

# JavaScript

prorowadzenie  
do języka



# Środowisko programistyczne

IDE (ang. Integrated Development Environment) to zintegrowane środowisko programistyczne. IDE jest to program/zbiór programów niezbędnych do tworzenia, rozwoju, modyfikacji i testowania oprogramowania. Ułatwia on pracę programistyczną oferując szybki dostęp do ważnych i często używanych funkcji oraz automatyzując często wykonywane zadania. Zintegrowane środowisko programistyczne zawiera w zależności od zakresu użycia kilka komponentów, takich jak na przykład edytor tekstowy, kompilator, debugger (narzędzie do testowania), tłumacz oraz inne.



Visual  
Studio



Visual Studio Code



# Pojęcie skryptu

Skrypt to zbiór instrukcji lub poleceń, które wykonuje komputer. Skrypty są często używane w programowaniu i służą do automatyzacji powtarzalnych zadań lub do wykonywania złożonych operacji. Używa się ich do tworzenia dynamicznych stron internetowych, gdzie są one wykonywane wewnątrz aplikacji.

Skrypty mogą być również używane w celu zmiany zachowania programów lub systemów operacyjnych.

Do tworzenia skryptów używa się skryptowych języków programowania. Interpreter języka JS wbudowany jest w przeglądarki

Pytanie 1 Co to jest skrypt?

Pytanie 2 Podaj przykładowe zastosowania skryptów.



# JavaScript

Skryptowy język programowania, którego typowe zastosowania w aplikacjach internetowych to:

- Służy do tworzenia interaktywnych stron
- Zapewnia obsługę zdarzeń
- Sprawdza poprawność wypełnienia formularzy po stronie klienta
- Obsługuje proste okna dialogowe
- Daje dostęp do DOM (ang. Document Object Model) zgodnie ze standardami W3C

Pytanie: Podaj zastosowania JS w aplikacjach internetowych



# Dołączanie skryptu JS do aplikacji

- I. Umieszczanie skryptu wewnątrz znacznika `<script>` i `</script>` zarówno w sekcji `<head>` jak i `<body>`. Znacznika tego można używać wielokrotnie, byle nie zagnieżdżać.
- II. W części `<head>` do znacznika `<script>` można dodać atrybut `src`, który definiuje względną ścieżkę dostępu do pliku z zawartością skryptu o rozszerzeniu `js`. Wygląda to wtedy następująco: `<script src = „skrypty/skrypt1.js” >`  
`</script>`. Tutaj również można go użyć wielokrotnie aby dodać wiele plików ze skryptami.

Pytanie 1: Podaj sposoby dołączenia skryptów js do strony

Pytanie 2: Ile skryptów JS można dołączyć do strony

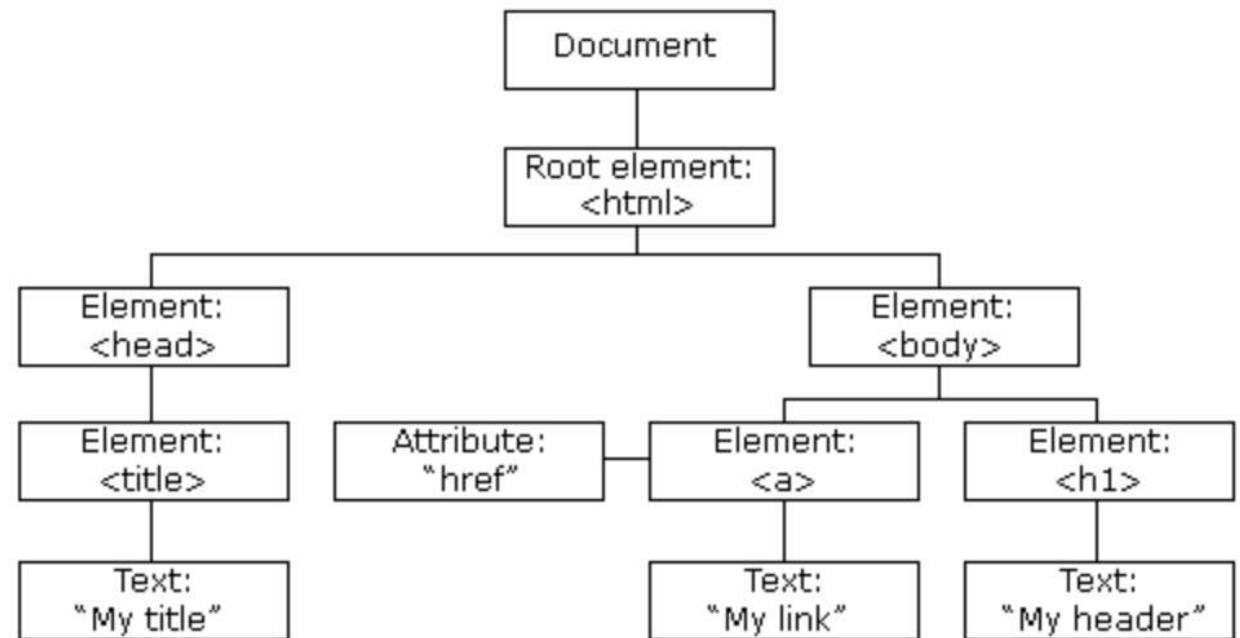
Zadanie: Dołącz puste skrypty JS do strony poznanymi sposobami

