

# JavaScript

**Witryny i aplikacje internetowe, cz-01**

*mgr. Marzena Parowińska*

# Spis treści

1. [Aplikacja internetowa](#)
2. [JavaScript - definicja](#)
3. [JavaScript – składnia języka](#)
4. [Ćwiczenia JavaScript](#)

# 1. Aplikacja internetowa

Wprowadzenie

# Aplikacja internetowa



- **Aplikacja internetowa** to program, który pracuje na serwerze i komunikuje się z użytkownikiem przez sieć komputerową z wykorzystaniem przeglądarki internetowej. Przeglądarka w tym wypadku pełni funkcję interfejsu użytkownika.
- Istotnymi cechami aplikacji internetowej są łatwość i szybkość dotarcia do informacji (komputer podłączony do Internetu lub sieci LAN i przeglądarka stron internetowych) oraz bezpieczeństwo danych (login i hasło, szyfrowanie bez połączenia oraz system uprawnień)

# Zalety aplikacji internetowych

Zalety aplikacji internetowych:

- Dostępność do wszystkich zainteresowanych bez ograniczeń związanych z czasem i miejscem,
- Praktycznie brak konieczności instalowania dodatkowego oprogramowania, aby korzystać z aplikacji internetowej, wystarczy przeglądarka,
- Brak konieczności zakupu dodatkowego sprzętu komputerowego; aplikacja może zostać zainstalowana na serwerze dostępnym w internecie,

# Zalety aplikacji internetowych cd

- Łatwość utrzymania i modernizacji, ponieważ zmiany i aktualizacje w aplikacji są wykonywane na serwerze bez udziału użytkowników aplikacji,
- Łatwość integracji aplikacji internetowej z innymi usługami,
- Niższe koszty wytwarzania, uruchomienia i utrzymania aplikacji internetowej w porównaniu z innymi rozwiązaniami, większość technologii stosowanych do tworzenia aplikacji internetowych jest bezpłatna.

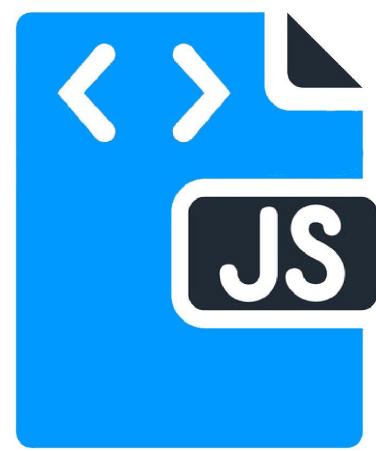
# Aplikacja internetowa

- Do prawidłowej pracy aplikacji internetowej potrzebne jest oprogramowanie zwane serwerem WWW (np. Apache, IIS) oraz oprogramowanie wspomagające (serwer bazodanowy np. MySQL, MS SQL) i środowisko do tworzenia programów (np. PHP, Java, C#)
- Z serwerem WWW komunikuje się przeglądarka internetowa, a serwer bazodanowy i interpreter umożliwiają udostępnianie dynamicznie tworzonych stron internetowych oraz zarządzanie bazami danych.

# Narzędzia aplikacji internetowych

- ***Narzędzia po stronie klienta*** - pozwalają klientowi wykonywać część zadań związanych z obsługą aplikacji. Należą do nich **JavaScript**, aplety Java, **AJAX** i inne.
- ***Narzędzia po stronie serwera*** – pozwalają wykonywać na serwerze zadania służące do obsługi żądań klienta, są to m.in. takie technologie jak technologie **ASP.NET (Active Server Pages)**, **JavaServer Pages (JSP)** i języki **PHP**, **Perl** i inne.

