

WPROWADZENIE DO PRZEDMIOTU

APLIKACJE INTERNETOWE

Prowadząca: Karina Czarkowska

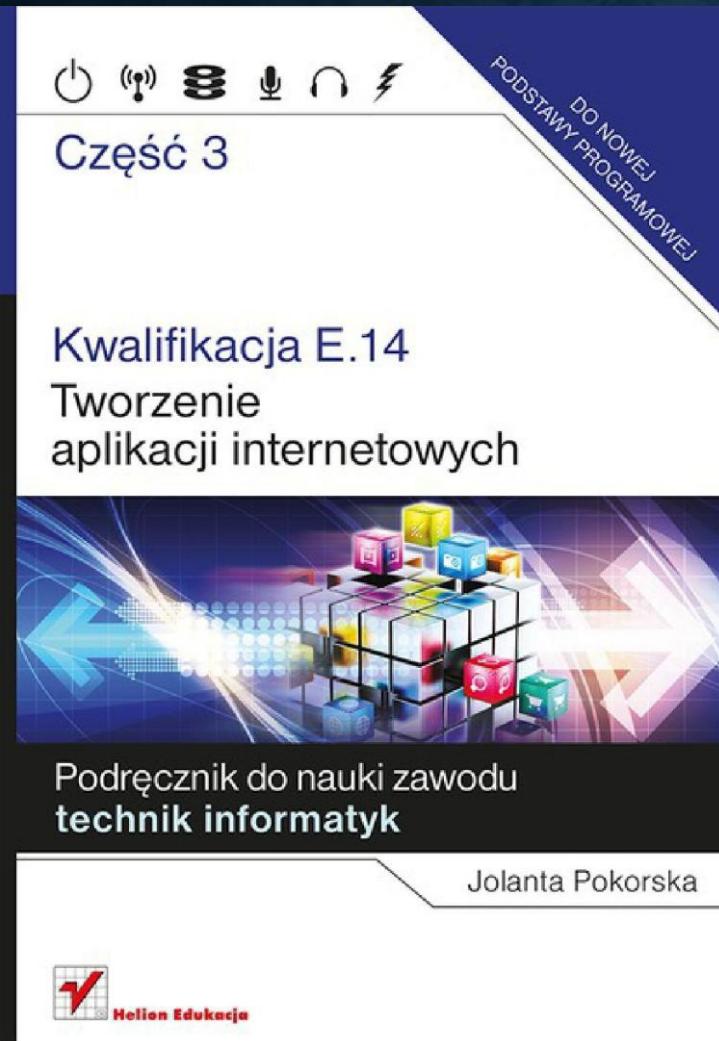
ZAKRES MATERIAŁU WYKŁADU

- 1. Plan nauczania przedmiotów**
- 2. Podręcznik do przedmiotu i literatura
uzupełniająca**
- 3. Wymagania dot. zaliczenia przedmiotu**
- 4. Podstawowe pojęcia**
- 5. Ważne skróty i ich rozwinięcia**

