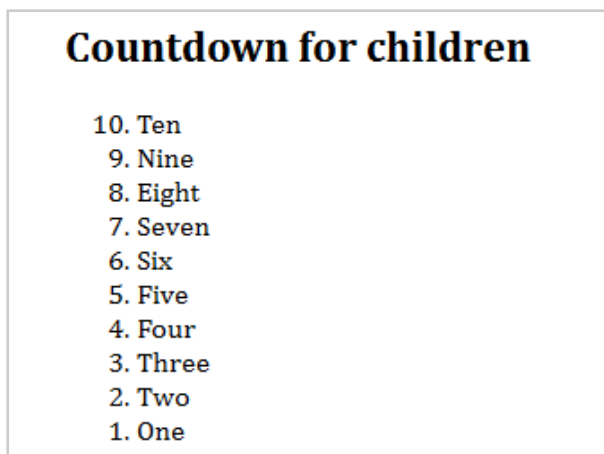


Рисунок 6

```
<ol reversed>
  <li>Ten</li>
  <li>Nine</li>
  <li>Eight</li>
  <li>Seven</li>
  <li>Six</li>
  <li>Five</li>
  <li>Four</li>
  <li>Three</li>
  <li>Two</li>
  <li>One</li>
</ol>
```

Примітка: *Internet Explorer* не підтримує цей атрибут.

Переглянути код прикладів можна в файлі *ordered-list-attributes.html*, який додається до уроку і прикріплен до цього PDF-файлу.

Маркований список

Маркований список відрізняється від нумерованого тим, що замість цифр має маркери (за замовчуванням у вигляді чорних кружечків). Також маркований список називається неупорядкованим (*unordered list*), так як нумерація у ньому не потрібна.

Формується маркований список аналогічно нумерованому двома тегам: **** (від англ. *unordered list*) та вже відомим вам **** (від англ. *list item*).

```
<ul>
  <li>Biographies (255)</li>
  <li>Action (1581)</li>
```

```

<li>Westerns (76)</li>
<li>War (237)</li>
<li>Detective (627)</li>
<li>Documental (64)</li>
<li>Dramas (2814)</li>
<li>Historical (237)</li>
<li>Comedy (1983)</li>
</ul>

```

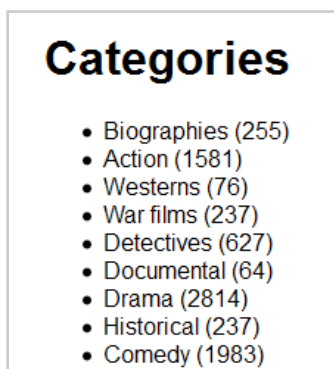


Рисунок 7

Файл прикладу — *unordered_list.html*.

За допомогою застарілого атрибуту **type** раніше визначався вид маркеру, який на даний момент замінюється CSS-властивістю **list-style-type**:

```
list-style-type: disc | circle | square | none
```

- **disc** — темний кружечок (значення за замовчуванням, можна не вказувати);
- **circle** — пустий кружечок;
- **square** — темний квадрат;
- **none** — маркери (нумерація) відсутні.

```
list-style-type: disc; (default)
```

- Lead in the development of hardening the web application in protecting PII (Personally Identifying Information)
- Work side-by-side with product owners to create products and features to help protect users information while remaining seamless to the experience
- Find and build creative solutions to satisfy general audiences and weather enthusiasts while building in compliance for GDPR
- Something about integrating with external data sources/CMS to generate consent flow screens
- Write fluid/liquid/responsive HTML and CSS that works across all three screens (phone, tablet, and desktop)
- Designing and implementing solutions collaboratively to add value to software development efforts in the most cost efficient way. Must be able to effectively communicate JavaScript/HTML/CSS designs to other team members for they could implement the design.
- Ability to define, document and write Unit Test for front-end web technologies
- Create custom JavaScript libraries and utilize APIs
- Knowledge and experience in any JavaScript frameworks, including Angular
- Strong aptitude for both giving and taking direction
- Strong documentation skills to work properly with the general audience (not necessarily technical)

Requirements:

```
list-style-type: circle;
```

- Core aspects of software development using HTML5, CSS and JavaScript
- 3+ years experience in web applications development
- Strong OOP and Functional software design ability. Able to develop systems that meet architectural objectives including reusable, scalable code
- Experience with Angular + NodeJS
- Exposure to build tools like Webpack and Jenkins
- Working knowledge of client- and server-side Javascript
- Ability to care about his/her products and the people who use them
- Ability to follow the timeline and meet deadlines
- Analytical and detail oriented, with the ability to prioritize, execute and deliver projects on time

As a Plus:

```
list-style-type: none;
```

Github profile and contributed projects
Development projects and code samples you can demo

Benefits:

```
list-style-type: square;
```

- Challenging and interesting projects for Fortune 500 companies
- Competitive salary & bonus system
- Your personal Career Advisor & individual career map
- Corporate doctors available 24/7
- Social package
- Certification center and language school
- SoftServe discount program
- A truly flexible working schedule
- Fully equipped gyms and relax zones
- Unforgettable corporate events and even more

Рисунок 8

Файл прикладу — *unordered-list-style.html*.

Список визначень

Списки визначень потрібні на тих сайтах, які, як правило, присвячені навчанню. Наприклад, чудовий сайт webref.ru використовує списки визначень для вказання атрибутів тегів або варіантів властивостей.

Attributes

type	Specifies the type of list marker
reversed	Sets descending order of the list (3, 2, 1) HTML5
start	Sets the first number of the numbered list HTML5

[Global attributes](#) and [events](#) are also available for this element.

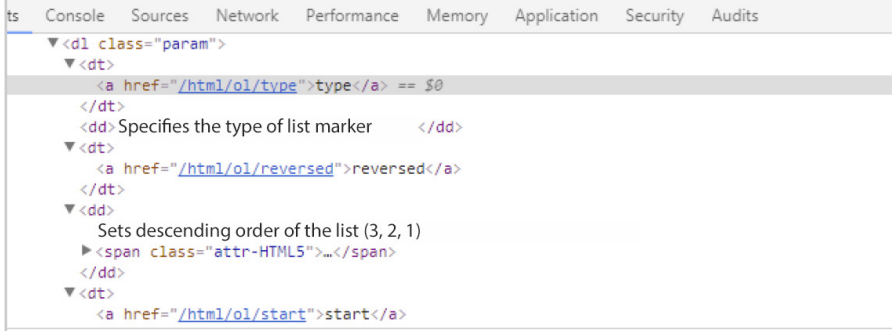


Рисунок 9

Створюється список за допомогою 3-х тегів:

- **dl** — *definition list* — основний або батьківський тег;
- **dt** — *definition termin* — вкладений тег;
- **dd** — *definition description* — вкладений тег.

