

# Welcome on frontend PRO

Sergey Savchuk

Fullstack developer

@Sergey\_Savchuk

# План заняття

- Знайомство.
- Взаємодія з користувачем через примітивні функції.
- Підключення скриптів, перший запуск програми.
- Коментарі.
- Git. Github.

# Тег `<script></script>`

Аргументи:

- `async`
- `defer`
- `type` – legacy, зараз використовується тільки у JS-модулях.
- `language` – JavaScript by default.
- `src` - абсолютний шлях до файлу, або відносний від місця розташування сторінки. Також може бути url до файлу.

# Strict mode

Оголошення - "use strict" or 'use strict'

Сучасний мод для js, введено після переходу на ECMAScript5.

Додасть:

- Обробку помилок;
- Виправлення помилок та росту працездатності обробки коду;
- Забороняє використовувати деякі елементи синтаксису;

# Сінтаксис

Декларація - це об'ява змінної.

Присвоєння - оператор " = ".

Передача даних у функції - ви можете передавати данні як аргументи у функції.

Текст і кавички - три типу "", "", ``.

Оператори математики:

- + — додавання;
- — — віднімання;
- \* — множення;
- / — ділення;
- % — ділення за модулем;
- ++ — оператор інкременту;
- -- — оператор декременту.

# Правила назв

1. Тільки англійська, ніякого трансліту.
2. Використовуйте короткі назви тільки локально.
3. Якщо вам потрібно ввести декілька слів, використовуйте snakeCase;
4. Ім'я змінної має відображати суть!

## Важливо пом'ятати

"Зайва" змінна — добро, а не зло.

# Змінні

Змінна - іменоване сховище даних. Наприклад коробка де ви зберігаєте речі.

Ключові слова:

- `var` - writable
- `let` - writable
- `const` - readable

# Примітивні функції

**alert(value)** - показує повідомлення та чекає, поки користувач натисне кнопку «OK».

**prompt(value, [default])** - відображає модальне вікно з текстом, полем для введення тексту та кнопками ОК/Скасувати.

**confirm(value)** - відображає модальне вікно з текстом запитання (**value**) і двома кнопками: ОК та Відміна.



# Коментарі

Однорядкові - //

Багаорядкові - /\* \*/

Вожені коментарі **не підтримуються!**

# Git

Існують два основні типи систем керування версіями: з централізованим сховищем та розподіленим.

Git – найпопулярніша CVS (Control Version System).

Git – це розподілена система контролю версій, яка дозволяє відстежувати історію розробки і спільно працювати над складними проектами з будь-якої точки світу.

Головна перевага Git – в тому, що він дуже швидкий і прозорий. Він зручний для нелінійної розробки і ефективний як для невеликих проектів, так і для великих систем з тисячами учасників.

На відміну від Perforce, CVS та інших, Git зберігає знімки репозиторіїв, а не списки змін в файлах, і внаслідок цього працює набагато швидше.

Git – розподілена система. Якщо сервер з віддаленим репозиторієм вийде з ладу, можна відновити код з локальної копії. Якщо локальна копія постраждає, можна завантажити код із сервера за кілька хвилин.

# Git

Генерація ssh

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

Додавання ssh

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account>

Тест ssh

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/testing-your-ssh-connection>

**Дякую за увагу**