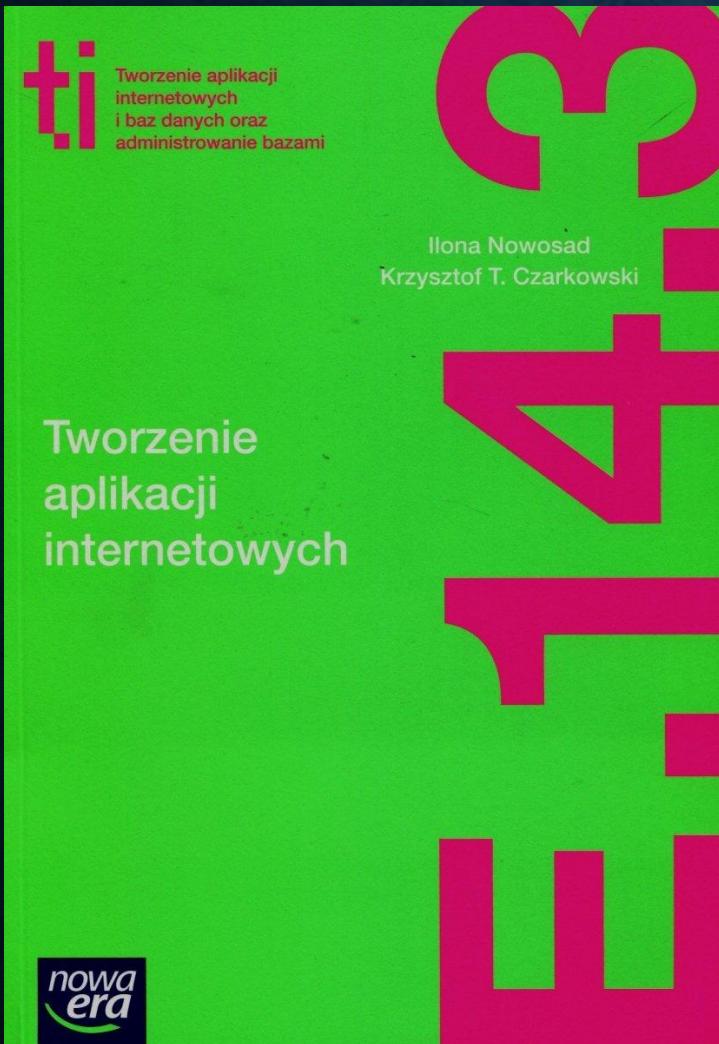
PLAN NAUCZANIA

	Klasa 2 Semestr II	Klasa 3 Semestr I	Klasa 3 Semestr II	Klasa 4 Semestr I	Ilość godzin
E.14.1 – Strony internetowe	2	2	2	---	180
E.14.2 – Bazy danych	---	2	2	1	150
E.14.3 – Aplikacje internetowe	---	---	2	3	150
EGZAMIN w II semestrze klasy 4 – styczeń/luty:					480

PODRĘCZNIK



Podstawy programowania
Aplikacje internetowe
Skrypty po stronie klienta – JavaScript
Biblioteka jQuery
Ajax
Serwery www
Język PHP
Technologia .NET
Systemy zarządzania treścią
Testowanie i dokumentowanie aplikacji