Код списку має вигляд:

```
<dl>
  <dt> Term 1 </dt>
  <dd> Description 1 </dd>

  <dt> Term 2 </dt>
  <dd> Description 2 </dd>

  <dt> Term 3 </dt>
  <dd> Description 3 </dd>
</dl>
```

Для прикладу використовуємо визначення декількох термінів:

Internet

a set of networks that use a single exchange protocol (more precisely, an extensive family of hundreds of protocols) for the data transfer

Local network

connection of several computers with the help of technical and software tools that allow combining the file systems of its computers

Global network

a connection of computers, which allows the use of information physically located on other computers

Network Service Provider

the owner of a particular network with which the client has legal relations

Рисунок 10

На скріншоті видно, що порівняно з терміном в елементі `dt`, елемент `dd` має зміщення від лівого краю вікна браузера на `40px` вправо. Цей відступ визначається властивістю `margin-left` або у браузері Chrome — `webkit-margin-start`. За рахунок цього, визначення візуально відрізняється від терміну та привертає до себе увагу.

```
dd {                                user agent stylesheet
  display: block;
  -webkit-margin-start: 40px;
}
```

Рисунок 11

Список визначень краще форматувати за допомогою CSS. Для сайту, який використовує багато варіантів таких списків, варто призначити CSS-властивості для селектора тегу, не вдаючись ні до використання селекторів класів,

ні до селекторів `id`. Наприклад, список на скріншоті вище можна відформатувати так:

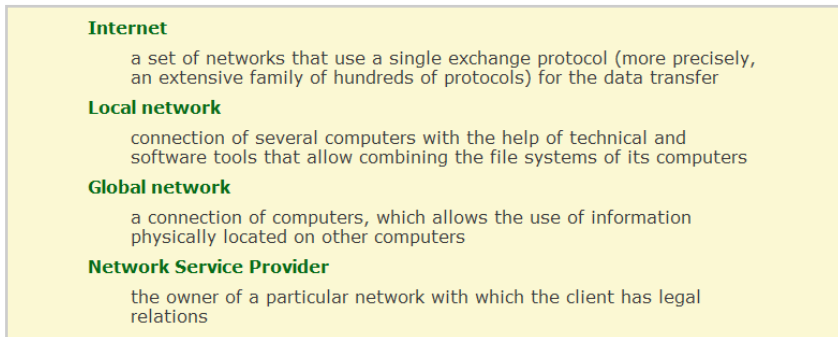


Рисунок 12

Код CSS-стилів:

```
<style>
  dt {
    font-weight: bold;
    color: #036011;
  }
  dd {
    color: #333;
    margin-top: 10px;
    margin-bottom: 10px;
  }
</style>
```

Переглянути даний приклад можна у файлі *definition-list.html*.

Посилання по темі:

1. <https://webref.ru/layout/html5-css3/list;>
2. [https://html5book.ru/html-lists/;](https://html5book.ru/html-lists/)
3. <http://htmlbook.ru/faq/theme/list.>

Багаторівневі або вкладені списки

Багаторівневим списком є список, усередині елементів `` якого розміщується список другого рівня. Тобто, другий список вкладається всередину першого, тому такі списки називають ще вкладеними (файл *inner_list.html*).

У багаторівневому списку можна використовувати як упорядковані, так і неупорядковані списки, а також списки визначень, змінюючи їх у будь-якій послідовності. Важливо, щоб вкладений список знаходився в елементі `` батьківського списку. Кількість рівнів у списках може бути будь-яким, хоча більше трьох рівнів вкладеності буває дуже рідко.

Приклад правильного коду:

```
<ol>
  <li> Squares
    <ul>
      <li> Palace Square </li>
      <li> Market Square </li>
    </ul>
  </li>
  <li> Castles
    <ul>
      <li> Royal Castle </li>
      <li> Wilanów Palace </li>
    </ul>
  </li>
  <li> Stare Miasto </li>
  <li> Museums
    <ul>
      <li> Historical Museum </li>
      <li> Polish Army Museum </li>
      <li> Cartoon Museum </li>
      <li> Marie Curie Museum </li>
    </ul>
  </li>
</ol>
```

```

        <li> Chopin Museum </li>
        <li> National Museum </li>
    </ul>
</li>
<li> Krakowskie Przedmieście </li>
</ol>

```

Варіант помилкового коду (файл *invalid-multilevel-list.html*):

```

<ol>
  <li> Squares </li>
  <ul>
    <li>Palace Square</li>
    <li>Market Square</li>
  </ul>
  <li>Castles</li>
  <ul>
    <li>Royal Castle</li>
    <li>Wilanów Palace</li>
  </ul>
  <li>Stare Miasto</li>
  <li>Museums</li>
  <ul>
    <li>Historical Museum</li>
    <li>Polish Army Museum</li>
    <li>Cartoon Museum</li>
    <li>Marie Curie Museum</li>
    <li>Chopin Museum</li>
    <li>National Museum</li>
  </ul>
  <li>Krakowskie Przedmieście</li>
</ol>

```

Помилка полягає в тому, що вкладений список вставлений не всередину елемента ``, а між двома сусідніми елементами, а це неприпустимо з погляду валідатора:

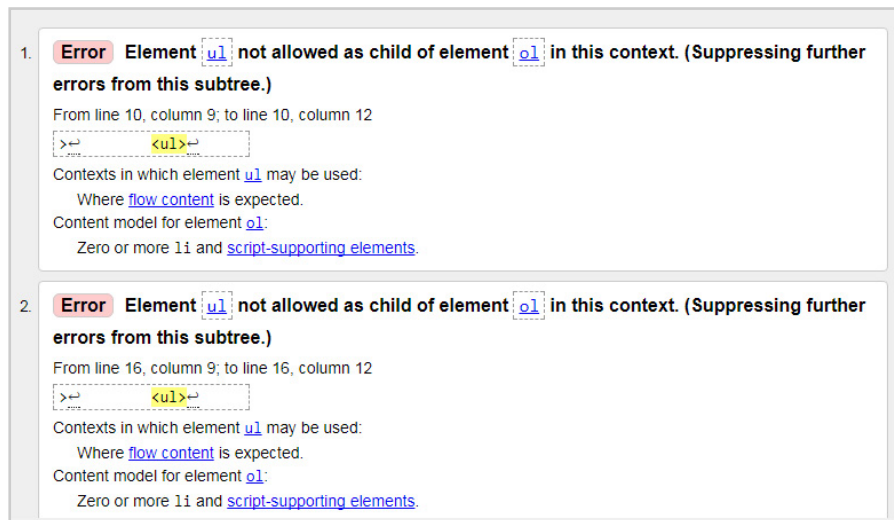


Рисунок 13

Візуально список виглядає так:



Рисунок 14

Цей список стане значно більшим, якщо його допов-
НИТИ:

Attractions of famous cities of the world

1. Attractions in Paris
 1. Eiffel Tower
 2. Notre Dame de Paris
 3. Triumphal Arch
 4. The National Residence of the Invalids
 5. Sacre Coeur
 6. Pantheon
 7. Alexander III Bridge
 8. The Louvre
 9. Conciergerie
 10. The Tower of Montparnasse
 11. Champs-Élysées
2. Attractions in Rome
 1. Appian Way
 2. Arch of Constantine
 3. Villa Borghese
 4. Borghese Gallery
 5. Spanish Steps
 6. Capitol Hill
 7. Coliseum
 8. Vatican Museum Complex
 9. Palatine
 10. Pantheon
 11. Squares
 - Piazza di Spagna
 - Piazza Navona
 - St. Peter's Square
 - Piazza del Popolo
 12. Roman forum
 13. Cathedrals
 - Santa Maria Maggiore
 - The Sistine Chapel
 - Saint Paul's Cathedral
 - Raphael's Staats
3. Attractions in Venice
 - Academy Gallery
 - Grand Canal
 - Ca'd'Oro
 - Café Florian in Venice
 - Bridge of Sighs
 - Rialto Bridge
 - Piazza San Marco
 - The Doges Palace

Рисунок 15

Файл прикладу — *multilevel-list.html*.

Зверніть увагу, що кожен вкладений список зміщений щодо свого батьківського вправо, причому при вкладенні один в одного маркованих списків змінюється стиль маркера — **disc** головного списку, **circle** першого дочірнього списку та **square** — другого дочірнього списку.

