Engineer

Кнопка

елемент, який натискають, щоб щось відбулося

Що це таке

Кнопка — частина інтерфейсу, яка діє, коли її активують: надсилає повідомлення, зберігає чернетку, відкриває діалог, вимикає звук у дзвінку. У вебі це зазвичай HTML-елемент `<button>`, а з погляду доступності — будь-що з роллю `button`. Старіша назва того самого — натискна кнопка, англійською «push button».

WAI-ARIA 1.2 визначає цю роль як «an input that allows for user-triggered actions when clicked or pressed», тобто елемент введення, яким користувач запускає дії клацанням або натисканням, і додає, що кнопки здебільшого використовують для окремих, завершених дій. Звичайна командна кнопка стану не має. Кнопка-перемикач, позначена `aria-pressed`, лишається увімкненою або вимкненою, доки її не натиснуть знову, а кнопка меню, позначена `aria-haspopup`, відкриває список подальших варіантів.

Найважливіша межа пролягає між кнопкою і посиланням. Посилання веде читача деінде; кнопка змінює щось там, де він уже є. ARIA Authoring Practices Guide називає ці дві функції чітко різними і, коли посилання намальовано як кнопку чи навпаки, радить узгодити візуальний дизайн із функцією, а не позичати роль іншого елемента.

HTML Living Standard описує елемент `<button>` як кнопку, підписану власним вмістом. Його атрибут `type` має три ключові слова. `submit` надсилає форму, до якої належить кнопка, `reset` повертає поля форми до початкових значень, а `button` сам нічого не робить і лишає дію скриптові. Коли `type` бракує або його написано з помилкою, кнопка перебуває в стані Auto, і така кнопка всередині форми, без прив’язаної команди, надсилає форму. Інтерактивного вмісту всередині неї бути не може: ні посилання, ні поля, ні другої кнопки.

Стандарт також дає кнопці змогу діяти на інший елемент без скрипту: `commandfor` називає ціль за її ID, а `command` каже, що з нею зробити, наприклад показати спливний елемент (popover) чи відкрити діалог як модальний. Наскільки це підтримує конкретний браузер, варто перевіряти в примітках до випуску самого браузера.

Застосуйте поняття до рішення

Застосуйте «Що таке кнопка?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Інтерфейси, що швидко завантажуються, працюють на телефоні й поводяться однаково в усіх браузерах ваших клієнтів.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Керувала full-stack-розробкою GIS-карт — full-stack-картографія — від бази даних до браузера

Керувала full-stack-розробкою GIS-карт Зв'язатися з нами →

Чому це важливо

Кнопка — місце, де читання стає дією: купівля, надсилання, збереження й видалення проходять саме через неї. Кнопка, яку частина людей не може знайти, досягти чи зрозуміти, відрізає їх від тієї частини продукту, яка оплачує все інше.

Справжній елемент приходить разом із поведінкою. `<button>` отримує фокус клавіатури, виконує дію на пробіл і на Enter, повідомляє допоміжним технологіям, що це кнопка і як вона називається, і перестає реагувати, коли має `disabled`. Документ Understanding до критерію успіху 4.1.2 у WCAG 2.2, Name, Role, Value, зазначає, що стандартні елементи керування HTML уже задовольняють критерій, коли їх використано згідно зі специфікацією. `<div>` з обробником клацання нічого з цього не має, і кожна відсутня частина когось відрізає.

Назва — це обіцянка. Екранний читач озвучує кнопку за її доступною назвою й роллю, тож назва має казати, що кнопка робить, а не як вона виглядає. APG додає правило для перемикачів, яке легко порушити: підпис лишається тим самим, коли змінюється стан. Приклад із посібника — кнопка «Mute», яку в увімкненому стані озвучують як «Mute toggle button pressed». Якщо дизайн віддає перевагу підпису, що змінюється між «Mute» і «Unmute», `aria-pressed` не потрібен, бо стан повідомляє сам підпис.

Розмір — вимога, а не справа смаку. Критерій успіху 2.5.8 у WCAG 2.2, Target Size (Minimum), рівня AA, вимагає, щоб ціль для вказівного пристрою мала щонайменше 24 на 24 CSS-пікселі. Винятки охоплюють малі цілі з достатнім простором навколо, повнорозмірний елемент деінде на сторінці, що робить те саме, посилання всередині речення, елементи, розмір яких задає браузер, і випадки, коли розмір суттєвий для змісту. Тремтіння руки, автобус у русі й великий палець на телефоні однаково ускладнюють влучання в малу ціль.

Фокус має потрапляти в розумне місце. Після натискання APG очікує, що фокус переміститься всередину діалогу, який відкрила кнопка, а коли діалог закриється, зазвичай повернеться до кнопки, що його відкрила. Якщо натискання лишає читача в тому самому контексті, як це робить кнопка «Застосувати», фокус зазвичай лишається на кнопці. Користувач клавіатури, чий фокус після кожного збереження стрибає на початок сторінки, щоразу проходить її спочатку.