LITERATURA UZUPEŁNIAJĄCA

Podstawowe narzędzia programowania

Podstawy programowania

Programowanie ogólne

Architektura aplikacji internetowych

Aplikacja po stronie klienta JavaScript

Baza danych – aplikacja po stronie serwera

Aplikacja po stronie serwera

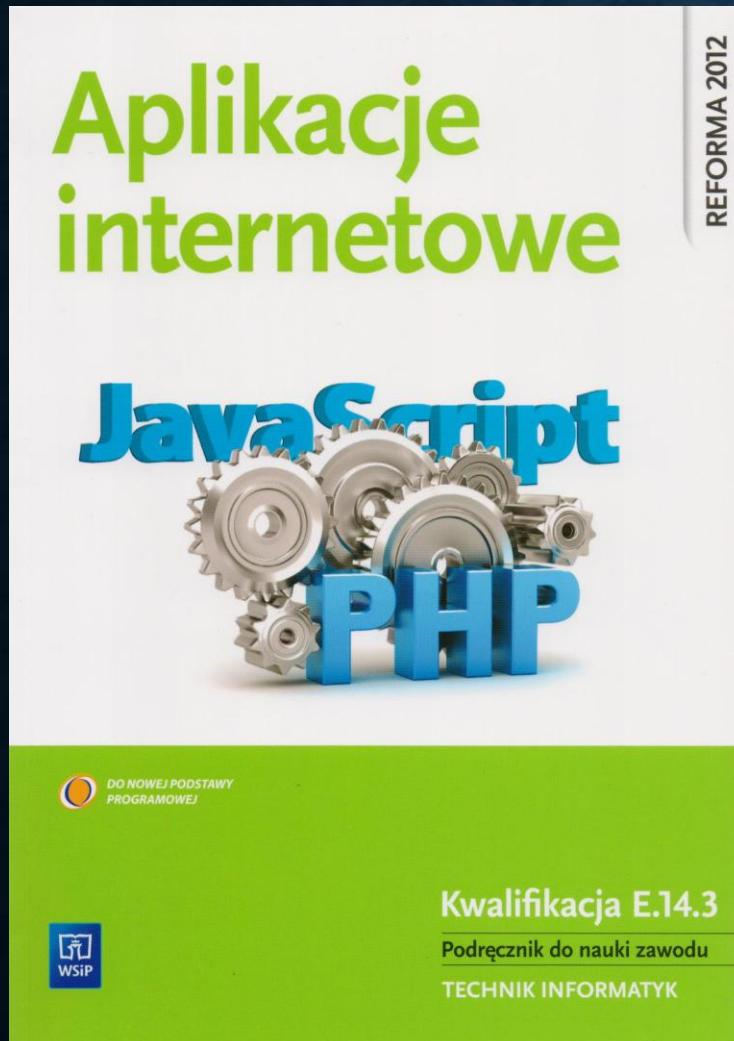
Technologia Ajax

Aplikacja po stronie serwera – PHP

Dokumentacja, testowanie i publikacja aplikacji internetowych

Framework Hibernate

LITERATURA UZUPEŁNIAJĄCA



JavaScript – skryptowy język programowania
PHP – obiektowy, skryptowy język programowania



LITERATURA UZUPEŁNIAJĄCA

Tworzenie stron internetowych

Tworzenie baz danych i administrowanie bazami danych

Tworzenie aplikacji internetowych

LITERATURA UZUPEŁNIAJĄCA

REPETYTORIUM
+TESTY
EGZAMIN ZAWODOWY



KWALIFIKACJA **E.14**



TECHNIK INFORMATYK

**Zestaw testów teoretycznych i zadań praktycznych
wraz z przykładowymi rozwiązaniami i kluczami
oceniania.**

ZALICZENIE PRZEDMIOTU

- **Frekwencja i aktywny udział w zajęciach**
- **Praca na lekcji, wykonywanie ćwiczeń praktycznych, zadań domowych i zaliczeniowych**
- **Wykonanie projektu witryny internetowej z wykorzystaniem skryptów JavaScript**
- **Udział w konkursach i olimpiadach o tematyce związanej z przedmiotem**

PODSTAWOWE POJĘCIA

Program jest to zbiór poleceń zapisanych w określonym języku programowania zgodnie z obowiązującymi w tym języku regułami.

Programowanie to proces tworzenia i testowania programu. Kod źródłowy programu jest napisany w języku programowania z użyciem określonych reguł. Przykładami używanych obecnie języków programowania są: C, C++, Java, Delphi, PHP, Perl, Python, Ruby.

PHP, Java – są popularne w programach korzystających z baz danych

Python – bywa wykorzystywany w administrowaniu systemami i na stronach www.

PODSTAWOWE POJĘCIA

Słowa kluczowe to zarezerwowane słowa, które w danym języku programowania mają ścisłe określone znaczenie i mogą zostać użyte tylko zgodnie z ich przeznaczeniem. Słowami kluczowymi są między innymi nazwy poleceń.

Kod źródłowy to ciąg instrukcji i deklaracji zapisany w języku programowania. Opisuje operacje, jakie powinien wykonać komputer. Kod źródłowy programu jest zapisywany w postaci tekstu, który przetwarzany jest za pomocą **kompilatora** na kod maszynowy lub kod pośredni. Możliwe jest również wykonywanie kodu źródłowego w locie za pomocą programu zwanego **interpretorem**.

