# Модуль 9. Розкладка CSS Grid.

background-size

* contain (зображення буде зберігати свої пропорції, займаючи простір по висоті і ширині)
* cover (пропорції зберігаються і зображення розтягується на все вільне місце по всій секції)

background-attachment

* fixed (зображення залишається нерухомим при скроллінгу)

скорочений запис властивостей фонового зображення: background: *url repeat position size*

## Градієнт

**Лінійний.**

background-image: linear-gradient(gold, blue);

(потрібно щонайменш два кольори, через кому можна додавати безліч кольорів, при цьому буде розраховуватись однакова величина для всіх градієнтів. Якщо треба вказати іншу величину, то цю величину, до якої треба застосовувати градієнт, треба вказати після кольору у відсотках).

background-image: linear-gradient(gold 50%, green 70%, blue 100%);

Зміна напрямку.

Вліво: linear-gradient(to left, gold 50%, green 70%, blue 100%);

Вправо: linear-gradient(to right, gold 50%, green 70%, blue 100%);

Вверх: to top;

Вниз: to bottom;

Також можна вказати конкретний кут нахилу градієнту у градусах, наприклад, 45град:

linear-gradient(45deg, gold 50%, green 70%, blue 100%);

Різкий перехід (без розмиття): linear-gradient(gold 50%, green 50%, green 70%, blue 70%, blue 100%);

**Радіальний.**

Починається від центру

radial-gradient(gold, green, blue)

## Тінь

box-shadow

має 5 параметрів: зсув по осі Х та У (може бути від’ємним), розмитість, розширення (відносно елементу до якого застосовується), колір: box-shadow: 2px 2px 4px 4px black;

багатошарова тінь: box-shadow: 2px 2px 4px 4px black, 4px 4px 8px 8px gold;

## Псевдоелементи

використовуються для додавання декоративних ефектів без створення додаткових тегів в розмітці

обов’язковою властивістю псевдоелемента є значення *content* воно відповідає за те, що саме буде відображатися всередині псевдоелемента.

Псевдоелемети є рядковими елементами, тому, якщо треба задати їм ширину та висоту їх треба зробити блоковими (display: block).

Текст у якості псевдоелементу не задається!

# Модуль 10. Декоративні ефекти.

Для додавання кольору .svg іконкам використовується властивість fill: *color*

властивість stroke: *color* – додає обводку

## Вставка іконки

<svg>

<use href=”*посилання на спрайт зображень*#*id зображення*”> </use>

</svg>

За дефолтом, браузер обмежує зображення іконки по висоті 150пкс (якщо не задані її розміри)

Колір svg-зображенню задається властивістю - fill:

Обводка іконки задається властивістю - stroke:

Розміри іконки можна задавати як властивостями в HTML, так і в CSS

Властивість - border-radius: 50% означає, що елемент округлиться повністю, тобто буде кругом

Прозорий фон - background-color: transparent

Важливо знати! Властивість fill – наслідується

# Модуль 11.

Властивість “flex-shrink: 0” відповідає за те, щоб елемент всередині контейнеру не стискався (або задає ступінь стискання).

align-items: center – вирівнює елементи в контейнері вздовж допоміжної осі

“flex-grow: 1” або “flex-basis: 50%”– ділить елементи в контейнері навпіл (50/50)

Коли глобально додається властивість “box-sizing: border-box” треба також додати її до псевдоелементів (::before/::after і т.п.)

# Модуль 12. Позиціонування

За замовчуванням будь-який елемент має статичну позицію – властивість “position: static”

## Відносне позиціонування. position: relative

При застосуванні відносного позиціонування елемент залишається в потоці сторінки (поток сторінки/документа – це те, як розташовуються елементи, якщо до них не додані ніякі стилі CSS), тобто сусідні елементи не займають його місце.

Для нього починають працювати властивості “top, left, right, bottom” – при цьому, зміщення відбуваються відносно початкового положення елементу.

## Абсолютне позиціонування. position: absolute

При абсолютному позиціонуванні елемент виходить з потоку сторінки – сусідні елементи займають його місце (нібито його не існує).

Властивості “top, left, right, bottom” відраховуються від найближчого предка з позиціонуванням, відмінного від “static”. Якщо такого предка не знайдено, то відлік йде від меж ”body”.

Абсолютне позиціонування корисне разом з відносним.

## Фіксоване позиціонування. position: fixed

При фіксованому позиціонуванні елемент виходить з потоку сторінки, та позиціонується відносно в’юпорту (вікна браузера)

## Липке позиціонування. position: sticky

Поєднує властивості відносного та фіксованого позиціонування.