Посилання по темі:

1. [https://webref.ru/layout/html5-css3/list/nested](https://webref.ru/layout/html5-css3/list/nested;);
2. https://puzzleweb.ru/html/11_lists2.php;
3. <http://html-plus.in.ua/vlozhennye-spiski/>.

2. CSS-стилі для списків

CSS-властивості, про які йтиметься нижче, призначають зазвичай для маркованих або нумерованих списків, причому вказувати їх можна як для батьківських елементів (`ul`, `ol`), так і для вкладених (`li`).

Одну зі стильових властивостей списків ми вже розглянули. Це властивість `list-style-type`, що визначає вид маркера чи цифри:

```
list-style-type: decimal | decimal-leading-zero |
                armenian | georgian | lower-alpha |
                lower-greek | lower-latin | lower-roman |
                upper-alpha | upper-latin | upper-roman |
                disc | circle | square | none;
```

У попередніх прикладах ми розділяли її за значеннями, які були різними для нумерованих та маркованих списків. Ви можете призначити будь-яке значення для будь-якого виду списку, проте логіка HTML-розмітки тоді загубиться.

Досить часто маркери або цифри у списках «приховують», використовуючи значення `list-style-type: none`. Наприклад, так роблять при форматуванні меню.

```
Emmet abbreviations: lst, lstdlz, lstlr, lstur, lstd,
                    lstc, lsts, lstn
```

Друга властивість — це позиція цифр або маркерів. Визначається властивістю `list-style-position` із такими значеннями:

```
list-style-position: outside | inside | inherit
```

За замовчуванням всі списки `list-style-position: outside`, тому використовувати це значення ви будете рідко. Воно передбачає, що маркери (цифри) розміщуються зліва від тексту елемента списку. Значення `list-style-position: inside` розміщує маркери (цифри) всередині тексту елемента списку.

<pre>1. list-style-position: inside</pre> <ol style="list-style-type: none"> Консультативная помощь пациенту на каждом этапе реабилитационного процесса, Жесткие стандарты проверки качества медицинских и дополнительных услуг. Выделение бюджета на инвестиции в образование и повышение квалификации сотрудников центра, Надежность результатов работы специалистов — основной принцип оказания медицинской услуги. 	<pre>1. list-style-position: outside; (default)</pre> <ol style="list-style-type: none"> Консультативная помощь пациенту на каждом этапе реабилитационного процесса, Жесткие стандарты проверки качества медицинских и дополнительных услуг. Выделение бюджета на инвестиции в образование и повышение квалификации сотрудников центра, Надежность результатов работы специалистов — основной принцип оказания медицинской услуги.
<pre>• list-style-position: inside;</pre> <ul style="list-style-type: none"> стойкое и долгосрочное улучшение качества медицинских услуг разработка терапевтических протоколов с учетом расширяющихся видов патологий пациентов центра гибкость и способность быстро адаптироваться к любым клинически сложным случаям научно-исследовательская работа и внедрение инноваций разработка и внедрение систем реабилитационных манипуляций и новых технологий 	<pre>• list-style-position: outside; (default)</pre> <ul style="list-style-type: none"> стойкое и долгосрочное улучшение качества медицинских услуг разработка терапевтических протоколов с учетом расширяющихся видов патологий пациентов центра гибкость и способность быстро адаптироваться к любым клинически сложным случаям научно-исследовательская работа и внедрение инноваций разработка и внедрение систем реабилитационных манипуляций и новых технологий

Рисунок 16

Значення `inherit` передбачає, що вкладений список успадковує це значення від батьківського елемента. Оскільки вкладені елементи списків зазвичай мають такі самі властивості, як і батьківські, тому для них використовують це значення не так вже й часто.

```
Emmet abbreviations: lsp, lsp_i, lsp_o
```

Переглянути приклад можна в файлі `list-style-position.html`.

Третьою, досить популярною властивістю є `list-style-image`, яка дозволяє встановити зображення як маркер списку:

```
list-style-image: none | url(шлях до файлу.розширення);  
list-style-image: url(markers/arrow.gif);
```

або

```
list-style-image: url(http://mysite.com/images/arrow.gif);
```

У першому прикладі шлях до файлу вказується відносно розташування HTML-файлу і папки із зображеннями для маркерів (вони повинні знаходитися у спільній папці). У другому прикладі шлях вказується з урахуванням доменного імені вашого сайту. Оскільки на даному етапі сайт у вас навряд чи існує, то єдиним варіантом для аудиторних та домашніх завдань буде використання першого шляху.

Щодо текстового редактора Brackets, то в ньому є дуже зручні підказки, які дозволяють легко вказати шлях до файлу зображення-маркера, а також подивитися при наведенні на рядок шляху до зображення, як виглядає файл і якого він розміру. Важливою умовою для цих підказок є відкриття папки як проєкт Brackets, доступний з контекстного меню провідника.

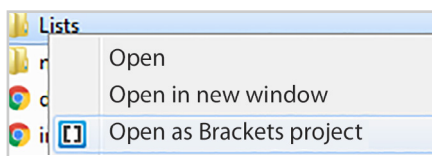


Рисунок 17

Як тільки ви написали властивість `list-style-image: url()` і залишили курсор у дужках, Brackets починає підказувати вам, показуючи папки, які є у вашому проєкті і файли у цих папках:



Рисунок 18

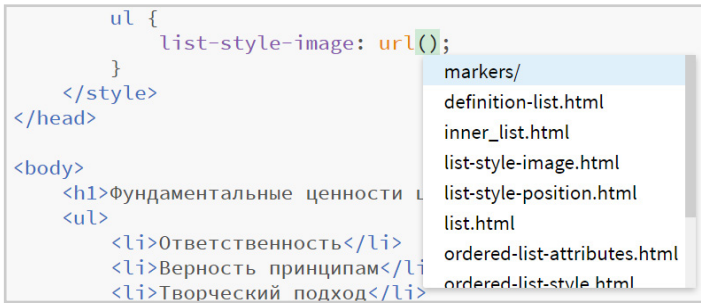


Рисунок 19

У списку будуть також всі файли, які є в поточній папці, тому будьте уважні. Свій вибір можна підтвердити, натиснувши клавішу **Enter** або зробивши клік лівою кнопкою миші.



Рисунок 20

При наведенні на текст у дужках ви побачите, як виглядає зображення та який його розмір (на скріншоті 25x25px):



Рисунок 21

Щоб змінити зображення на будь-яке інше, потрібно просто поставити курсор після слешу і натиснути **Ctrl + пробіл**:



Рисунок 22

Для заміни одного зображення іншим, потрібно просто вибрати його та натиснути **Enter** — і назву файлу буде автоматично замінено, навіть якщо не було виділено.

При виборі різних файлів для переміщення по списку можна використовувати не тільки мишу, але й клавіші зі стрілками вгору і вниз.

В результаті отримаємо наступний вид списку (рис. 25):



Рисунок 23

Файл прикладу *list-style-image.html*.

Примітка: властивість `list-style-image` має пріоритет над `list-style-type`. Тобто зображення завжди буде перекривати маркер або номер елементу списку.