PODSTAWOWE POJĘCIA

Translator to program służący do tłumaczenia programu zapisanego w języku programowania z postaci źródłowej do postaci wynikowej. Translatory dzielimy na: **kompilatory** tłumaczące programy zapisane w językach wysokiego poziomu oraz **asemblery** tłumaczące programy zapisane w językach symbolicznych.

Kompilator to program służący do tłumaczenia kodu napisanego w języku źródłowym na odpowiadający mu kod w języku wynikowym. Najczęściej jest to program do tłumaczenia kodu źródłowego napisanego w wybranym języku programowania na kod maszynowy.

Interpreter to program, który analizuje kod źródłowy instrukcja po instrukcji i każdy przeanalizowany fragment kodu wykonuje na bieżąco.

PODSTAWOWE POJĘCIA

Moduł to wydzielony fragment programu komputerowego. Podzielenie dużych programów na moduły ułatwia pracę z programem, szczególnie gdy każdą część programu opracowuje inny programista. Każdy moduł jest komplikowany osobno.

Linker (konsolidator) to narzędzie służące do łączenia przekompilowanych modułów w jeden plik wykonywalny. Odpowiada on za poprawne połączenie modułów ze sobą. Dobry linker powinien umożliwiać tworzenie plików wykonywalnych o różnych formatach i dla różnych systemów.

Konsolidacja to proces polegający na połączeniu skomplikowanych modułów i utworzeniu pliku wynikowego. Podczas konsolidacji do pliku wynikowego mogą być dołączone biblioteki i inne informacje, dotyczące np. formatu pliku wykonywalnego.

PODSTAWOWE POJĘCIA

Aplikacja to program użytkowy, wykonujący konkretne zadania i oferujący interfejs użytkownika. Określenie program komputerowy jest często stosowane zamiennie z określeniem aplikacja. Podstawową różnicą między tymi pojęciami jest to, że program komputerowy nie musi oferować żadnego interfejsu użytkownika.

Aplikacja internetowa (ang. web application) lub aplikacja webowa to program komputerowy, który pracuje na serwerze i komunikuje się z użytkownikiem poprzez sieć komputerową z wykorzystaniem przeglądarki internetowej. Niektóre aplikacje webowe mogą również działać offline, wykorzystując wszystkie swoje zalety.

PODSTAWOWE POJĘCIA

Aplet jest to niewielki program komputerowy, którego wykonanie jest możliwe tylko z poziomu innej aplikacji. Aplety najczęściej są pisane w języku Java lub jako komponenty Active-X i są wykorzystywane na stronach internetowych.

Ich zadaniem może być wykonywanie prostych czynności, np. uruchomienie aplikacji, przeprowadzenie obliczeń. W zaawansowanych zastosowaniach mogą być stosowane, np. jako przeglądarkowe wersje demo w przypadku gier pisanych w Javie lub do podpisywania przelewów w bankowości internetowej.

WAŻNE SKRÓTY I ICH ROZWINIĘCIA

JS (ang. JavaScript) – skryptowy język programowania, pozwalający na rozszerzenie dokumentu HTML i wprowadzenie interakcji z użytkownikiem, działa po stronie przeglądarki.

JSP (ang. JavaServer Pages) – wchodzi w skład platformy Java Platform, umożliwia tworzenie dynamicznych dokumentów www w formatach: HTML, XHTML, DHTML ORAZ XML.

PHP (ang. Personal Home Page) - interpretowany skryptowy język programowania, zaprojektowany do generowania stron internetowych i budowania aplikacji webowych w czasie rzeczywistym, działa po stronie serwera, jest wykorzystywany również przy pracy nad aplikacjami uruchamianymi niezależnie od serwera www.s

AJAX (ang. Asynchronous JavaScript and XML) – to technologia uzupełniająca język JavaScript, działa po stronie klienta, ale w tle komunikuje się z serwerem.

