

## Зміст

<b>КРОК 1. Вибір технології.</b>	<b>2</b>
<b>КРОК 2. Установка.</b>	<b>2</b>
Обнуління.....	2
box-sizing із значенням border-box .....	2
<b>КРОК 3. Створення контейнера сітки.</b>	<b>4</b>
<b>КРОК 4. Розрахунок ширини колонок.</b>	<b>5</b>
Проблема 1. Визначитись з одиницею виміру.....	5
Проблема 2. Як бути з зовнішніми однобічними відступами в колонках?	6
Проблема 3. Як вирішити проблему з відступами на більшій кількості рядків? .....	6
<b>КРОК 5. Створення шаблону сітки (на 12 колонок)</b>	<b>8</b>
<b>КРОК 6. Адаптивні сітки.</b>	<b>8</b>
Підготовча робота. ....	9
- Прибиваємо футер донизу .....	9
Адаптивна сітка для грід-контейнеру на 3 колонки.....	9
Адаптивна сітка для грід-контейнеру на 4 колонки.....	10
Адаптивна сітка flex на 2 колонки + grid на 9 елементів у 3 колонки ....	11
Адаптивний грід у 6 колонок в лінію .....	12
Адаптивна сітка для 5 блоків .....	13
Адаптивні модифікатори для різних розмірів стовпчиків, зміщення, вирівнювання і розподілення по ширині області перегляду .....	15
<b>КРОК 7. Додаткові методи для спрощення адаптації:</b>	<b>18</b>
Використання змінних .....	18
Формула розрахунку адаптивного (резинового) шрифту .....	21
<b>ВИСНОВОК.</b>	<b>22</b>

Адаптивна сітка - один із кращих інструментів для створення унікального дизайну. Ми можемо налаштувати все, що потрібно - кількість і ширину колонок, відступи і навіть контрольні точки, при досягненні яких перебудовується розкладка сторінки. Тоді почнемо?

Основні питання, які ми намагались вирішити:

1. Дизайн сітки;
2. Як сітка поводить себе на різних в'юпортах;
3. Використовуємо CSS- або HTML-сітку.

У цій розробці ми об'єднали ідеї побудови *HTML*-сітки та *CSS*-сітки.

Отже,

## КРОК 1. Вибір технології.

Вирішили зупинитись на *CSS*-ґрідах та флексах.

## КРОК 2. Установка.

*/\*Обнуління\*/*

**! Обов'язково звертаємо увагу на властивість  
box-sizing із значенням border-box**

```
*{  
  
    padding: 0;  
    margin: 0;  
    border: 0;  
}  
  
html {  
    -moz-box-sizing: border-box;  
    -webkit-box-sizing: border-box;  
    box-sizing: border-box;  
}
```

```
*,
*:before,
*:after {
    box-sizing: inherit;
}

:focus,:active{outline: none;}

a:focus,a:active{outline: none;}

nav,footer,header,aside{display: block;}

html,body{
    height: 100%;
    width: 100%;
    font-size: 100%;
    line-height: 1;
    font-size: 14px;
    -ms-text-size-adjust: 100%;
    -moz-text-size-adjust: 100%;
    -webkit-text-size-adjust: 100%;
}

input,button,textarea{font-family:inherit;}

input::-ms-clear{display: none;}

button{cursor: pointer;}

button::-moz-focus-inner {padding:0;border:0;}

a,a:visited{text-decoration: none;}

a:hover{text-decoration: none;}

ul li{list-style: none;}

img{vertical-align: top;}

h1,h2,h3,h4,h5,h6{font-size:inherit;font-weight: 400;}
```

## КРОК 3. Створення контейнера сітки.

```
.container {  
  --width: calc((100% - 20px * 12) / 12); //Для 12-колоночної сітки//  
  width: var(--width);  
  --max-width: 1000px;  
  max-width: var(--max-width);  
  margin: 0 auto;  
}  
  
@media screen and (max-width: 480px) {  
  .container {  
    padding: 0px 15px;  
  }  
}
```

**Примітка:** для кращої доступності та адаптивності можна використовувати замість пікселів відносні одиниці виміру на зразок `em` або `rem`.

**Пояснення:** в даній формулі ми використали змінні (`--width`, `--max-width`), які в проєкті можна зазначати як, наприклад, `width: var(--width)`; `max-width: var(--max-width)`;

*Таке застосування дозволить більш гнучко використовувати зазначені параметри і скоротить код (при умові підключення параметрів бібліотеки).*

## КРОК 4. Розрахунок ширини колонок.

Ми створюємо колонки за допомогою властивостей:

- width;
- margin або padding.

