

Kod szkolenia: **CSS/STYLE**

Tytuł szkolenia: **Zaawansowane style CSS i postprocesory CSS w praktyce**

Dni: 2

## Opis:

### Adresaci szkolenia

Szkolenie jest adresowane do osób uczących się tworzenia stron internetowych, posiadających już podstawową wiedzę w tym zakresie oraz do developerów chcących pogłębić wiedzę o nowości w zakresie CSS.

### Cel szkolenia

Celem szkolenia jest dogłębne poznanie CSSa, umożliwiające tworzenie rozbudowanych, estetycznych layoutów stron i aplikacji internetowych w technologii RWD.

### Mocne strony szkolenia

Szkolenie kompleksowo przygotowuje do pracy z CSSem z wykorzystaniem najnowszych standardów i trendów.

### Wymagania

- znajomość podstaw HTML lub wybranego systemu CMS z edytorem HTML
- znajomość podstaw CSS

### Program szkolenia:

#### 1. (Nieoczywiste) podstawy CSS

- selektory
- grupowanie selektorów
- !important
- kolory
- jednostki
- elementy następne (+, ~)
- stylowanie w zależności od atrybutów np. [type="text"]
- priorytety reguł
- pseudo-selektory (nth-child, first-child, odd, even, last-child, first-of-type, etc.)
- pseudo-elementy i elementy formularzy oraz obrazki
- @import css i jego wady

- stylowanie kursora
- zapisy skrócone
- 2. Stylowanie formularzy
  - domyślne style (webkit-appearance\*, border, input: text, number, range)
  - stosowanie klas do określania stanu elementu
  - stylowanie pseudo-elementów wbudowanej walidacji formularzy
  - zagnieżdżanie elementów celem zwiększenia możliwości stylowania
  - stylowanie ostrzeżeń walidacji natywnej w HTML
- 3. Canvas
  - domyślne rozmiary
  - CSS a Canvas
- 4. Multimedia w CSS
  - osadzanie video
  - nakładanie efektów na obraz dzięki blend-mode
- 5. Animacje
  - transform
  - transition
  - definiowanie klatek kluczonych
  - biblioteki do animacji
  - animacje poprzez nadawanie i zdejmowanie klas
- 6. Dodatkowe efekty graficzne
  - mieszanie warstw CSS
  - Clipping mask (Chrome)
  - SCG clipping mask
  - CSS reflex
- 7. RWD i Media Queries
  - ogólne zasady RWD
  - Automatyzacja RWD z wykorzystaniem SCSS
  - RWD tabel (przewijanie wewnętrzne, ukrywanie kolumn)
  - wykrywanie orientacji ekranu (portrait i landscape)
  - rozpoznawanie touchable i clickable device
- 8. "Mobile" CSS
  - CSS, a usability
  - Wzorce do stosowania na urządzeniach mobilnych
  - ViewPort
  - Obrazki na wyświetlaczach Retina
  - RWD
- 9. Style pod druk
  - wycinanie kolorów, tła
  - ustawianie tekstu
  - definicje kartek papieru
- 10. Braki w CSSie
  - jeśli element jest większy niż 500px to...
  - brak informacji o strukturze HTML na poziomie definiowania reguł CSS
  - brak możliwości "chwytania" rodziców i elementów poprzednich
  - sam HTML określa przez strukturę wygląd



- wsparcie standardu\* CSS w różnych przeglądarkach
  - problemy z urządzeniami mobilnymi
  - “Webkit” to nie cała przeglądarka
11. Usability
    - obszary aktywne większe niż się wydaje
    - szybkie kliknięcia i CSS
  12. Wydajność CSSa
    - rendering
    - operacje na DOM
    - zasobożerność efektów graficznych
    - recalculate layout, repaint, composition
  13. CSS dla osób niewidomych
    - podkład muzyczny podczas odczytywania tekstu
    - określenie balansu przy odtwarzaniu dźwięku
  14. Dobre praktyki
    - wersjonowanie GIT
    - utrzymanie dużych projektów CSS - Sposoby dzielenia plików
    - konwencje nazewnicze
    - definicje bez efektów ubocznych
  15. Preprocesory CSS
    - LESS vs SASS / SCSS vs Stylus
    - parsowanie SCSS - NetBeans, Compass, node.js SASS
    - formaty wyjściowe
    - minifikacja
    - tworzenie plików map
  16. SCSS - podstawowe pojęcia
    - zmienne
    - operatory
    - zagnieżdżanie
    - dziedziczenie / rozszerzanie klas
  17. Instrukcje sterujące
    - @import
    - pętle
    - instrukcje warunkowe
  18. SCSS - tworzenie mixinów
    - parametry wejściowe
    - wyłuskiwanie wartości zmiennych
    - mixiny łączące wiele funkcjonalności
    - @content
  19. SCSS - Korzystanie z funkcji
    - funkcje wbudowane
    - pisanie własnych funkcji
    - funkcja czy mixin
  20. SCSS - zagadnienia dodatkowe
    - klasy abstrakcyjne / placeholders
    - tworzenie rozbudowanych mixinów i stosowanie ich



- @extend vs @include
- pułapki i ich obchodzenie
- @debug
- dobre praktyki