Для реалізації, по-перше, треба вказати точку, відносно якої буде виконуватись позиціонування (наприклад, ”top: 0;”), по-друге, контейнер повинний бути більший ніж ліпкий елемент.

## Накладання елементів. z-index

Ця властивість управляє глибиною (порядком) елементів на сторінці по осі Z. Дозволяє накладати елементи один на другий.

Значення може бути як цілим так і від’ємним

z-index не працює з елементом у якого позиціонування static (або, що те ж саме, взагалі не вказано – властивість за замовчанням)

Коли в елементів є властивість position то вони накладаються один на одного у тому порядку, у якому вони йдуть в HTML.

Елемент з більшим z-index розташовується вище.

Елементи з однаковими значеннями z-index будуть накладатись в порядку, у якому вони йдуть в HTML.

## Властивість overflow

Дана властивість управляє тим як поводиться вміст елемента всередині якщо його розміри більше ніж в елемента, в якому він знаходиться.

За замовчуванням значення “overflow: visible”, тобто, контент є видимим.

Сховати можна за допомогою значення “hidden” – ховається повністю, або “auto” – ховається з можливістю скролу (повзунок скролу прихований – відображається при наведенні, якщо контенту мало - взагалі не відображається), або “scroll” (повзунок скролу видимий постійно, навіть якщо контент вже менше за елемент).

Також є дві підвластивості “overflow-x”, “overflow-y” – задає скрол по осі Х чи У. (у значенні “ overflow: auto” обидві ці підвластивості додаються автоматично)

Якщо додається “position: absolute” то автоматично рядкові елементи стають блоковими. (відпадає потреба у властивості “display: block”)

## Додавання backdrop (затемнений фон)

Це фіксований елемент, який розтягується на всю ширину екрану

position: fixed;

top: 0;

left: 0;

right: 0;

bottom: 0;

Значення “display: none” у властивості “backdrop.is-hidden”– при закритті ховає повністю модальне вікно. Якщо треба застосувати до модального вікна анімацію, треба використовувати код:

backdrop.is-hidden {

opacity: 0 (робить вікно прозорим)

visibility: hidden (ховає вікно)

pointer-events: none (відключає кліки по вікну)

}

# Модуль 13. Анімація.

За замовчуванням трансформація відбувається відносно центру елементу.

## Переміщення

transform: translate(-px, -px) – дозволяє зсувати елемент по осях X та Y

transform: translateX(-px) – зміщення тільки по осі X

transform: translateY(-px) – зміщення тільки по осі Y

## Обертання

transform: rotate(-deg) – обертання на задану величину градусів “deg” або обертів “turn”

## Масштабування

transform: scale() – масштабування на заданий коефіцієнт (1.5, 2 і т.д.)

коли до елемента додається властивість “transform” то для нього також починає працювати властивість z-index

## Перекіс

transform: skew(-deg, -deg) – перекіс елемента по осях X та Y у градусах

## CSS переходи

“transition-property” – властивість, яку ми хочемо анімувати

“transition-duration” – час, швидкість переходу (у секундах “s” чи мілісекундах “ms”)

“transition-timing-function” – спеціальна функція розподілу часу. Значення – linear; ease-in; ease-out; steps (6, end); cubic-bezier ();

Згенерувати властивості функції http://cubic-bezier.com

“transition-delay” – затримка перед початком переходу (у секундах “s” чи мілісекундах “ms”)

Властивість transition треба задавати на самому елементі, до якого вона буде застосовуватись, а не на псевдоклас (hover)

.item.red {

  width: 150px;

  height: 150px;

  background-color: red;

  transition-property: background-color;

  transition-duration: 2s;

}

.item.red:hover {

  background-color: pink;

  }

Скорочений запис

.item.red {

  width: 150px;

  height: 150px;

  background-color: red;

  transition: background-color 2s;

}

.item.red:hover {

  background-color: pink;

  }

Якщо треба анімувати декілька властивостей, вони пишуться через кому:

.item.red {

  width: 150px;

  height: 150px;

  background-color: red;

  transition: background-color 2s, color 2s;

}

.item.red:hover {

  background-color: pink;

  color: black;

  }

## CSS анімація

На відміну від переходів, анімації не потрібен ніякий ініціатор (hover, тощо)

Оголошується директивою @keyframes

Ця директива повинна мати описове ім’я, та мінімум, для того щоб відбувалась анімація, задати початковий та кінцевий стан елементу. На додаток можна задати декілька проміжних станів.