У випадку, якщо потрібно забрати в одному з елементів списку призначений маркер-зображення, використовують значення `none`:

```
list-style-image: none;
```

```
Emmet abbreviations: lsi, lisi, lisin
```

Універсальна властивість `list-style` дозволяє задати через пробіл усі властивості з перелічених вище, тобто вказати стиль маркера, його позицію і зображення. Перелік йде у такій послідовності:

```
list-style: list-style-type list-style-position
            list-style-image;
```

Наприклад:

```
list-style: upper-latin inside url(markers/somepic.png);
```



Рисунок 24

Оскільки зображення перекриває маркер, має сенс обмежитися двома властивостями:

```
list-style: upper-latin inside;
```



Рисунок 25

або

```
list-style: inside url(markers/somepic.png);
```

Можна вказати в цій CSS-властивості тільки одне значення:

```
list-style: url(markers/actor-icon.png);
```

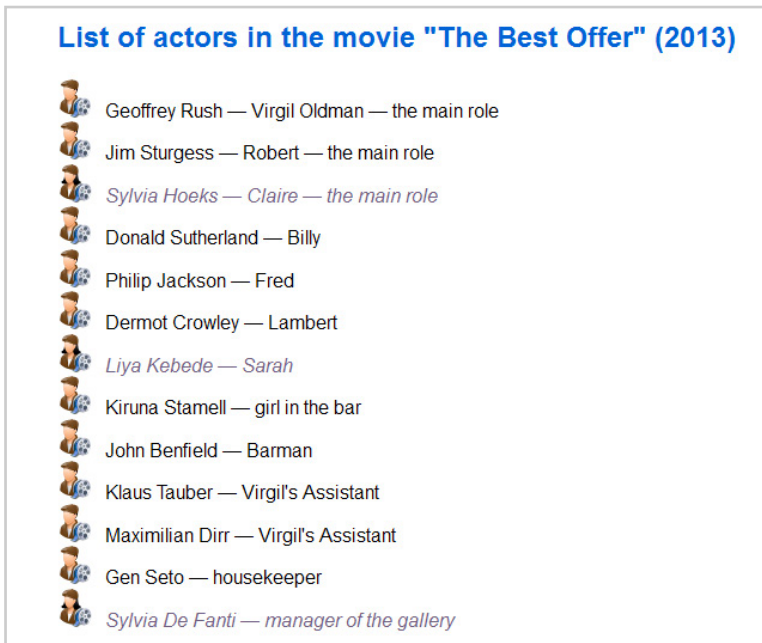


Рисунок 26

Значення за замовчуванням: **disc outside none**;
Можна також вказати значення **inherit**.

```
Emmet abbreviations: lis, lsn
```

Файл з прикладами *list-style.html*.

Використання псевдоелемента `::before` для стилів списків

Найчастіше хочеться змінити нумерацію у впорядкованому списку, поставивши для нього дужки після номеру, а для невпорядкованого списку — тире замість маркера. Для цього необхідно вимкнути стандартну нумерацію чи маркери за допомогою властивості `list-style-type: none`. Після цього використати псевдоелемент `::before` для елемента ``.

Що таке псевдоеlementи `::before` та `::after`?

Псевдоеlementи мають приставку псевдо- перед знайомим вам вже словом «елемент», тому що в початковому HTML-кодi їх немає. До речі, їх і не буде в кодi взагалі, тому що вони з'являються виключно за допомогою CSS. Існує декілька псевдоеlementів, а саме `::first-line`, `::first-letter`, `::selection`, `::before` і `::after`. Причому їм доступний синтаксис як із двома двокрапками перед найменуванням (CSS3), так і з однією (CSS2), оскільки у версії CSS3 було прийнято рішення відокремити псевдоеlementи від псевдокласів, додавши до псевдоеlementів другу двокрапку. Тому не дивуйтеся, якщо ви побачите різні їхні написання.

З приводу `::before` і `::after` можна сказати, що ці псевдоеlementи дозволяють додати певний контент (текстовий чи візуальний вміст) перед текстом елемента або в кінці елемента, тобто після вмісту елемента. Причому текст псевдоеlementу не виділятиметься, а CSS-властивості для нього можна задати такі ж, як і для звичайних елементів.

За замовчуванням `:before` та `:after` є рядковими елементами і мають такий самий шрифт, колір тексту та накреслення, що й у основного елемента.

Наприклад, псевдоеlementи `:before` і `:after` для абзацу, у якого в стилях зазначено курсивне накреслення, теж будуть виведені курсивом:

```
p { font-style: italic; }

p:before, p:after { background-color: #ccc; }
p:before { content: 'Start'; }
p:after { content: "End"; }
```

Start Lorem ipsum dolor sit amet, consectetur adipiscing elit. Illo qui quod eius, vitae accusamus eveniet! Ipsum doloribus dolore sed numquam! *End*

Start Quas quia veniam autem delectus, non repellat consequatur, amet corporis ullam odit deserunt, ab itaque ipsam assumenda aliquid. Nulla, qui. *End*

Рисунок 27

В прикладі (файл `list-before.html`) псевдоеlementи виділені сірим фоном. У коді властивості `content` вказано в лапках (подвійних або одинарних) текст, який буде виведений перед та після тексту абзацу.

Приймітка: властивість `content` є обов'язковою для псевдоеlementів `:before` та `:after`. Якщо ця властивість не буде вказана, навіть з порожніми лапками (" " або " "), то на сторінці ці псевдоеlementи не відобразяться, які б іще властивості ви для них не писали.

Посилання по темі:

1. <https://webref.ru/css/before;>
2. <https://webref.ru/css/after;>
3. [http://html-plus.in.ua/psevdo-elements-what-is-before-and-after/;](http://html-plus.in.ua/psevdo-elements-what-is-before-and-after/)
4. [http://html-plus.in.ua/stili-dlya-spiskov/.](http://html-plus.in.ua/stili-dlya-spiskov/)

Що ж до впорядкованих списків, то в них зазвичай використовують тільки псевдоелемент `:before` для зміни нумерації, вказуючи певне ім'я лічильника (наприклад, `number`) в декількох властивостях:

```
ol { list-style-type: none;
    counter-reset: number; }

ol li::before {
    content: counter(number) ' ) ';
    counter-increment: number; }
```

Результат:

Rotational motion

- 1) Angular momentum (Introduction)
- 2) Angular velocity
- 3) Centrifugal force
- 4) Centripetal force
- 5) Circular motion
- 6) Tangential velocity
- 7) Torque

Рисунок 28

Для невпорядкованих списків у властивості `content` вказують, наприклад, текст із пробілом замість маркеру:


```
ul { list-style-type: none;
      font-style: italic; }
ul li::before { content: '- ';
                  color: #04b548; }
```

Conservation of energy and momentum

- *Conservation of energy*
- *Elastic collision*
- *Energy*
- *Inelastic collision*
- *Inertia*
- *Kinetic energy*
- *Moment of inertia*
- *Momentum*
- *Potential energy*
- *Rotational energy*

Рисунок 29

Оформлення багаторівневих списків.

Вкладені селектори

Для оформлення багаторівневого списку недостатньо написати CSS-правила лише для селектора елемента (`ol`, `ul`, `li`). Найчастіше для них (і не лише) доведеться застосовувати контекстні (або вкладені) селектори. Вони мають на увазі, що CSS-властивості призначаються для тегу, вкладеного в інший тег, наприклад, селектор

```
ol ul {color: red}
```

передбачає, що червоного кольору буде лише той маркований список (`ul`), який вкладено в нумерований список (`ol`).

Кількість селекторів може бути будь-якою, але слід розуміти, що CSS-правила призначаються для самого ос-

таннього селектору, але з урахуванням його вкладеності в усі попередні. Тобто, читається такий селектор справа наліво. Наприклад, код

```
.season ul li { font-style: italic; }
```

говорить про те, що курсивне накреслення буде тільки у елементів `li` маркованого списку `ul`, що знаходиться в елементі з класом `season`.