# Technologie tworzenia stron internetowych



# Technologie po stronie klienta -1

- Podstawowa technologią stosowaną po stronie klienta jest **JavaScript**. Jest to język skryptowy, którego kod można wstawić w kod dokumentu HTML w celu uzyskania dynamiki strony, umożliwiając interakcji z użytkownikiem oraz sprawdzanie poprawności informacji wprowadzanych przez użytkowników.
- JavaScript jest obecnie wspierany przez wszystkie przeglądarki i nie wymaga dodatkowego oprogramowania.
  - Język skryptowy
  - Nie wymaga kompilowania przed uruchomieniem
  - Wykonywany przez program zwany interpreterem/parserem
  - Parser jest wbudowany w większość przeglądarek

# Technologie po stronie klienta -2

- Inna technologią, która może być stosowana po stronie klienta są aplikacje Java, a do ich wykonania potrzebne jest zainstalowanie w przeglądarce wtyczki Java Virtual Machine. Aplikacje Java są coraz rzadziej stosowane.
- Technologia **Flash** to narzędzie które może być stosowane po stronie klienta. Dostarcza ono narzędzi do tworzenia animacji i efektów graficznych. Technologia Flash wymaga instalowania w przeglądarce dodatkowej wtyczki Adobe Flash Player.
- **AJAX** (ang. *Asynchronous JavaScript and XML*) to technologia uzupełniająca język JavaScript. Oferuje ona te same działania po stronie klienta co JavaScript, ale w technologii AJAX komunikacja z serwerem jest realizowana w tle.

# Technologie po stronie serwera -1

Obecnie popularne są technologie, które generują strony WWW poprzez interpretowanie kodu osadzonego wewnątrz strony. Do technologii tego typu należą:

- **ASP.NET** – technologia (Microsoft). Przeznaczona została do tworzenia aplikacji w różnych formatach HTML, WML oraz XML. Tworzone dokumenty zawierają statyczną treść, ale w ramach statycznego kodu istnieje możliwość dodania kodu dynamicznego.
- **JSP (ang.*JavaServer Pages*)** wraz z innymi elementami wchodzącyymi w skład platformy Java Platform firmy Sun umożliwia tworzenie dynamicznych dokumentów WWW w formatach HTML, XHTML, DHTML oraz XML z wykorzystaniem języka Java.

# Technologie po stronie serwera -2



- PHP wraz z serwerem WWW Apache oraz z serwerem baz danych **MySQL** tworzy wydajną i niezależną od systemu platformę do budowania dynamicznych stron WWW. Platforma ta określana jest jako platforma AMP. W systemach Linux znana jest pod nazwą LAMP, a w systemie Windows pod nazwą WAMP. Istnieje również wieloplatformowa wersja **XAMPP**.
- PHP jest systemem otwartym, dzięki czemu powstają coraz nowsze moduły rozszerzające jego możliwości. Jest to również język skryptowy, wykorzystywany przy pracy nad aplikacjami uruchamianymi niezależnie od serwera WWW.

# JavaScript dynamiczne strony HTML

- Techniki służące do dynamicznej zmiany treści wyglądu lub zachowania się dokumentu HTML umożliwiają interakcję użytkownika ze stroną internetową. W skład dynamicznego HTML wchodzą technologie: DOM, CSS, HTML, SVG, języki skryptowe (JavaScript, Vbscript i inne)
- Języki skryptowe to języki interpretowane, zaprojektowane z myślą o interakcji z użytkownikiem. Skrypty wykonywane wewnętrz określonej aplikacji w odróżnieniu od programów nieskryptowych wykonują się niezależnie od innych aplikacji.

