

# Internet- технології

Каскадні таблиці стилів.  
Позиціювання елементів

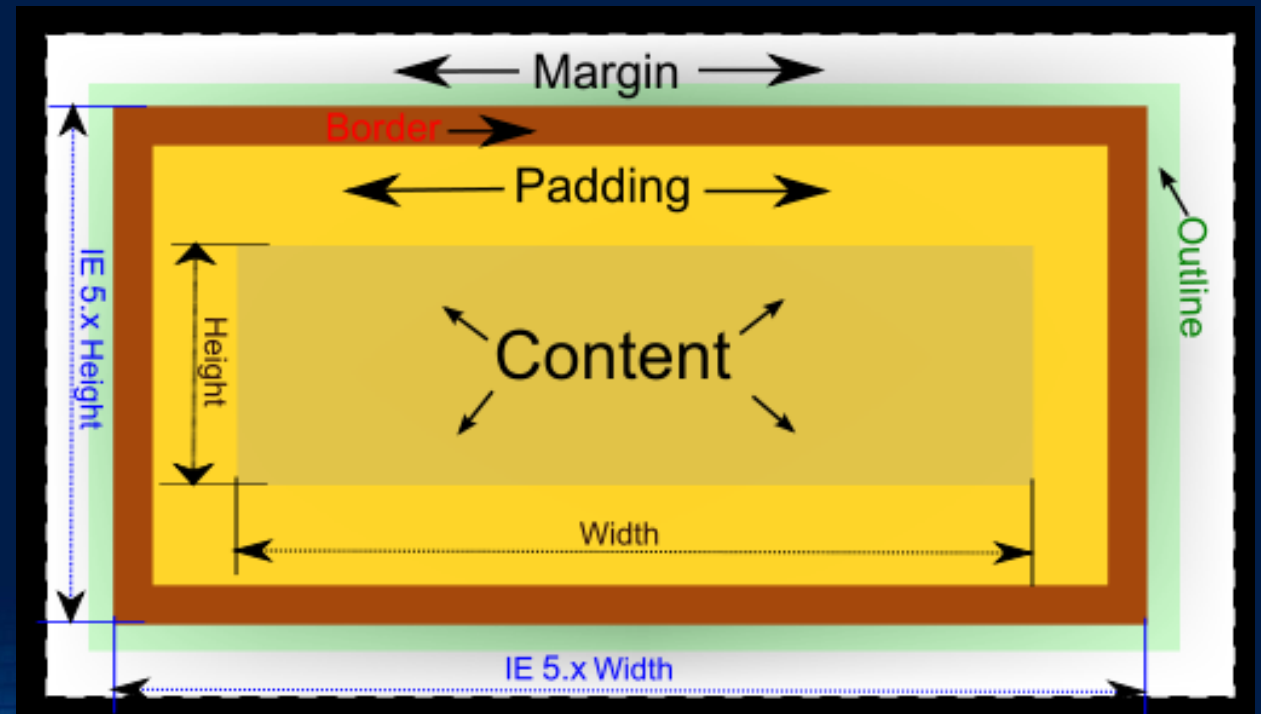
# Рекомендовані джерела

- Малювання <https://habrahabr.ru/company/paysto/blog/251933/>
- Як працює position: sticky <http://shpargalkablog.ru/2017/04/position-sticky.html>
- Flexbox <http://html5.by/blog/flexbox/>, <https://habr.com/ru/post/425977/>, <https://css-live.ru/articles/vizualnoe-rukovodstvo-po-svoystvam-flexbox-iz-css3.html>, ігровий ознайомчий курс <http://flexboxfroggy.com/>

# Блокова модель

Визначає формування розмірів окремих елементів на сторінці та їх взаємодію між собою. Характеризується такими властивостями:

- border та outline – межа та зовнішня межа; (друга властивість не бере участі у формуванні розмірів блоку).
- height та width – висота та ширина елемента.
- content – вміст блоку;
- padding – внутрішній відступ;
- margin – зовнішній відступ;