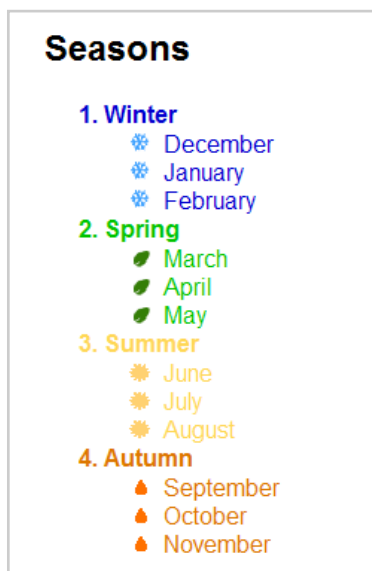


Рисунок 30

Наприклад, треба оформити список з переліком пір року так, як вказано на зображенні.

Якщо проаналізувати його зовнішній вигляд, то стане зрозуміло, що це нумерований список з 2-ма рівнями вкладеності, при чому вкладений список може бути як нумерованим, так і маркованим, тобто, цифри чи марке-

ри все рівно будуть перекриті зображеннями. Оскільки нумерація не буде видно, має сенс зробити вкладений список маркованим. Тому початкова HTML-розмітка буде такою:

```
<h2> Seasons </h2>
<ol>
  <li> Winter
    <ul>
      <li> December </li>
      <li> January </li>
      <li> February </li>
    </ul>
  </li>
  <li> Spring
    <ul>
      <li> March </li>
      <li> April </li>
      <li> May </li>
    </ul>
  </li>
  <li> Summer
    <ul>
      <li> June </li>
      <li> July </li>
      <li> August </li>
    </ul>
  </li>
  <li> Autumn
    <ul>
      <li> September </li>
      <li> October </li>
      <li> November </li>
    </ul>
  </li>
</ol>
```

Seasons

1. Winter
 - o December
 - o January
 - o February
2. Spring
 - o March
 - o April
 - o May
3. Summer
 - o June
 - o July
 - o August
4. Autumn
 - o September
 - o October
 - o November

Рисунок 31

Seasons

1. Winter
 - o December
 - o January
 - o February
2. Spring
 - o March
 - o April
 - o May
3. Summer
 - o June
 - o July
 - o August
4. Autumn
 - o September
 - o October
 - o November

Рисунок 32

Зовнішній вигляд списку представлений на скріншоті.

Тепер потрібно зробити так, щоб шрифт батьківського списку `` був жирним, а дочірнього списку `` — звичайним або нормальним. Оскільки елементи `` успадковують властивості батьківського елемента, призначимо для селектору `ol` наступну властивість:

```
ol { font-weight: 600; }
```

І отримаємо в результаті список, в якому всі елементи отримали жирне накреслення.



Рисунок 33

Щоб вкладений список «повернув» звичайне накреслення, необхідно вказати наступне CSS-правило:

```
ul { font-weight: 200; }
```

Тепер список має наступний вигляд: елементи батьківського списку — з жирним накресленням, елементи вкладеного списку — зі звичайним або нормальним накресленням.

Наступним кроком буде призначення кольору для різних блоків тексту: синього — для місяців зими, зеленого — для весняних місяців і т.д.

Найпростіше це зробити, задавши клас або `id` кожному з елементів `li` батьківського списку. Оскільки значення кольорів використовуватимемо індивідуально для кожного блоку місяців, використаємо атрибут `id`:

```
<ol>
  <li id = "winter"> Winter
    <ul>
      <li> December </li>
      <li> January </li>
      <li> February </li>
    </ul>
  </li>
  <li id = "spring"> Spring
    <ul>
      <li> March </li>
      <li> April </li>
      <li> May </li>
    </ul>
  </li>
  <li id = "summer"> Summer
    <ul>
      <li> June </li>
      <li> July </li>
      <li> August </li>
    </ul>
  </li>
```

```

<li id = "autumn"> Autumn
  <ul>
    <li> September </li>
    <li> October </li>
    <li> November </li>
  </ul>
</li>
</ol>

```



Рисунок 34

Тепер можна призначити кольори для кожного селектору **id**:

```

#winter { color: rgb(0, 0, 200); }
#spring { color: rgb(0, 190, 0); }
#summer { color: #ffd24c; }
#autumn { color: #d96c00; }

```

Зовнішній вигляд блоків змінився відповідно до призначених кольорів.

Останньою є задача з визначення маркерів у вигляді різних зображень. Можна було б призначити кожному з 3-х елементів `` атрибут клас і задати йому властивість `list-style-image`, але простіше використати тут контекстний селектор такого виду:

```
#winter li { list-style-image: url(markers/snow.png); }
#spring li { list-style-image: url(markers/leaf.png); }
#summer ul { list-style-image: url(markers/sun.png); }
#autumn ul { list-style-image: url(markers/rain.png); }
```



Рисунок 35

В результаті отримаємо необхідний для нас вид списку.

Зверніть увагу, що у перших двох селекторах ми призначили маркер-зображення елемента `li`, вкладеного в певний селектор `id`, а третьому і четвертому — для елемента `ul`. При цьому різниця візуально непомітна. Сталося це з тієї причини, що властивості, призначені для всього списку (`ul`)

у прикладі), успадковуються його елементами **li**, а властивості елементів **li** застосовуються безпосередньо до них.

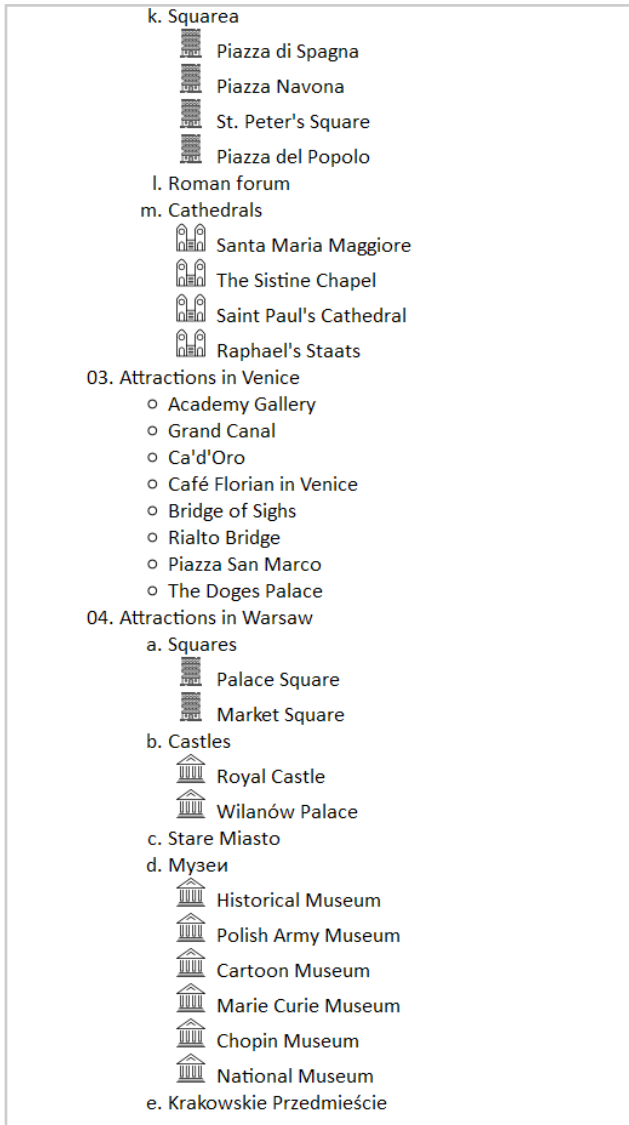


Рисунок 36

Файл прикладу *seasons.html*.

В файлі *multilevel-list-style.html* ви знайдете CSS-формування для багаторівневого списку, розглянутого раніше.