# DOM (ang. Document Object Model)

W trakcie wczytywania strony przez przeglądarkę tworzony jest DOM – skonstruowany jako drzewo obiektów. Model ten opisuje jak zaprezentować tekstowe dokumenty HTML (pisane jako tekst w edytorze) w postaci modelu obiektowego w pamięci komputera.

(rys. ze strony [https://www.w3schools.com/js/js\\_htmlDOM.asp](https://www.w3schools.com/js/js_htmlDOM.asp))

## The HTML DOM Tree of Objects



# Dostęp do obiektów DOM

Obiekty mają swoje właściwości i metody. Metody to w uproszczeniu funkcje zdefiniowane dla obiektu i na nim wykonywane. Dostęp do nich: nazwa obiektu, kropka, nazwa metody, po której (jak dla każdej funkcji) następuje lista argumentów w nawiasach okrągłych, czyli **obiekt.metoda(argumenty)**;

```
document.write(„Ala ma kota”);
```

```
window.alert(„Pora na zadania”);
```

UWAGA. Obiekt *window* jest wyjątkiem i można pominąć go w zapisie. Zostanie wtedy: **alert**(„Pora na zadania”);

Pytanie: W jaki sposób wywołać metody obiektów DOM w JS

Zadanie 1: Do istniejącego skryptu dopisać podane metody

Zadanie 2: Sprawdzić działanie znaczników HTML wewnątrz metody `document.write()`. Można użyć przykładów z podręcznika str. 38





# Składnia języka JavaScript

W językach składania to zbiór pewnych reguł, które są istotne dla danego języka. I tak jak w językach stosowanych w komunikacji należy przestrzegać pewnych zasad gramatyki, tak w językach programowania składnia odgrywa istotną rolę, a przestrzeganie ustalonych reguł jest niezbędne dla prawidłowego działania programu.

Aby prawidłowo używać języka należy znać jego składnię.

Składnie wielu języków programowania są podobne i inspirowane językiem C++.





# Instrukcje




Średnik jest ogranicznikiem, może być jednak pominięty, gdy następny jest znak nowej linii. Bezpieczniej jest go stawiać zawsze.

Dla czytelności kodu dobrze jest pisać po jednej instrukcji w jednej linii.



Nadmiarowe spacje są pomijane, ale ich użycie może zwiększać czytelność kodu.

Nawiasy klamrowe służą do grupowania instrukcji w bloki, gdzie są wykonywane wtedy razem jedna po drugiej, np. w pętlach.



# Słowa kluczowe

Każdy język zawiera własny zestaw słów kluczowych

To słowa o określonym znaczeniu oznaczającym jakąś instrukcję, rozkaz, deklarację itp.

Są zastrzeżone, co oznacza, że nie można tych słów używać do nazywania własnych elementów w programie.

Są pisane małymi literami. W nazwach metod stosowany jest zapis wielbłądzi.



# Komentarze

Komentarze to fragment w kodzie pomijany przez translator (kompilator, interpreter)

Użycie komentarzy zwiększa rozumienie kodu przez czytającego co niesie istotne korzyści przy jego modyfikowaniu i rozwijaniu i jest w dobrym tonie.

Znak // to komentarz liniowy, który wyznacza jego początek. Końcem jest koniec linii.

Znaki /\* oraz \*/ są odpowiednio początkiem i końcem komentarza blokowego. Może być on wielolinijkowy.



# Pytania i zadania

Pytanie 1: Podaj znaki i zakres stosowania komentarzy

Pytanie 2: Jakie jest znaczenie średnika w języku JS.

Pytanie 3: Do czego służą nawiasy klamrowe.