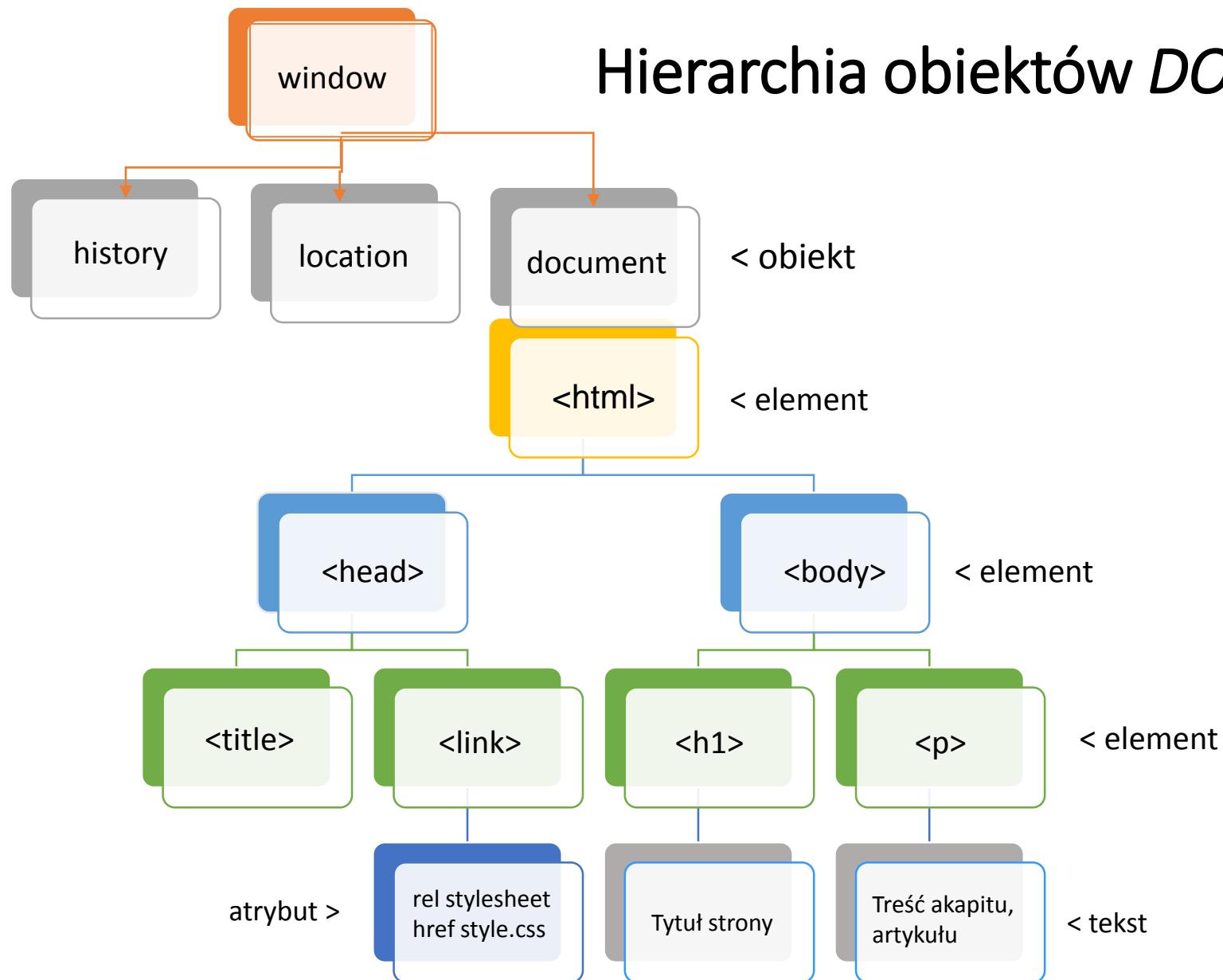
# JavaScript dynamiczne strony HTML

- **SVG** (ang. *Scalable Vector Graphics*) to uniwersalny format grafiki wektorowej (statycznej i animowanej) powstały z myślą o zastosowaniu na stronach internetowych. Może być integrowany z językami opartymi na technologii XML.
- Został on opracowany przez organizację W3C na potrzeby internetu. W SVG oprócz standardowych obiektów (prostokąty, elipsy, krzywe) można opisywać efekty specjalne, maski przezroczystości oraz sposób animacji elementów.