Посилання по темі:

1. <https://webref.ru/css/selector/descendant>;
2. <http://html-plus.in.ua/vlozhennye-spiski/>.

3. CSS-властивості для керування відступами і полями

Відступи між блоками тексту дуже важливі під час верстки. Причому немає значення, мається на увазі верстка книги чи журналу чи web-сторінки. Неможливо читати текст, який не має відступів, оскільки у цьому випадку текст сприймається суцільним блоком, який просто лякає своїм «насиченим» виглядом. Тому в CSS передбачені два види відступів — зовнішні, які називаються **margin**, і внутрішні — **padding**.

Слід зазначити, будь-які види відступів ми розглядатимемо для блокових елементів, оскільки рядкові елементи відображають їх дещо інакше. Ця різниця буде розглянута більш детально в уроці «Блочна модель елементів».

Внутрішні та зовнішні відступи. Загальна інформація

Властивість **padding** визначає відстань від вмісту (контенту) блокового елемента до його межі, а **margin** — відстань між межами елементів, розташованих поруч.

Зовнішні відступи прозорі, а внутрішні можна заливати фоновим кольором. Зовнішні відступи мають ефект схлопування, тобто поєднання значень поруч розміщених елементів. Зовнішні відступи можуть мати від’ємні значення, а внутрішні — лише додатні. Внутрішні та

зовнішні відступи можна записувати або в скороченій формі, або для кожної із сторін окремо.

Відступи можна призначати з 4-х сторін елемента: верхньої (англ. назва — *top*), правої (англ. назва — *right*), нижньої (англ. назва — *bottom*) та лівої (англ. назва — *left*):

- **padding-top/margin-top** — верхній відступ (внутрішній/зовнішній);
- **padding-right /margin-right** — відступ праворуч (внутрішній/зовнішній);
- **padding-bottom /margin-bottom** — нижній відступ (внутрішній/зовнішній);
- **padding-left /margin-left** — відступ зліва (внутрішній/зовнішній);
- **padding/margin** — відступ (внутрішній/зовнішній; скорочена форма).

Зовнішні відступи — margin

Зовнішні відступи можна встановити для всіх сторін елемента, наприклад:

```
margin: 10px;
```

Можна вказати відступи для верхнього та нижнього краю елемента (перше значення) та через пробіл — для лівої та правої сторони елемента (друге значення):

```
margin: 10px 2em;
```

Три значення використовують для:

- Верхньої сторони,
- Лівої та правої сторін,

- Нижньої сторони елемента

```
margin: 10px 1.5% 30px;
```

Чотири значення визначають значення зовнішніх відступів в наступній послідовності: зверху — праворуч — знизу — ліворуч.

```
margin: 15px 10% 30pt 0;
```

Порядок вказування величин відступів починається з лівого верхнього кута та йде за годинниковою стрілкою:

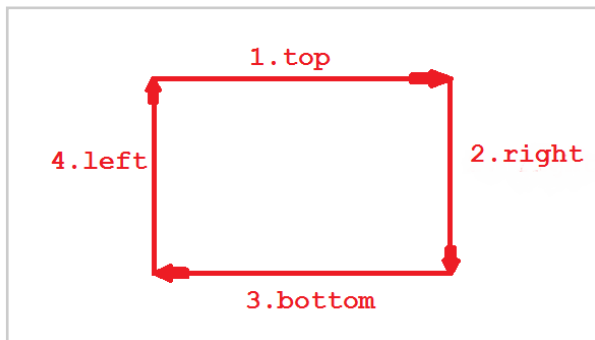


Рисунок 37

Задають відступи у будь-яких одиницях вимірів (**px**, **pt**, **em**, **rem**, **% in.**). Можна вказувати від'ємні величини, **auto** (браузер розраховує величину автоматично) або **0** (відступу немає). У останньому випадку одиниці виміру не вказуються, оскільки **0** у **%**, у **px** і **em** — однаково дорівнює **0**.

Можна також призначити лише один із відступів:

```
margin-right: 20px;  
margin-left: 2%;
```

```
margin-top: 10pt;
margin-bottom: 2em;
```

Ще можна призначити загальний відступ для всіх сторін, а потім скасувати або перевизначити його для однієї з них:

```
margin: 20px;
margin-right: 0;
```

Варіанти призначення властивості **margin** ви знайдете у файлі *margin.html*, де відступи відображаються бежевим кольором, а блок, до якого вони застосовуються — білим.

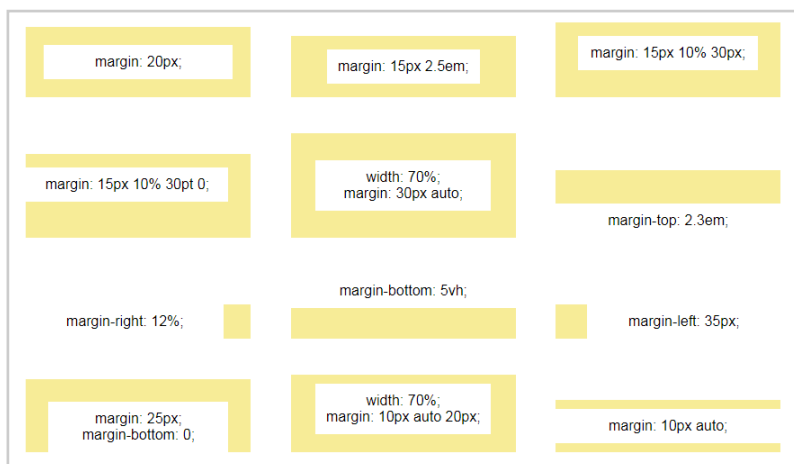


Рисунок 38

Цікавим є значення **auto**. Використовуючи його разом із властивістю **width**, ви надаєте браузеру розрахувати величину відступу самостійно, виходячи з розміру, вказаного для ширини селектору.

Наприклад, при вказівці для **body** таких властивостей:

```
body { width: 1000px;
      margin: auto;}
```

при ширині браузеру в **1340px** браузер залишить **1000px** на відображення контенту, а **340px**, що залишилися, розділить і сформує відступи в **170px** праворуч і ліворуч від контенту, центруючи його. Так само станеться, якщо ширина вказана у **%**. Різниця полягає тільки в тому, що спочатку браузер розрахує ширину елемента **px**, виходячи з ширини браузеру або ширини батьківського елемента, а потім розрахує величину відступів.

Зазвичай, таким чином вказують відступи не для **body**, а для якогось елемента-«обгортки», який часто має ім'я класу або **id** **wrap**, **wrapper**, **container**, **content** тощо. Для **body margin** встановлюють рівними **0**, оскільки за замовчуванням **body** має зовнішні відступи у **8px**.

```
body { margin: 0}
.wrapper {width: 80%; margin: auto;}
```

Ще однією особливістю властивості **margin** є «схлопування» вертикальних відступів у елементів, що йдуть один за одним. Наприклад, якщо задати такі CSS правила для елементів з класом **block**:

```
.block { margin: 40px 0; }
```

можна очікувати, що між блоками, які йдуть один за одним, відступи будуть **80px**. Однак насправді відступи

накладаються, і загальний відступ буде в 40px (файл *margin-collapse.html*).