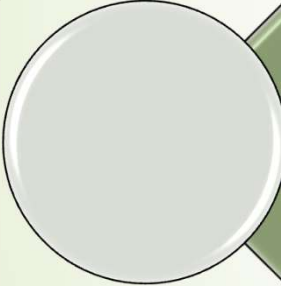
Pytanie 4: W jaki sposób zapisywane są słowa kluczowe.

Zadanie 1: Dopisz do swojego skryptu komentarz liniowy zawierający aktualną datę i komentarz blokowy zawierający Twoje imię i nazwisko.

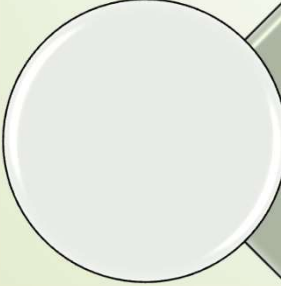
# Zmienne



Służą do przechowywania wszelkich danych

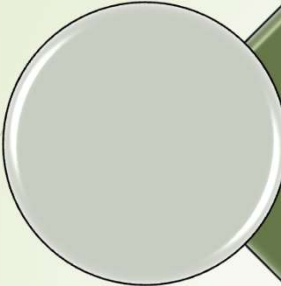
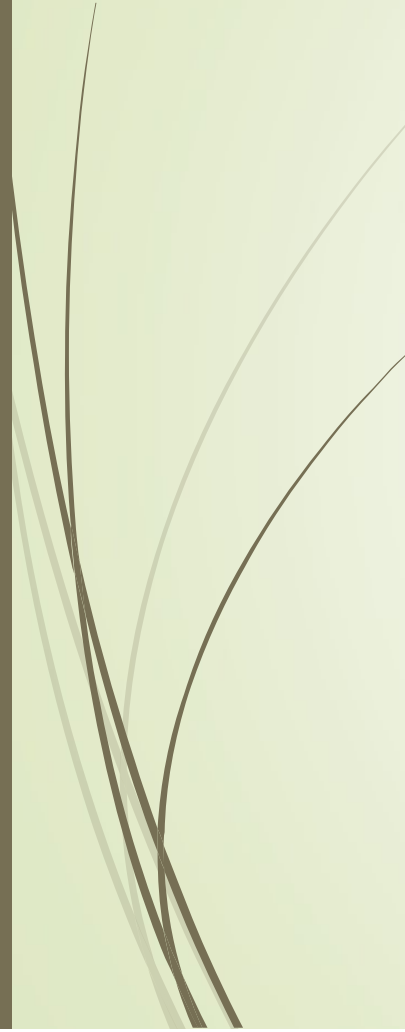



Przed użyciem zmiennej należy ją zadeklarować. W czasie deklaracji nie podajemy typu zmiennej, a ten może ulec zmianie w trakcie działania skryptu.

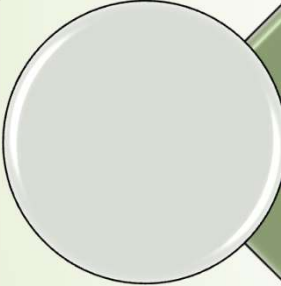


Nazwa zmiennej nie może być słowem kluczowym, składa się z liter, cyfr, podkreślenia, nie może zaczynać się od cyfry. Wielkość liter ma znaczenie.

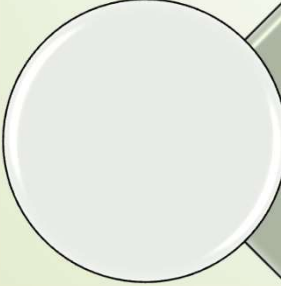
# Deklaracja zmiennych



var - służy do deklarowania zmiennych o globalnym lub lokalnym zakresie widoczności, które mogą być modyfikowane w dowolnym momencie.



let - służy do deklarowania zmiennych o lokalnym zakresie widoczności, które mogą być modyfikowane.



const - służy do deklarowania stałych o lokalnym zakresie widoczności, których wartość nie może być zmieniona po zainicjowaniu.

# Korzystanie ze zmiennych

Aby móc korzystać ze zmiennych należy ją zainicjować. Można to zrobić za pomocą standardowe przypisania czyli znaku =. Można to zrobić zarówno na etapie deklaracji (sposób II), jak i później (sposób I):

**I. let y;**

**y = „Ala ma kota”;**

**II. let x = 2;**

Pytanie: Dla zmiennych x, y zainicjowanych wyżej podaj co wyświetli się na stronie po wykonaniu następujących instrukcji skryptu

```
document.write(x) ;  
document.write(y) ;  
document.write(y + " a nawet " + x + " koty.") ;
```





