

Programowanie aplikacji w języku JavaScript

kryteria oceny

Plan wykład

- 1: Wprowadzenie do przedmiotu. HTML5 i CSS3.
- 2-4: Język JavaScript (wprowadzenie, typy danych, zmienne, instrukcje sterujące, funkcje, tablice, obiekty, zagadnienia zaawansowane).
- 5-6: Zdarzenia, manipulacja zawartością strony, DOM.
- 7-8: Node.js, REST, Ajax, Web APIs.
- 9-12: Biblioteki i frameworki do tworzenia aplikacji bogatego klienta (Vue).
- 13-14: Grafika, animacje, biblioteki do tworzenia gier.
- 15: Zaliczenie wykładu.

Forma zaliczenia wykładu

- Test pisemny (w formie quizu) w terminie ostatniego wykładu.
- Osoby, które uzyskały pozytywną ocenę z PS mogą zdecydować się na przepisanie oceny z górnym ograniczeniem do 4.0.
- Pisanie testu nie anuluje oferty przepisania oceny (nie wiąże się z ryzykiem).
- Aby skorzystać z przepisania, nie ma konieczności uzyskania wpisu przed ostatnim wykładem (uzupełnione oceny będą później przepisane z USOS).

Plan PS

- 1-5: niepunktowane ćwiczenia wykonywane w trakcie zajęć, dotyczące tematów omawianych na poprzednim wykładzie
- 6-8: mini-projekt wymagający bezpośrednio wykorzystania wiedzy z dotychczasowych wykładów
- 9-15: większe zadanie projektowe polegające na zaprojektowaniu i zaimplementowaniu aplikacji webowej
- 15: zaliczenie (ocena)

Zasady oceniania PS

- 25% - mini-projekt,
- 75% - projekt:
 - projekty realizowane są w zespołach 2-3 osobowych,
 - szczegółowy kryteria oceny zostaną podane wraz z tematami do wyboru.

Skala ocen

- od 51% - dst

- od 61% - dst+
- od 71% - db
- od 81% - db+
- od 91% - bdb

Odniesienia do efektów uczenia:

- EU1 jest weryfikowany na podstawie testu zamkniętego (zaliczenie wykładu) – wymagane jest zdobycie przynajmniej 50% punktów
- EU2 jest weryfikowany na podstawie mini-projektu – wymagane jest zdobycie przynajmniej 5%
- EU3 jest weryfikowany na podstawie realizacji projektu zespołowego - wymagane jest stworzenie aplikacji webowej typu „bogatego klienta” działającej w przeglądarce
- EU4 jest weryfikowany na podstawie realizacji projektu zespołowego - wymagane jest, by stworzona aplikacja komunikowała się z przygotowanym backendem