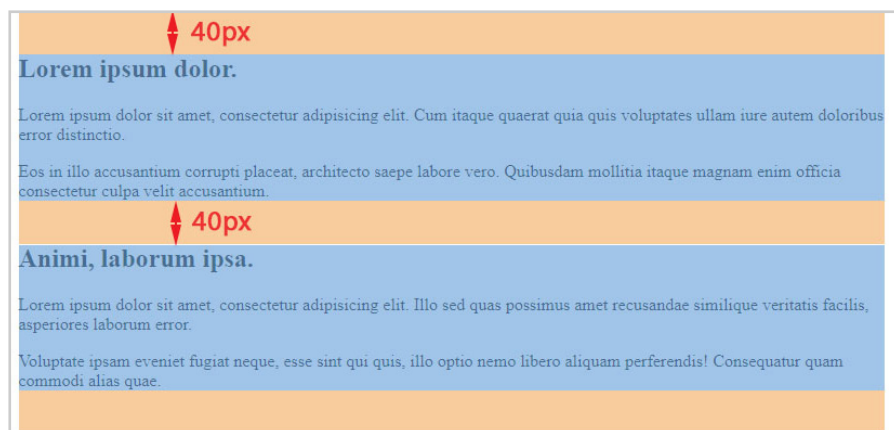


Рисунок 39

Схлопування спрацьовує лише для вертикальних відступів, але не застосовується для горизонтальних. Крім того, схлопування не спрацьовує:

- для тегу `<html>`;
- для рядкових елементів;
- для елементів, які мають властивість `float: left` або `right`;
- у абсолютно позиціонованих елементів;
- для елементів, в яких на стороні схлопування визначена властивість `padding`;
- для елементів, в яких на стороні схлопування визначена властивість `border`.

Emmet abbreviation: m20, m2em, mb2%, mt, ml, mr:7, m:a, m20px10

Внутрішні відступи – padding

Внутрішні відступи поширюються від контенту блокового елемента до його межі (CSS-властивість **border**), якщо вона задана. Використовуючи заливку елемента фоновим кольором **padding** зазвичай також має цей колір.

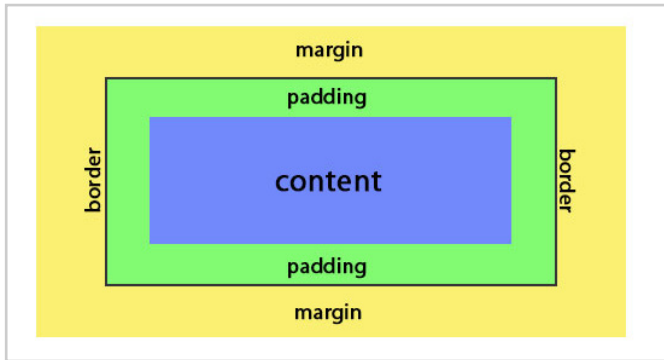


Рисунок 40

Внутрішні відступи можуть бути невід’ємними і вони не схлопуються як зовнішні. Подібність цих двох відступів у тому, що **padding** можна задавати в **px**, **pt**, **em**, **rem**, **%**, тощо, а також робити це по 4-м сторонам блоку так само, як і для **margin**. Також можна використовувати значення **inherit**, яке показує, що воно успадковується у батьківському елементі.

Одне значення для всіх сторін:

```
padding: 20px;
```

Відступи з горизонтальних (зверху і знизу) і вертикальних (праворуч і ліворуч) сторін:

```
padding: 2em 10px;
```

Відступ зверху, відступ для лівої і правої сторони, та відступ знизу:

```
padding: 2vw 3% 10px;
```

Відступ зверху, праворуч, знизу, ліворуч:

```
padding: 1rem 5% 0 20px;
```

Окремий відступ по кожній із сторін:

```
padding-top: 15px;
padding-bottom: 1rem;
padding-left: 5%
padding-right: 14pt;
```

Загальне значення відступів з його перевизначенням з однієї (лівої) сторони:

```
padding: 20px; padding-left: 3px;
```

Візуально блоки зі всіма цими значеннями будуть виглядати так:

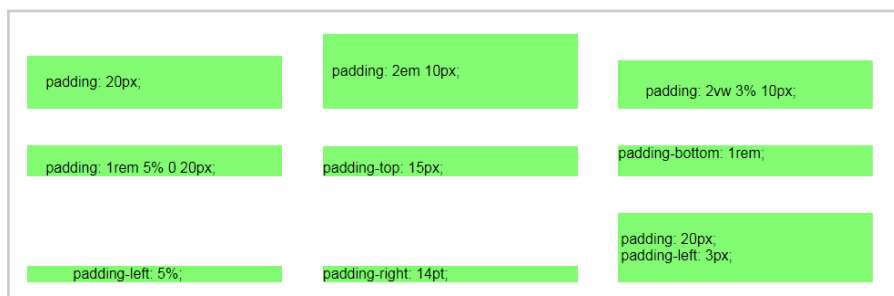


Рисунок 41

Усі значення ви знайдете у файлі *padding.html*.

Ще одна відмінність внутрішніх відступів від зовнішніх полягає у тому, що властивість `padding: auto` не застосовується, оскільки не існує значення `auto` для `padding`.

Emmet abbreviation: `p20, p2em, pb2%, pt, pl, pr:7, p20px6`

Примітка: при використанні величин відступів у % слід розуміти, що відсотки розраховуються від ширини батьківського елемента, а не поточного. Тому відступ у 5% для елемента шириною 200px, який поміщений усередину блоку з шириною 1000px, становитиме 50px ($1000px * 5\% / 100\% = 50px$), а не 10px ($200px * 5\% / 100\% = 10px$).

Значення відступів за замовчуванням

Для того, щоб різні HTML-елементи на сторінці мали правильний вигляд навіть без CSS-форматування, вони мають містити ряд CSS-властивостей, наданих за замовчуванням. Вони підтягуються з таблиці стилів браузера (user agent stylesheet), що створюється розробниками з урахуванням рекомендацій корпорації W3C. У тому числі значення за замовчуванням є в таких властивостях, як `margin` і `padding`.

Розглянуті сьогодні елементи списків мають такі відступи за замовчуванням:

```
ul, ol {margin-top: 1em;
        margin-bottom: 1em;
        padding-left: 40px; }
```

Важливо зазначити, що вкладені елементи **ol** або **ul** у багаторівневих списках НЕ мають відступів зверху та знизу. У списках визначень значення відступів за замовчуванням такі:

```
dl { margin-top: 1em;
    margin-bottom: 1em;
}
dd { margin-left: 40px; }
```

Для заголовків також використовуються зовнішні відступи:

```
h1 { margin-top: 0.67em;
    margin-bottom: 0.67em;
}
h2 { margin-top: 0.83em;
    margin-bottom: 0.83em;
}
h3 { margin-top: 1em;
    margin-bottom: 1em;
}
h4 { margin-top: 1.33em;
    margin-bottom: 1.33em;
}
h5 { margin-top: 1.67em;
    margin-bottom: 1.67em;
}
h6 { margin-top: 2.33em;
    margin-bottom: 2.33em;
}
```

Тут спостерігається така тенденція — чим більша цифра заголовка, тим менший розмір шрифту, але більший верхній і нижній зовнішній відступ.

Відступи у заголовків часто доводиться змінювати, тому що у psd-макетах дизайнери їх або збільшують, або зменшують.

Абзаци за замовчуванням також мають зовнішні, горизонтальні відступи такої ж величини, як у [h3](#):

```
p { margin-top: 1em;
    margin-bottom: 1em;
}
```

Для блокової цитати відступи за замовчуванням надані з усіх боків, але з різними значеннями:

```
blockquote {
    margin-top: 1em;
    margin-bottom: 1em;
    margin-left: 40px;
    margin-right: 40px;
}
```

Для елемента [body](#), як вже зазначалося раніше, зовнішні відступи за замовчуванням становлять [8px](#):

```
body { margin: 8px;}
```

Такі елементи як [html](#), [div](#), [section](#), [article](#), [aside](#), [main](#) за замовчуванням НЕ мають ні зовнішніх, ні внутрішніх відступів, тому, за необхідності, ви можете їх надати самостійно.