### Проблема 1. Визначитись з одиницею виміру

Для отримання «резинової» сітки, нам потрібна одиниця виміру, яка залежить від ширини контейнера: це відсотки. Нижче, приклади розрахування розмірів колонок (блоків) у відсотках залежно від необхідної кількості.

```
.grid1 {  
  width: calc(100% - 5px);  
}  
  
.grid2 {  
  width: calc((100% - 5px) / 2);  
}  
  
.grid3 {  
  width: calc((100% - 5px) / 3);  
}  
  
.grid4 {  
  width: calc((100% - 5px) / 4);  
}  
  
.grid5 {  
  width: calc((100% - 5px) / 5);  
}  
  
.grid6 {  
  width: calc((100% - 5px) / 6);  
}  
  
.grid7 {  
  width: calc((100% - 5px) / 7);  
}  
  
.grid8 {  
  width: calc((100% - 5px) / 8);  
}  
  
.grid9 {  
  width: calc((100% - 5px) / 9);  
}
```

```
.grid10 {
  width: calc((100% - 5px) / 10);
}

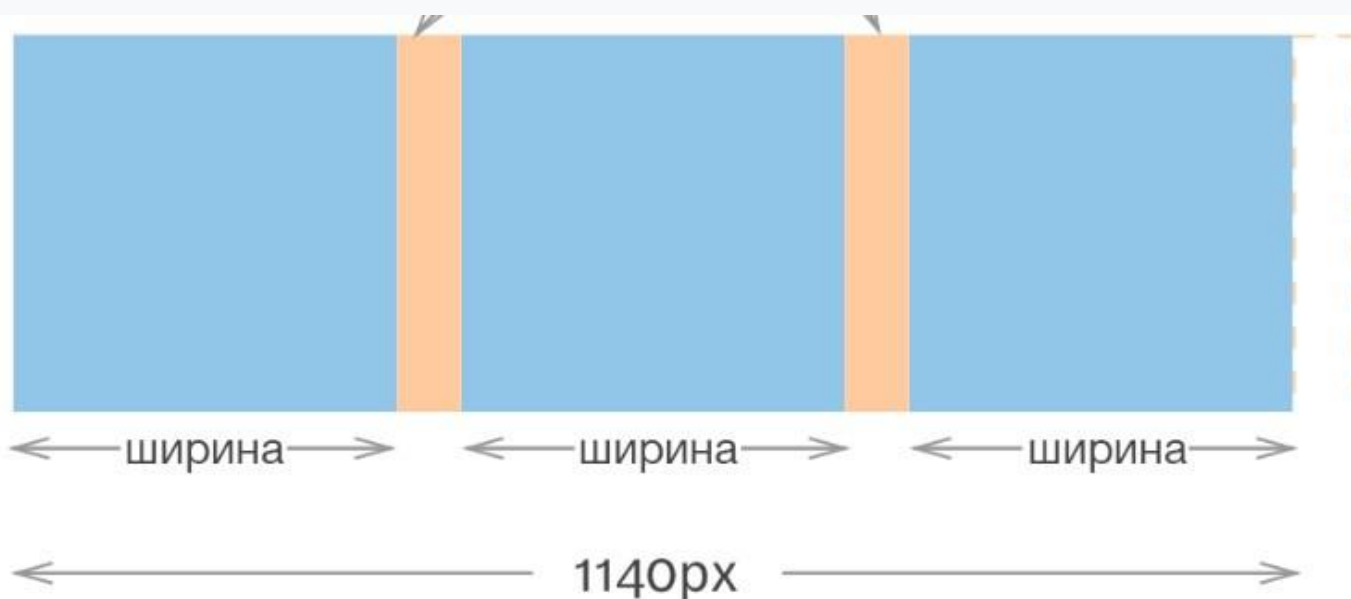
.grid11 {
  width: calc((100% - 5px) / 11);
}

.grid12 {
  width: calc((100% - 5px) / 12);
}
```

**Прим.** Де 5px - розмір зовнішніх відступів контейнеру, а grid(N) – ширина КОЛОНКИ

## Проблема 2. Як бути з зовнішніми одnobічними відступами в колонках?

Наприклад, є 3 колонки та 2 внутрішніх відступи.