# Pytania i zadania

Pytanie 1: Podaj czynność niezbędną do wykonania przed użyciem zmiennej w JS

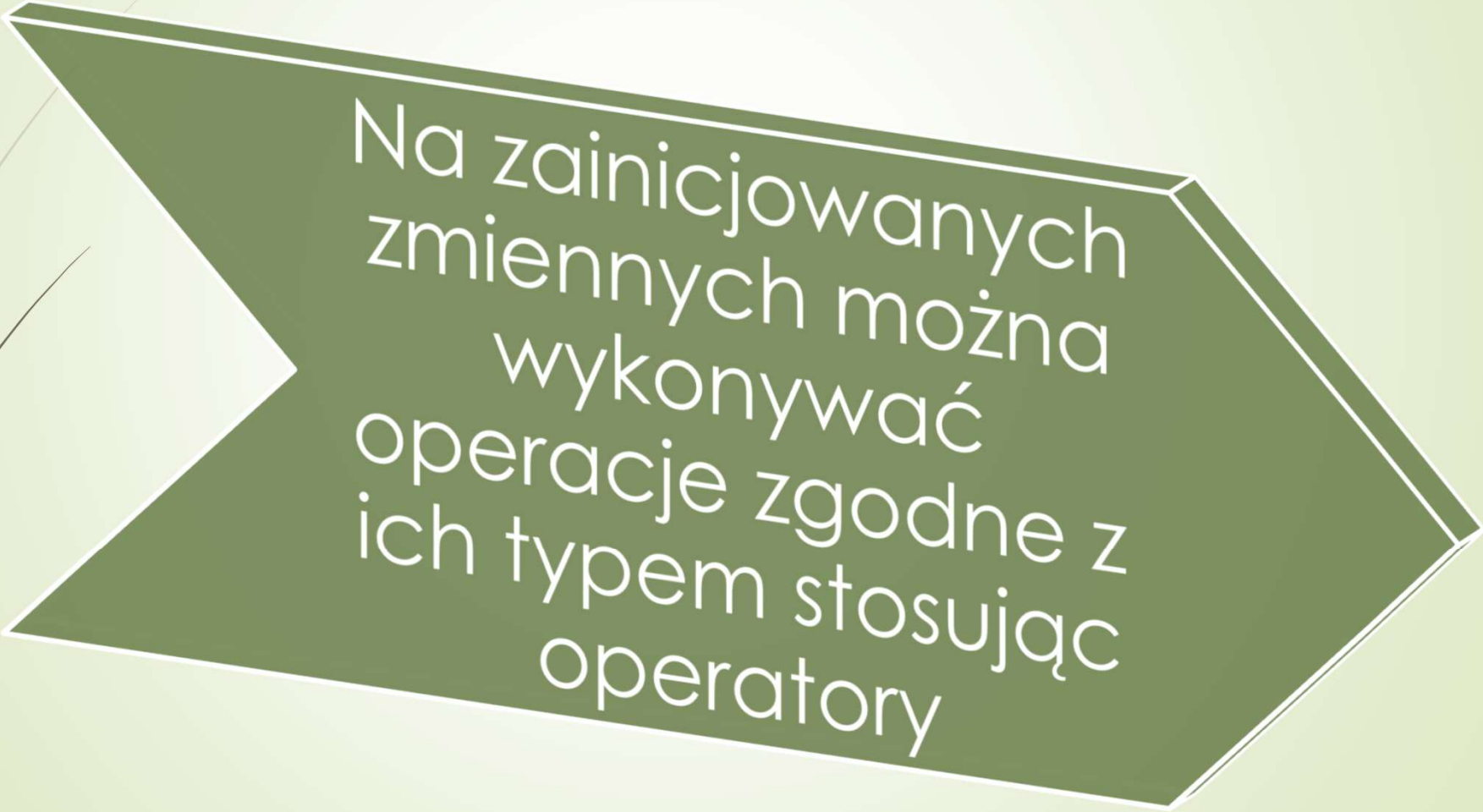
Pytanie 2: W jaki sposób podajemy typ zmiennej w JS

Pytanie 3: Wymień zasady dotyczące nazywania zmiennych w JS


Pytanie 4: Jakie słowa kluczowe służą do deklaracji zmiennych w JS. Podaj ich znaczenie.



## Operacje na zmiennych



Na zainicjowanych  
zmiennych można  
wykonywać  
operacje zgodne z  
ich typem stosując  
operatory



# Operatory arytmetyczne – stosowane do zmiennych liczbowych

Operator	Operacja
+	dodawanie
-	odejmowanie
*	mnożenie
/	dzielenie
**	potęgowanie
%	modulo (reszta z dzielenia)
++	inkrementacja (wzrost o 1)
--	dekrementacja (ubytek o 1)

# Pytania i zadania

Pytanie 1: Wymień poznane operatory arytmetyczne.