# Позиціювання елементів

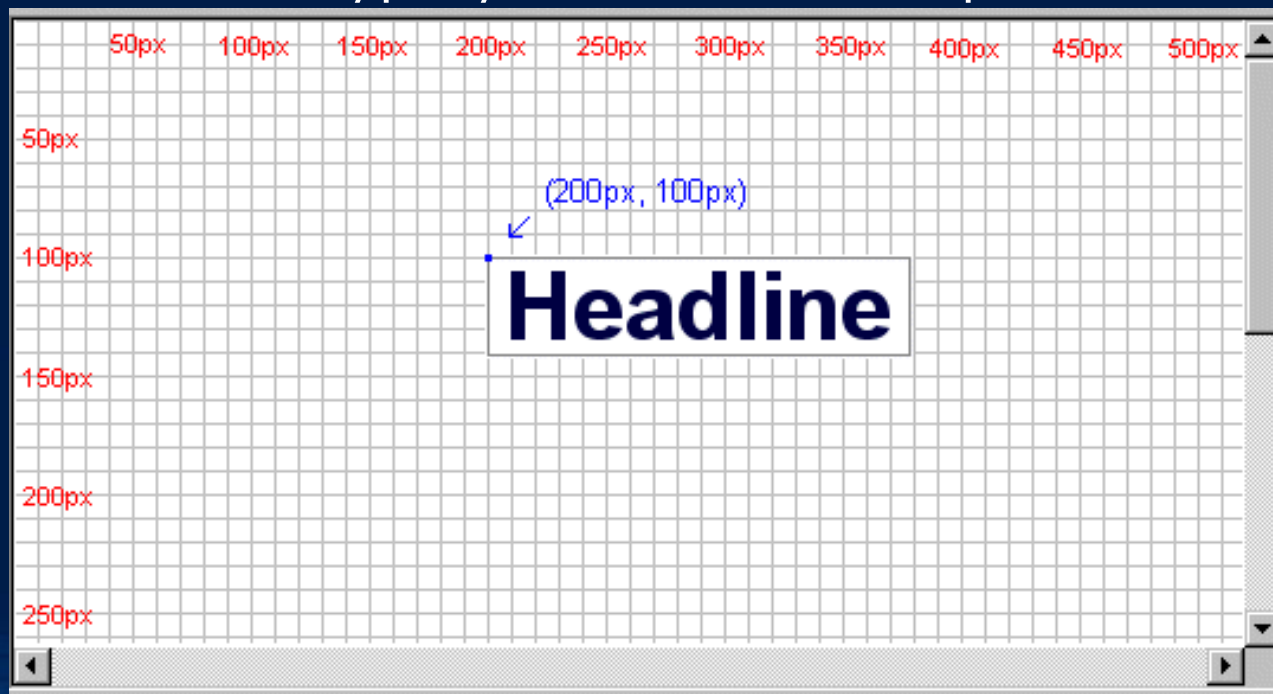
За допомогою CSS-позиціонування можна розмістити елемент точно в потрібному місці сторінки. Використовуються такі властивості:

**position** – встановлює спосіб позиціонування елемента щодо вікна браузера чи інших об'єктів на веб-сторінці.

**bottom** (або **left**, **right**, **top**) – встановлюють положення нижнього (лівого, правого, верхнього) краю вмісту елемента без урахування товщини рамок та відступів.

**z-index** – задає положення елементів по z-осі.

```
h1 {  
  position: absolute;  
  top: 100px;  
  left: 200px;  
}
```





# Позиціювання елементів

Властивість **position** визначає спосіб позиціонування елемента щодо вікна браузера, меж документа чи інших об'єктів на веб-сторінці. Можливі такі значення:

**absolute** – елемент абсолютно позиціонований, при цьому інші елементи відображаються на веб-сторінці ніби абсолютно позиціонованого елемента і взагалі немає. Положення елемента задається властивостями left, top, right і bottom, також на положення впливає значення властивості position батьківського елемента.

**fixed** – на відміну від **absolute**, прив'язується до вказаного властивостями left, top, right і bottom положенню на екрані і не змінює свого положення при прокручуванні веб-сторінки.

**relative** – положення елемента встановлюється щодо вихідного.

**static** – елементи відображаються "як завжди". Використання властивостей left, top, right та bottom не призводить до будь-яких результатів.