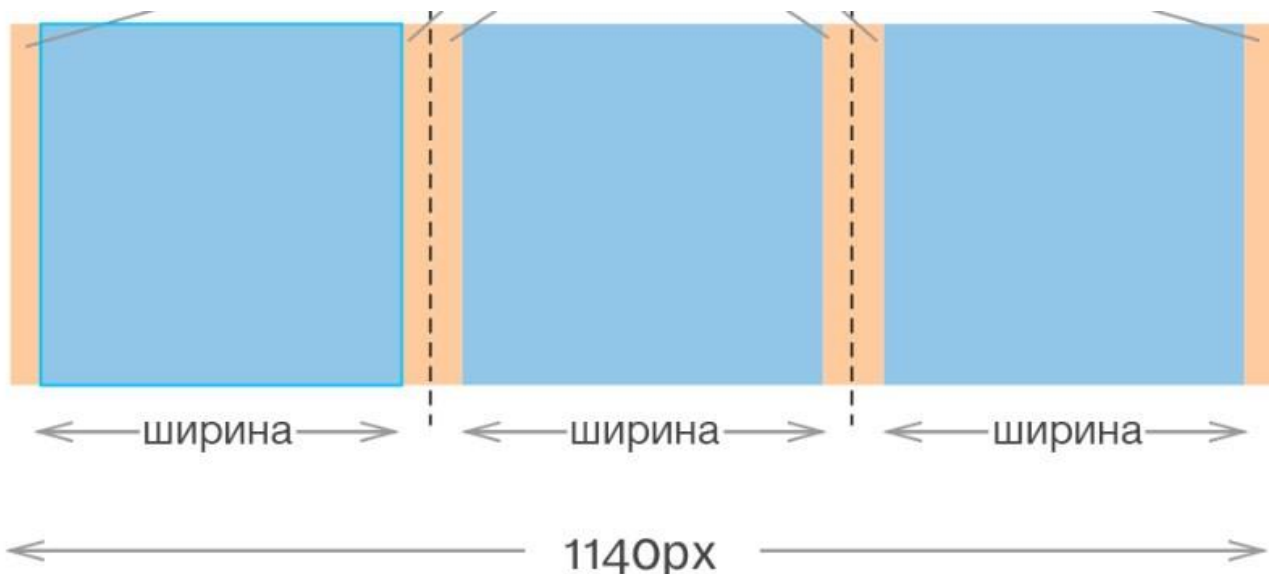
КОД CSS:

```
.grid-item {
  width: calc((100% - 20px * 2) / 3);
}
```

## Проблема 3. Як вирішити проблему з відступами на більшій кількості рядків?

Цей код працює, коли сітка містить лише один рядок. Але він не справляється, якщо рядків з елементами більше, ніж один 😞

Тоді ми можемо розділити зовнішні однобічні відступи на дві частини і розмістити по половині з кожного боку колонки. Код буде виглядати приблизно так: `.grid-item { margin-right: 10px; margin-left: 10px; }`



#### КОД HTML:

```
<div class="container">
  <div class="grid4__contents">
    <div class="grid-item">Елемент сітки</div>
    <div class="grid-item">Елемент сітки</div>
    <div class="grid-item">Елемент сітки</div>
    <div class="grid-item">Елемент сітки</div>
    <div class="grid-item">Елемент сітки</div>
    <div class="grid-item">Елемент сітки</div>
  </div>
</div>
```

#### КОД CSS:

```
.container {
  max-width: 1140px;
  margin: 0 auto;
}

.grid4__contents {
  display: flex;
  flex-wrap: wrap;
}

.grid-item {
  width: calc((100% - 20px * 3) / 3);
  margin: 20px 10px 0 10px;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 80px;
  text-align: center;
  background-color: rgb(235, 36, 36);
}
```

**Пояснення:** **width** елемента розраховується по формулі :  $100\% - (\text{сума відступів зправа та зліва, тобто } \text{margin-right: } 10\text{px} + \text{margin-left: } 10\text{px}) * (\text{кількість не розділених зовнішніх відступів}) \text{ та поділених на кількість колонок у рядку.}$

**Отже**, завдання з шириною колонок та відступами вирішено! Але адаптивність лише «резинова», що не підходить для мобільних пристроїв..

## КРОК 5. Створення шаблону сітки (на 12 колонок)

**КОД HTML:**

```
<div class="template-grid">
  <div class="column"></div>
  <div class="column"></div>
  <div class="column"></div>
  <div class="column"></div>
  <div class="column"></div>
  <div class="column"></div>
  <div class="column"></div>
  <div class="column"></div>
  <div class="column"></div>
  <div class="column"></div>
  <div class="column"></div>
  <div class="column"></div>
  <div class="column"></div>
</div>
```

**КОД CSS:**

```
.column {
  width: calc((100% - 20px * 12) / 12);
  margin: 20px 10px 0 10px;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 80px;
  text-align: center;
  background-color: rgb(235, 36, 36);
}
.template-grid{
  display: flex;
}
```

Використовуючи цей шаблон сітки, вносимо в неї зміни для створення адаптивності.

## КРОК 6. Адаптивні сітки.

До нас на допомогу поспішають CSS-ґіди та флекси.



Підготовча робота.