Pytanie 2: Co się pojawi na ekranie w wyniku działania skryptu

```
1  let a = 8;  
2  let b = 2;  
3  let c = 4;  
4  document.write(a+b + "<br>");  
5  document.write(c/b + "<br>");  
6  document.write(a-b-c + "<br>");  
7  document.write(b%c + "<br>");  
8  document.write(c**b + "<br>");  
9  document.write(a++ + "<br>");  
0  document.write(c-- + "<br>");
```

# Obiekt window

- Reprezentuje otwarte okno w przeglądarce
- Jest nadrzędny wobec innych obiektów hierarchii DOM
- Tworzony automatycznie podczas otwierania okna
- Jako jedyny nie wymaga podania nazwy obiektu
- Posiada sporo ważnych właściwości i metod
- [https://www.w3schools.com/jsref/obj\\_window.asp](https://www.w3schools.com/jsref/obj_window.asp)

# Okna dialogowe

Metody obiektu  
window

alert(„Czas na  
praktykę”);

Komunikat z bieżącej strony  
Czas na praktykę

OK

confirm(„Czy  
to koniec?”);

Komunikat z bieżącej strony  
Czy to koniec?

OK

Anuluj

prompt(„Podaj  
liczbę x:”);

Komunikat z bieżącej strony  
Podaj liczbę x:

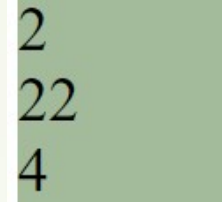
OK

Anuluj

# Metoda window.prompt

```
<script>
  let x= prompt("Podaj liczbę x:");
  document.write(x + "<br>");
  document.write(x+x + "<br>");
  document.write(x*x + "<br>");
</script>
```

Zakładając, że użytkownik wprowadzi 2 w okienku tekstowym skryptu z lewej strony, zobaczy na ekranie:



```
2
22
4
```

Jest to spowodowane, faktem że okienko jest tekstowe zatem wprowadzamy znak 2, a nie liczbę 2.

Znak + oznacza konkatencję (łącznie) napisów stąd wynik 22.

Użycie operatora \* dokonało konwersji zmiennej x na typ liczbowy i stąd dobry wynik 4. Ma ona czasowy charakter.



# Metoda Number()

Przykładowy sposób wprowadzania liczby:

```
let x= Number(prompt("Podaj liczbę x:"));  
document.write(x + "<br>");  
document.write(x+x + "<br>");
```

Metoda globalna

Konwertuje różne wartości na liczby

Zwraca liczbę lub NaN

# Rzutowanie typów

Do zamiany typów można użyć tak zwanego rzutowania. W przypadku rzutowania na typ liczbowy mamy dwie metody globalne:

- `parseInt(argument)` – zamienia stringa na liczbę całkowitą
- `parseFloat(argument)` – zamienia stringa na liczbę zmiennoprzecinkową

```
let x= parseInt(prompt("Podaj liczbę x:"));  
let y= parseFloat(prompt("Podaj liczbę y:"));
```

# Pytania

Pytanie 1: Wymień rodzaje okien dialogowych w JS

Pytanie 2: Podaj metody do tworzenia okien dialogowych w JS. Jakiego obiektu są to metody?

Pytanie 3: Omów postać, i zwracaną wartość okna dialogowego, którym można wprowadzić informację do aplikacji.

Pytanie 4: Jakiego rodzaju metody służą do zamiany typów zmiennych

Pytanie 5: Wymień metody służące do zmiany typów na liczbowe.

Pytanie 6: Dla każdej metody do zmiany typów na liczbowe podaj możliwe typy argumentów i typ zwracanej wartości.



# Zadania

W poniższych zadaniach napisz skrypty będące realizacją zagadnienia w zadaniu. Dla uproszczenia przyjmujemy, że użytkownik wprowadza poprawne dane. Działanie skryptów przetestuj na stronie internetowej, pisząc przy każdym skrypcie którego zadania on dotyczy.

Zadanie 1: Dla podanej przez użytkownika podstawy trójkąta i wysokości na nią opadającej, wylicz pole trójkąta.

Zadanie 2: Napisz skrypt, który przeliczy podaną wielkość w bajtach na bity.

Zadanie 3: Napisz skrypt, który wczytując od użytkownika aktualny kurs euro i podaną kwotę w euro przeliczy jej wartość na naszą walutę.