**inherit** – успадковує значення елемента-батька.

# Приклад абсолютного та відносного позиціонування

```
<head>
<style>
.absolute_box {
  position: absolute;
  top: 50px;
  left: 50px;

  height: 50px;
  width: 100px;
  background: red;
}
```

```
.relative_box {
  position: relative;
  top: 0px;
  left: 100px;

  height: 50px;
  width: 100px;
  background: green;
}
</style>
</head>
```

# Приклад абсолютного та відносного позиціонування

```
<body>  
  <div style="position: absolute; top: 50px; left:  
20px; border: 4px double black; height:50%;  
width:50%;"> Батьківський блок  
    <div class="absolute_box"> Absolute</div>  
    <div class="relative_box"> Relative</div>  
    <div class="relative_box" style="top: 50px;">  
Relative2</div>  
  </div>  
</body>
```



# Малювання. Приклад

```
#rectangle {  
  width: 200px;  
  height: 100px;  
  background: red;  
}
```

```
#triangle-up {  
  width: 0;  
  height: 0;  
  border-left: 50px solid transparent;  
  border-right: 50px solid transparent;  
  border-bottom: 100px solid red;  
}
```

```
#triangle-left {  
  width: 0;  
  height: 0;  
  border-top: 50px solid transparent;  
  border-right: 100px solid red;  
  border-bottom: 50px solid transparent;  
}
```

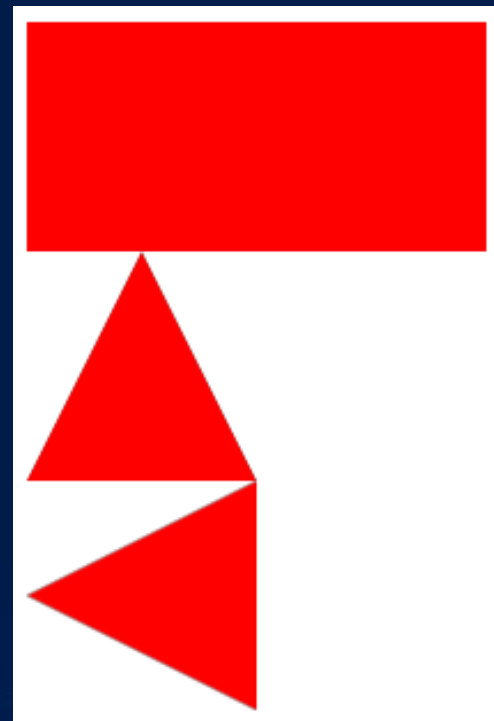
```
<body>
```

```
<div id="rectangle"></div>
```

```
<div id="triangle-up"></div>
```

```
<div id="triangle-left"></div>
```