.item.green {

  width: 100px;

  height: 100px;

  background-color: green;

  animation-name: spin;

  animation-duration: 1s;

  }

@keyframes spin {

  from {

transform: rotate(0);

  }

  to {

transform: rotate(360deg);

  }

}

Проміжний стан:

.item.green {

  width: 100px;

  height: 100px;

  background-color: green;

  animation-name: spin;

  animation-duration: 1s;

  }

@keyframes spin {

  0% {

transform: rotate(0);

  }

  50% {

transform: rotate(10deg);

  }

  100% {

transform: rotate(360deg);

  }

}

Анімація обертання та кольору

.item.green {

  width: 100px;

  height: 100px;

  background-color: green;

  animation-name: spin;

  animation-duration: 1s;

  }

@keyframes spin {

  0% {

transform: rotate(0);

  }

  50% {

transform: rotate(20deg);

background-color: red;

  }

  100% {

transform: rotate(360deg);

  }

}

Для повторювання анімації певну кількість разів використовується властивість “animation-iteration-count: кількість повторів”. Якщо треба, щоб вона тривала безкінечно “animation-iteration-count: infinite”

“animation-timing-function: linear” – робить анімацію плавнішою

Усі властивості окрема задавати немає необхідності, наприклад:

.item.green {

  width: 100px;

  height: 100px;

  background-color: green;

  animation-name: spin;

  animation-duration: 1s;

  animation-iteration-count: infinite;

  animation-timing-function: linear;

  }

Можна скоротити

.item.green {

  width: 100px;

  height: 100px;

  background-color: green;

  animation: spin 1s infinite linear;

}

Призупинення анімації при наведенні на елемент hover

.item.green:hover {

  animation-play-state: paused;

}

Якщо усередині батьківського елементу треба посунити усі дочірні, то можна просто додати padding

opacity – відповідає за прозорість елемента, visibility – за те, чи буде він видимий для браузера

щоб не вискакувало модальне вікно при перезавантажуванні сторінки – треба додати клас is-hidden до бекдропу

# Модуль 14. Створення веб-форм

<form> - контейнер для всіх пов’язаних елементів форми.

Має наступні необовʼязкові атрибути:

➔ name – унікальне на поточній веб-сторінці ім'я форми

➔ autocomplete – визначає, чи може браузер автоматично заповнювати

значення всіх елементів форми

➔ novalidate – говорить браузеру не перевіряти валідність введених

даних під час відправки форми

➔ action – вказується шлях до файлу або абсолютний url до файлу який обробляє цю форму (як правило скріпти php)

<input> - універсальний елемент для створення полів форми, його тип визначається атрибутом type, який може мати значення:

➔ text

➔ email

➔ password

➔ tel

➔ radio/checkbox

➔ range

➔ datetime-local

<label> - описовий текст, що логічно пов’язаний з полем введення. До <label> додається for а до поля ID. Або між тегами <label>:

<label>

Введіть ваше ім’я

<input type=”text” />

</label>

Для того, щоб працювала вибірка (радіокнопка) треба задати їм ім’я, тобто, у кожної радіокнопки повинен бути атрібут name=”--” (однаковий для всіх)

Для чекбоксу атрібут name має відрізнятися один від одного, тому що саме значення name буде приходити у базу даних.

Для відправки форми на сервер, для кнопки застосовується атрібут <submit>

<textarea> - створює багаторядкове поле. За замовчуванням вікно може розтягуватись користувачем. Щоб відмінити таку поведінку у стилях треба задати “resize: none”

<select> - створює випадаюче меню. Значення, яке буде відправлятися на сервер вказується в атрибуті value властивості option:

<select>

<option value= “L” >Size L</option>

<option value= “XL” >Size XL</option>

…

</select>

<fieldset> і <legend> - контейнер для угруповання кількох пов'язаних елементів у формі та його заголовок.

Висота <input> задається паддінгами

Підказка для поля форми – “plaseholder”

Для візуального приховування елементів використовується спеціальний стиль visually-hidden

Абсолютне позиціонування “position: absolute” робить елемент блоковим за замовчуванням

Псевдоклас “focus-within” спрацьовує у випадку, коли всередині елементу, до якого він додається є елемент у фокусі

# Модуль 15. Валідація веб-форм.

Атрібут “required” позначає поле форми як обов’язкове

Якщо у форми стоїть атрибут “novalidate”, форма не буде валідуватися, навіть, якщо поля позначені, як обов’язкові.

<form name="contact-form" autocomplete="off" novalidate></form>