WAŻNE SKRÓTY I ICH ROZWINIĘCIA

CGI (ang. Common Gateway Interface) – pierwsza technologia działająca po stronie serwera, umożliwiająca m.in. dynamiczne generowanie dokumentów przed wysłaniem ich do przeglądarki.

ISAPI (ang. Internet Server API) – technologia filtrów, których działanie jest sterowane zdarzeniami serwera, a nie żądaniami klienta, działa w systemach Windows.

ASP.NET – wydajna i nowoczesna technologia opracowana przez firmę Microsoft. Przeznaczona została do tworzenia różnych aplikacji internetowych w formatach HTML, WML, XML.

WAŻNE SKRÓTY I ICH ROZWINIĘCIA

XAMPP - darmowy, wieloplatformowy, zintegrowany pakiet, składający się głównie z serwera Apache, bazy danych MySQL i interpreterów dla skryptów napisanych w PHP i Perlu. Nazwa XAMPP jest akronimem od **X** (ang. cross-platform), **Apache**, **MySQL**, **PHP**, **Perl**.

HTTP – (ang. Hypertext Transfer Protocol) protokół transportowy hipertekstu.

FTP – (ang. File Transfer Protocol) protokół transferu plików na serwer.

Obecnie popularne technologie to: **ASP.NET**, **JSP** i **PHP**, generują strony www poprzez interpretowanie kodu osadzonego wewnątrz strony.

STRUKTURA JĘZYKA JAVASCRIPT

Prowadząca: Karina Czarkowska

ZAKRES MATERIAŁU WYKŁADU

- 1. Umieszczanie skryptów w dokumencie**
- 2. Komentarze w skryptach**
- 3. Instrukcja `dokument.write`**
- 4. Okna dialogowe**

UMIESZCZANIE SKRYPTÓW W DOKUMENCIE

Pierwszym sposobem na umieszczenie skryptu w dokumencie HTML jest wprowadzenie go bezpośrednio do kodu przez wykorzystanie znacznika **<script> ... </script>** (plik osadzony).

Można go wprowadzić w nagłówku **head** lub w głównej części dokumentu **body**.

```
<script type="text/javascript">
```

kod skryptu

```
</script>
```

UMIESZCZANIE SKRYPTÓW W DOKUMENCIE

Drugim sposobem jest umieszczenie skryptu w osobnym pliku zewnętrznym, który może mieć dowolną nazwę oraz charakterystyczne rozszerzenie ***.js**

Plik zewnętrzny zostaje powiązany z dokumentem HTML za pomocą znacznika **script** z dodatkowym oznaczeniem lokalizacji i nazwy pliku wraz z rozszerzeniem.

```
<script type="text/javascript" src="plik.js">  
</script>
```

UMIESZCZANIE SKRYPTÓW W DOKUMENCIE

W jednym dokumencie można umieścić kilka skryptów, zarówno osadzonych , jak i zewnętrznych.

```
<head>
```

```
...
```

```
</head>
```

```
<body>
```

```
...
```

```
<body>
```

KOMENTARZE W SKRYPTACH

W skryptach można umieszczać komentarze:

- **Jednowierszowe** (liniowe) – rozpoczynają się od znaków **//** i działają do końca danej linii skryptu;
- **Wielowierszowe** – rozpoczynają się od znaków **/*** i kończą się ***/**. Można je wykorzystać np. do czasowego wyłączenia fragmentu kodu podczas wyszukiwania błędów.

Komentarze są ignorowane przez interpreter i użytkownik nie widzi ich na stronie.

KOMENTARZE W SKRYPTACH

Przykład: Komentarz jednoliniowy

```
1 // To jest komentarz jednoliniowy.  
2 var intAge = 37; //To jest inny komentarz
```

Przykład: Komentarz wieloliniowy

```
1 /* To jest komentarz wieloliniowy.  
2  Może być dowolnie długi i zawierać dowolne znaki,  
3  za wyjątkiem innych komentarzy wieloliniowych. */
```

INSTRUKCJA DOKUMENT.WRITE

Składnia instrukcji w języku JavaScript jest zapisywana zgodnie z ogólną zasadą: **obiekt.metoda (argumenty metody)**.

