# WPROWADZENIE DO TECHNOLOGII WEBOWYCH

Tematem nadrzędnym kursu jest pojęcie **komponowalności** (composibility).

**Kurs od zera** - nie zakłada wcześniej wiedzy w zakresie programowania skryptowego. Dwoma przewodnimi wątkami kursu są **nauka języka TypeScript oraz narzędzi deweloperskich w przeglądarce.**

Pierwszy wątek - poza nauką konkretnego języka - koncentruje się na przybliżeniu zagadnień natury ogólnej i fundamentalnej:

- danych,
- operacji na danych
- oraz asynchroniczności.

Będziemy przyswajać możliwie dużo koncepcji i dobrych praktyk programistycznych. Drugi wątek zaznajamia uczniów **z wewnętrznym życiem przeglądarki internetowej.** Będziemy dociekać co się dzieje podczas wyświetlania stron i aplikacji webowych. Poznamy:

- anatomię dokumentów HTML,
- różnice między statycznym a dynamicznym kontentem
- oraz podstawowe pojęcie dotyczące bezpieczeństwa i prywatności.

Pierwszy wątek wykorzystuje podejście od szczegółu do ogółu; drugi zaś od ogółu do szczegółu.

Pierwszy wątek jest abstrakcyjny, drugi konkretny. Pomimo to, oba wątki się przeplatają ze sobą.

**Po ukończeniu kursu, uczestnik:**

- zna podstawy języka TypeScript,
- jest zaznajomiony z podstawowymi pojęciami z zakresu technologii web takimi jak: HTML/DOM, CSS, JavaScript, URL, HTTP, adres IP,
- potrafi korzystać z narzędzi programistycznych dostępnych w przeglądarce,
- jest w stanie tworzyć proste aplikacje interaktywne.

kodii

# TEMATY SPOTKAŃ:

1. **Wprowadzenie.** Formalności. Poznajemy się. Otwieramy narzędzia deweloperskie chrome.
2. **Od "zera".** Typ prosty: liczba. Operacje arytmetyczne. Typ prosty: bool (0/1). Narzędzie programistyczne: konsola JavaScript.
3. **Przez "jedność".** Typ prosty: string (tekst). Typ złożony: mapa. Kształt danych. Narzędzie programistyczne: inspekcja HTML/DOM/CSS.
4. **Po "mnogość".** Typy złożone: array (tablica), tuple (krotka). Cykl życia dokumentu HTML wyświetlanego w przeglądarce.
5. **Jeszcze raz.** Formalizujemy pojęcie funkcji. Dane a funkcje. Funkcja matematyczna. Kontekst wykonywania kodu JavaScript.
6. **Oddzwonimy.** Callback. Asynchroniczność. Żądania HTTP. Narzędzie developerskie: sieć.
7. **W DOM-u.** Programistyczne tworzenie i wyświetlanie elementów HTML. Podstawy layout-u. Zaczynamy projekt kalkulator.
8. **Wracamy do punktu wyjścia.** Zarządzanie stanem i zmianą. Elementy interaktywne. Kontynuujemy projekt kalkulator.
9. **Stylowo.** Formalizujemy CSS oraz stylowanie elementów HTML. Kończymy projekt kalkulator.
10. **To zależy.** Wstęp do programowania reaktywnego albo... tematyka wybrana przez uczestników.

**Program w trakcie kursu może ulec niewielkim zmianom wynikającym z dopasowania zajęć do tempa pracy i umiejętności uczestników.**

kodii