# DOM - *Document Object Model*

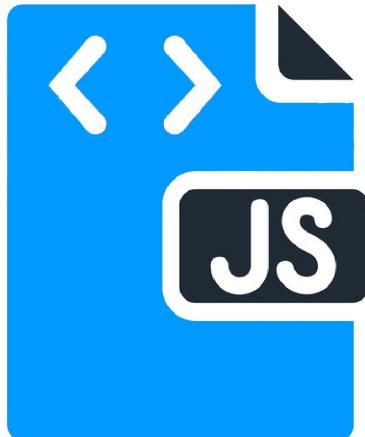
- **DOM** (ang. *Document Object Model*) to **obiektowy model dokumentu**, czyli sposób reprezentacji dokumentów XML i XHTML w postaci modelu obiektowego. Jest on niezależny od platformy i języka programowania. Standard zdefiniowany jako DOM przez organizację W3C opiera się na zespołach klas i interfejsów pozwalających na dostęp do struktury dokumentów oraz jej modyfikacje przez tworzenie, zmienianie tzw. węzłów (ang. *modes*)
- W modelu obiektowym DOM każdemu znacznikowi HTML, np. body, form, input, div, odpowiada **obiekt** (ang. *object*). Element HTML jest traktowany jako obiekt modelu DOM, może posiadać określone właściwości (ang. *properties*) oraz metody (ang. *methods*).

# Hierarchia obiektów DOM



# 2. JavaScript

# JavaScript - czym jest?



- **JavaScript** jest obiektowym skryptowym językiem pozwalającym na rozszerzenie standardowych dokumentów HTML m.in. o możliwość interakcji z użytkownikiem czy sprawdzanie poprawności danych wprowadzanych przez użytkowników. Jest językiem interpretowanym, co oznacza że efekty jego użycia można zobaczyć bez konieczności kompilowania kodu. Potrzebna jest tylko przeglądarka internetowa, która obsługuje język JavaScript.

# JavaScript

Tworzone skrypty zapewniają interaktywność strony internetowej przez reagowanie na zdarzenia, budowanie elementów nawigacji oraz sprawdzenie poprawności formularzy. Typowe zastosowania języka JavaScript:

- Dynamiczne modyfikowanie wyglądu bieżącego dokumentu,
- Wyświetlanie prostych okien dialogowych,
- Reagowanie na akcje podejmowane przez użytkownika (kliknięcie, wpisanie tekstu, ruch myszką)
- Kontrola poprawności wypełnienia formularza,
- Manipulowanie informacją dotyczącą daty i czasu,
- Lokalne generowanie dokumentów HTML.

# JavaScript, edytor

- Do tworzenia programu wystarczy zwykły **edytor tekstu** (np. Notatnik) lub dowolnie wybrane narzędzie wspomagające tworzenie stron WWW np. Notepad++ , Pajęczek, a także edytor Aptana Studio, czy Visual Studio Code.
- Wyniki działania programu obserwujemy w przeglądarce internetowej. Warto jednak pamiętać, żeby testować dany program w kilku dostępnych przeglądarkach (istnieją różnice interpretacji).

# 3. JavaScript

Składnia języka

# JavaScript - dołączanie kodu script I

- Pierwszym sposobem na umieszczenie skryptu w dokumencie HTML - ***skrypt osadzony*** - jest wprowadzenie go bezpośrednio do kodu przez wykorzystanie znacznika **<script> ... </script>** umieszczając go w nagłówku *head* lub w głównej części dokumentu *body*.  
**<script type=„text/javascript”> kod skryptu </script>**
- Parametr *type* określa rodzaj języka skryptowego. Nie jest on jednak wymagany. Większość współczesnych przeglądarek zaakceptuje sam znacznik *script*, warto jednak stosować zapis ***script*** w całości.

