

Лекція: Дата, RegExp та Chrome DevTools

- Дисципліна: JavaScript та фронтенд динаміка

План заняття

- 1) Date API: створення, форматування, арифметика, часові пояси
- 2) RegExp: база, групи, флаги, lookahead
- 3) Chrome DevTools: Console, Sources, Network, Performance, Memory
- 4) Міні-демо та мікро-вправи

Цілі:

- Коректно створювати/форматувати/порівнювати дати
- Засвоїти RegExp: синтаксис, групи, флаги
- Використовувати Chrome DevTools для налагодження та аналізу

Date: створення та базові поняття

- `new Date()`: поточний час
- `new Date(iso)`, `new Date(y,m,d,h,mi,s,ms)` — місяці 0–11
- Момент зберігається в UTC, відображається в локальній TZ
- Ключові методи: `getFullYear`, `getMonth`, `getDate`, `getDay`, `getTime`

```
const now = new Date();  
const d1 = new Date('2025-09-25T09:00:00Z');  
const d2 = new Date(2025, 8, 25, 12); // 25 вер 2025 12:00 локально
```

Форматування дат (Intl)

- Intl.DateTimeFormat(locale, options) — локалізоване форматування
- Опції: dateStyle, timeStyle, timeZone, weekday, month
- Легко показувати різні часові пояси

```
const fmtUA = new Intl.DateTimeFormat('uk-UA', { dateStyle:'full',  
timeStyle:'short' });  
const fmtNY = new Intl.DateTimeFormat('uk-UA', { dateStyle:'medium',  
timeStyle:'short', timeZone:'America/New_York' });  
console.log(fmtUA.format(new Date()));  
console.log(fmtNY.format(new Date()));
```

Арифметика дат, різниця, нормалізація

- Нормалізуйте час до 00:00 перед підрахунком днів
- Додавання днів: `new Date(y, m, d + n)`
- Обережно з DST: для «календарних» днів працюйте без часу

```
function daysBetween(a,b){  
  const A = new Date(a), B = new Date(b);  
  A.setHours(0,0,0,0); B.setHours(0,0,0,0);  
  return Math.round((B - A)/86_400_000);  
}
```

Практикум: «Дні до дедлайну» та «Робочі дні місяця»

- Підрахунок днів до дати
- Список робочих днів (пн-пт) для місяця

```
const deadline = new Date(2025, 10, 30);  
console.log('Залишилось днів:', daysBetween(new Date(), deadline));
```

```
function workdaysOfMonth(year, month){  
  const res=[]; let d=new Date(year, month, 1);  
  while(d.getMonth()===month){  
    const day=d.getDay();  
    if(day>=1 && day<=5) res.push(new Date(d));  
    d = new Date(d.getFullYear(), d.getMonth(), d.getDate()+1);  
  }  
  return res;  
}
```

Регулярні вирази: база

- Класи: `\d \w \s`; Якорі: `^ $`; Квантифікатори: `+ * {m,n}`
- Групи: `(...)` та `(?:...)` — незахоплююча; Альтернація: `a|b`
- Флаги: `g, i, m, u, s`

```
const re = /\b\w{3,}\b/gi; // слова довжиною 3+  
console.log('JS makes regex fun'.match(re));
```

Групи, іменовані групи, перевірка формату

- Іменовані групи: (?<name>...)
- Backreference: \1 або \k<name>
- Валідація простих форм: дата YYYY-MM-DD, телефон, email

```
const reDate = /^(?<y>\d{4})-(?<m>\d{2})-(?<d>\d{2})$/;  
const m = '2025-09-25'.match(reDate);  
if(m){ console.log(m.groups.y, m.groups.m, m.groups.d); }
```

Lookaround: приклади

- Погляд уперед/назад: (?=...), (?!...), (?<=...), (?<!...)
- Перевірка пароля: великі букви, цифри, спецсимволи

```
const strong = /^(?=.*[A-Z])(?=.*\d)(?=.*[!@#$%^&*]).{8,}$/;  
console.log(strong.test('Super$Pass1')); // true
```

Практикум RegEx: email/телефон/події

- Email (спрощено)
- Телефон України +380XXXXXXXXX
- Подія: YYYY-MM-DD: Назва

```
const emails = text.match(/[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-z]{2,}/g);  
const phone = /^\\+380\\d{9}$/;  
const line = '2025-09-25: Консультація';  
const pat = /^(\\d{4})-(\\d{2})-(\\d{2}):(\\s*(.+))$/;
```

Chrome DevTools: огляд

- Відкриття: F12 або Ctrl+Shift+I
- Панелі: Elements, Console, Sources, Network, Performance, Memory, Application
- Command Menu: Ctrl+Shift+P

Console: форматований вивід і таймінг

- `console.log/warn/error, console.table, console.group(End)`
- `console.time/timeEnd, performance.now()`
- `$0` — виділений елемент, `copy(obj)` — у буфер

```
console.group('Аналіз');  
console.table(data);  
console.time('op'); expensiveOp(); console.timeEnd('op');  
console.groupEnd();
```

Sources: Snippets, breakpoints, debugger

- Snippets — збережені фрагменти JS
- Умовні/XHR/DOM breakpoints
- Ключове слово `debugger` для зупинки з коду

```
function calc(){ debugger; /* ... */ }
```

Network & Performance

- Network: таймінги, заголовки, кеш, throttling (повільний 3G)
- Performance: трейс, FPS, long tasks, CPU throttling
- Coverage: непотрібні байти CSS/JS

Memory & Application

- Memory: heap snapshot, allocations, пошук витоків
- Application: Storage/IndexedDB, кеші, Service Workers
- WeakMap/WeakSet у дизайні для уникнення витоків

Міні-демо: «Валідатор подій» у DevTools

- Вхід: масив рядків "YYYY-MM-DD: Назва"
- Кроки: RegExp → розбір → перевірка дати → console.table

```
const events = [  
  '2025-09-21: Лекція',  
  '2025-09-31: Консультація' // невалідна дата  
];  
const pat = /^(\\d{4})-(\\d{2})-(\\d{2}):\\s*(.+)$/;  
const rows = events.map(str=>{  
  const m = str.match(pat); if(!m) return {raw:str, valid:false};  
  const [, y, mo, d, title] = m; const dt = new Date(+y, +mo-1, +d);  
  const valid = dt.getFullYear()==+y && dt.getMonth()==+mo-1 && dt.getDate()==+d;  
  return { date:`${y}-${mo}-${d}`, title, valid };  
});  
console.table(rows);
```

Підсумки

- Дати: Intl для форматування, нормалізація для арифметики
- RegExp: починаємо просто, групи/флаги, поступова перевірка
- DevTools: Console, Sources, Network/Performance/Memory для діагностики
- Домашка: міні-дашборд із кнопками Run/Clear і табличним виводом