Щоб прибрати відступи, необхідно надати значення [0](#):

```
body { margin: 0;}
```

Якщо потрібно прибрати відступи за замовчуванням у всіх елементів, використовують «найпростіше скидання», встановлюючи правила для універсального селектору:

```
* { margin: 0; padding: 0; }
```



Рисунок 42

Зверніть увагу, що при такому підході маркери списків «вийдуть» за межі браузера, якщо в **body** або в іншому батьківському елементі не буде додано **margin-left**.

Також, для приведення стилів усіх елементів сторінки до одноманітного вигляду слугує файл *reset.css*, запропонований CSS-гуру — [Еріком Меєром](#). Це відбувається за рахунок видалення відступів та призначення

одного розміру шрифту. Зараз найчастіше використовують інший підхід — нормалізацію вигляду елементів за рахунок використання файлу [normalize.css](https://necolas.github.io/normalize.css/), в якому впроваджено одноманітні стилі для всіх елементів, CSS-форматування яких відрізняється залежно від браузера та його версії.

Дізнатися, які CSS-властивості у певного елемента сторінки, ви можете за допомогою інструментів розробника, передбачених нині у будь-якому браузері, але про це ми докладно поговоримо вже у наступному уроці.

Посилання по темі:

1. <https://meyerweb.com/eric/tools/css/reset/>;
2. <https://webref.ru/course/css-basics/css-reset>;
3. <https://necolas.github.io/normalize.css/>;
4. <https://htmlacademy.ru/blog/64-about-normalize-css>.

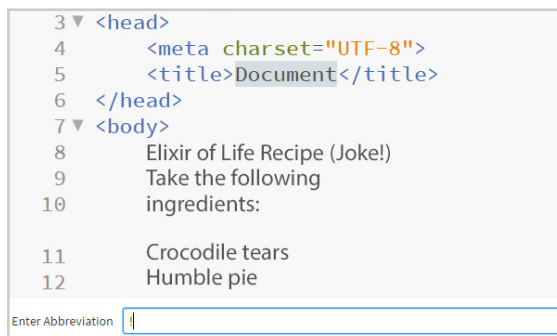
4. Домашнє завдання

Вдома вам доведеться створити та відформатувати за допомогою CSS-правил декілька багаторівневих списків. В *папці* з домашнім завданням ви знайдете зображення зі списками і тексти до них в txt-файлах, та зображення-маркери в папці *markers* (матеріали до домашнього завдання прикріплені до цього PDF-файлу).

Для центрування контенту використовуйте поєднання правил `width` та `margin: auto`.

Аби легше було поміщати текст з txt-файлу в HTML-теги, має сенс використовувати «Emmet-обгортку», а саме:

- Виділити потрібний текст;
- Натиснути `Ctrl+Shift+A`;
- Ввести необхідну аббревіатуру;
- Натиснути клавішу `Enter`.



```
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6 </head>
7 <body>
8   Elixir of Life Recipe (Joke!)
9   Take the following
10  ingredients:
11
12  Crocodile tears
   Humble pie
```

Enter Abbreviation

Рисунок 43

Наприклад, щоб додати теги структури за допомогою такої обгортки Emmet, потрібно виділити весь текст

(**Ctrl+A**), натиснути **Ctrl+Shift+A**, а потім ввести знак оклику (!) (рис. 43).

Щоб помістити текст списку всередину тегів списку, слід його виділити та використати таку аббревіатуру:

```
ol>li*
```

У цьому випадку теги **ol** з'являться на початку і в кінці виділеного тексту, а теги **** «обгорнуть» усі рядки, відокремлені один від одного в txt-файлі клавішею **Enter**.

Зовнішній вигляд завдань

Звертайте, будь ласка, увагу на сімейство шрифту та його накреслення (рис. 44).



Рисунок 44

Для стильового оформлення використовуйте контекстні селектори, селектори класів та **id** (рис. 45).

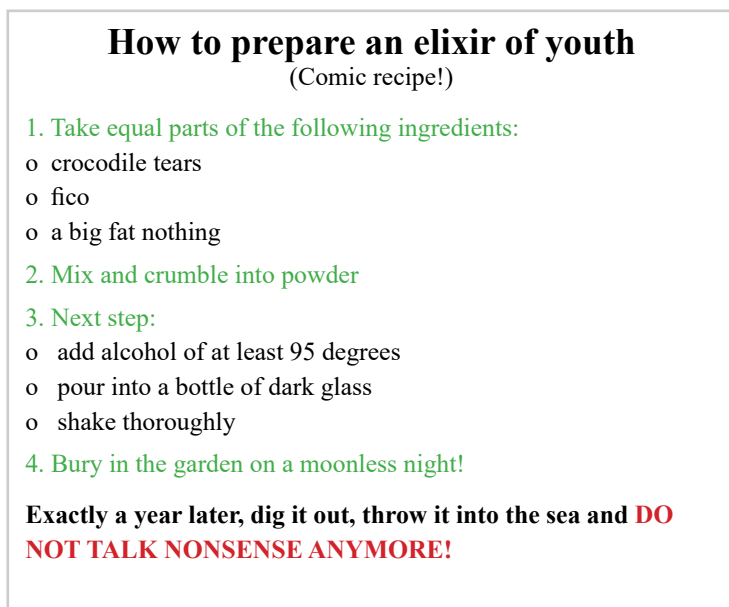


Рисунок 45

Зверніть увагу, що заголовок «Мапа сайту» зміщений праворуч і має звичайне накреслення. Використовуйте відступи (рис. 46).

Site Map

- News
- Contacts
- About company
 - ➔ Our mission
 1. Advantages of working with us
 2. Methodology
 - ➔ About us
 - ➔ Feedback
 - ➔ Team
- Language training
 - ➔ German
 - ➔ English
 1. English by specialization
 2. Preparation for exams
 3. Business English
 4. General English
 5. Conversation Club
 6. Special intensive courses
 7. New competition "Master Mind"
- Services
 - ➔ English for children
 - ➔ Trainings
 - ➔ Translation Agency
 - ➔ Study abroad
- Loyalty programs
- Our newsletter
- Practical tips
- FAQ
- Site Map

Рисунок 46



Урок 3

Списки. CSS відступи та поля

© Олена Слуцька.

© STEP IT Academy, www.itstep.org

Усі права на фото-, аудіо- і відеотвори, що охороняються авторським правом і фрагменти яких використані в матеріалі, належать їх законним власникам. Фрагменти творів використовуються в ілюстративних цілях в обсязі, виправданому поставленим завданням, у рамках учбового процесу і в учбових цілях, відповідно до законодавства про вільне використання твору без згоди його автора (або іншої особи, яка має авторське право на цей твір). Обсяг і спосіб цитованих творів відповідає прийнятим нормам, не завдає збитку нормальному використанню об'єктів авторського права і не обмежує законні інтереси автора і правовласників. Цитовані фрагменти творів на момент використання не можуть бути замінені альтернативними аналогами, що не охороняються авторським правом, і відповідають критеріям добросовісного використання і чесного використання.

Усі права захищені. Повне або часткове копіювання матеріалів заборонене. Узгодження використання творів або їх фрагментів здійснюється з авторами і правовласниками. Погоджене використання матеріалів можливе тільки якщо вказано джерело.

Відповідальність за несанкціоноване копіювання і комерційне використання матеріалів визначається чинним законодавством.