Прибиваємо футер донизу:

**А. за допомогою ґрів:**

```
html{
  height: 100%;
}
body{
  min-height: 100%;
  display: grid;
  grid-template-rows: 1fr auto;
}

footer{
  grid-row-start: 2;
  grid-row-end: 3;
}
```

**В. за допомогою флексів:**

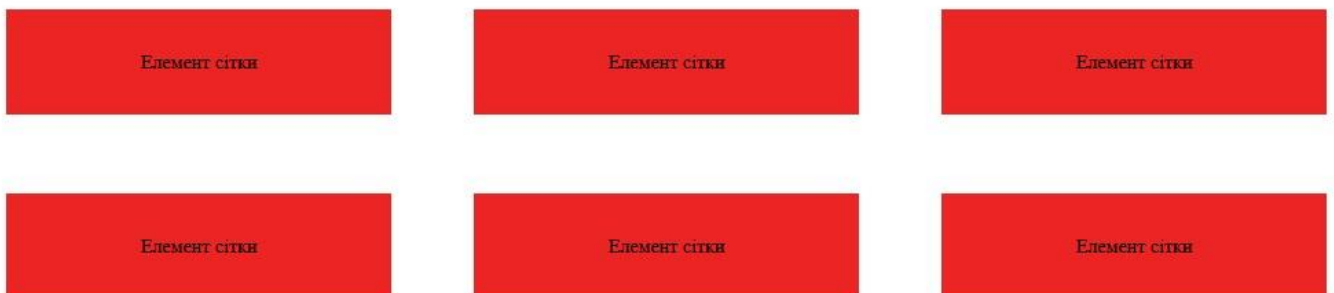
```
html,
body {
  height: 100%;
}

body {
  display: flex;
  flex-direction: column;
}

main{
  flex: 1 1 auto;
}

footer {
  flex-shrink: 0;
}
```

Адаптивна сітка для ґрід-контейнеру на 3 колонки



**КОД CSS:** //для ґрід-контейнеру//

```
.grid3__content {
  display: -ms-grid;
  display: grid;
}
```

```

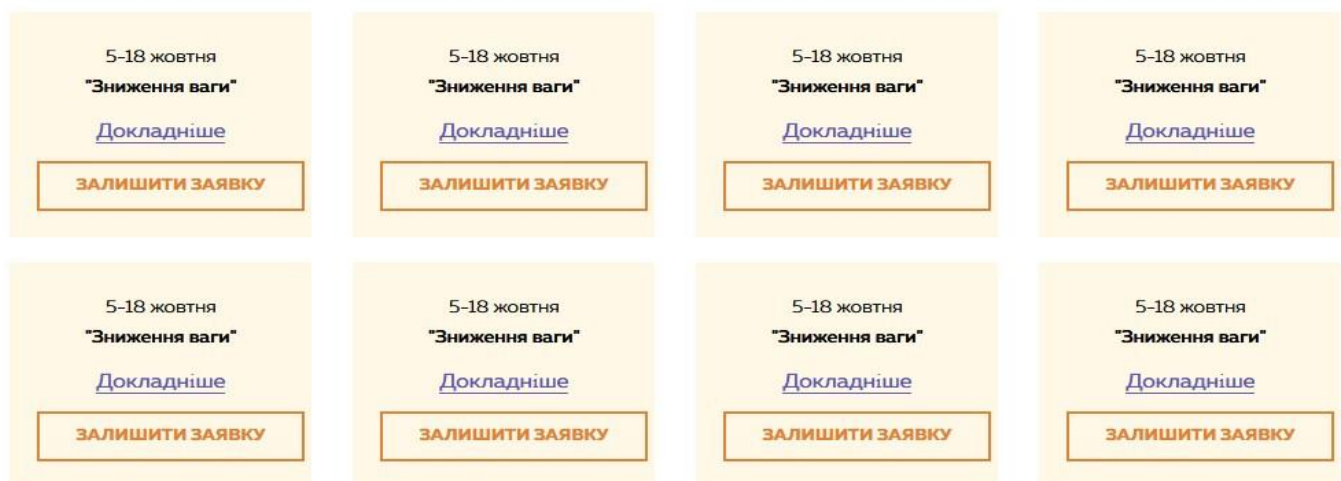
-ms-grid-columns: 1fr 1fr 1fr;
grid-template-columns: 1fr 1fr 1fr;
grid-column-gap: 20px;
grid-row-gap: 20px;
}

@media screen and (max-width: 1000px) {
.grid3__content {
-ms-grid-columns: 1fr 1fr;
grid-template-columns: 1fr 1fr;
grid-row-gap: 25px;
padding-bottom: 20px;
}
}

@media screen and (max-width: 480px) {
.grid3__content {
display: block;
}
}