Obiektem może być dokument reprezentujący aktualną stronę. Metoda określa funkcję, która ma zostać wykonana. Na końcu instrukcji umieszcza się średnik.

Do wyświetlenia na aktualnej stronie dowolnego tekstu służy instrukcja **dokument.write**. Argumenty instrukcji umieszcza się w nawiasie okrągły.

INSTRUKCJA DOKUMENT.WRITE

Jeżeli argument jest tekstem, dodatkowo umieszcza się go w cudzysłowie; jeżeli jest liczbą, podaje się ją bez cudzysłowu, np.:

dokument.write(„Witaj”);
dokument.write(123);

Do połączenia wielu argumentów funkcji **write** w jeden łańcuch znaków służy symbol „+”. Za jego pomocą można łączyć teksty i liczby, np. **dokument.write("Mam „+18” lat!")**. Wyświetlana zawartość jest traktowana jak kod języka HTML. Aby rozpocząć pisanie od nowej linii, należy w argumencie wprowadzić znacznik **
**.

INSTRUKCJA DOKUMENT.WRITE

Za pomocą zagnieżdżonych znaczników HTML można również wprowadzić dowolne sposoby formatowania tekstu, a także umieszczać na stronie inne obiekty, np. graficzne. W tym celu trzeba jako argument funkcji **dokument.write** użyć znacznika wykorzystywanego do wstawiania grafiki w HTML, np.:

**dokument.write("
Ala ma kota")**

OKNA DIALOGOWE

Okna dialogowe są narzędziem pozwalającym na nawiązywanie interakcji z użytkownikiem. JavaScript umożliwia wykorzystywanie trzech rodzajów okien dialogowych:

- **Okno informacyjne,**
- **Okno decyzyjne,**
- **Okno tekstowe.**

OKNA DIALOGOWE - OKNO INFORMACYJNE

Jego zadaniem jest przekazanie określonej informacji. Nie ma ono wpływu na dalsze działanie skryptu. Okno wyświetla tekst określony jako argument metody **alert** i ma jeden przycisk **OK**, powodujący zamknięcie okna.

```
<script type="text/javascript">  
    alert("Uwaga! Okno dialogowe!")  
</script>
```

OKNA DIALOGOWE - OKNO DECYZYJNE

Odpowiada za wyświetlanie treści komunikatu stanowiącego argument metody **confirm**. Udostępnia dwa przyciski **OK** oraz **Anuluj**, które po wciśnięciu zwracają wartość logiczną **true** lub **false**.

```
<script type="text/javascript">  
    confirm("Czy chcesz przejść do strony www?")  
</script>
```

Ponieważ skrypt nie ma żadnej funkcji podpiętej do okna decyzyjnego, wciśnięcie dowolnego klawisza nie wywoła żadnej reakcji.

OKNA DIALOGOWE - OKNO TEKSTOWE

Wyświetla treść komunikatu stanowiącego argument metody **prompt** oraz pole umożliwiające wprowadzenie danych przez użytkownika. W trakcie wywoływania okna tekstowego w polu może pojawić się tekst domyślny.

```
<script type="text/javascript">  
document.write("JavaScript to ");  
document.write(prompt("JavaScript to:", "skryptowy  
język programowania"));  
</script>
```

OKNA DIALOGOWE - OKNO TEKSTOWE

Dzięki zagnieżdżeniu metody **prompt** wewnątrz instrukcji **dokument.write** informacje wpisane w polu okna tekstowego, po zatwierdzeniu przyciskiem **OK**, zostaną wyświetcone w oknie przeglądarki.