# JavaScript - dołączanie kodu script II

- Istnieje możliwość umieszczania skryptu w osobnym pliku – **skrypt zewnętrzny**. Może on mieć dowolna nazwę oraz rozszerzenie: **nazwa.js** plik ten 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>
```

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

# Znacznik noscript

```
<noscript>
```

```
  <p>
```

Twoja przeglądarka nie obsługuje JavaScriptu. Aby zobaczyć stronę w pełnej funkcjonalności, zainstaluj inną przeglądarkę lub włącz opcję wyświetlania JS w przeglądarce

```
  </p>
```

```
</noscript>
```

# JavaScript jako kalkulator

```
<html>
<head><title>kalkulator</title></head>
<body>
    <script type="text/javascript">
        document.write(3*(2+1));
        alert(3*(2+1));
    </script>
</body>
</html>
```

# Jak to działa?

## Jak to działa?

- przeglądarka próbuje wyświetlić stronę napisaną w HTML
- natyka na znacznik script
- zaczyna interpretować program zawarty w znaczniku script
- funkcja **document.write** – wpisuje argument do wyświetlonego okienka
- funkcja **alert** – pokazuje nowe okienko

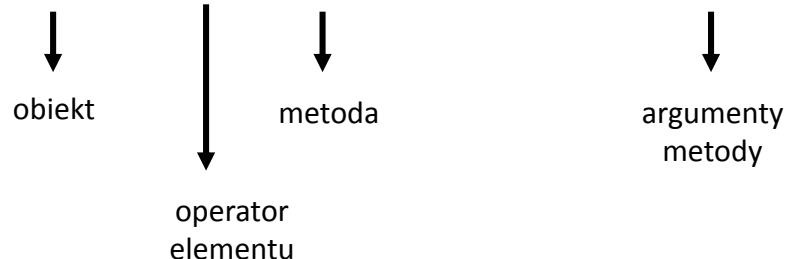
# Składnia języka JavaScript -1

- Składnia języka jest ścisłe określonym zbiorem reguł, których należy przestrzegać. W języku JavaScript stosuje się następujące kategorie pojęć składniowych:
  - **wyrażenia**
  - **zmienne**
  - **literały**
- Do elementów języka należy również zbiór predefiniowanych **obiektów i funkcji** (tablice, data, funkcje matematyczne)

# Składnia języka JavaScript - instrukcje

- **Instrukcje** w języku JavaScript mogą być pisane pojedynczo w jednym wierszu lub w kilku wierszach. Średnik odgrywa rolę ogranicznika instrukcji, ale nie jest wymagany jeśli po instrukcji nastąpi znak nowego wiersza. Jeżeli kilka instrukcji zostanie zapisanych w jednym wierszu, muszą być oddzielone średnikiem.

```
document.write("Pierwszy skrypt!");
```

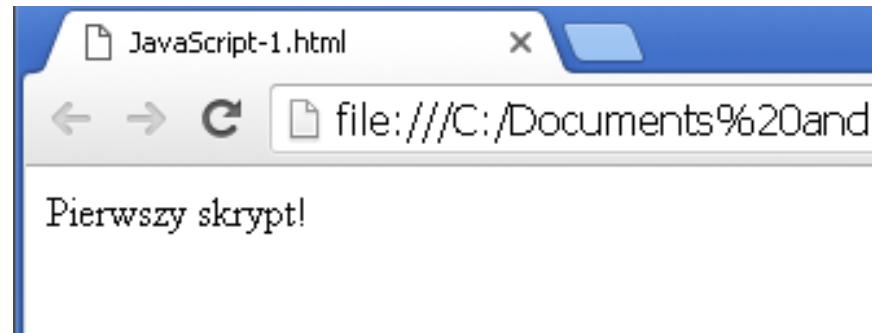


# Składnia języka JavaScript – instrukcje cd

Zapisem ogólny instrukcji:

**obiekt.metoda("argumenty metody");**

W instrukcji **document.write("treść wyświetlna")**; jak widać, **document** jest obiektem. Reprezentuje aktualną stronę. Metoda **write** to funkcja działająca na obiekcie document, której zadaniem jest wyświetlenie argumentów (tekstu, wartości liczbowych) w oknie przeglądarki.



# Składnia języka JavaScript - wielkość liter

**Wielkość liter** w programie JavaScript przeważnie ma znaczenie:

- Słowa kluczowe pisane są małymi literami np. **for**.
- Nazwy obiektów wbudowanych pisane są od dużej litery, pozostałe litery są małe np. **Date**.
- Nazwy obiektów DOM pisane są małymi literami, ale w nazwach metod tych obiektów są małe i duże litery, np. **toLowerCase**
- Definiowanym, zmiennym, obiektom, funkcjom można nadawać nazwy pisane w dowolny sposób. Oprócz liter w nazwach można stosować cyfry i znak podkreślenia.
- Białe znaki, typu spacja znak tabulacji itp. są nieistotne.

# Składnia języka JavaScript - komentarze

Komentarze są ignorowane w trakcie przetwarzania kodu.  
Mogą być umieszczane w dowolnym miejscu kodu.

- Komentarzem jest każdy wiersz rozpoczynający się od znaków **// tutaj jest komentarz**, można go także umieszczać po instrukcji np. `a = b + c // suma wartości`
- Jeżeli komentarz zawiera wiele wierszy można użyć ograniczników **/\*** (do rozpoczęcia komentarza) i **\*/** (do zakończenia komentarza).
- Tak zapisane komentarze dopuszczalne są tylko wewnątrz znaczników `<script> </script>`

# Składnia języka JS – okno dialogowe 1

Okno dialogowe jest narzędziem pozwalającym na nawiązanie interakcji z użytkownikiem. JavaScript umożliwia wykorzystanie trzech rodzajów okien dialogowych: informacyjnego, decyzyjnego i tekstowego.

- **Okno informacyjne** – wyświetla ono tekst określony jako argument metody **alert** i ma jeden przycisk **OK** powodujący zamknięcie okna.