```

## Адаптивна сітка для грід-контейнеру на 4 колонки



### КОД CSS: //для грід-контейнеру//

```

.grid4__contents {
--grid-gap: 20px;
display: -ms-grid;
display: grid;
-ms-grid-columns: 1fr 1fr 1fr 1fr;
grid-template-columns: 1fr 1fr 1fr 1fr;
grid-gap: var(--grid-gap);
grid-auto-rows: minmax(180px, auto);
justify-items: center;
}

@media screen and (max-width: 1000px) {
.grid4__contents {
-ms-grid-columns: 1fr 1fr 1fr;
grid-template-columns: 1fr 1fr 1fr;
}
}

```

```

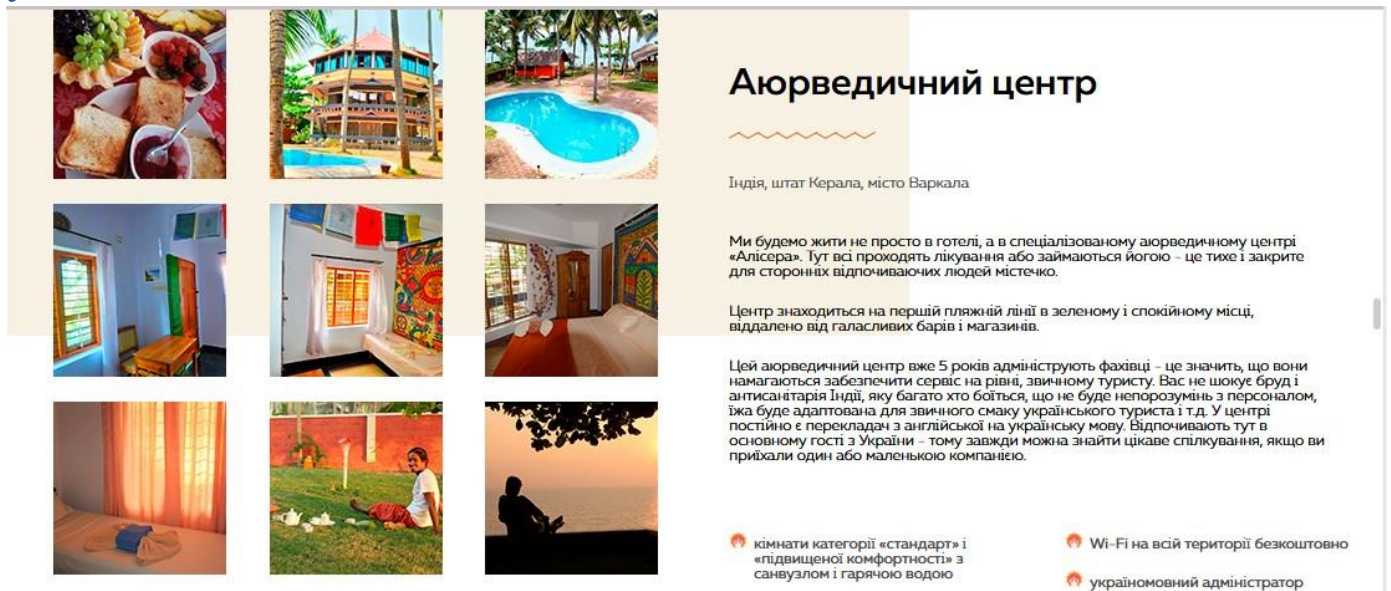
@media only screen and (max-width: 767px) {
  .grid4__contents {
    -ms-grid-columns: 1fr 1fr;
    grid-template-columns: 1fr 1fr;
  }
}
@media screen and (max-width: 560px) {
  .grid4__contents {
    --grid-gap: 10px;
    -ms-grid-columns: 1fr;
    grid-template-columns: 1fr;
    grid-gap: var(--grid-gap);
  }
}

```

Окремо стилізуються елементи контейнеру.

**Пояснення:** в даному коді ми також використали змінні.

Адаптивна сітка flex на 2 колонки + grid на 9 елементів у 3 колонки



**КОД HTML:**

```

<div class="block2">
  <div class="grid-gall">
  </div>
  <div class="text">
  </div>
</div>

```

**КОД CSS:**

```

.block2 {
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  -webkit-box-align: start;
  -ms-flex-align: start;
}