Як це виглядає на практиці

Найпоказовіший випадок починається з одного озвучення. У застосунку на SwiftUI, побудованому із системних символів, екранний читач озвучував кнопку надсилання як «стрілка вгору коло, кнопка», бо символ без мітки озвучується своєю внутрішньою назвою. Коли усім сімнадцяти елементам керування з самою лише піктограмою дали назви для екранних читачів, кожен отримав мітку, що називає дію, а не форму, а там, де значення елемента залежить від стану, мітка змінюється разом зі станом. Довговічніша частина — перевірка: лінтер знаходить конструкцію, що створює елемент із самою лише піктограмою, і вимагає мітку доступності не далі ніж за вісім рядків від неї. Вісім рядків обрано на основі вимірювань, і це навмисно грубо: правило доводить, що мітка є поруч, а не що вона правильна, і хтось досі має її прослухати.

Другий урок — про те, чого кнопка зробити не може. У рівні платежів і прав доступу, побудованому на Stripe і на покупках у застосунках Apple і Google, окрема схема доступу з 11 таблиць вирішує, чи може обліковий запис користуватися каталогом компаній, пошуком і експортом даних. Сервер перевіряє цю відповідь заново під час кожного запиту, а сторінка пояснює причину одним рядком: прихована кнопка — це люб’язність, а не засіб контролю.

Кнопки важливі й для читачів, які не є людьми. Сторінка сайту про готовність до ШІ-агентів зазначає, що багато агентів, які керують браузером, читають сторінку через її дерево доступності, так само як екранний читач. Справжні кнопки й підписані поля форм — частина того, що дає агентові змогу знайти форму й заповнити її.

Власні шаблони цього сайту взагалі не містять елемента `<button>`. Заклик «Зв'язатися з нами» із суцільним тлом — це елемент `<a>`, оформлений як кнопка, бо він веде на сторінку контактів, а його правило стилів задає `min-height` 24 пікселі з коментарем, що посилається на критерій успіху 2.5.8. Головне меню — елемент `<details>`, який відкриває й закриває його `<summary>`, і коментар у шаблоні каже, що відкритим станом і семантикою кнопки володіє браузер, без жодного скрипту.

Де виникають проблеми

- **Клікабельний блок замінює елемент.** `<div>` чи `<span>` з обробником клацання відповідає на мишу і майже ні на що інше: фокусу не отримує, пробіл і Enter ігнорує, а озвучується як звичайний текст. `role="button"` виправляє лише озвучення; фокус і клавіші доводиться дописувати вручну.
- **Посилання виконує роботу кнопки або навпаки.** Посилання, що видаляє запис, можна відкрити в новій вкладці чи додати до закладок, і видалення повториться або загубиться з поля зору. Кнопку, яка лише переводить на іншу сторінку, не відкрити в новій вкладці, і адреси, яку можна скопіювати, вона не має.
- **Кнопка без типу надсилає форму.** Оскільки брак `type` ставить кнопку в стан Auto, кнопка «Додати рядок» чи «Показати пароль» усередині форми надсилає всю форму, щойно її натиснуть. `type="button"` на кожній кнопці, яка не має надсилати, прибирає цю несподіванку.
- **Піктограма — це весь підпис.** Кошик, олівець чи стрілка щось означають для того, хто їх намалював. Без доступної назви екранний читач прочитає назву файлу, назву символу або просто «кнопка». Найкраще — видиме слово; де для нього бракує місця, прогалину заповнює доступна назва, що описує дію.
- **Приховану чи вимкнену кнопку вважають захистом.** Якщо прибрати кнопку зі сторінки, це зупинить лише чесних користувачів; запит за нею можна надіслати й так, і сервер має сам вирішити, чи може цей обліковий запис це зробити.

Що почитати далі

- WAI-ARIA 1.2: button role — визначення ролі, її варіанти для перемикачів і меню, а також стани й властивості, які вона підтримує.
- HTML Living Standard: the button element — модель вмісту, типи submit, reset і button, стан Auto та атрибути `command` і `commandfor`.
- ARIA Authoring Practices Guide: Button Pattern — кнопки й посилання, поведінка клавіатури, куди переходить фокус після натискання і правило для підписів перемикачів.
- Understanding SC 2.5.8: Target Size (Minimum) — мінімум 24 на 24 пікселі, перевірка відстані та інші винятки.
- Understanding SC 4.1.2: Name, Role, Value — чому кожен елемент керування потребує назви й ролі, які можуть прочитати програми, і коли стандартні елементи HTML уже задовольняють критерій.

Словник

Обирайте за фактичними умовами

Опишіть робочий процес, наявні інтерфейси та користувачів, яким потрібна зміна. Ми можемо визначити перевірюваний результат і спосіб передачі, який дасть вашій команді змогу підтримувати його.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке кнопка?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/button/