```
<script type="text/javascript">  
    alert("treść komunikatu");  
</script>
```

# Składnia języka JS – okno dialogowe 2

- **Okno decyzyjne** odpowiada za wyświetlenie 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("treść komunikatu");  
</script>
```

# Składnia języka JS – okno dialogowe 3

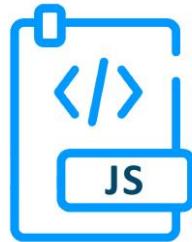
- **Okno tekstowe** wyświetla treść komunikatu stanowiącego argument metody **prompt** oraz pole umożliwiające wprowadzenie danych przez użytkownika. W trakcie wywoływania okna tekstowego w polu może pojawić się tekst domyślny.

```
<script type="text/javascript">  
    prompt("treść komunikatu", "tekst domyślny");  
</script>
```

# 4. Ćwiczenia JS

Ćwiczenia 01-03

# Ćwiczenie JavaScript-01

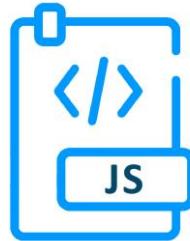


- Przygotuj plik **index.html** z podstawowymi znacznikami html i dodaj do niego znaczniki umożliwiające obsługę skryptów JS.

```
JavaScript-01.html
1 <html>
2 <head>
3   <title>JavaScript-01</title>
4   <script type="text/javascript">
5     <!--
6       // --
7     </script>
8   </head>
9
10 <body>
11   <script type="text/javascript">
12     <!--
13       // --
14     </script>
15   </body>
16 </html>
```

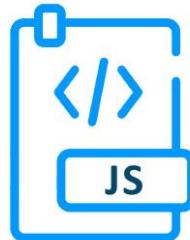
The screenshot shows a code editor window titled "JavaScript-01.html". The code is a basic HTML document structure. It includes an HTML tag, a head section containing a title and a script tag, and a body section containing another script tag. Both script tags contain comments starting with '--'. The code is color-coded: HTML tags are in blue, script tags are in red, and comments are in green. The editor interface at the bottom shows file statistics: "length : 226", "lines : 17", "Ln : 17", "Col : 1", "Pos : 227", and encoding information: "Windows (CR LF)", "UTF-8", and "INS".

# Ćwiczenie JavaScript-02



- Korzystając z pliku utworzonego w ćwiczeniu-01 napisz skrypt wyświetlający napis: „Witaj świecie”. Do wyświetlenia komunikatu użyj funkcji **alert()**.
- Utwórz skrypt z komunikatem wymagającym potwierdzenia przez użytkownika korzystając z funkcji **confirm()**.