```

```

    align-items: flex-start;
    position: relative;
}

.grid-gall {
  --grid-gap: 20px;
  -webkit-box-flex: 1;
  -ms-flex: 1 0 50%;
  flex: 1 0 50%;
  margin-right: 50px;
  display: -ms-grid;
  display: grid;
  -ms-grid-columns: 1fr 1fr 1fr;
  grid-template-columns: 1fr 1fr 1fr;
  grid-gap: var(--grid-gap);
  justify-items: center;
}

@media screen and (max-width: 1000px) {
  .grid-gall {
    --margin-r: 20px;
    margin-right: var(--margin-r);
    grid-gap: 10px;
    -ms-grid-columns: 1fr 1fr;
    grid-template-columns: 1fr 1fr;
  }
}

@media screen and (max-width: 560px) {
  .grid-gall {
    --margin-r: 0px;
    -ms-grid-columns: 1fr 1fr 1fr;
    grid-template-columns: 1fr 1fr 1fr;
    margin-right: var(--margin-r);
  }
}

@media screen and (max-width: 480px) {
  .grid-gall {
    -ms-grid-columns: 1fr 1fr;
    grid-template-columns: 1fr 1fr;
  }
}

```

## Адаптивний ґрид у 6 колонок в лінію

### КОД CSS:

```

.grid6-line {
  display: -ms-grid;
  display: grid;
  -ms-grid-columns: 1fr 1fr 1fr 1fr 1fr 1fr;
  grid-template-columns: 1fr 1fr 1fr 1fr 1fr 1fr;
  -webkit-box-pack: justify;
  -ms-flex-pack: justify;
}

```

```

justify-content: space-between;
}

@media screen and (max-width: 980px) {
  .grid6-line {
    -ms-grid-columns: 1fr 1fr 1fr;
    grid-template-columns: 1fr 1fr 1fr;
  }
}

@media screen and (max-width: 480px) {
  .grid6-line {
    -ms-grid-columns: 1fr 1fr;
    grid-template-columns: 1fr 1fr;
    grid-column-gap: 10px;
  }
}

```

## Адаптивна сітка для 5 блоків

(на екранах розмірами max-width:840px; має вигляд 2-1-2;

max-width:560px; має вигляд 1-1-1-1-1)



Провести відпустку нестандартно



Пізнати культуру Індії зсередини



Покращити здоров'я



Позбавитися стресів  
та "перезавантажитися"



Відпочити у гарній компанії та  
знайти однодумців

## КОД CSS:

```

.wrapper, .block-top, .block-bottom {
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
}

.wrapper{
  text-align: center;

```

```
-webkit-box-orient: vertical;
-webkit-box-direction: normal;
-ms-flex-direction: column;
flex-direction: column;
}

.block-top {
  -ms-flex-wrap: wrap;
  flex-wrap: wrap;
}

.block-top .item {
  -webkit-box-align: center;

  -ms-flex-align: center;
  align-items: center;
  -webkit-box-flex: 0;
  -ms-flex: 0 1 33.333%;
  flex: 0 1 33.333%;
}

.block-top .item , .block-bottom .item {
  margin-bottom: 10px;
}

.block-bottom {
  margin-top: 45px;
}

.block-bottom .item {
  -webkit-box-flex: 0;
  -ms-flex: 0 1 50%;
  flex: 0 1 50%;
}

.block-bottom .item:nth-child(1) {
  -webkit-box-align: end;
  -ms-flex-align: end;
  align-items: flex-end;
  padding-right: 70px;
}

.block-bottom .item:nth-child(2) {
  -webkit-box-align: start;
  -ms-flex-align: start;
  align-items: flex-start;
  padding-left: 70px;
}

@media screen and (max-width: 840px) {

  .block-top .item {
    -webkit-box-flex: 0;
    -ms-flex: 0 1 50%;
    flex: 0 1 50%;
  }
}
```

```

.block-top .item:nth-child(3) {
  -webkit-box-flex: 0;
  -ms-flex: 0 1 100%;
  flex: 0 1 100%;
  margin-top: 45px;
}
.block-top .item:nth-child(3) {
  margin-top: 20px;
}
}

@media screen and (max-width: 560px) {

  .block-top {
    display: block;
  }

  .block-top .item {
    margin: 20px 0;
  }

  .block-top .item:nth-child(3) {
    margin-top: 0px;
  }

  .block-bottom {
    display: block;
    margin-top: 10px;
  }

  .block-bottom .item:nth-child(1),
  .block-bottom .item:nth-child(2) {
    -webkit-box-align: center;
    -ms-flex-align: center;
    align-items: center;
  }

  .block-bottom .item:nth-child(1) {
    padding-right: 0px;
  }

  .block-bottom .item:nth-child(2) {
    padding-left: 0px;
  }
}

```

**Адаптивні модифікатори** для різних розмірів стовпчиків, зміщення, вирівнювання і розподілення по ширині області перегляду