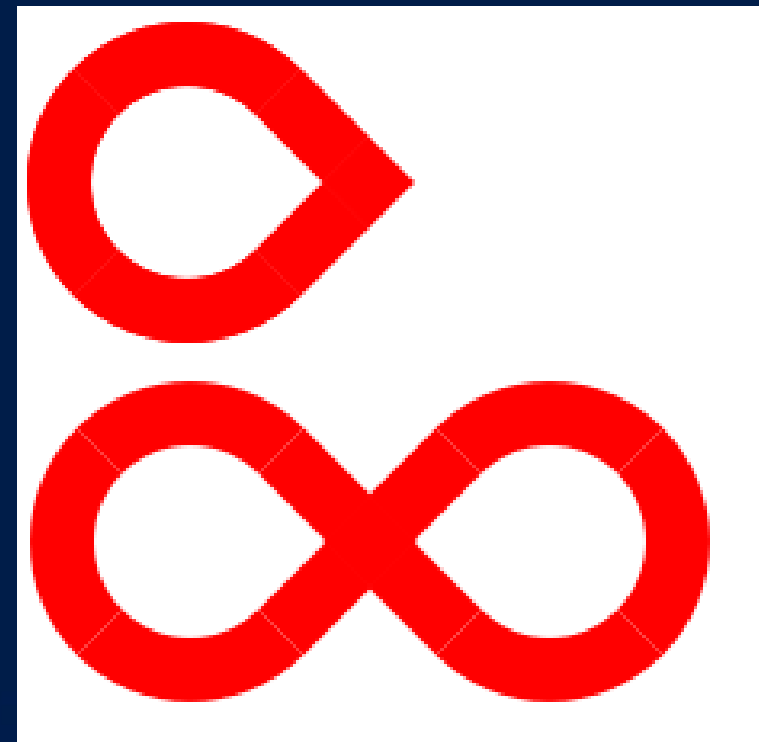
```
</body>
```





# Малювання. Приклад

```
#infinity {  
  position: relative; width: 212px; height: 100px;  
}  
#infinity:before, #infinity:after {  
  content: ""; position: absolute;  
  top: 0; left: 0; width: 60px; height: 60px;  
  border: 20px solid red;  
  -moz-border-radius: 50px 50px 0 50px;  
  border-radius: 50px 50px 0 50px;  
  transform: rotate(-45deg);  
}  
#infinity:after {  
  left: auto; right: 0;  
  -moz-border-radius: 50px 50px 50px 0;  
  border-radius: 50px 50px 50px 0;  
  transform: rotate(45deg);  
}
```



# Позиціювання елементів. Властивість *float*

Властивість **float** визначає, по якій стороні вирівнюватиметься елемент, при цьому інші елементи будуть обтікати його з інших сторін.

Використовуються такі значення:

**left** – вирівнює елемент по лівому краю, а всі інші елементи, (наприклад, текст), обтікають його праворуч.

**right** – вирівнює елемент з правого краю, а всі інші елементи обтікають його з лівого боку.

**none** – обтікання елемента не задається.

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

`clear: none | left | right | both`

# Властивість *float*. Приклад

```
.layer1 {  
  float: left;  
  background: #fdo;  
  border: 1px solid black;  
  padding: 10px;  
  margin-right: 20px;  
  width: 40%;  
}
```

Lorem ipsum  
dolor sit amet,  
consectetuer  
adipiscing elit,  
sed diam  
nonummy nibh  
euismod  
tincidunt ut  
laccet dolore  
magna aliquam  
erat volutpat.

Duis autem dolor  
in hendrerit in  
vulputate velit esse  
molestie  
consequat, vel  
illum dolore eu  
feugiat nulla  
facilisis at vero  
eros et accumsan  
et iusto odio  
dignissim qui  
blandit praesent  
luptatum zzril

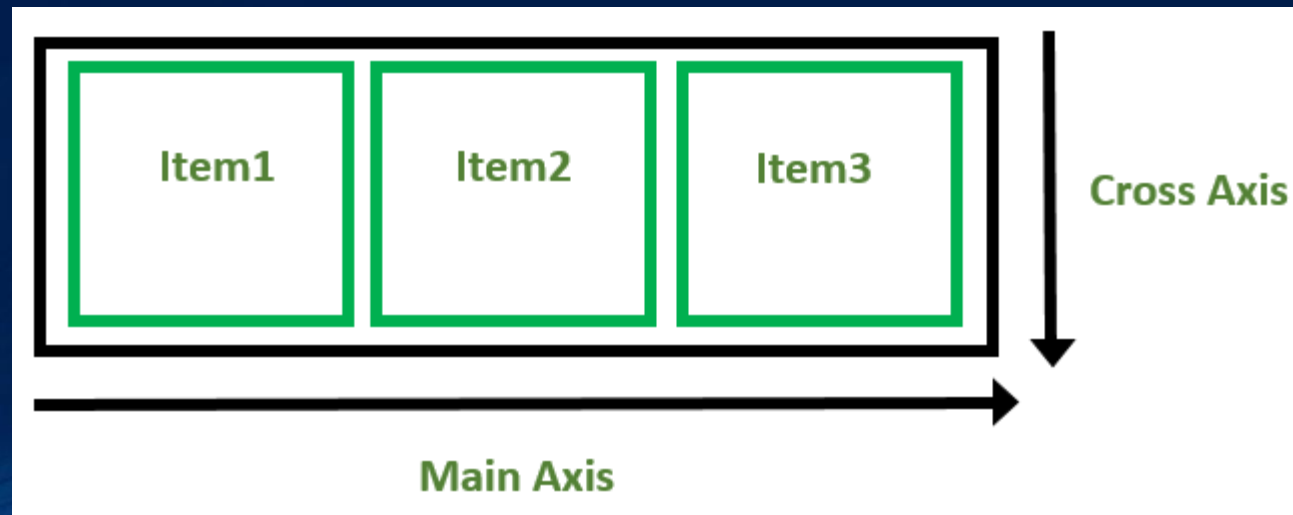
delenit au gue duis dolore te feugiat nulla  
facilisi.