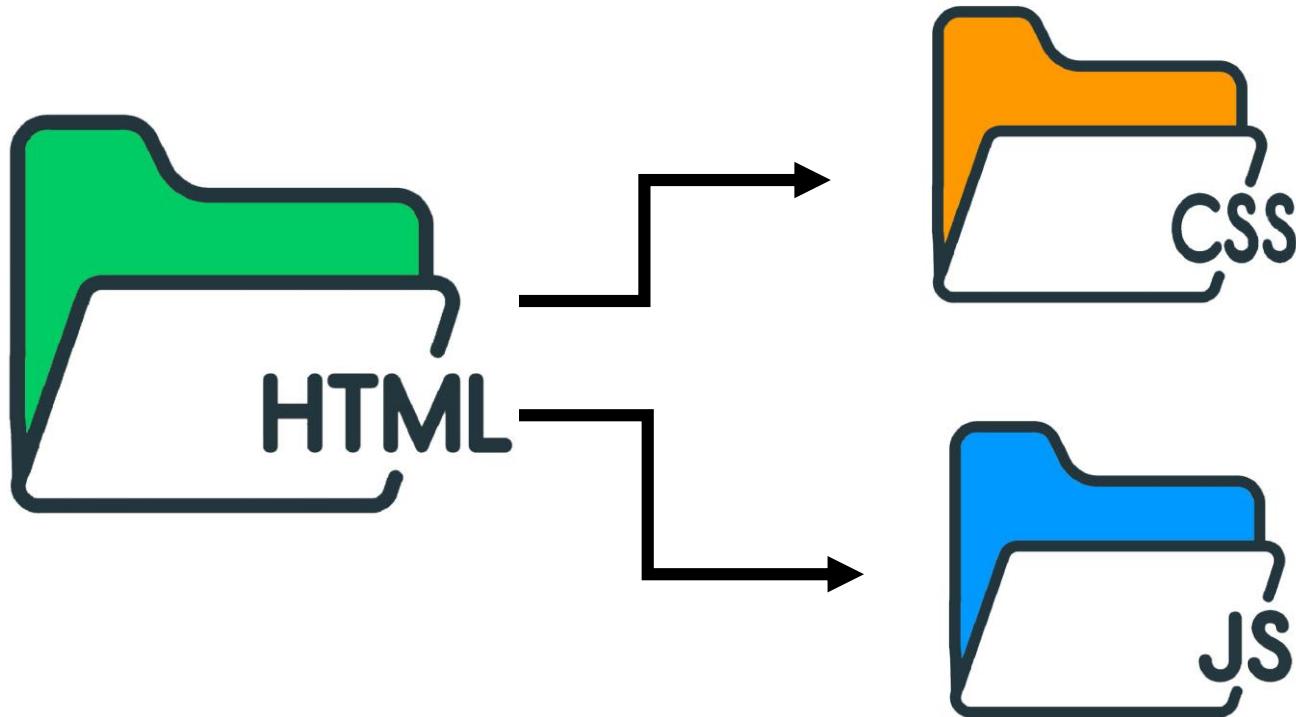
# Ćwiczenie JavaScript-03



- Zmodyfikuj skrypt z poprzedniego ćwiczenia, tak aby na stronie pojawił się komunikat "Witaj *imie*". Zmienną *imie* utwórz przez przypisanie wyniku wywołania funkcji **prompt()** która pyta użytkownika i imię.
- Do wypisania danych na ekranie użyj funkcji **document.write()** dopisującej treść do strony html.



# Struktura katalogów projektu witryny



# Informacje

## Bibliografia

1. Jolanta Pokorska, Kwalifikacja INF.03 Tworzenie i administrowanie stronami i aplikacjami internetowymi oraz bazami danych. Część 3. Podręcznik do nauki zawodu technik informatyk i technik programista, Helion 2020



## Legenda

1. Wykład
2. Notatka z wykładu
3. Ćwiczenie JavaScript