**Xs** (max-width:1140px),  
**sm** (max-width:540px),  
**md** (max-width:720px) ,  
**lg** (max-width:960px).

// змістити стовпець //

```
<div class="row">  
  <div class="col-xs-offset-3 col-xs-9">  
    <div class="box">offset</div>  
  </div>  
</div>
```

// Додайте будь-яку кількість стовпців для автоматичного визначення розміру в рядку //

```
<div class="row">  
  <div class="col-xs">  
    <div class="box">auto</div>  
  </div>  
</div>
```

//Вкладеність блоків //

```
<div class="row">  
  <div class="col-xs">  
    <div class="box">  
      <div class="row">  
        <div class="col-xs">  
          <div class="box">auto</div>  
        </div>  
      </div>  
    </div>  
  </div>  
</div>
```

//Додайте класи для вирівнювання елементів на початку або в кінці рядка, а також у верхній, нижній частині чи по центру стовпця //

```
<div class="row start-xs">  
  <div class="col-xs-6">  
    <div class="box"> start </div>  
  </div>  
</div>
```

```
<div class="row center-xs">  
  <div class="col-xs-6">  
    <div class="box">start </div>  
  </div>  
</div>
```

```
<div class="row end-xs">  
  <div class="col-xs-6">  
    <div class="box">end </div>  
  </div>  
</div>
```



//Ось приклад використання модифікаторів спільно для досягнення різного вирівнювання при різних розмірах вікна перегляду.

```
<div class="row center-xs end-sm start-lg">
  <div class="col-xs-6">
    <div class="box">All together now </div>
  </div>
</div>
```

```
<div class="row top-xs">
  <div class="col-xs-6">
    <div class="box">top </div>
  </div>
</div>
```

```
<div class="row middle-xs">
  <div class="col-xs-6">
    <div class="box">center </div>
  </div>
</div>
```

```
<div class="row bottom-xs">
  <div class="col-xs-6">
    <div class="box">bottom </div>
  </div>
</div>
```

Додайте класи для розподілу вмісту рядка чи стовпця. //

```
<div class="row around-xs">
  <div class="col-xs-2">
    <div class="box">around </div>
  </div>
  <div class="col-xs-2">
    <div class="box">around </div>
  </div>
  <div class="col-xs-2">
    <div class="box">around </div>
  </div>
</div>
```

```
<div class="row between-xs">
  <div class="col-xs-2">
    <div class="box">between </div>
  </div>
  <div class="col-xs-2">
    <div class="box">between </div>
  </div>
</div>
```

//Додайте класи для упорядкування стовпців.//

```
<div class="row">
  <div class="col-xs-2">
    <div class="box">1 </div>
  </div>
  <div class="col-xs-2">
    <div class="box">2 </div>
  </div>
</div>
```

```

    </div>
    <div class="col-xs-2 first-xs">
      <div class="box">3 </div>
    </div>
  </div>

  <div class="row">
    <div class="col-xs-2 last-xs">
      <div class="box">1 </div>
    </div>
    <div class="col-xs-2">
      <div class="box">2 </div>
    </div>
    <div class="col-xs-2">
      <div class="box">3 </div>
    </div>
  </div>

  //Reversing//
  <div class="row reverse">
    <div class="col-xs-2">
      <div class="box">1 </div>
    </div>
    <div class="col-xs-2">
      <div class="box">2 </div>
    </div>
    <div class="col-xs-2">
      <div class="box">3 </div>
    </div>
  </div>

```

## КРОК 7. Додаткові методи для спрощення адаптації:

1. Використання змінних
2. Адаптивний (резиновий) шрифт

## ВИКОРИСТАННЯ ЗМІННИХ

- для шрифтів

```

:root {
  --global-font-size: 16px;
  --font-sz1: 3.750em;
  /*60px*/
  --font-sz2: 1.250em;

```

```

/*20px*/
--font-sz3: 1.125em;
/*18px*/
--font-sz4: 1.000em;
/*16px*/
--font-sz5: 1.875em;
/*30px*/
--font-sz6: 3.000em;
/*48px*/
--font-sz7: 0.750em;
/*12px*/
--font-sz8: 0.875em;
/*14px*/
--font-sz9: 1.375em;
/*22px*/
--font-sz10: 1.500em;
/*24px*/
--font-sz11: 1.625em;
/*26px*/
--font-sz12: 1.750em;
/*28px*/
--font-sz13: 2.250em;
/*36px*/
}