# Flexbox

Flexbox дозволяє контролювати розмір, порядок та вирівнювання елементів по кількох осях та розподіл вільного місця між елементами. Flexbox визначає набір CSS властивостей для контейнера (flex-контейнер, властивість **display:flex**) та його дочірніх елементів (flex-блоків).

Головною віссю flex-контейнера є напрям, відповідно до якого розташовуються всі його дочірні елементи, що задається за допомогою властивості **flex-direction**.

Поперечною віссю називається напрямок, перпендикулярний головній осі.

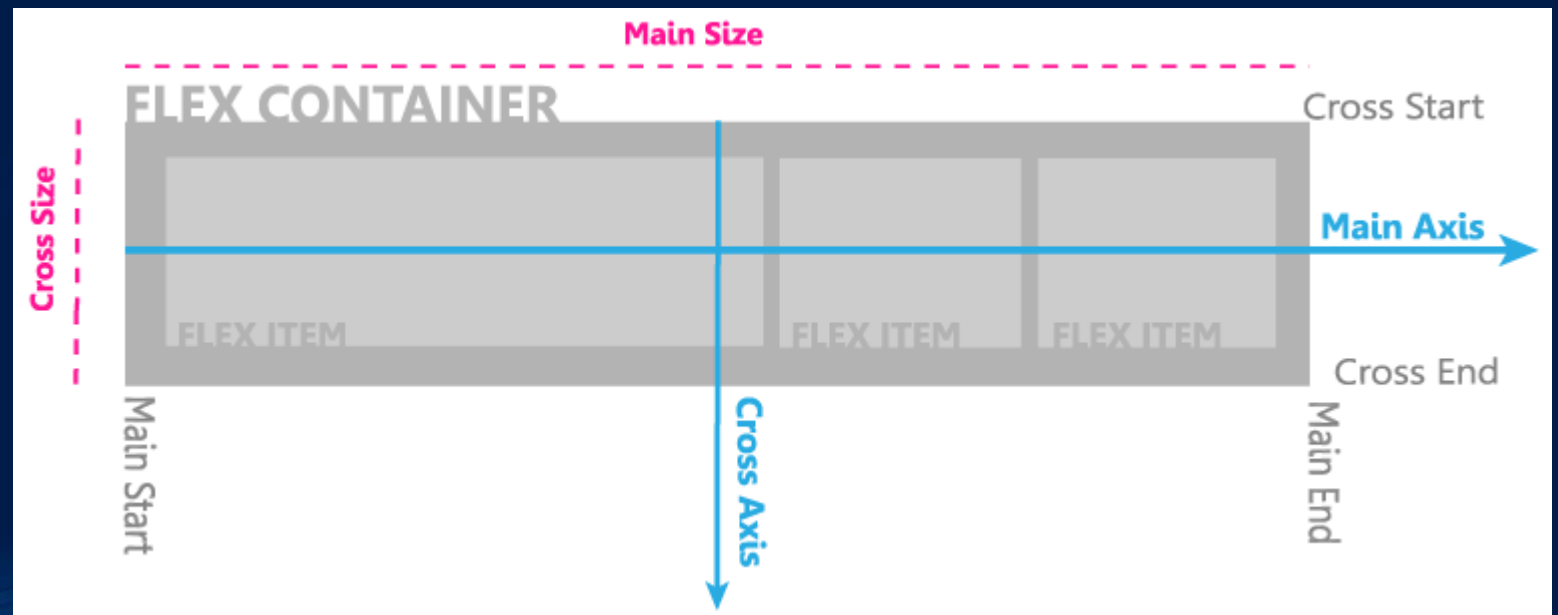




# Flexbox

Основні переваги CSS Flexible Box Layout Module (Flexbox):

- гумова верстка всіх блоків. Елементи можуть стискатися та розтягуватися за заданими правилами, займаючи потрібний простір.
- вирівнювання по вертикалі та горизонталі, базової лінії тексту.
- можливість змінювати задане в HTML розташування елементів.
- автоматичне розміщення елементів займаючи все вільне місце на сторінці.
- можливість автоматичної зміни розташування елементів праворуч-наліво.