```

## Typography | Global

```

:root {
Body  font size
  --text-base-size: 1em;
type scale
  --text-scale-ratio: 1.2;
  --text-xs: calc((1em / var(--text-scale-ratio)) / var(--text-scale-ratio));
  --text-sm: calc(var(--text-xs) * var(--text-scale-ratio));
  --text-md: calc(var(--text-sm) * var(--text-scale-ratio) * var(--text-scale-ratio));
  --text-lg: calc(var(--text-md) * var(--text-scale-ratio));
  --text-xl: calc(var(--text-lg) * var(--text-scale-ratio));
  --text-xxl: calc(var(--text-xl) * var(--text-scale-ratio));
}

@media (min-width: 64rem) {

  //чуйне рішення, застосоване до всіх текстових елементів//
  :root {
    --text-base-size: 1.25em;
    --text-scale-ratio: 1.25;
  }
}

h1,
.text-xxl {
  font-size: var(--text-xxl, 2.074em);
}

h2,
.text-xl {

```

```

    font-size: var(--text-xl, 1.728em);
}

h3,
.text-lg {
    font-size: var(--text-lg, 1.44em);
}

h4,
.text-md {
    font-size: var(--text-md, 1.2em);
}

.text-base {
    font-size: 1em;
}

small,
.text-sm {
    font-size: var(--text-sm, 0.833em);
}

.text-xs {
    font-size: var(--text-xs, 0.694em);
}

```

- Для зовнішніх відступів margin

```

:root {
    --margin-c: 0 auto;
    /* вирівнювання по центру */
    --margin-r: 10px;
    /* відступ з правого боку */
    --margin-l: 10px;
    /* відступ з лівого боку */
    --margin-t: 100px;
    /* відступ зверху */
    --margin-b: 30px;
    /* відступ знизу */
}

```

- Для внутрішніх відступів padding

```

:root {
    --padding: 10px;
    /*з усіх боків*/
    --padding-r: 10px;
    /*зправа*/
}

```

- Для: flex-basis: var(--grid-N);

```
:root {  
  --grid-1: 100%;  
  --grid-2: 50%;  
  --grid-3: 33.333333%;  
  --grid-4: 25%;  
  --grid-5: 20%;  
  --grid-6: 16.666666%;  
  --grid-7: 14.2857142%;  
  --grid-8: 12.5%;  
  --grid-9: 11.1111111%;  
  --grid-10: 10%;  
  --grid-11: 9.090909%;  
  --grid-12: 8.3333333%;  
}
```

-Для кольору :var(--name-color):

```
:root {  
  --white: #fff;  
  --black: #000;  
  --black2: #383838;  
  --gray: #f8f2e4;  
  --red: red;  
  --orange: #db883c;  
  --yellow: yellow;  
  --green: green;  
  --blue: blue;  
  --brown: #412525;  
}
```

## Формула розрахунку адаптивного (резинового) шрифта

```
h1{  
  font-size: calc((var(--min-font-size)) + (var(--font-size) - var(--min-font-size)) * (100vw / var(--max-width))));  
}
```

**ПРИМІТКА.** --min-font-size мінімальний розмір шрифту на мобіл, --font-size розмір шрифту за макетом, --max-width максимальний розмір макету. Наприклад, h1- 40px,min - 24px, тоді font-size: calc(24px + 16 \* (100vw / 1280));

```
@media (max-width:767px){  
  h1{  
    font-size: calc((var(--min-font-size)) + ((var(--font-size) - var(--min-font-size)) + ((var(--font-size) - var(--min-font-size) * 70%)) * ((100vw - var(--min-width)) / var(--max-width))));  
  }  
}  
@media (min-width:767px) {
```

```
h1 {  
  font-size: calc((var(--min-font-size)) + ((var(--font-size) - var(--min-font-size)) * (100vw / var(--max-width))));  
}
```

**ПРИМІТКА.** --min-font-size мінімальний розмір екрану, на який встановлюється розмір шрифту (наприклад, мобіл 320px), h1- 40px, min - 24px, тоді font-size: calc(24px + (16 + 16 \* 0.7) \* ((100vw - 320px) / 1280));

## ВИСНОВОК.

Запропонована **НАША бібліотека** при використанні розроблених в ній рішень дозволить проектам адекватно працювати на всіх пристроях і платформах: телефони, смартфони, планшети, ноутбуки, стаціонарні персональні комп'ютери, з повною підтримкою всіх популярних браузерів. Це дасть можливість широкого охоплення клієнтів. Адже кожний бажаючий буде мати можливість відкривати сторінку з будь-якого пристрою і мати можливість ознайомитися з інформацією та пропозиціями.

Ідеї, викладені в цій роботі, були перевірені на реальному проєкті Yoga-тур: [https://mykhailopylka.github.io/group\\_project\\_Victory-team.github.io/](https://mykhailopylka.github.io/group_project_Victory-team.github.io/)

**Запрошуємо до спіробітництва! Будемо раді почути Ваші відгуки та побажання!**