# Flexbox. Властивості

**flex-direction** – властивість елемента-контейнера, встановлює основну вісь, яка визначає напрямок flex-елементів, що поміщаються у flex-контейнер. Приймає значення **row** або **row-reverse** (для розміщення елементів у рядок) та **column** або **column-reverse** (для розміщення в колонку).

**justify-content** и **align-items** – властивості елемента-контейнера, які визначають вирівнювання елементів вздовж головної та поперечної осей.

**order** дозволяє змінити порядок відображення конкретного елемента у flex-контейнері, набуває негативних або позитивних значень, що вказують зміщення порядку щодо початкового.

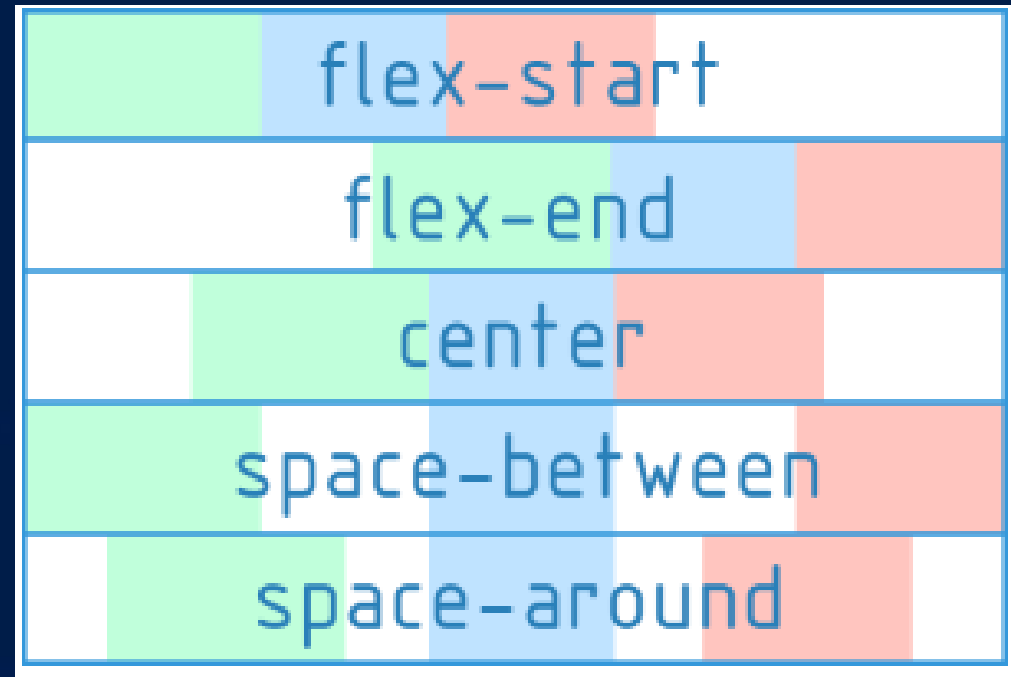
**flex-wrap** визначає можливість перенесення елементів на новий рядок, за потреби (якщо не поміщаються в одному). Приймає значення **nowrap**, **wrap**, **wrap-reverse** (зліва направо та **знизу вгору**).

Скорочена властивість **flex-flow** поєднує **flex-direction** та **flex-wrap**. Наприклад, **flex-flow: row wrap** вказує, що елементи повинні розташовуватись у ряд і автоматично переноситися на новий рядок.

# Flexbox. Властивість justify-content

Властивість justify-content вирівнює flex-елементи по головній осі поточного рядка flex-контейнера для розподілу вільного простору, що залишився.

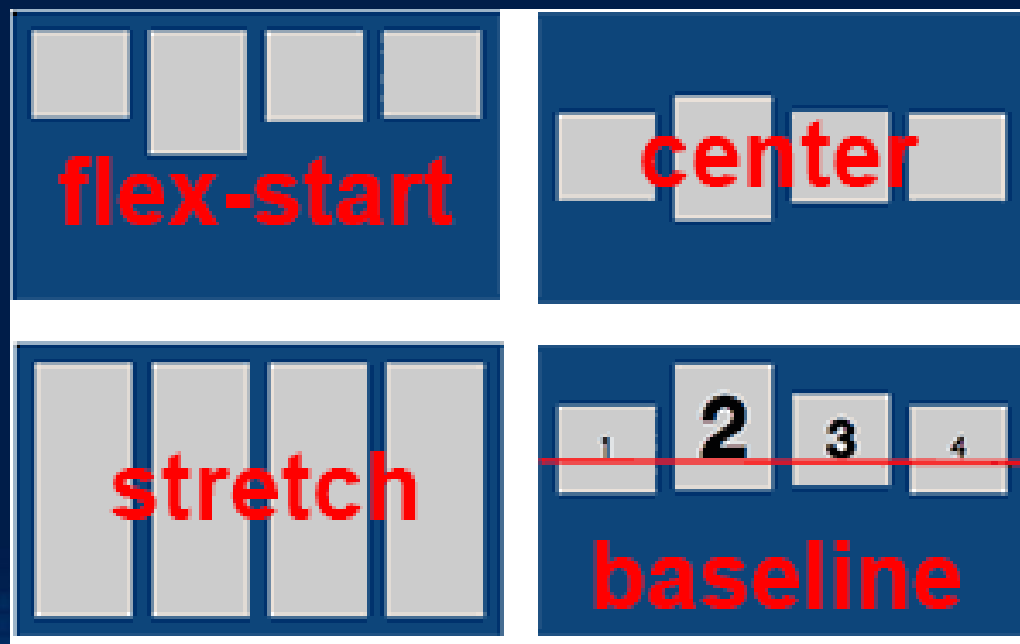
- flex-start – елементи вирівнюються по лівій стороні контейнера.
- flex-end – елементи вирівнюються по правій стороні контейнера.
- center – елементи вирівнюються по центру контейнера.
- space-between – елементи відбиваються з однаковими відступами між ними.
- space-around – елементи відображаються з однаковими відступами довкола них.



# Flexbox. Властивість *align-items*

- flex-start – елементи вирівнюються по верхньому краю контейнера.
- flex-end – елементи вирівнюються по нижньому краю контейнера.
- center – елементи вирівнюються вертикально по центру контейнера.
- baseline – елементи відображаються на базовій лінії контейнера.
- stretch – елементи розтягуються, щоб заповнити контейнер.

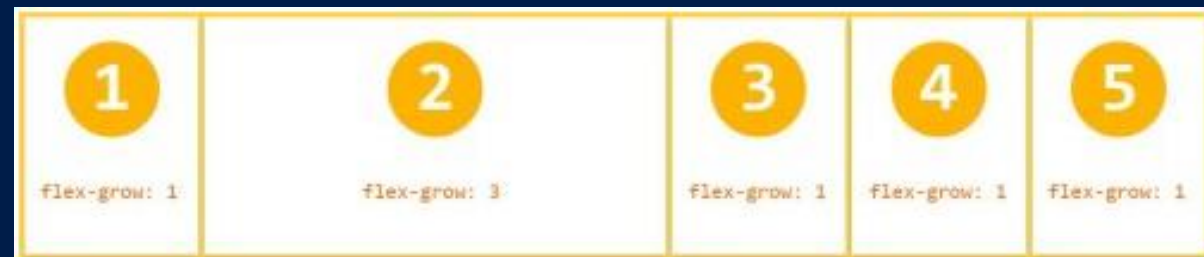
Властивість **align-self**  
дозволяє керувати  
вирівнюванням окремого  
елемента.





# Flexbox. Розміри елементів

**flex-grow** – визначає, наскільки буде розтягнутий flex-елемент щодо flex-елементів, що залишилися у flex-контейнері при розподілі вільного простору.



**flex-shrink** – визначає, наскільки буде звужений flex-елемент щодо flex-елементів, що залишилися в контейнері при розподілі вільного простору. За замовчуванням будь-які flex-елементи можуть бути звужені (коли простору не вистачає), але якщо `flex-shrink: 0` (не звужуються), вони збережуть початковий розмір.

**flex-basis** – вказує вихідний, ймовірний розмір flex-елемента, перш ніж розподілятиметься вільний простір.

# *Контрольні запитання*

- У чому різниця між абсолютним та фіксованим позиціонуванням?
- Призначення властивості float.
- Які переваги CSS Flexible Box Layout Module?
- У чому різниця між властивостями align-content і align-items?
- Чи можна елемент флекс-контейнера самого зробити флекс-контейнером?